First Workshop on Secure Control Systems (SCS)

April 12, 2010

Royal Institute of Technology (KTH)

Stockholm, Sweden

TRUST
Team for Research in Ubiquitous Secure Technology

KTH
VETENSKAP
OCH KONST

CONTENTS

WELCOME MESSAGE

On behalf of the Workshop Program Committee, it is my pleasure to welcome you to the ***First Workshop on Secure Control Systems (SCS)*** in conjunction with CPSWeek 2010 at the Royal Institute of Technology (KTH) in Stockholm, Sweden.

This workshop will bring together researchers and practitioners from academia, industry, and the government to discuss system theoretic approaches to enhance the security and resilience of control and monitoring systems. These systems govern the operation of critical infrastructure systems such as power transmission, water distribution, transportation networks, building automation systems and process control systems.

The workshop will include paper presentations as well as a tutorial session on the need for resilient control systems lead by researchers from the Idaho National Laboratory (INL) which will focus on the importance of resilient control systems and their related processes as well as SCADA test-beds. The workshop will also include a panel

In addition to presentations and a tutorial, the workshop will include a panel discussion session focused on enhancing security and privacy of networked control systems. The panel will consist of academic researchers and industry practitioners and is intended to complement the research talks and technical tutorial to provide a better idea of the pressing issues in this nascent field.

We greatly appreciate your participation and hope you find the workshop informative.

Sincerely,

S. Shankar Sastry, Ph.D.
Workshop Co-Chair
University of California, Berkeley

Miles McQueen, Ph.D.
Workshop Co-Chair
University of Idaho

WORKSHOP OVERVIEW

Over the last few years, researchers have gathered at CPSWeek to define the research scope for cyber-physical systems (CPS) which are systems that integrate computation, networking, and physical processes. Modern control technology is based on embedded computers and networked systems that monitor and control large-scale physical processes. The use of internet-connected devices and commodity IT solutions to enhance scalability and performance of control systems on one hand and the malicious intents of hackers and cybercrime networks on the other have made control systems more vulnerable now than before.

Despite numerous attempts to develop guidelines for the design and operation of security policies for control systems, much remains to be done in order to arrive at a principled approach to enhance security, trustworthiness, and dependability of control systems. In light of the nascent state of this discipline, the scope of this workshop is to discuss theories and methodologies that encompass ideas from:

- Robust, fault-tolerant and networked control systems
- Heterogeneous composition, abstraction and verification methods
- Information security and privacy.

Included to both enliven and broaden the discussion across disciplines during the workshop, will be a tutorial that discusses the current state of software security, the human importance and weakness in aiding system security and robustness, and an example demonstrating the difficulty of assessing potential damage to systems under cyber attack. As the practicality and benefit of new technologies require both simulated and experimental environments to handle interdependencies, an overview of the process and opportunities for verification will be suggested. The tutorial will emphasize that both resilience and security need to be integrated to achieve the desired protection of our critical physical processes.

In doing so, the workshop aims to foster collaborations between interested researchers from the fields of control and systems theory, software verification and computer security. A secondary goal of the workshop is to discuss the establishment of dedicated benchmarks and test-beds that can help in accelerating the development of new theories and tools. The scope of the workshop includes, but is not restricted to, the following topics:

- Taxonomy of attacks and attack models for control systems
- Novel security challenges in control systems
- Decision and game theoretic approaches to security analysis
- Design architectures for prevention and resilience/robustness against attacks
- Risk assessment and verification of security properties
- Detectability and diagnosis of attacks
- Complexity and resilience in control systems

Approaches that can be applied to particular critical infrastructure systems such as power grid, water distribution, transportation systems and process control systems are particularly emphasized.

TRUST CENTER OVERVIEW

The **Team for Research in Ubiquitous Secure Technology (TRUST)** is focused on the development of cyber security science and technology that will radically transform the ability of organizations to design, build, and operate trustworthy information systems for the nation's critical infrastructure. Established as a National Science Foundation Science and Technology Center (STC), TRUST is addressing technical, operational, legal, policy, and economic issues affecting security, privacy, and data protection as well as the challenges of developing, deploying, and using trustworthy systems.

TRUST activities are advancing a leading-edge *research* agenda to improve the state-of-the art in cyber security; developing a robust *education* plan to teach the next generation of computer scientists, engineers, and social scientists; and pursuing *knowledge transfer* opportunities to transition TRUST results to end users within industry and the government.

TRUST is addressing technical, operational, privacy, and policy challenges via interdisciplinary projects that combine fundamental science and applied research to deliver breakthrough advances in trustworthy systems in three "grand challenge" areas:

Financial Infrastructures – Creation of a trustworthy environment that links and supports commercial transactions among financial institutions, online retailers, and customers.

Health Infrastructures – Technology that advances "Healthcare Informatics" to enable engaged patients, personalized medicine, providers as coach-consultants, and agile evidence-based care.

Physical Infrastructures – Advances that support Next Generation Supervisory Control and Data Acquisition (SCADA) and control systems, including power, water, and telecommunications.

TRUST is led by the University of California, Berkeley with partner institutions Carnegie Mellon University, Cornell University, Mills College, San Jose State University, Smith College, Stanford University, and Vanderbilt University. TRUST projects have a holistic view that addresses computer security, software technology, analysis of complex interacting systems, and economic, legal, and public policy issues. As such, TRUST draws on researchers is such diverse fields as Computer Engineering, Computer Science, Economics, Electrical Engineering, Law, Public Policy, and the Social Sciences.

More information on TRUST is available at **http://www.truststc.org**.

CONFERENCE PAPERS

The VIKING Project – Towards more Secure SCADA Systems

Gunnar Björkman
Senior Consultant and
Project Coordinator VIKING
ABB Network Management
gunnar.bjoerkman@de.abb.com

Abstract — The purpose of this paper is to give an overview of the VIKING project including its motivation and background. The VIKING project has been started to investigate the increased cyber security risks for deliberate attacks on critical infrastructures coming from SCADA systems and to propose mitigation.

The second part of this paper describes the principle design of modern SCADA systems in order to give a better understanding of this technology.

The VIKING project is an EU financed Framework 7 Collaborative STREP Project and is part of themes 4, ICT, and 10, Security. VIKING stands for Vital Infrastructure, NetworKs, INformation and Control Systems ManaGement and aims on making SCADA system more resilient against cyber attacks.

I. INTRODUCTION

Society is increasingly dependent on the proper functioning of the electric power system, which in turn supports most other critical infrastructures: water and sewage systems; telecommunications, internet and computing services; air traffic, railroads and other transportation. Many of these other infrastructures are able to operate without power for shorter periods of time, but larger power outages may be difficult and time consuming to restore. Such outages might thus lead to situations of fully non-functioning societies with devastating economical and humanitarian consequences.

The operation and management of the electric power system depend on computerized industrial control systems, so called SCADA systems standing for System for Control And Data Acquisition. Keeping these systems secure and resilient to external attacks as well as to internal operational errors is thus vital for uninterrupted service. However, this is challenging since the control systems are extremely complex. Yet, the systems are operating under stringent requirements on availability and performance: If control and supervision are not done in real-time, the power network may come to a collapse.

The VIKING project will take a holistic approach and investigates the risks for cyber attacks on all parts of the SCADA system including substation control systems, communication networks and central control systems. The project aims to model the whole chain from cyber attacks, modeled via attack trees, over architectural SCADA system models, power system models down to societal models in order to evaluate the cost and consequences for the society from cyber attacks on SCADA systems. One important part of the project is to verify models and proposed mitigation methods on a test bench.

A second part of this paper will discuss the principle design of modern computerized control systems including data acquisition, event and alarm handling, user interface, process models, optimization and simulation in order to increase the understanding how these systems are designed.

The VIKING project results could potentially have substantial, practical importance for the implementation and usage of SCADA systems. Therefore, the VIKING consortium includes the industrial partners ABB, E.ON and Astron. ABB is one of the world leading SCADA vendors and E.ON is one of the major energy utilities in Europe. Astron is a system integrator working in Hungary. Possible benefits resulting from the new approaches in VIKING will be described from an industrial view point in the last section of the paper.

II. THE VIKING PROJECT

The VIKING project will concentrate its research on computerized system for the supervision and control of electrical transmission and distribution networks. One of the reasons to limit the project to the electrical process is that the results of VIKING is believed to be applicable to other SCADA systems used for other critical infrastructures like gas, water, telecommunication, etc. and, as described in the Introduction, the vital importance of the electricity supply.

These computerized control systems include functions for remote collection of vast amounts of real-time data coming from measurement devices placed at strategic points, e.g. in transformer substations, in the geographically widely spread process and for the remote control of process devices. Many SCADA systems include computerized models of the supervised process which enables simulation of alternatives process states and optimization. Due to legal and environmental constraints, e.g. for building of new high voltage power lines or power stations, the primary process itself is difficult to expand which in its turn leads to higher and higher utilization of the existing transmission, distribution and generation resources. The process is, in

other words, operated closer to its physical limits. Thus the SCADA systems are becoming increasingly critical for the operation of the process and therefore are becoming a critical component for the availability, safety and security of the supervised infrastructure.

The objective of the VIKING project is to develop, test and evaluate methodologies for the analysis, design and operation of resilient and secure industrial control systems for critical infrastructures. Methodologies will be developed with a particular focus on increased robustness of the control system. As mentioned, the focus is on power transmission and distribution networks. The project combines a holistic management perspective—in order to counteract sub-optimization in the design—with in-depth analysis and development of security solutions adapted to the specific requirements of networked control systems.

The traditional approach to verify the security of SCADA systems has been ad-hoc testing of existing commercial SCADA system in laboratory environments. The systems to be examined have been installed in different labs and tested by skillful people searching for cyber attacks vulnerabilities. The focus in these tests has been on the protection of the central computer system of the SCADA system, since the central computer system has most connections to the outside world through office networks, vendor links and Internet.

In the VIKING project we will take an alternative and complementary approach to SCADA system security. Firstly we will study the whole control system from the measurement points in the process itself over the communication network to the central computer system as illustrated in the following picture with the yellow exclamation marks indicating potential targets for cyber attacks.

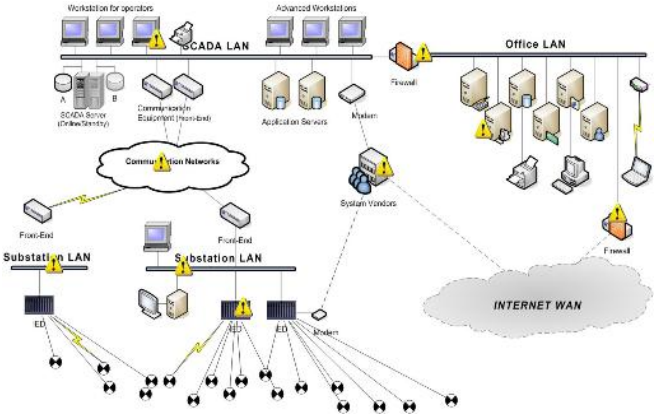


Figure 1 – SCADA System vulnerabilities

Secondly, and more importantly, we take a model-based approach to investigating SCADA system vulnerability. Models are defined for the SCADA system, for the electrical process as well as of for the society that is dependent on the electricity supply. The society models are used to evaluate the economic consequences coming from disturbances in the electricity supply and to give load scenarios for the simulations. The power system models are in turn used to evaluate the effects on the electricity supply caused by SCADA system misbehavior. Finally, SCADA system architectural and cyber-physical models are employed to assess the effect on SCADA system behavior caused by cyber attacks. Based on analysis performed on these models, VIKING will propose mitigation actions to be taken to decrease or to eliminate these risks. The results of the project will be evaluated on a test-bed that can be configured to simulate cyber attacks on the power network coming from SCADA and the corresponding consequences in the virtual society.

The modeling approach is indicated in the following picture.

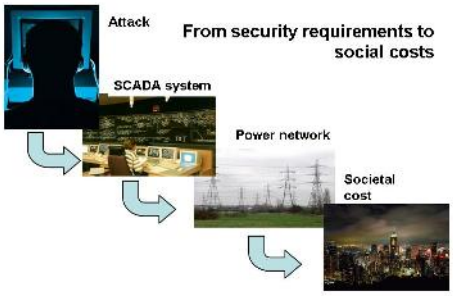


Figure 2 – Modeling approach

With this approach the project hopes to achieve the following research results.

- Estimates of the security risk and consequences (in terms of monetary loss for the society) based on threats trees, graphical system architecture and society models
- Comparable, quantitative results for IT security for different control system solutions and implementations
- Use of existing model based application as application level Intrusion Detection Systems to detect manipulation of data
- Use of innovative and existing communication solutions to secure power system communication
- Help with identifying ”weak spots” and how to mitigate them
- An environment for performing what-if analyses of the security risk impact of different architecture solutions.

III. Security Incidents

The number of security incidents reported in the area of critical infrastructures has increased significantly over the last few years. Even attacks on control systems are becoming more frequent and sophisticated. The diagrams shown below were published in the Pipeline and Gas Journal [3]

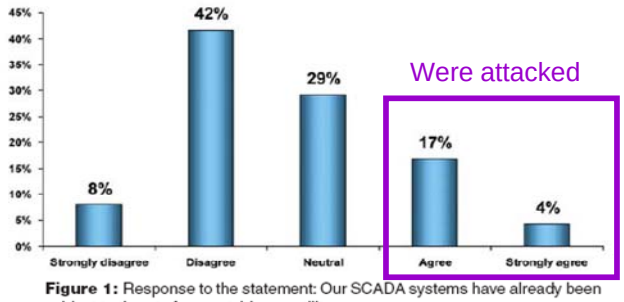


Figure 1: Response to the statement: Our SCADA systems have already been subject to threats from outside our utility.

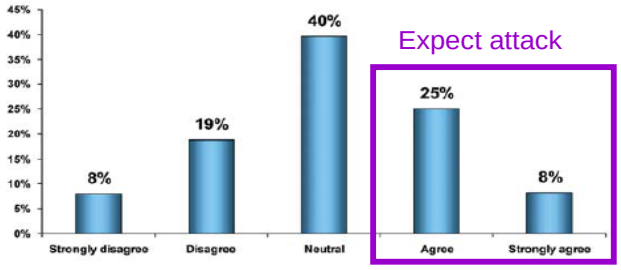


Figure 2: Response to the statement: A utility's SCADA system or energy distribution system will be attacked or compromised in the next 24 months.

Figure 3 – Attacks on control systems

The incidents in the diagrams above include directed and malicious intrusion attempts as well as unintentional security breaches done by mistake. In addition to the threats from viruses and hackers breaking into computer systems, there is a growing concern over the possibility of network based terrorist attacks against infrastructure and critical process industries.

Many of the reported incidents were initiated by people with legitimate access to the network. In general, these attacks are the most difficult ones from which to protect a system, because insiders (or former insiders) are the most likely persons to have access to passwords, codes, and systems, and to have knowledge about the nature of the system and its potential vulnerabilities. Recently, however, the share of externally sourced incidents has increased drastically, particularly in the form of virus and worm infections. In many cases, virus and worm infections are caused by connecting a portable computer or storage device that has previously been connected to an infected environment [4].

There is no single solution or technology for network security that fits the needs of all organizations and applications. While basically all computer systems are exposed to intrusion attempts, the potential consequences of such attempts are vastly different for different types of applications.

Cyber security measures aim at protecting the confidentiality, integrity, and availability of a computer system from being compromised through deliberate or accidental attacks. This is accomplished by implementing and maintaining a suitable set of controls to ensure that the security objectives of the organization are met. These controls should include policies, practices, procedures, and organizational structures, as well as software and hardware implemented security functions.

The security measures that are applied to a specific installation should be proportional to the assessed risk in terms of probability of a successful attack and the potential consequences. For a small system with a few users controlling a non-critical process this risk is obviously smaller than for a large system spanning multiple sites with safety critical processes in several countries and continents and thousands or even tens of thousands of users.

It is not possible to achieve 100% security in an interconnected environment. A network that is arranged with state-of-the-art security measures may still be vulnerable through connections to the networks of suppliers, contractors or partners. Even a network that is perceived as being totally isolated from the outer world is vulnerable to security intrusions from different sources, such as the occasional connection of portable computers, modems that are not properly disconnected or unauthorized installation of software.

IV. Restructuring of the Power Industry and its Impact on SCADA System

The security of computer systems in general and of SCADA systems in particular, has become increasingly critical. Changes resulting from electric power industry restructuring have increased the need for heightened information security efforts in this industry. In many countries, unbundling of the power generation function from the power delivery and retail functions, as well as deregulation of the power generation market, have motivated power plant and network owners/operators to reduce costs and improve plant and network operating efficiency. To do this, these owner/operators have implemented several changes and new practices that can potentially affect the cyber security of their power plants.

The most significant impacts of these changes relate to the SCADA systems that are used to operate many transmission and distribution networks and power plants worldwide. For example, many transmission network owner/operators have added connections between their corporate office networks and their SCADA systems. This interconnectivity allows corporate decision makers to obtain instant access, for example, to critical data about the status of their operating assets. However, this interconnection also opens new vulnerabilities to the SCADA system, as the corporate network then becomes a potential additional access point to the control system.

Other practices to improve network efficiency include enabling remote access to SCADA systems by company engineers, contractors, vendors, and others via dial-in

modems and other means. But, like the interconnection with the corporate network, this practice introduces new access points to the SCADA system. Visiting employees from other locations, hired contractors, and other authorized parties also need to access the corporate network from their laptop computers to gather information to aid decision making and maintenance activities. Such access may, in turn, unleash viruses or malicious code on the SCADA systems.

The drive to improve network operation is also leading to increasing standardization of SCADA technologies. SCADA systems are increasingly implemented on Microsoft Windows and UNIX/Linux operating system-based platforms, enabling a broad range of third parties to offer software that can help optimize plant operation and maintenance techniques. Similarly, most SCADA systems comply with OPC (a Microsoft-based standard for open connectivity) and with the standardized protocols of major manufacturers of Remote Terminal Units (RTUs).

Providing and managing enterprise-wide Information System (IS) security is a moving and dynamic target, complicated by continuous technical, organizational, and political changes, global interconnections, and new business models such as Internet-based e-commerce. IT security is a complex challenge requiring procedural as well as technical measures.

V. Principle design of SCADA systems

To improve the understanding of what a SCADA system really is, and how it is designed, a short introduction on SCADA systems architecture is included in this report.

Modern SCADA systems designed for geographically widespread processes have a principle design as shown already in Figure 1 above. We will in the following discuss the major components one by one.

A. Process connection

Measurement devices like voltage and current transformers are placed in the supervised process and will measure analogue process values like active/reactive power, and voltages and also digital values like open/close state of breakers, isolators and transformer tap changers positions. The signals from the measurement devices are connected to electronic units, so called Remote Terminal Units (RTUs), which are placed in transformer substations or in power stations. These RTUs have the primary task of transforming the measured signals into digital format and transmit them to the central control system and to receive command and setpoints signals from the control centre and to execute these control orders to the primary equipment.

The number of signals in one RTU can vary from several dozen to a few thousand. These signals are traditionally connected via separate signal cables from each sensor to the RTU input board via marshalling cabinets.

In recent years it has become increasingly more common for RTUs to be equipped with interfaces to a serial bus on which various types of secondary station equipment, e.g. protective relays, are connected. These Intelligent Electronic Devices (IEDs) contain information about the process values which is transmitted directly from these devices over the station bus and a gateway to central control system without the need to use dual sensors and extra wiring. An international standardization work has been ongoing for a number of years and has resulted in a number of standards for station bus communication. IEC 870-5-103 is now an established standard in station control systems and IEC 61850 is a coming standard with higher ambition and scope. The intent of this standardization is, of course, to enable mixing of equipment from different manufacturers that can communicate with each other and with the central control system.

Modern generation and transformer stations frequently have their own local SCADA system including operator consoles with advanced man-machine communication to allow local control of the station. These local SCADA systems have access to all local information via the station bus and process buses. The connection to the central control system is handled by a communication node (gateway) on the station bus, which converts the local information to a communication protocol to the central operations center. Many transformer stations are unmanned, which means that those local SCADA systems are only used occasionally.

B. Communications

RTUs and station control systems communicate with the central control system over different type's communication networks and are using many types of media. Traditionally, the utilities are the owner of the communication networks since the process owners want to have full control of the networks to ensure that communication, especially during major process disturbances, is always available. However, this practice, especially in distribution networks, is becoming increasingly difficult to motivate when public communication services are becoming cheaper.

The networks used for SCADA communication are characterized by relatively low transmission speeds, typically 1200 to 9600 bits per second. Because of the low communication speed and the high requirements for data security, all SCADA vendors did traditionally use their own, proprietary communication protocol between the control center and the RTUs. Since no telegrams should be wrong, especially when commanding the process, protocols have been highly secure with many parity bits. The design rule has been that all single bit errors can be corrected and all dual bit errors are detected. Since these protocols used to be proprietary it was difficult to mix RTUs from different manufacturers in the same system.

This fact has driven a standardization process for RTU protocol and today most of the newly installed SCADA systems and modern RTUs support the international

standard IEC 60870-5-101 and IEC 60870-5-104 (TCP/IP based) and the de-facto standards DNP3.0 and MODBUS. Because of the long life of RTU installations (up to 30 years) there are still lots of older types of vendor specific protocols in operation.

A clear trend today is toward more fiber-based networks with higher communication speeds and TCP/IP based protocols. This will in the long term lead to changes in SCADA system structure and in the distribution of functionality between central control centers and substation systems, but this trend is slow.

C. Central Control Centre

The Central Control Center systems in all modern implementations are built around a Local Area Network (LAN) based on Ethernet to which all servers, workstations and other equipments are connected. This LAN can be single or redundant, but is most commonly doubled for availability reasons. For the same reason are all application servers redundant and operator workstations so designed that they can take over process operation from each other if anyone should fail. One of the main design criteria for control center configurations is to avoid that a single device failure could bring down the entire control center. Figure 4 below shows a typical, bigger control centre configuration with redundant LAN and servers.

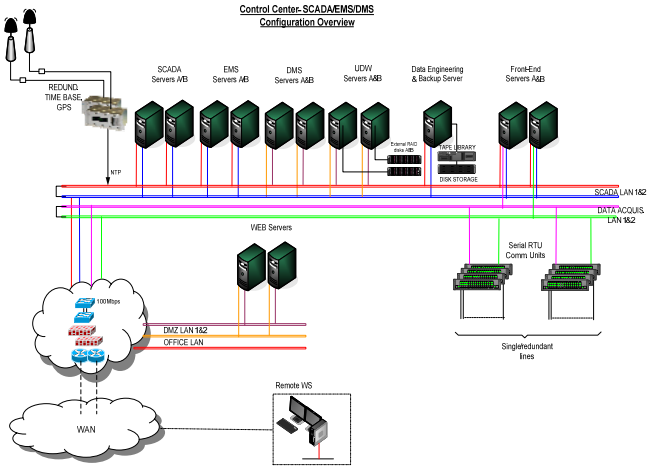


Figure 4 – Central Control System configuration

Redundant Front-End servers are responsible for the communication with RTUs and Station Control System. The Front-Ends poll the RTUs for new information which is sent to the SCADA servers. The Front-Ends are also responsible for monitoring and managing data acquisition network.

The process information from the Front-End servers are sent to the SCADA servers and stored in a real-time database. The real-time database maintains an image of the current process state as accurate as possible. The time difference between the real process state and the information in the real-time database is normally in the range of a few seconds.

SCADA servers are today implemented on commercially available computers based on UNIX/Linux or Windows with vendor specific SCADA software and comprehensive applications. The reason that virtually all major SCADA vendors use proprietary real-time databases is that the performance at process disturbances and for operator picture call-up can not be solved with the technology available today in commercial relational databases.

The main task of SCADA is to monitor the process data for significant changes and to alert operators about these changes. Such a significant change can be a breaker opening initiated by line protection or a voltage measurement over an alarm limit. On the other hand, a small change of active power within the permissible limits is just stored in the database for presentation in process displays but do not call for special attention. Significant changes are collected chronologically in Event and Alarm lists including local time stamps. Event lists record all what happens in the process, all operator actions and all events in the control system, while Alarm lists requires an active acknowledgment from the operators to confirm their attention.

The operators use process displays on the workstations to supervise the status of the real-time process and to command breakers and isolators or to send new setpoints to local processing units. Automatic commands for "close-loop" regulations exist but are rare, mostly the operators close the loop of active regulation.

Man-machine workstations are the tools for the operator to monitor and control the process. They are usually based on PC computers equipped with multiple VDUs and are using modern, full graphic technologies to present process information in different views. The information on operator workstations is automatically updated by the SCADA servers as soon as any significant change is detected in the process. Modern presentation techniques, e.g. multiple windows, information zoom (declutter) and layers, is employed to give the best possible view of the process to the operators under all circumstances. Figure 5 illustrates one example of dynamically updated station display.

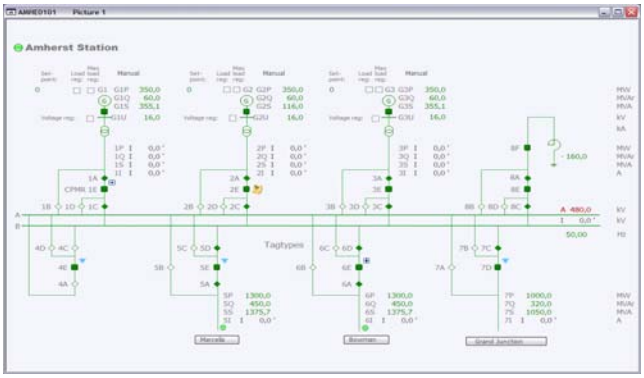


Figure 5 – Single Line Station Display

An important task for SCADA systems is to record and archive all incoming data from the process and all operator actions. For this purpose the SCADA systems use special archive servers (Data Warehouses) usually based on a commercial relational database such as Oracle. Based on the relational database, there is a variety of data mining, reporting and programming tools available for different types of users ranging from the normal control room operators to system users and application experts.

Archiving functions must have the capacity to record all information coming from the process through RTUs, station computers, and other control centers, plus archiving of event and alarm and calculation results. This means that many thousand of data points have to be stored every second. The archive function uses many types of media such as DVD or tape robots for automatic long-term storage when disk space is no longer enough. In this way virtually unlimited storage can be achieved.

It is common for a control center to be a part of a control hierarchy within the company or a country. For example, a regional control center will be connected to a national control center. The data exchange between centers will in this case take place using standardized protocols for inter-center communication where the most common today is ICCP (or TASE.2). This protocol includes functions for data acquisition and command and is, like RTU protocols, normally not encrypted.

D. Office Connection

Today it is common for office network to be connected to the SCADA systems to enable normal PC users in the office network to be connected to the SCADA real-time database and archives. This allows the office users to build specific reports and applications working directly with real-time and historical data. This, of course, means that a strict authority scheme must be applied combined with firewalls. SCADA systems are nowadays, to further increase security, configured with so-called "Demilitarized Zones' (DMZs), an isolated network part between the office network and the SCADA. The DMZ prohibits data access directly from the office network to the real-time LAN. This means that replicas of the real-time database and the historical archive have to be placed in the DMZ and protected by firewalls on both sides.

E. Process Models

So far we have only discussed SCADA systems as pure data acquisition and control systems. All measurements are acquired independently of each other and the process knowledge, i.e. how various objects of the process are interconnected are only shown in pictures. The same SCADA system can be used independent of the monitored process since the basic SCADA functions, i.e. data acquisition and control and event and alarm reporting, are used in all types of process monitoring.

If the SCADA system should include a more intelligent understanding of the monitored process, models of the various process objects must be introduced. These models are, of course, different depending on the type of process that is targeted. There is a difference in behavior between a compressor in a gas pipeline and a transformer in an electrical network although they perform similar functions in the networks. It is also important to define the connectivity, i.e. how the various process objects are connected to each other, for example, on which bus a certain circuit breakers is connected. This connectivity is combined with real time measurements of breaker and isolator states to create a dynamically updated “bus-branch” model.

Defining these models for a certain utility is a substantial work because of the vast number of monitored objects. Special tools have been developed by the SCADA suppliers to support the users in defining and maintaining these models efficiently. Support for imports from other computer systems, for example, a GIS system (Geographical Information System) where the data is maintained are often included.

F. Advanced Applications

Using the dynamic topological models discussed in the previous section, it is possible to implement advanced applications. These applications will be unique for each process type based on the different physical characteristics of the process. Perhaps the most clear examples of such applications exists for electrical SCADA systems, where for a long time such applications have been used based on relatively simple mathematical models of the electricity grid, i.e. Ohms and Kirchhoff’s laws. It is not the purpose of this report to discuss the different types of applications in detail but a brief description how these applications work can provide a better understanding of their use and importance.

By using a non-complete real-time measurement vector (not all points in the network have measurements) the complete power flow state of the network can be calculated (State Estimation) provided that the network is observable, i.e. enough measurements are available. This calculated state defines all voltages and phase angels in all nodes and can be used to calculate the active and reactive power flows for the whole network. The resulting network state can also be used to simulate new situations in the network, for example, after disconnecting a transformer for maintenance or for other operational changes in the grid. These studies are normally done in a parallel database, a so-called study database, in order not to disturb real-time operation. Load flow calculations can automatically be done for all possible errors in the network (N-1 analysis) and warnings or recommendations for grid changes to avoid dangerous situations can be issued before the contingencies actually occur.

These electrical applications make it possible to calculate

the optimal flows in the network in order to minimize, for example, the active losses which could mean substantial economical savings. Similarly, the optimal production schedules with regard to the best use of resources (nuclear power, oil, gas, water) can be calculated and applications for automatic control of generating units according to the approved schedules are available.

Process models and applications can also be used to achieve realistic operator training environments with instructor consoles and operators to be trained in normal and disturbed process situations.

VI. Practical Implication of VIKING

Already now, in the middle of the VIKING project, it is possible to see potential, practical implementations as a result of the research work. Some of these implications will be direct results of VIKING project as deliverables and others could be spinoffs that could explored by the academic or industrial partners. In the context of this paper we will only indicate some rough ideas which have been discussed within the project. We are convinced that many more possible results of VIKING will emerge during the continued project work.

The generic SCADA architecture that will be developed in the VIKING project could be applied to individual commercial SCADA system. Such an analysis could be performed in a specialized tool which would lead analyzers of a certain SCADA system by asking specific questions, e.g. have you applied a DeMilitarized Zone in your configuration or do you use encryption in your Inter-Center Communication and many more. The tool would, as an end results, give a quantitative value of the cyber security of the analyzed system. It would be possible to use such a tool for testing specific actions to improve security, e.g. how much would the IT security improve by adding encryption for parts of the RTU traffic or introducing an Intrusion Detection System.

Another possible application of the VIKING research work would be to use the existing advanced application for an application level Intrusion Detection System. The SCADA systems are in the somewhat unique position to have model based applications that know how the process should behave. This knowledge can be used to detect manipulated process data or make the effort to fool the system with false data much higher.

We have also looked into the possibility to use multiple communication paths to the RTUs and Substation Automation Systems. Such additional paths exist in almost all real installations because the communication network structure follows the meshed structure of the electrical network. Traffic could be divided in such way that it would be impossible to reconstruct a telegram or to enter false telegrams by accessing the traffic on a single communication line. Such a solution could make the management of encryption keys unnecessary which is, in itself, a risk in geographically wide spread processes.

The industrial partners ABB, E.ON and Astron have joined the VIKING project with the intention of improving and extending their product and service offerings or, as in the case of E.ON, to make their existing and future installations of SCADA systems more secure. E.ON is already today very active in the area of SCADA system security and hopes to further improve their knowledge base by an active participation in the VIKING project.

Acknowledgment

This VIKING project has been partly funded by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 225643. The author would simply like to thank all the other members of the projects and other associated persons that have actively contributed to the ongoing work of VIKING.

References

[1] A. Giani, S. Sastry, K.H. Johansson and H. Sandberg
The VIKING Project: An Initiative on Resilient Control of Power Networks
[2] F. Wu, K. Moslehi, A, Bose and H. Sandberg
Power System Control Centers: Past, Present and Future
Proceeding of the IEEE, Vol. 93, No 11, November 2005
[3] Oildom Publishing Company of Texas, Inc, Pipeline and Gas Journal, February 2006
[4] E Byres, D Leversage, N Kube; Security incidents and trends in SCADA and process industries. [Online] The Industrial Ethernet Book: http://ethernet.industria-networking.com/articles/articledisplay.asp?id=1823
[5] http://www.vikingproject.eu
[6] http://cordis.europa.eu/fp7/ict/security/projects_en.html
[7] http://cordis.europa.eu/fp7/ict/security/home_en.html
[8] http://www.abb.com
[9] http://www.eon.com
[10] http://www.mml.se
[11] http://www.astron.hu

Detecting False Data Injection Attacks on DC State Estimation

Rakesh B. Bobba, Katherine M. Rogers, Qiyan Wang, Himanshu Khurana, Klara Nahrstedt and Thomas J. Overbye
University of Illinois, Urbana-Champaign
Email: {rbobba, krogers6, qwang26, hkhurana, klara, overbye}@illinois.edu

***Abstract*—State estimation is an important power system application that is used to estimate the state of the power transmission networks using (usually) a redundant set of sensor measurements and network topology information. Many power system applications such as contingency analysis rely on the output of the state estimator. Until recently it was assumed that the techniques used to detect and identify bad sensor measurements in state estimation can also thwart malicious sensor measurement modification. However, recent work by Yao *et al.* [1] demonstrated that an adversary, armed with the knowledge of network configuration, can inject false data into state estimation that uses DC power flow models without being detected. In this work, we explore the detection of false data injection attacks of [1] by protecting a strategically selected set of sensor measurements and by having a way to independently verify or measure the values of a strategically selected set of state variables. Specifically, we show that it is necessary and sufficient to protect a set of *basic measurements* to detect such attacks.**

I. Introduction

The power grid is a complex system of interconnected networks each of which consists of electric power generators and power consumers (loads) connected by transmission and distribution lines. To ensure safe and reliable operation of the power grid, each of the interconnected networks is continuously monitored and controlled by a control center[1] using an industrial control system known as Supervisory Control and Data Acquisition (SCADA) system. SCADA system collects measurements from sensors in the network, every 2 to 4 seconds. These sensor measurements are fed into a State Estimator which, as the name indicates, estimates the state of the power network based on the sensor measurements. Local grid operators use this estimate of the current state to take corrective control actions if necessary and to plan for any contingencies (*e.g.*, loss of a transmission line or generator). Thus state estimation plays an important role in the reliable operation of the power grid.

The power grid, being critical infrastructure, is an attractive attack target. Adversaries may attempt to manipulate sensor measurements, insert fake control commands, delay measurements and/or control commands, and resort to other malicious actions. Therefore, it is crucial to protect power system applications against such malicious activity to ensure safe and reliable operation of the power grid. Until recently, it was generally assumed that the techniques used to detect, identify and correct [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16] bad sensor measurements in state estimation are sufficient to detect and recover from sensor measurement manipulation. However, recent work by Yao *et al.* [1] demonstrated that an adversary, armed with the knowledge of the network configurations, can inject false data into state estimation that uses DC power flow models without being detected.

One of the key reasons behind such attack demonstrations is that current bad data detection techniques were designed to deal with errors and not coordinated malicious activity. Therefore, there is a need to develop advanced defense strategies for protecting state estimation and other power system applications. The aim of this work is to take the first step in this direction and develop defense strategies for protecting DC state estimation against the false data injection attacks proposed in [1]. While the work of Yao *et al.* [1] presented the false data injection attacks from an adversary's point of view and showed what it takes for an adversary to launch a successful attack, we look at the problem from the power grid operator's point of view and ask what it takes to defend against such attacks. Intuitively, there are two approaches to protecting control applications such as state estimation. The first is to design robust control algorithms that can detect or tolerate malicious data modification. The second is to protect the sensor measurements and other data from being manipulated. These two approaches are not necessarily mutually exclusive but can complement each other.

The first approach of handling malicious data injection at the application layer might mean reduced application efficiency with higher development costs. Furthermore, changing algorithms that the grid operators are used to and have gained significant experience with is not lightly done. More often than not, new algorithms are first introduced as research and development prototypes and are not commissioned for production use until the operators gain some experience and get comfortable with using the new algorithm. Thus, the second approach of fundamentally thwarting sensor data manipulation at the lower layer is the only alternative until the new algorithm is accepted for production use. This second approach can mean simpler power applications and higher performance. However, it may not be feasible to protect all sensor measurements,

[1]In order to ensure reliability of the interconnected networks as a whole, designated entities known as reliability coordinators monitor the network over a wide region and provide oversight and reliability coordination between control centers.

either due to budgetary constraints or the legacy nature of the measurement device and its communications. In this work, we explore bringing application awareness to the second approach in order to reduce the burden of protecting all sensor measurements.

Specifically, we investigate whether it is possible to significantly reduce the risk of undetectable false data injection attacks of [1] against DC state estimation by 1) protecting a carefully chosen subset of the sensor measurements, and 2) having ways to independently verify or measure the values of a carefully chosen set of state variables, both for a given network topology. The intuition behind this approach is that for a given topology some sensor measurements influence more state variables than others and hence might provide better cost to benefit ratio when protected. Similarly, some state variables are dependent on more sensor measurements than others and hence independently verifying their estimate might limit the attackers ability in manipulating sensor measurements without being detected.

Such an analysis is very useful for state estimation, and any deployed control algorithm in general, as it allows grid operators to make informed decisions regarding how to invest their protection budget. Even when a grid operator is willing and has the resources to protect all sensor measurements and upgrade their state estimation algorithm, this change will not be effected overnight. Thus, our investigation is useful in prioritizing which sensor measurements to protect first from a security point of view. Besides, this approach is general in nature in the sense that it can be combined with solutions that handle malicious data at the upper layers.

Our results show that our defense strategy is very effective in thwarting undetectable false data injection attacks of [1] on DC state estimation. Our main contribution in this context is showing that protecting a set of *basic measurements* is a necessary and sufficient condition for detecting false data injection attacks of [1] on DC state estimation. A set of *basic measurements* is composed of the minimum number of measurements needed to ensure *observability* of the power network, *i.e.*, to ensure that the state variables can be estimated using the measurements. For DC state estimation, the size of a set of basic measurements is equal to the number of state variables, n, that need to be estimated, while the number of measurements, m, is often larger than that, *i.e.* $m > n$. For example, for the IEEE 300-bus test system, the number of measurements is 1122 while the number of state variables to be estimated is 299[2]. The additional measurements provide redundancy and are useful for traditional bad sensor measurement detection and identification methods mentioned above.

The rest of this paper is organized as follows. In Section II, we provide a brief background on state estimation and associated bad data detection mechanisms and the false data injection attacks proposed in [1]. We motivate and present our general approach in Section III. We discuss approaches to identifying a set of sensors and state variables to protect and then present our results in Section IV. We discuss practical issues, limitations and future directions in V.

[2]Here we are excluding the slack bus angle

TABLE I: Notations

m	The number of measurements
n	The number of state variables
$\mathbf{H}$	$m \times n$ Jacobian matrix representing the topology
$\mathbf{x}$	$n \times 1$ vector of state variables
$\mathbf{z}$	$m \times 1$ vector of measurements
$\mathbf{e}$	$m \times 1$ vector of measurement errors, s.t., $\mathbf{z} = \mathbf{Hx} + \mathbf{e}$
$\hat{\mathbf{x}}$	$n \times 1$ vector of estimated state variables
$\mathbf{W}$	$m \times m$ diagonal matrix, s.t., $w_{i,i} = \sigma_i^{-2}$, where σ_i^2 is the variance of the i-th measurement ($1 \leq i \leq m$)
τ	Threshold for the L_2-norm based detection of bad measurements
$\mathbf{z_a}$	$m \times 1$ measurement vector with bad measurements
$\mathbf{a}$	$m \times 1$ attack vector, s.t., $\mathbf{z_a} = \mathbf{z} + \mathbf{a}$
$\mathbf{c}$	$n \times 1$ vector of estimation errors introduced due to $\mathbf{a}$
$\mathcal{M}$	The set of sensor or measurement indices
$\mathcal{V}$	The set of state variable indices
$\mathcal{I}_{\tilde{m}}$	The set of indices of protected sensor measurements
$\mathcal{I}_{\tilde{v}}$	The set of indices of independently verified state variables
$\mathcal{I}_m$	The set of indices of potentially manipulated measurements
$\mathcal{I}_v$	The set of indices of potentially manipulated state variables
p	The number of protected measurements, i.e., $\lvert\mathcal{I}_{\tilde{m}}\rvert$
q	The number of independently verified state variables, i.e., $\lvert\mathcal{I}_{\tilde{v}}\rvert$

II. Background

In this section, we briefly discuss DC state estimation including bad data detection and the false data injection attacks proposed in [1]. Table I shows the notation used.

A. DC State Estimation [17]

Here, we present a common formulation of the state estimation problem when using a DC power flow model.

$$\mathbf{z} = \mathbf{Hx} + \mathbf{e} \tag{1}$$

In (1), $\mathbf{x} = (x_i, x_2, \ldots, x_n)^T$ represents the true states of the system that are to be estimated, $\mathbf{z} = (z_i, z_2, \ldots, z_m)^T$ represents the sensor measurements, $\mathbf{H}$ is an $m \times n$ Jacobian matrix where $\mathbf{Hx}$ is a vector of m linear functions linking measurement to states, and $\mathbf{e} = (e_i, e_2, \ldots, e_m)^T$ represents random errors in measurement.

The power network is considered *observable* if there are enough measurements to make state estimation possible. There are many sensor placement algorithms that can identify the set of sensor measurements that ensure observability of a power network [17]. Typically, there are more sensors in the power network than those needed for observability, *i.e.* $m > n$. The minimum set of measurements needed to estimate the n state variables is commonly referred to as a set of *basic measurements* or *essential measurements*. The remaining set of measurements are referred to as *redundant measurements*. The redundant measurements are useful in identifying bad sensor measurements [17]. Note that for DC state estimation, any set of n measurements whose corresponding rows in $\mathbf{H}$ are linearly independent are sufficient to solve for the n state variables and hence constitute a set of *basic measurements*. In other words, n independent linear equations are sufficient

to solve for n variables. When m is greater than n, as is the typical case, state estimation involves solving an over-determined system of linear equations. It can be solved as a weighted least squares problem to arrive at the following estimator:

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

where $\mathbf{W}$ is a diagonal matrix whose elements are the measurement weights. It is common to base $\mathbf{W}$ on the reciprocals of the variance of measurement error. As pointed out in [1], as long as the sensor measurement error is assumed to be normally distributed with zero mean, other commonly used estimation criteria, namely, maximum likelihood criterion and minimum variance criterion also lead to the estimator in (2).

1) Bad Measurement Detection: Sensor measurements used for state estimation might be inaccurate because of device misconfiguration, device failures, malicious actions or other errors and can adversely affect the estimate of state variables. Thus, it is extremely valuable for power system operators to detect the presence of bad measurements and identify them. Many schemes for detecting, identifying and correcting bad measurements have been proposed [18], [17].

A common approach [18], [17] for detecting the presence of bad data is by looking at L_2-norm of measurement residual which is defined as follows:

$$||\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}|| \tag{3}$$

In, equation (3), $\hat{\mathbf{x}}$ is the state estimate and $\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}$ is the measurement residual, which is the difference between the vector of observed measurements and estimated measurements. Intuitively, when observed measurements, $\mathbf{z}$, contain bad data, the $L_2 - norm$ of the measurement residual will be high. Thus if the value of expression in (3) is greater than a certain threshold τ it is assumed that bad data is present. Assuming that all state variables are mutually independent and that the sensor errors follow a normal distribution, it can be shown that $(\|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\|)^2$ follows a *chi-squared distribution* with $\nu = m - n$ degrees of freedom [18]. Threshold value τ can then be determined through a hypothesis test with a significance level α.

B. False Data Injection Attacks [1]

False data injection attacks on state estimation [1] are those in which an attacker[3] manipulates the sensor measurements to induce an arbitrary change in the estimated value of state variables without being detected by the bad measurement detection algorithm of the state estimator. In [1], Yao *et al.* present false data injection attacks that can bypass the bad measurement detection algorithm described in Section II-A1. Here, we summarize the basic attack principle, attack scenarios and goals from [1].

[3]We use the terms attacker and adversary interchangeably throughout this paper

1) Attack Principle: Let $\mathbf{a} = (a_1, a_2, \ldots, a_m)^T$ be an attack vector representing the malicious data added to the original measurement vector $\mathbf{z} = (z_1, z_2, \ldots, z_m)^T$. Let $\mathbf{z_a} = \mathbf{z} + \mathbf{a}$ represent the resulting modified measurement vector. Let $\hat{\mathbf{x}}_\mathbf{bad}$ and $\hat{\mathbf{x}}$ represent the estimates of $\mathbf{x}$ when using the manipulated measurements $\mathbf{z_a}$ and original measurements $\mathbf{z}$ respectively. Then $\hat{\mathbf{x}}_\mathbf{bad}$ can be represented as $\hat{\mathbf{x}} + \mathbf{c}$, where $\mathbf{c}$ is the estimation error introduced by the attacker.

Theorem 1 in [1] shows that if the adversary chooses the attack vector, $\mathbf{a}$, to be equal to $\mathbf{Hc}$, then resulting manipulated measurement $\mathbf{z_a} = \mathbf{z} + \mathbf{a}$ can pass the bad measurement detection algorithm described in Section II-A1 as long as the original measurement $\mathbf{z}$ can pass it. To see this, consider the L_2-norm of the measurement residual with manipulated data

$$\begin{aligned} \|\mathbf{z_a} - \mathbf{H}\hat{\mathbf{x}}_\mathbf{bad}\| &= \|\mathbf{z} + \mathbf{a} - \mathbf{H}(\hat{\mathbf{x}} + \mathbf{c})\| \\ &= \|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}} + (\mathbf{a} - \mathbf{Hc})\| \quad (4) \\ &= \|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}\| \quad (5) \\ &\quad \text{when} \quad \mathbf{a} = \mathbf{Hc} \quad (6) \end{aligned}$$

2) Attack Scenarios and Goals or Adversary Model: It is assumed that the adversary has access to $\mathbf{H}$ which is determined by the power network topology and line impedances. It is also assumed that the adversary has the capability to manipulate sensors measurements, either by compromising the sensor or the communication between the sensor and the control center. However, this capability of the attacker is constrained as follows:

- **Scenario I:** The attacker is restricted to accessing only specific sensors. This takes into account the possibility that some sensors may be protected or beyond the reach of the attacker for other reasons.
- **Scenario II:** The attacker has limited resources to compromise sensors. That is, the attacker can compromise any sensor but is restricted to compromising only a limited number, say k, out of all sensors.

For both of the above scenarios, two attack goals are considered, namely, *random false data injection* and *targeted false data injection*. In *random false data injection*, the adversary aims to find any attack vector that injects arbitrary errors into the estimates of state variables. In *targeted false data injection*, the adversary aims to find an attack vector that injects specific errors into the estimates of specific state variables chosen by him. For *targeted false data injection*, two cases are considered: *constrained* and *unconstrained*. In the *constrained* case, the adversary aims to find an attack vector that injects specific errors into the estimates of specific state variables but does not pollute the estimates of other state variables. This case represents situations where the control center may have independent ways to verify the estimates of certain state variables, and to avoid detection, the adversary does not want to pollute them. In the *unconstrained* case, the adversary has no such concerns regarding polluting other state variables.

Methods to identify attack vectors for both of the above described attack goals and in each of the above described attack scenarios, as well as the effectiveness of those methods on the IEEE 9-bus, 14-bus, 30-bus, 118-bus and 300-bus test systems, are presented in [1]. We refer the readers to [1] for details.

III. Motivation and Approach

While the work of Yao *et al.* [1] presented the false data injection attacks from an adversary's point of view and showed what it takes for an adversary to launch a successful attack, we look at the problem from the power grid operator's point of view and ask what it takes to defend such attacks. An obvious approach is to protect all sensor measurements from being manipulated. However, this is not always feasible, and in this work we explore the feasibility of detecting false data injection attacks without having to protect measurements from all sensors. Specifically, for a given $\mathbf{H}$, we aim to identify a set of sensors and a set of state variables such that, when the measurements from the sensors in the chosen set are protected and when the values of state variables from the chosen set can be verified independently, then an adversary cannot find attack vectors that can inject false data without being detected. Furthermore, we would like to identify the smallest of such sets.

The existence of a set of sensors such that, when the measurements from those sensors are protected, an adversary cannot inject false data without being detected is evident from the results in [1]. For a given integer k, Figure 2 in [1] shows the estimated success probability of an attacker in injecting false data without being detected when he picks k measurements at random to manipulate. This success probability was estimated using multiple trials of picking k measurements at random to manipulate. If the success probability of an attacker is less than 1 for a given k, it implies that there exist sets of $m-k$ measurements such that when they are protected an attacker cannot inject false data without being detected. For example, an adversary has to compromise close to 80% and 50% of the total measurements, for the IEEE 9-bus and 300-bus systems respectively, before his success probability approaches 1. That means there exist sets consisting of more than 20% and 50% of the sensors for the IEEE 9-bus and 300-bus systems respectively, such that when the measurements from those sensors are protected against compromise an adversary cannot inject false data without being detected.

However, it is useful to identify the smallest set of sensors that need to be protected for detecting false data injection attacks. According to Theorem 2 in [1], if an attacker can compromise k sensor measurements, where $k \geq m-n+1$, there always exist attack vectors that can inject false data without being detected even when the attacker has no control over which k sensors he can compromise. This provides a lower bound on the number of sensors that need to be protected. That is, a necessary condition for detecting false data injection is protecting at least n sensors. However, it does not seem to be a sufficient condition. Results in [1] show that protecting any n out of m sensors doesn't guarantee detection of false data injection, and that sometimes more than n sensors need to be protected. For example, consider the IEEE 300-bus test system where there are $m = 1122$ measurements and $n = 299$ state variables[4]. For this system, according to Theorem 2 of [1], if the adversary compromises any $m-n+1 = 824$ measurements, then he can always find an attack vector for *random false data injection* (without being detected). But, as mentioned above, experimental results presented in Figure 2 of [1] show that an attacker is able to find an attack vector with probability 1, *i.e.*, found an attack vector in all the trials, when manipulating about 50% of measurements, *i.e.* 561 measurements, picked at random. Thus, it seems the grid operator is forced to protect more than 561 measurements to detect false data injection, and even then, if the set of protected measurements is not carefully chosen, the attacker may still succeed in injecting false data without being detected.

While selectively protecting a little more than 50% of the total measurements is more cost-effective than having to protect all sensor measurements, we explore the possibility of further reducing this burden by leveraging the operators ability to independently verify the values of a few chosen state variables. Intuitively, the ability to independently verify the value of a state variable provides some measure of indirect protection for the sensor measurements that most influence the value of the state variable. One way to independently verify the value of state variables is through the deployment of Phasor Measurement Units (PMUs). PMUs can directly measure both the magnitude and phase angles of currents and voltage at a bus and the measurements are GPS timestamped. There are already about 200 networked PMUs deployed in North America and another 800+ are slated to be deployed with support from Department of Energy (DOE) Smart Grid Investment Grants. It might be better to use the measurements from these PMU devices as an independent way to verify the value of a state variable and potentially save on the cost of protecting measurements from multiple legacy sensors.

A. Adversary Model

We assume that the adversary has access to the topology matrix $\mathbf{H}$ which is determined by the power network topology and line impedances. We also assume that the adversary has the capability to manipulate sensors measurements, either by compromising the sensors or the communication between the sensors and the control center. However, the attacker is restricted to compromising the measurements from only specific sensors denoted by the set $\mathcal{I}_m$. This takes into account the fact that the remaining measurements are protected by the grid operator. Furthermore, as discussed above, we assume that the grid operator can independently verify the values of a few chosen state variables, denoted by the set $\mathcal{I}_{\bar{v}}$, and that

[4]Based on the topology matrix $\mathbf{H}$ of the IEEE 300-bus test system obtained from MATPOWER [19], a MATLAB package for solving power flow equations. All the topology matrices used in this work are obtained from MATPOWER package.

the adversary, in order to avoid detection, is constrained not to inject false data into those variables.

B. Detecting False Data Injection

Let $\mathcal{M}$ denote the set of measurement indices. Let $\mathcal{I}_{\bar{m}} = \mathcal{M} \setminus \mathcal{I}_m$ denote the set of indices of measurements that are protected by the grid operator. Let $\mathcal{V}$ denote the set of state variables indices. Let $\mathcal{I}_v = \mathcal{V} \setminus \mathcal{I}_{\bar{v}}$ denote the set of indices of state variables that the attacker may inject false data into.

Since the measurements of sensors in $\mathcal{I}_{\bar{m}}$ cannot be manipulated by the attacker, the corresponding elements a_i for $i \in \mathcal{I}_{\bar{m}}$ in the attack vector $\mathbf{a} = (a_1, a_2, \ldots, a_m)^T$ are zero. Similarly, if $\mathbf{c} = (c_1, c_2, \ldots, a_n)^T$ represents the estimation error introduced by the attack vector $\mathbf{a}$, then c_j for $j \in \mathcal{I}_{\bar{v}}$ are also zero. Thus, to launch a false data injection attack without being detected, the attacker needs to find an attack vector $\mathbf{a} = (a_1, a_2, \ldots, a_m)^T$ such that it satisfies the following three conditions:

$$\mathbf{a} = \mathbf{Hc} \tag{7}$$

$$a_i = 0 \qquad \text{for} \quad i \in \mathcal{I}_{\bar{m}} \tag{8}$$

$$c_j = 0 \qquad \text{for} \quad j \in \mathcal{I}_{\bar{v}} \tag{9}$$

On the other hand, from the grid operator's perspective, in order to ensure that false data injection attacks are always detected, the grid operator needs to identify a set of sensors, $\mathcal{I}_{\bar{m}}$, and a set of state variables, $\mathcal{I}_{\bar{v}}$, such that an adversary *cannot* find an attack vector that satisfies the above three conditions. Ideally, the operator should find the smallest such sets. How to select the set of sensors, $\mathcal{I}_{\bar{m}}$, and the set of state variables, $\mathcal{I}_{\bar{v}}$, is described in the following section.

IV. Identifying Optimal $\mathcal{I}_{\bar{m}}$ and $\mathcal{I}_{\bar{v}}$

A. Approach I: Brute-Force Search

In our first attempt at identifying optimal $\mathcal{I}_{\bar{m}}$ and $\mathcal{I}_{\bar{v}}$, we tried a straight forward brute-force approach. Let $p = |\mathcal{I}_{\bar{m}}|$ and $q = |\mathcal{I}_{\bar{v}}|$. The grid operator can pick at random a fixed q out of n state variables to populate $\mathcal{I}_{\bar{v}}$ and a fixed p out of m sensors to populate $\mathcal{I}_{\bar{m}}$, and check if any attack vectors that satisfy the above three conditions exist for this choice, as follows.

Let $\mathbf{H} = (\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n)$, where $\mathbf{h}_i$ denotes the ith column vectors of $\mathbf{H}$. Set $\mathbf{H}_s = (\mathbf{h}_{j_1}, \mathbf{h}_{j_2}, \ldots, \mathbf{h}_{j_{n-q}})$ and $c_s = (c_{j_1}, c_{j_2}, \ldots, c_{j_{n-q}})^T$ where $j_i \notin \mathcal{I}_{\bar{v}}$ for $1 \le i \le n-q$. Let $\mathbf{P}_s = \mathbf{H}_s(\mathbf{H}_s^T\mathbf{H}_s)^{-1}\mathbf{H}_s^T$ and $\mathbf{B}_s = \mathbf{P}_s - \mathbf{I}$. Then,

$$\begin{aligned}
\mathbf{a} = \mathbf{Hc} \Leftrightarrow \mathbf{a} &= \sum_{i \in \mathcal{I}_v} \mathbf{h}_i c_i + \sum_{j \in \mathcal{I}_{\bar{v}}} \mathbf{h}_j c_j = \mathbf{H}_s \mathbf{c}_s \\
&\text{since} \quad c_j = 0 \text{ for } j \in \mathcal{I}_{\bar{v}} \\
&\Leftrightarrow \mathbf{P}_s\mathbf{a} = \mathbf{P}_s\mathbf{H}_s\mathbf{c}_s \\
&\Leftrightarrow \mathbf{P}_s\mathbf{a} = \mathbf{H}_s\mathbf{c}_s = \mathbf{a} \\
&\Leftrightarrow \mathbf{P}_s\mathbf{a} - \mathbf{a} = 0 \Leftrightarrow (\mathbf{P}_s - I)\mathbf{a} = 0 \\
&\Leftrightarrow \mathbf{B}_s\mathbf{a} = 0
\end{aligned} \tag{10}$$

This means an attack vector $\mathbf{a}$ satisfies (10) if and only if it satisfies conditions (7) and (9). Now to take into account condition (8), let $\mathbf{B}_s = (\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m)$, where $\mathbf{b}_i$ $(1 \le i \le m)$ denote the column vectors of $\mathbf{B}_s$. Set $\mathbf{B'}_s = (\mathbf{b}_{i_1}, \mathbf{b}_{i_2}, \ldots, \mathbf{b}_{i_{m-p}})$ and $\mathbf{a'} = (a_{i_1}, a_{i_2}, \ldots, a_{i_{m-p}})^T$, where $i_r \notin \mathcal{I}_{\bar{v}}$ for $1 \le r \le m-p$. Then,

$$\begin{aligned}
\mathbf{B}_s\mathbf{a} = 0 \Leftrightarrow \sum_{i \in \mathcal{I}_m} \mathbf{b}_i a_i + \sum_{j \in \mathcal{I}_{\bar{m}}} \mathbf{b}_j a_j = 0 \\
\Leftrightarrow \mathbf{B'}_s\mathbf{a'} = 0 \\
\text{since} \quad a_j = 0 \text{ for } j \in \mathcal{I}_{\bar{m}}
\end{aligned} \tag{11}$$

Thus, for a given topology matrix $\mathbf{H}$ and sets $\mathcal{I}_{\bar{m}}$ and $\mathcal{I}_{\bar{v}}$, to find an attack vector that can inject false data without being detected, an attacker needs to (1) compute $\mathbf{a'}$ that satisfies (11), and (2) set $\mathbf{a} = (0, \ldots, 0, a_{i_1}, 0, \ldots, 0, a_{i_2}, 0, \ldots, 0, a_{i_{m-p}}, 0, \ldots, 0,)^T$, where a_{i_r} occupy the appropriate places denoted by i_r for $1 \le r \le m-p$. Note that $\mathbf{B'}_s$ is a $m \times (m-p)$ matrix. If the rank of $\mathbf{B'}_s$ is $m-p$ then equation (11) has no non-zero solutions and thus no error can be injected into state estimation without being detected, but if rank of $\mathbf{B'}_s$ is less than $m-p$ then an infinite number of solutions exist. The operator thus needs to find the smallest possible sets of $\mathcal{I}_{\bar{m}}$ and $\mathcal{I}_{\bar{v}}$ such that the rank of $\mathbf{B'}_s$ is $m-p$ in order to be able to detect false data injection attacks of [1].

Such a brute-force approach to identifying $\mathcal{I}_{\bar{m}}$ and $\mathcal{I}_{\bar{v}}$ needs to search through $\binom{m}{p} * \binom{n}{q}$ combinations for a given choice of p and q, where $0 \le p \le m$ and $0 \le q \le n$. Thus the potential search space for finding the smallest possible sets is quite large. However, in practice an operator may not have an independent way to verify the estimated value of a state variable for more than 10% of the total state variables, *i.e.*, $q \le \frac{n}{10}$. Similarly, with a way to verify the estimated value of some state variables, it is most likely that one could detect false data injection attacks by protecting less than half the sensors, *i.e.*, $p \le \frac{n}{2}$. Furthermore, the lower bound of n on the number of sensor measurements that need to be protected (when there are no verifiable state variables), as indicated by Theorem 2 of [1], provides a good starting point around which to search for a solution.

We implemented this approach using Matlab and analyzed the IEEE 9-bus system. The results are summarized in Table II. For the IEEE 9-bus system, there are 8 state variables and 27 measurements, *i.e.* $m = 27$ and $n = 8$. Thus, according to the Theorem 2 of [1], when an adversary is allowed to compromise more than or equal to $m - n + 1 = 20$ sensors he can always find successful attack vectors that inject false data without being detected. As seen in Table II, when only 7 sensors are protected, *i.e.* 20 sensors are allowed to be compromised, and no state variables are verifiable, there are no defensible configurations. That is, when $\mathcal{I}_{\bar{v}}$ is null, there exists no set, $\mathcal{I}_{\bar{m}}$, of size 7 that can prevent an adversary from finding a successful undetectable attack vector. However, when 8 sensors are protected, there are 329245 or 14% of

TABLE II: Number of protected sensors and verifiable state variables needed to detect false data injection attacks for IEEE 9-bus system

Number of Protected Sensors	Number of Verifiable State Variables	Number of Defensible Configurations *i.e.* those that can detect attacks	Percentage of Defensible Configurations
7	0	0	0
8	0	329245	14%
9	0	1991771	35%
6	1	0	0
7	1	18954135	75%
6	2	12288444	62%

the total combinations of 8 sensors, that provide defensible configurations. That is, an adversary cannot inject false data without being detected when one of the 329245 possible sets of 8 sensors is selected as $\mathcal{I}_{\bar{m}}$ even when $\mathcal{I}_{\bar{v}}$ is null.

When one state variable is verifiable, then defensible configurations can be found even when only 7 sensors are protected, and it turns out that 75% of all the possible combinations (*i.e.* $\binom{27}{7} * \binom{8}{1}$) are defensible configurations. However, protecting any less than 7 sensors when there is only one verifiable state variable yields no defensible configurations. Thus, for the IEEE 9-bus system we do not seem to be gaining much in terms of reduction in the number of sensors to be protected by having a way to verify state variables. However, the number of defensible configurations increases considerably compared with the case where there are no verifiable state variables. This provides a lot of flexibility to the operator in terms of the set of sensors he can choose to protect.

While this approach was tractable for IEEE 9-bus system, the search space got very large for the IEEE 14-bus system even with small p and q. When we ran a parallelized version, using Matlab Parallel Computing Toolbox, of our algorithm with $p = 12$ and $q = 1$ on an Intel Xeon dual processor quad-core 64-bit machine, the analysis did not complete even after two days. It is worth noting that, given a $m \times m$ matrix $\mathbf{B}_s$ as in equation (10), and p, the number of sensors to be protected, the problem of identifying the set of protected sensors of size less than or equal to p such that an adversary cannot inject false data without being detected is NP-hard. To see this, let $\mathcal{U} = \{u_i\}$ denote the set of matrices such that (1) each u_i is a sub-matrix of $\mathbf{B}_s$ and contains no more than $m - p$ columns of $\mathbf{B}_s$, and (2) if u_i contains x columns, then $rank(u_i) \leq x - 1$. To find the set of measurements to be protected, we need to find a sub-matrix h of $\mathbf{B}_s$ such that, (1) it has no more than p columns, and (2) for each $u_i \in \mathcal{U}$, $h \cap u_i \neq \emptyset$, where $\cap$ between two sub-matrices returns the common columns in them. Clearly, this problem is reducible to the *hitting set problem* which is NP-complete.

B. Approach II: Protecting Basic Measurements

While existing approximate algorithms for the *hitting set problem* could have been leveraged to analyze larger IEEE test systems using the approach in the preceding section, we wanted to find a more intuitive solution. The alternate approach described below provides such a solution and leverages the concept of *basic measurements*.

For a given $\mathcal{I}_{\bar{m}}$ set of protected sensors, let $\mathbf{H}' = (\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n)^T$, where $\mathbf{r}_i$ $(1 \leq i \leq m)$ are the row vectors of $\mathbf{H}'$, denote a Jacobian matrix obtained by re-arranging the rows of $\mathbf{H}$ such that the rows corresponding to the sensors in $\mathcal{I}_{\bar{m}}$ appear as the first p rows of $\mathbf{H}'$. Thus, $\mathcal{I}'_{\bar{m}}$ corresponding to $\mathbf{H}'$ is simply $\{1, 2, \ldots, p\}$. Then, equation (7) can be written as

$$\begin{bmatrix} \mathbf{a}'_p \\ \mathbf{a}'_k \end{bmatrix} = \begin{bmatrix} \mathbf{H}'_{pn} \\ \mathbf{H}'_{kn} \end{bmatrix} \begin{bmatrix} \mathbf{c}' \end{bmatrix} \tag{12}$$

In equation (12), $\mathbf{a}'$ and $\mathbf{c}'$ are appropriately re-arranged versions of $\mathbf{a}$ and $\mathbf{c}$ with $\mathbf{a}'_p$, and $\mathbf{a}'_p$ being column vectors of length p and $k = m - p$ respectively; $\mathbf{H}'_{pn}$ is a $p \times n$ matrix, and $\mathbf{H}'_{kn}$ is a $k \times n$ matrix. Taking equation (8) into account ($\mathbf{a}'_p$ becomes a zero vector) and splitting equation (12) into two matrix equations we arrive at the following:

$$\mathbf{0} = \mathbf{H}'_{pn}\mathbf{c}' \tag{13}$$

$$\mathbf{a}'_k = \mathbf{H}'_{kn}\mathbf{c}' \tag{14}$$

Let us for now assume that there are no verifiable state variables, *i.e.* $\mathcal{I}_{\bar{v}} = \emptyset$. Then, for an undetected false data injection attack to be possible, there must be a $\mathbf{c}'$ in the null space of $\mathbf{H}'_{pn}$ such that $\mathbf{a}'_k = \mathbf{H}'_{\mathbf{kn}}\mathbf{c}'$ is satisfied. Conversely, no attacks are possible if $\mathbf{H}'_{pn}$ has full column rank, *i.e.*, $rank(\mathbf{H}'_{pn}) = n$. Since the rank of a $m \times n$ matrix is always less than or equal to $min(m, n)$, $rank(\mathbf{H}'_{pn})$ can be equal to n only if $p \geq n$. However, $p \geq n$ does not guarantee the detection of attacks since the $rank(\mathbf{H}'_{pn})$ may still be less than n. This result is captured in the following corollary.

Corollary 4.1: It is necessary but not sufficient to protect at least n measurements in order to be able to detect false data injection attacks.

The above result is in line with Theorem 2 and experimental observations of [1]. In order for the rank of $\mathbf{H}'_{pn}$ to be equal to n, at least n rows of $\mathbf{H}'_{pn}$ should be linearly independent vectors. That is, $\mathbf{H}'_{pn}$, should contain rows corresponding to at least one set of what are referred to as *basic measurements* (refer to Section II-A). A set of *basic measurements* in state estimation is a minimum set of measurements which is

sufficient to ensure *observability* (refer to Section II-A). The following theorem states this result.

Theorem 4.1: When there are no verifiable state variables, it is necessary and sufficient to protect a set of **basic measurements** *in order to be able to detect false data injection attacks.*

For DC state estimation, the size of the set of basic measurements is equal to the number of state variables which is n. The remaining $m-n$ measurements provide redundancy and help with bad measurement identification. Note that the choice of a set of basic measurements is not unique. The existence of multiple sets of basic measurements is obvious since if there are two independent measurements of a state variable, it does not matter which is taken to be the basic measurement and which is taken to be redundant. Thus, the optimal number of sensor measurements to protect in order to detect false data injection attacks is n.

Theorem 4.1 seems to contradict the findings in Figure 2 of [1] and as such needs some clarification. As discussed in Section III, Figure 2 in [1] shows the estimated success probability of an attacker in injecting false data without being detected when he picks k measurements to manipulate at random. Figure 2 of [1] shows that, the probability of success of an adversary is 1, *i.e.*, always able to find an attack vector, even when about 561 (> 299) and 171 (> 117) measurements are protected in IEEE 300-bus and 118-bus test systems respectively. This apparent contradiction is due to the fact that the success probability shown was an estimated value, estimated using multiple trials of picking k measurements at random to manipulate. We observe that the discrepancy is very stark only for IEEE 300-bus and 118-bus systems and not for the other systems, namely IEEE 9-bus, 14-bus and 30-bus, that were also analyzed in [1]. We also note that only 100 trials were used in estimating success probabilities for the IEEE 300-bus and 118-bus systems, in order to reduce simulation time, as opposed to using 1000 trials as was done for the other smaller bus systems. Given the large search space for the IEEE 300-bus and 118-bus systems and the small number of trials used, it is very likely that the set of protected measurements picked by the simulations in the small number of trials did not contain a set of basic measurements. In fact, for IEEE 9-bus, 14-bus, 30-bus, 118-bus and 300-bus test systems, we picked a set of basic measurements (using the method outlined in Section IV-B1) and verified that there are no attack vectors, *i.e.*, the rank of $\mathbf{B'}_s$ in equation (11) is 0. Thus, the probability of success of an attacker cannot be 1 when the number of protected measurements is greater than n.

Now suppose we also have verifiable state variables. Without loss of generality, let us say we have only one verifiable state variable and it is the jth state variable. Taking equation (9) into account, equations (13) and (14) can be written as

$$\mathbf{0} = \mathbf{H''}_{pn'}\mathbf{c''} \tag{15}$$

$$\mathbf{a'}_k = \mathbf{H''}_{kn'}\mathbf{c''} \tag{16}$$

where $\mathbf{H''}_{pn'}$ and $\mathbf{H''}_{kn'}$ are derived from $\mathbf{H'}_{pn}$ and $\mathbf{H'}_{kn}$ respectively by removing the jth column, $n' = n-1$ and $\mathbf{c''}$ is a column vector of length n' derived from $\mathbf{c'}$ by removing the jth element. If $\mathbf{H'}_{pn}$ was a full column rank matrix, *i.e.*, rank n matrix, then $\mathbf{H''}_{pn'}$ will also be a full column rank matrix, *i.e.*, rank $n-1$ matrix, and thus no undetectable false data injection attacks are possible. Since $p \geq n$, it is possible to remove a few measurements from the protected measurements without compromising attack detectability as long as (1) p', the size of the resulting set of protected measurements, is equal to $n-1$ and (2) the resulting set of measurements are sufficient to ensure observability of the $n-1$ state variables (*i.e.*, excluding the jth state variable).

Corollary 4.2: If there are q verifiable state variables it is necessary and sufficient to protect a set of basic measurements *corresponding to the remaining $n-q$ state variables in order to be able to detect false data injection attacks.*

Thus, while a protected basic measurement may be replaced by a verifiable state variable, it is clear that the minimum required number of protected or verifiable quantities is equal to n, *i.e.*, the number of state variables.

1) Determining the Protected Set: The importance of protecting a set of basic measurements has been made clear. We now discuss how a defender of the system can identify such a set of measurements. Much work has been done to determine sensor placement for observability of a power network, *i.e.*, to determine a set of basic measurements, including [20], [21], [22], [23], [17], [24]. Another straight forward but brute-force approach is to pick a set of n measurements out of m at random and see if the rows corresponding to them in $\mathbf{H}$ are linearly independent. In this work however, we leverage a more computationally efficient approach described and justified in [16] and [25] respectively. In this approach, the measurements in the system are mapped to a new equivalent state space where identification of basic and redundant measurements is easily accomplished. This approach is briefly described below.

To obtain the equivalent states, an LU decomposition is performed on $\mathbf{H}$,

$$\widetilde{\mathbf{H}} = \mathbf{P} \cdot \mathbf{H} = \mathbf{L_{AA}} \cdot \mathbf{U_b} \tag{17}$$

where $\mathbf{P}$ is a row permutation matrix which maps the original rows of $\mathbf{H}$ to the new rows of $\widetilde{\mathbf{H}}$. The new basis is given by

$$\mathbf{L'_{AA}} = \begin{bmatrix} \mathbf{I_n} \\ \mathbf{R} \end{bmatrix} \tag{18}$$

where $\mathbf{I_n}$ is the $n \times n$ identity matrix. Rows of $\mathbf{I_n}$ correspond to the n basic measurements, and rows of $\mathbf{R}$ correspond to redundant measurements. Columns correspond to equivalent states. We compute the LU decomposition of $\mathbf{H}$ and use $\mathbf{P}$ to map the first n measurements in the new basis back to the original measurements. This gives us one set of basic measurements.

Other basic measurement sets may be derived after the first one is found. Furthermore, the matrix $\mathbf{L'_{AA}}$ can tell us which measurements we may switch out. As an example, consider this $\mathbf{L'_{AA}}$ from [16]:

$$\mathbf{L'_{AA}} = \begin{matrix} p_2 \\ p_3 \\ p_{24} \\ p_{12} \\ p_{34} \\ p_{23} \end{matrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0.5 & 0.5 & 0 \\ 0 & -0.5 & 0.5 & 0 \end{pmatrix}$$

The basic measurements are (p_2, p_3, p_{24}, p_{12}), but p_3 and p_{24} could be replaced by either p_{34} or p_{23}. Thus, another basic measurement set is (p_2, p_{34}, p_{23}, p_{12}). The key is that the rows switched out of the redundant measurement set must be linearly independent, otherwise they could not both be made into basic measurements. Following this line of reasoning, in an incremental manner, we can switch a basic measurement with one of its redundant measurements in $\mathbf{H}$ and recompute $\mathbf{L'_{AA}}$ to obtain a new set of basic measurements. Note that the LU decomposition can be computed quickly, even for large systems. This is beneficial, especially when compared to a brute-force search of the entire space for meters and state variables to protect. We implemented this approach of identifying a basic set of measurements using Matlab, and identified multiple sets of basic measurements for the the IEEE 9-bus, 14-bus, 30-bus, 118-bus and 300-bus test systems. We verified that no attack vectors exist when these sets of measurements are protected using the approach described in Section IV-A.

In summary, there are many choices of sets of minimal size n which may be protected. Ultimately, the choice comes down to the interests of the defenders or owners of the system. The owners may have particular measurements that they would like to include in a basic measurement set if possible, or they may already know which particular state variables will be made verifiable. Their selection process may proceed as follows: (1) Determine a satisfactory set of basic measurements for the system using the approach described above. This is the candidate set of protected items. (2) Decide which state variables will be made verifiable. Add these state variables to the candidate set, and optionally remove an equal number of protected measurements.

V. Discussion and Future Directions

Protecting Sensor Measurements: So far, we have focused on identifying a set of sensors whose measurements need to be protected in order to detect the false data injection attacks of [1]. Here, we discuss what it means to protect sensor measurements in this context. Clearly, we want the measurements from the sensors be authentic. That means manipulation of the sensor measurements either by 1) physically tampering with the device or 2) by tampering with the communication between the sensor and control center needs to be prevented. That is, sensors should be protected from unauthorized access (both physical and remote access), and measurements from the sensor should be authenticated and integrity protected. However, this may not be sufficient if one would like to only protect measurements in the smallest required set. This is because the operator may not be able to detect false data injection attacks when a measurement from one of the protected sensors is unavailable. Thus, it is also essential that the measurements from the protected sensors be available at all times. This latter requirement may be relaxed a bit by protecting a few more strategically selected sensors than the smallest set necessary. Identification of this larger set of sensors and studying the associated trade-offs are left to be addressed in a future work.

Considering Topology Changes: In this work, we focus on identifying measurements and state variables to protect, for a given $\mathbf{H}$, but do not consider how topology changes such as line outages would affect these decisions. In reality, the defender of the system needs to deploy a protected set so that in the event of any expected topology change, false data injection attacks are still detectable.

If any line l is opened, and that line has a protected measurement, the measurement is no longer valid. Thus, the corresponding protected measurement row in $\mathbf{H'}_{pn}$ needs to be replaced. Otherwise, the resulting $\mathbf{H'}_{pn}$ is not full rank, and attacks are possible. The measurement m that replaces the lost measurement must have a row in $\mathbf{H}$ which is linearly independent with the remaining protected measurement rows in $\mathbf{H'}_{pn}$, so that when it is added, the resulting $\mathbf{H'}_{pn}$ is again full rank. In this case, the set of measurements required for the system to be protected before and after a line outage of l is of size $(n+1)$. Each considered line outage will thus increase the size of the required protected set by at least one. Suppose that there is a basic measurement set $\{1, 2, 3, 4, 5\}$ and another basic measurement set which is $\{2, 3, 4, 5, 6\}$. If measurement 1 is lost, it can be replaced by measurement 6, and the system will be protected. For the system to be protected both with and without line 1, measurements $\{1, 2, 3, 4, 5, 6\}$ must all be protected. As before, one basic measurement may be substituted for one verifiable state variable. There are different ways [24], [26] to identify a set of measurements such that full observability is maintained with most common/frequent topology changes.

Generic False Data Injection Attacks: So far in this work, we have focused on strategies to detect false data injection attacks proposed in [1]. The basic principle behind such attacks, as discussed in Section II-B1, is that when the adversary sets his attack vector $\mathbf{a}$ to be equal to $\mathbf{Hc}$, then the bad measurement detection algorithm described in Section II-A1 fails to detect attacks. However, this is not the only way to inject false data without being detected. To see this, consider the equation (4). It is clear from equation (4) that, even if $\mathbf{a} \neq \mathbf{Hc}$, as long as the adversary chooses his attack vector, $\mathbf{a}$, such that the following equation (19) is true then the attacker could still inject false data without being detected.

$$\|\mathbf{z} - \mathbf{H}\hat{\mathbf{x}} + (\mathbf{a} - \mathbf{Hc})\| \leq \tau \tag{19}$$

However, in this case, apart from knowing $\mathbf{H}$, the adversary has to know $\mathbf{z}$, *i.e.* values of all sensor measurements, and $\hat{\mathbf{x}}$. The adversary can compute $\hat{\mathbf{x}}$ using equation (2) but then

needs to know $\mathbf{W}$. Thus, this form of false data injection attack imposes higher burden on the adversary than the one described in [1]. Furthermore, it may be possible to protect against these attacks by protecting the confidentiality of sensor measurements, *i.e.* preventing the adversary from knowing $\mathbf{z}$. Thus, by incorporating the requirement of confidentiality into our definition of "sensor measurement protection" discussed above, we might be able to detect generic false data injection attacks. A detailed analysis of such attacks and defense strategies will be the subject of future work.

ACKNOWLEDGEMENTS

The authors would like to thank Yao Liu for sharing with us the Matlab code used in the analysis of [1]. The authors would also like to thank William H. Sanders, Tim Yardley and the anonymous reviewers for their helpful comments. This material is based upon work supported by the National Science Foundation under Grant No. CNS-0524695. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

REFERENCES

[1] Y. Liu, M. K. Reiter, and P. Ning, "False data injection attacks against state estimation in electric power grids," in *CCS '09: Proceedings of the 16th ACM conference on Computer and communications security*. New York, NY, USA: ACM, 2009, pp. 21–32.

[2] H. Merrill and F. Schweppe, "Bad data suppression in power system static state estimation," *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-90, no. 6, pp. 2718–2725, Nov. 1971.

[3] E. Handschin, F. Schweppe, J. Kohlas, and A. Fiechter, "Bad data analysis for power system state estimation," *Power Apparatus and Systems, IEEE Transactions on*, vol. 94, no. 2, pp. 329–337, Mar 1975.

[4] D. Falcao, P. Cooke, and A. Brameller, "Power system tracking state estimation and bad data processing," *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-101, no. 2, pp. 325–333, Feb. 1982.

[5] W. Kotiuga and M. Vidyasagar, "Bad data rejection properties of weughted least absolute value techniques applied to static state estimation," *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-101, no. 4, pp. 844–853, April 1982.

[6] X. Nian-de, W. Shi-ying, and Y. Er-keng, "A new approach for detection and identification of multiple bad data in power system state estimation," *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-101, no. 2, pp. 454–462, Feb. 1982.

[7] A. Monticelli and A. Garcia, "Reliable bad data processing for real-time state estimation," *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-102, no. 5, pp. 1126–1139, May 1983.

[8] T. Van Cutsem, M. Ribbens-Pavella, and L. Mili, "Hypothesis testing identification: A new method for bad data analysis in power system state estimation," *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-103, no. 11, pp. 3239–3252, Nov. 1984.

[9] X. N. de, W. Shi-ying, and Y. Ers-keng, "An application of estimation-identification approach of multiple bad data in power system state estimation," *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-103, no. 2, pp. 225–233, Feb. 1984.

[10] W. Peterson and A. Girgis, "Multiple bad data detection in power system state estimation using linear programming," in *System Theory, 1988., Proceedings of the Twentieth Southeastern Symposium on*, Mar 1988, pp. 405–409.

[11] F. Wu, W.-H. Liu, and S.-M. Lun, "Observability analysis and bad data processing for state estimation with equality constraints," *Power Systems, IEEE Transactions on*, vol. 3, no. 2, pp. 541–548, May 1988.

[12] I. Slutsker, "Bad data identification in power system state estimation based on measurement compensation and linear residual calculation," *Power Systems, IEEE Transactions on*, vol. 4, no. 1, pp. 53–60, Feb 1989.

[13] A. Abur, "A bad data identification method for linear programming state estimation," *Power Systems, IEEE Transactions on*, vol. 5, no. 3, pp. 894–901, Aug 1990.

[14] B. Zhang, S. Wang, and N. Xiang, "A linear recursive bad data identification method with real-time application to power system state estimation," *Power Systems, IEEE Transactions on*, vol. 7, no. 3, pp. 1378–1385, Aug 1992.

[15] E. Asada, A. Garcia, and R. Romero, "Identifying multiple interacting bad data in power system state estimation," in *Power Engineering Society General Meeting, 2005. IEEE*, June 2005, pp. 571–577 Vol. 1.

[16] J. Chen and A. Abur, "Placement of pmus to enable bad data detection in state estimation," *Power Systems, IEEE Transactions on*, vol. 21, no. 4, pp. 1608–1615, Nov. 2006.

[17] A. Monticelli, *State estimation in electric power systems: a generalized approach*. Kluwer Academic Publishers, 1999.

[18] A. Wood and B. Wollenberg, *Power Generation, Operation, and Control*, 2nd ed. John Wiley and Sons, 1996.

[19] R. Zimmerman, C. Murillo-Sanchez, and R. Thomas, "Matpower's extensible optimal power flow architecture," july 2009, pp. 1 –7.

[20] V. Quintana, A. Simoes-Costa, and A. Mandel, "Power system topological observability using a direct graph-theoretic approach," *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-101, no. 3, pp. 617 –626, march 1982.

[21] E. Fetzer and P. Anderson, "Observability in the state estimation of power systems," *Power Apparatus and Systems, IEEE Transactions on*, vol. 94, no. 6, pp. 1981 – 1988, nov. 1975.

[22] G. Krumpholz, K. Clements, and P. Davis, "Power system observability: A practical algorithm using network topology," *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-99, no. 4, pp. 1534 –1542, july 1980.

[23] A. Monticelli and F. Wu, "Network observability: Identification of observable islands and measurement placement," *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-104, no. 5, pp. 1035 –1041, may 1985.

[24] A. Abur and F. H. Magnago, "Optimal meter placement for maintaining observability during single branch outages," *IEEE Transactions on Power Systems*, vol. 14, no. 4, pp. 1273 – 1278, 1999, least absolute value estimations;. [Online]. Available: http://dx.doi.org/10.1109/59.801884

[25] J. London, L. Alberto, and N. Bretas, "Analysis of measurement-set qualitative characteristics for state-estimation purposes," *Generation, Transmission Distribution, IET*, vol. 1, no. 1, pp. 39 –45, January 2007.

[26] A. Abur and F. Magnago, "Optimal meter placement against contingencies," vol. 1, no. SUMMER, Vancouver, BC, Canada, 2001, pp. 424 – 428, network observability;Optimal meter placement;. [Online]. Available: http://dx.doi.org/10.1109/PESS.2001.970061

Simulation of Network Attacks on SCADA Systems

Rohan Chabukswar*, Bruno Sinópoli*, Gabor Karsai†, Annarita Giani‡, Himanshu Neema† and Andrew Davis†
*Carnegie Mellon University
†Vanderbilt University
‡University of California Berkeley

***Abstract*—Network security is a major issue affecting SCADA systems designed and deployed in the last decade. Simulation of network attacks on a SCADA system presents certain challenges, since even a simple SCADA system is composed of models in several domains and simulation environments. Here we demonstrate the use of C2WindTunnel to simulate a plant and its controller, and the Ethernet network that connects them, in different simulation environments. We also simulate DDOS-like attacks on a few of the routers to observe and analyze the effects of a network attack on such a system.**

I. Introduction

Supervisory Control And Data Acquisition (SCADA) systems are computer-based monitoring tools that are used to manage and control critical infrastructure functions in real time, like gas utilities, power plants, chemical plants, traffic control systems, etc. A typical SCADA system consists of a SCADA Master which provides overall monitoring and control for the system, local process controllers called Remote Terminal Units (RTUs), sensors and actuators and a network which provides the communication between the Master and the RTUs.

A. Security of SCADA Systems

SCADA systems are designed to have long life spans, usually in decades. The SCADA systems currently installed and used were designed at a time when security issues were not paramount, which is not the case today. Furthermore, SCADA systems are now connected to the Internet for remote monitoring and control making the systems susceptible to network security problems which arise through a connection to a public network.

Despite these evident security risks, SCADA systems are cumbersome to upgrade for several reasons. Firstly, adding security features often implies a large downtime, which is not desirable in systems like power plants and traffic control. Secondly, SCADA devices with embedded codes would need to be completely replaced to add new security protocols. Lastly, the networks used in a SCADA system are usually customized for that system and cannot be generalized.

Security of legacy SCADA systems and design of future systems both thus rely heavily on the assessment and rectification of security vulnerabilities of SCADA implementations in realistic settings.

B. Simulation of SCADA Systems

In a SCADA system it is essential to model and simulate communication networks in order to study mission critical situations such as network failures or attacks. Even a simple SCADA system is composed of several units in various domains like dynamic systems, networks and physical environments, and each of these units can be modeled using a variety of available simulators and/or emulators. An example system could include simulating controller and plant dynamics in Simulink or Matlab, network architecture and behavior in a network simulator like OMNeT++, etc. An adequate simulation of such a system necessitates the use of an underlying software infrastructure that connects and relates the heterogeneous simulators in a logically and temporally coherent framework.

II. C2WindTunnel

One infrastructure suitable for such an application is the Command and Control WindTunnel ([3]). The C2WindTunnel is an integrated, graphical, multi-model simulation environment for the experimental evaluation of congruence between organizational and technical architectures in large-scale C2 systems. It enables various simulation engines to interact and transmit data to and from one another and log and analyze the real time simulation results.

The C2WindTunnel framework uses the discrete event model of computation as the common semantic framework for the precise integration of an extensible range of simulation engines. Each simulation model, when incorporated into the overall simulation environment of C2WindTunnel, requires integration on two levels: the API level and the interaction level. API level integration provides basic services such as message passing, and shared object management, whereas interaction level integration addresses the issues of synchronization and coordination. C2WindTunnel offers a solution for multi-model simulation by decomposing the problem into model integration and experiment integration tasks. It facilitates the rapid development of integration models and use of these models throughout the lifecycle of the simulated environment. An integration model defines all interactions between federated models and captures other design intent, such as simulation engine-specific parameters and deployment information. This information is leveraged to streamline and automate significant portions of the simulation lifecycle. The integration modeling language combined with various sophisticated generation tools provides a robust environment for users to rapidly design and synthesize complex, heterogeneous command and control simulations.

A. High Level Architecture (HLA)

C2WindTunnel is based on the High-Level Architecture (HLA) IEEE standard 1.3 ([2], [4] and [5]) initially designed by Department of Defense (DoD) to ensure interoperability and reusability of models and simulation components. Reusability implies individual simulation models can be employed in different simulation scenarios, while interoperability implies an ability to incorporate simulations on different types of distributed computing platforms, with real-time operation.

A complex simulation can be considered as a hierarchy of components with increasing levels of aggregation. At the lowest level is the model of a system component implemented in software to produce a simulation, referred to as a federate. Several such federates form a part of an HLA compliant simulation, called a federation. There are three components of an HLA:

1) HLA rules to ensure proper interaction among federates and to delineate the respective responsibilities.
2) Object Model Template (OMT) to prescribe format and syntax for recording and communicating information.
3) Interface specification to define Run Time Infrastructure services and interfaces and federate callback functions.

B. Run-Time Infrastructure (RTI)

The software implementation of HLA is called a Run-Time Infrastructure (RTI). There are several commercial and open-source RTIs available in the market, some of which have been verified by the US Defense Modeling and Simulation Office.

The RTI is basically a collection of software that provides a set of commonly required services, described by the HLA Interface Specification, to multiple simulation systems. Apart from federation and object management, the RTI handles time management and co-ordinates the exchange of interacting events and data among the federates in a system.

1) Time Management: The time management services provided by the RTI ensure advancement of the simulation time in an orderly fashion among all the federates. Initially, the federate manager uses HLA-specified synchronization points to guarantee that all federates are ready to proceed with the simulation. Only when each federate has reported readiness to proceed with the simulation, does the federate manager allow all federates to commence the simulation.

2) Event and Data Interaction: A publish and subscribe mechanism is used by the HLA to manage the distribution of messages between the federates in a federation. Each federate defines to the federation what data is to be published for each update or event. Each federate declares to the federation which updates and interactions it is interested in receiving by subscribing to those messages.

C. C2WindTunnel Modeling Environment

C2Wind Tunnel uses a custom developed domain-specific modeling language (DSML) for the definition of integration models and the design details for the simulation environment. A simulation environment is composed of multiple federates each of which includes a simulation model, the engine upon which it executes, and some amount of specialized glue code to integrate the engine with the simulation bus. Both the engine configuration and the integration code needed for each federate is highly dependent upon the role the federate plays in the environment as well as the type of simulation engine being utilized. While manually developing the glue code is possible, by leveraging the integration model, C2WindTunnel is able to synthesize all of the code, greatly reducing errors and effort. A suite of tools called model interpreters, integrated directly with the DSML automatically generates engine configurations, glue code, as well as scripts to automate simulation execution and data collection. The integration model DSML combined with the generation tools provides a robust environment for users to rapidly define complex, heterogeneous command and control simulations.

The Generic Modeling Environment is the foundation for the C2WindTunnel environment. GME is a meta-programmable model-integrated computing toolkit that supports the creation of rich domain-specific modeling and program synthesis environments. Configuration is accomplished through meta models, expressed as UML class diagrams, specifying the modeling paradigm of the application domain. Meta models characterize the abstract syntax of the domain-specific modeling language, defining which objects are permissible in the language and how they compose. The meta model is a schema or data model for all of the possible models that can be expressed by a language.

Figure 1 shows the structure of a simulation undertaken using C2WindTunnel

III. Simulation

In this section, we demonstrate the simulation of network security attacks on a SCADA system simulated using C2WindTunnel.

The SCADA system chosen was a simplified version of the famous Tennessee Eastman Control Challenge Problem, proposed by N. Lawrence Ricker ([1]). The original challenge problem requires co-ordination of three unit operations, with 41 measured output variables (with added measurement noise) and 12 manipulated variables. The control challenge presented by this case study is quite complex, but a simplified version was proposed by Ricker in 1993.

The process schematic is shown in Figure 2. It consists of an isothermal fixed volume reactor with a combined separation system, in which a single irreversible reaction occurs:

$$A + C \rightarrow D.$$

The reactants A and C are non-condensible, and the product D is a non-volatile liquid. The reaction rate depends only on the partial pressures of A and C. There are two controlled feeds to the reactor chamber. Feed 1 consists of the reactants A and C, and traces of an inert gas B. Feed 2 consists of pure A, which is used to compensate for disturbances in the partial pressures of A and C in feed 1. The solubilities of A, B and C in D are negligible, so the vapor phase can be assumed to consist only of A, B and C, and the liquid, pure D. Thus, the

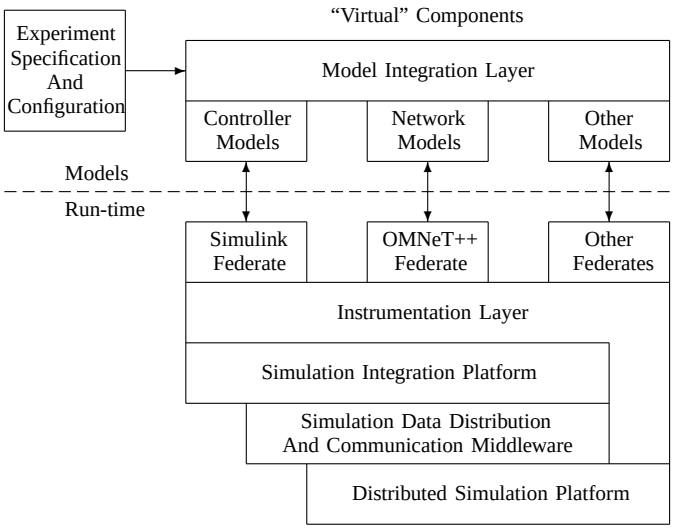


Fig. 1. C2WindTunnel Simulation Architecture

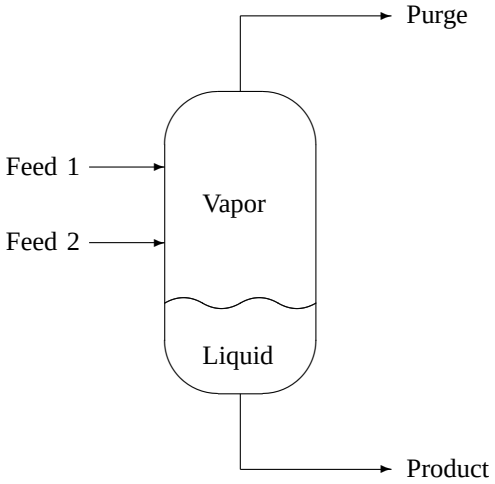


Fig. 2. Process Schematic, taken from [1]

only disturbance variables are the mole fractions of A, B and C in feed 1.

Isothermal conditions are maintained using independent controls. The product flow rate is adjusted using a proportional feedback controller which responds to variations in the liquid inventory. The purge rate depends on the pressure in the vessel and the position of the purge control valve.

Measured outputs include the four flow rates, pressure, liquid holdup volume, and the fractions of A, B and C in the purge flow. This composition measurement offers the pure time delay as in the original TE problem, as the simulated chromatograph operates on a 6 minute cycle.

The control problem is to maintain the product flow rate at a specified value by manipulating flows of feed and purge streams, and the liquid holdup volume. The restrictions come from the physical aspects of the plant: the operating pressure must be kept below the shutdown limit of 3 MPa, and the flow rates saturate at some point. A higher level control objective is to minimize operating costs, which are a function of the purge losses of A and C.

Tables I, II, III and IV list the different variables of the system.

In his paper, Ricker derives a linear time-invariant dynamic model of the plant at the base-case state. The LTI model matches the impulse response of the nonlinear plant well. The robust model-predictive controller is one of several proposed in [1]. It uses only four out of the ten available sensor outputs, and controls all four manipulated variables. The variables which necessarily must be monitored include the production rate (F_4), the pressure (P) and the liquid inventory (V_L). Failure to do so might upset other variables and profitability, or will allow violation of bounds on pressure and liquid holdup volume. The fourth variable chosen is the amount of reactant A in the purge flow (y_{A3}), though amounts of any of the other two components would perform just as well. The final structure of the simplified model as derived in [1] is:

$$\mathbf{y} = \begin{pmatrix} F_4 \\ P \\ y_{A3} \\ V_L \end{pmatrix} = \mathbf{G}\mathbf{u} = \begin{pmatrix} g_{11} & 0 & 0 & g_{14} \\ g_{21} & 0 & g_{23} & 0 \\ 0 & g_{32} & 0 & 0 \\ 0 & 0 & 0 & g_{44} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix}. \quad (1)$$

The individual transfer functions are given below (the unit

TABLE I
STATE VARIABLES (TAKEN FROM [1])

Variable	Nominal Value	Description	Symbol	Units
x_1	44.49999958429348	Molar holdup of A	N_A	kmol
x_2	13.53296996509594	Molar holdup of B	N_B	kmol
x_3	36.64788062995841	Molar holdup of C	N_C	kmol
x_4	110.0	Molar holdup of D	N_D	kmol
x_5	60.95327313484253	Feed 1 valve position	χ_1	%
x_6	25.02232231706676	Feed 2 valve position	χ_2	%
x_7	39.25777017606444	Purge valve position	χ_3	%
x_8	47.03024823457651	Product valve position	χ_4	%

TABLE II
MANIPULATED VARIABLES (TAKEN FROM [1])

Variable	Nominal Value	Purpose	Range
u_1	60.95327313484253	Changes feed 1 valve position	0 – 100%
u_2	25.02232231706676	Changes feed 2 valve position	0 – 100%
u_3	39.25777017606444	Changes purge valve position	0 – 100%
u_4	44.17670682730923	Liquid inventory set point	0 – 100%

TABLE III
OUTPUT VARIABLES (TAKEN FROM [1])

Variable	Nominal Value	Description	Symbol	Units	Range
y_1	201.43	Feed 1 flow measurement	F_1	kmol/hr	0 – 330.46
y_2	5.62	Feed 2 flow measurement	F_2	kmol/hr	0 – 22.46
y_3	7.05	Purge flow measurement	F_3	kmol/hr	Complicated
y_4	100.00	Product flow measurement	F_4	kmol/hr	Complicated
y_5	2700.00	Pressure	P	kPa	< 3000
y_6	44.18	Liquid inventory	V_L	% of maximum	0 – 100
y_7	47.00	Amount of A in purge	y_{A3}	mol %	0 – 100
y_8	14.29	Amount of B in purge	y_{B3}	mol %	0 – 100
y_9	38.71	Amount of C in purge	y_{C3}	mol %	0 – 100
y_{10}	0.2415	Instantaneous cost	C	\$/kmol	> 0

TABLE IV
DISTURBANCE VARIABLES (TAKEN FROM [1])

Variable	Nominal Value	Description	Units
y_{A1}	0.485	Mole fraction of A in feed 1	—
y_{B1}	0.005	Mole fraction of B in feed 1	—

of s is assumed to be hr^{-1}):

$$g_{11} = \frac{1.7}{0.75s+1}, \tag{2}$$

$$g_{21} = \frac{45\,(5.667s+1)}{2.5s^2+10.25s+1}, \tag{3}$$

$$g_{23} = \frac{-15s-11.25}{2.5s^2+10.25s+1}, \tag{4}$$

$$g_{32} = \frac{1.5}{10s+1}e^{-0.1s}, \tag{5}$$

$$g_{14} = \frac{-3.4s}{0.1s^2+1.1s+1}, \tag{6}$$

$$g_{44} = \frac{1}{s+1}. \tag{7}$$

(The transfer function g_{23} is not given in [1], probably due to oversight. It was estimated using the method described in the paper.)

Ricker provides a FORTRAN simulator of the plant model. This was converted to C code and used as a S-Function block to create a Simulink model of the plant.

The plant has a very high time constant, as is characteristic of chemical plants. To properly study the effects of a network attack, we need a system which can respond to disturbances during network attacks of duration in minutes. To this end, the time constants for the plant were multiplied by 60, so that changes which took hours now take the same number of minutes to occur. The transfer functions were then suitably modified to convert the unit of time to seconds. The final

equations used were:

$$g_{11} = \frac{0.02833}{45s+1}, \tag{8}$$

$$g_{21} = \frac{45\,(340s+1)}{9000s^2+615s+1}, \tag{9}$$

$$g_{23} = \frac{-900s-11.25}{9000s^2+615s+1}, \tag{10}$$

$$g_{32} = \frac{1.5}{600s+1}e^{-6s}, \tag{11}$$

$$g_{14} = \frac{-3.4s}{360s^2+66s+1}, \tag{12}$$

$$g_{44} = \frac{1}{60s+1}. \tag{13}$$

Using Matlab, a minimal state space model of the system was constructed, which was then discretized to run at one-hundredth of a second. This is the system assumed in order to implement the controller as a Discrete State Space System block in a separate Simulink model. The A, B, C and D matrices for the controller were then calculated by using a Kalman state estimator and linear quadratic state feedback regulator system.

To complete the system, an Ethernet network was added for communication between the plant and its controller. The network was designed to be a realistic implementation of one in a chemical plant, where a single router would collect data from plant sensors which are physically close to it (and to each other). Similarly, it would send manipulation data to the valves which are physically close to it. There are four routers which route data from different sensors and to different control inputs. There is a three-level hierarchy in the network map, with a master controller distributing data to and collecting data from other routers at the plant site. Two relay routers are employed between the master router at the plant site and the router that communicates to the controller. To keep the network model simple, no redundancy was employed. The network model was simulated in OMNeT++, a generic discrete event simulation package using INET network protocols ([6]). The schematic of the network model is given in Figure 3.

A. Model Integration

The integration of models in different simulation environments has been described in more detail in [3].

1) OMNeT++: NetworkSim, a network simulator based on OMNeT++, provides a set of high-level communication protocols while maintaining full network stack simulation internally. NetworkSim utilizes network models built using OMNeT++. It translates messages from the RTI into appropriate network actions and vice versa, and injects these messages onto the correct simulated network node. This mechanism isolates the simulated network traffic from the general RTI traffic. Each OMNeT++ model deployed onto NetworkSim must have some code synthesized for integration with the RTI. When simulated via NetworkSim, some of the connected nodes in OMNeT++ become end-points, responsible for passing messages between the RTI and the OMNeT++ engine. The code that implements the communication between the RTI and these end nodes must be generated by the integration software.

A GME-based interpreter traverses the C2WT integration model and generates the C++ code needed for end-point nodes within an OMNeT++ model. For each federate, the integration model provides information about which interactions may be sent or received and which objects attributes may be published or updated. The interpreter understands these relationships and synthesizes code for each end-point in an OMNeT++ federate. The generated code builds upon the OMNeT++ API and is compiled directly into NetworkSim. Apart from the glue code, evolution of the OMNeT++ internal simulation clock must also be synchronized with the RTI. As a part of HLA compliance, NetworkSim includes a reusable class that extends the basic OMNeT++ scheduler. The function `getNextEvent()` is called by OMNeT++ to determine the next event, originating either internally or externally. If the timestamp on the next message places it outside of the window of time granted by the RTI, then a time advance is requested. An internal dispatch mechanism routes all RTI interactions to the appropriate OMNeT++ protocol module which interprets them and can schedule new internal OMNeT++ messages. A similar mechanism interprets and routes OMNeT++ messages bound for external dispatch into the RTI. Using these mechanisms both the evolution of time and message passing within an OMNeT++ federate is tightly coordinated via the RTI with the federation.

2) Simulink: Like in the integration of the OMNeT++ simulation engine, all of the engine-specific glue code is generated based on the overarching integration model. The GME-based model interpreter generates code that, in conjunction with several reusable Java generic classes, is used to directly integrate any Simulink model with a C2WT federation. The generic classes provide all of the fundamental RTI integration requirements: providing interfaces for converting between Simulink types and RTI types, encapsulating interfacing with the RTI for initializing the federate, synchronizing the Simulink engine's simulation clock, and managing any publish-and-subscribe relationships with other federates.

Within any given Simulink model the user must insert an S-function block for each interaction to which the model either publishes or subscribes via which blocks that the Simulink engine can interact with the rest of the federation. The modeler specifies whether the block either publishes or subscribes an interaction by instantiating the corresponding sender or receiver S-function from those that were generated from the integration model. The modeler must also specify which interaction the S-function block should call by passing the name of the interaction via a string parameter to the block. The naming convention of the `.m` files and of the parameters is standardized and easily derived from the primary C2WT model.

Once the S-function blocks have been incorporated and their values set, no further manual steps are typically necessary to prepare the model to be integrated. Some effort has to be spent to properly order the signals entering and exiting the S-

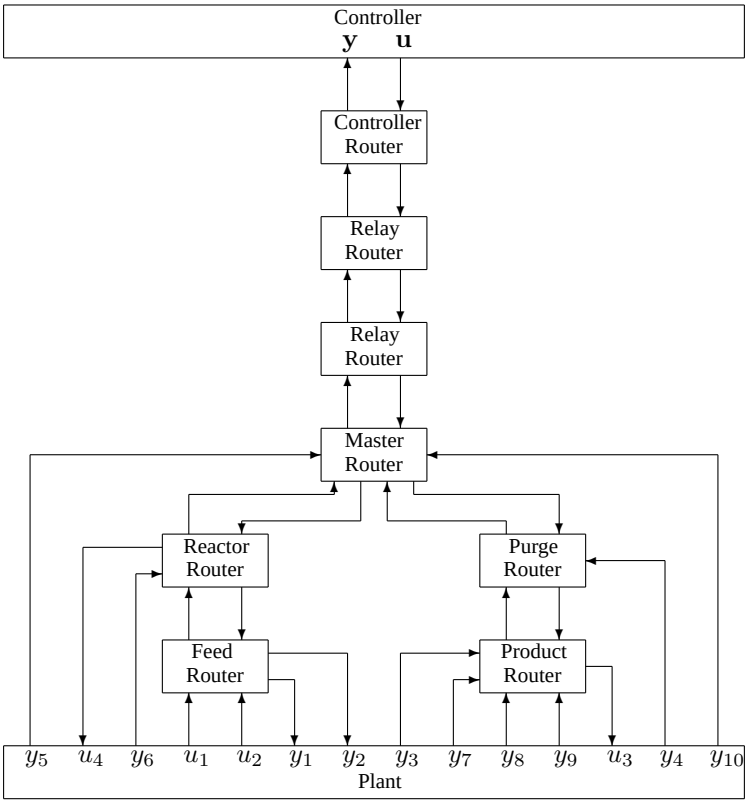


Fig. 3. Network Map

function blocks so that they correspond to the attribute ordering of the corresponding RTI interaction. The key mechanism for synchronizing the clock progression of the Simulink model with that of the RTI is the basic time-progression model for S-function blocks. During its execution, the Simulink engine consults each block in a model about when it can generate an output. With all S-function blocks, code must be supplied to respond to this request from the engine. The synthesized integration code in an S-function block uses this method to synchronize the model with the RTI and allow simulation time within Simulink to progress only when the RTI allows it to proceed. Until the RTI allows federation time to progress, we do not return from the method call within the S-function block, thus not allowing the Simulink engine to progress. We keep the Simulink engine step-size low (typically 0.1 seconds) to minimize any event timing errors due to the passing of input and output events between the Simulink model and the HLA. For incoming events, the glue code uses a polling scheme at every time step to check if the federate has received an input from the remainder of the federation. Very small step-sizes in any Simulink model can lead to a significant slowdown in simulation speed. In the context of the C2WT, possible performance penalties due to having small step-sizes must be weighed against minimizing timing errors due to overly large time-steps.

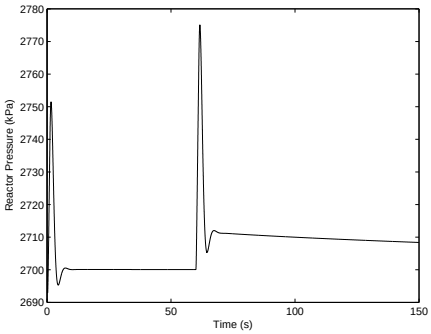
B. Attacks

Several DDOS-like attacks are simulated on the SCADA system, targeting various routers of the network. In each such attack, the target is saturated with external communication requests from a large number of zombie nodes so that it cannot handle the legitimate traffic of the system, or at least, is rendered so slow in handling the traffic, that it is effectively unavailable for transfer of legitimate data. The targets, durations and the number of attacks in the simulation are specified beforehand. In these simulations, the controller, feed and product routers are attacked. In each case, the simulation was run for 150 seconds, and attack started at the 30-second mark and continued till the 60-second mark.

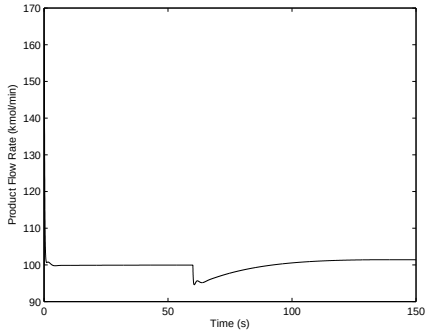
IV. Observations

When one of the routers is under a full-fledged DDOS attack, the network is essentially broken at that point. The controller will be rendered blind to sensors from which the router collects data. The plant will also be rendered unresponsive to such controller commands as are handled by that router. This will result in a loss of the regulatory function of the controller, which can potentially cause a variety of damage to the plant, from an unwanted change in the operating cost and production rate, to physical damage of plant equipment.

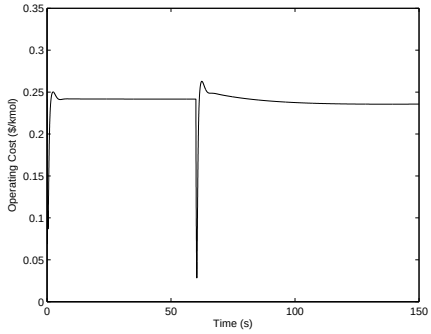
In case the target is one of the routers which handle all of the data (controller, master or relay routers), such an attack causes a complete loss of communication between the plant and



(a) Pressure

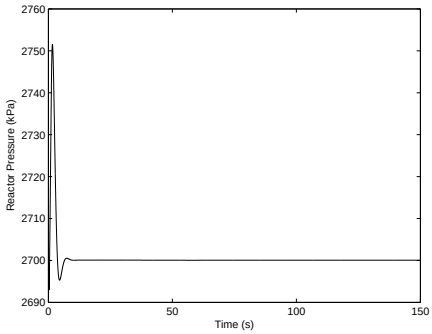


(b) Product Flow Rate

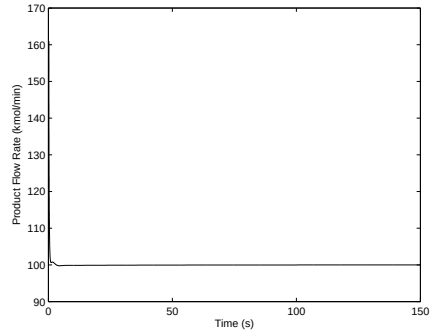


(c) Operating Cost

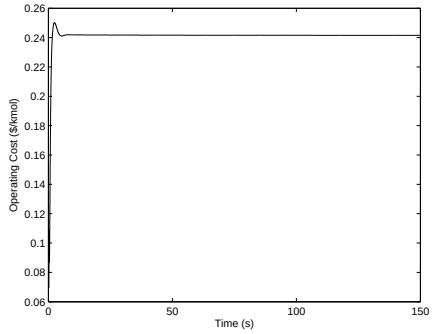
Fig. 4. Attack on Controller Router, from 30s through 60s



(a) Pressure



(b) Product Flow Rate



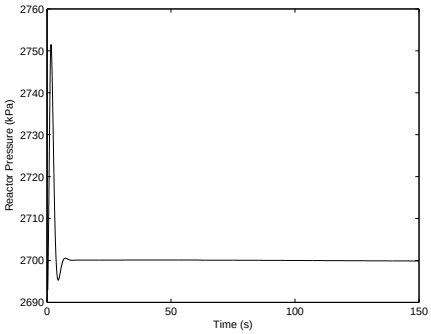
(c) Operating Cost

Fig. 5. Attack on Feed Router, from 30s through 60s

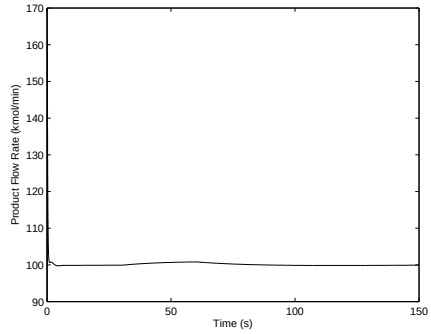
controller. The plant undergoes a severe change of state when the attack begins, from which it recovers and resumes normal operation . Such an attack is the first to be simulated. The effects can be observed in the change in pressure, production rate and operating cost over time, especially during the attack (Figures 4a, 4b and 4c). In case other routers are targeted, the controller will generate outputs based on the sensors outputs it has, and will try to control the inputs which are not unresponsive. These effect can be observed by monitoring the operating cost during the simulation.

The next simulation involves attacking the feed router, which blocks the feed 1 and feed 2 flow measurement sensors (which are not used by the controller), and the valve 1 and valve 2 controllers (u_1 and u_2). The controller is thus not blind to any of the required sensors, but its regulation function could be hampered by it not being able to control the two valves. The effects can be observed by monitoring the same sensor outputs over time (Figures 5a, 5b and 5c). We see that due to the robust nature of the controller, an attack on the feed router as no effect on the state of the plant, which continues to operate normally despite the attack.

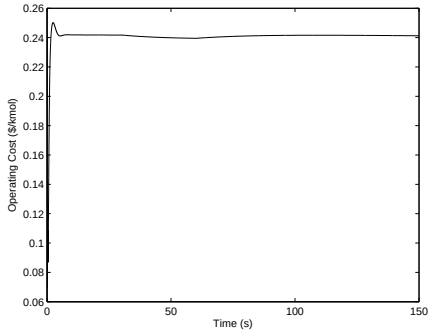
The last simulation involves attacking the product router, which blocks several sensors, the only required one of which being the amount of A in purge (y_7), and the purge valve controller (u_3). The controller is thus blind to one of the required sensors, and it is not able to control the purge valve. The effects of this attack are different than the previous two simulations. The plant goes into an uncontrolled state for



(a) Pressure



(b) Product Flow Rate



(c) Operating Cost

Fig. 6. Attack on Purge Router, from 30s through 60s

the duration of the attack, from which it can recover and resume normal operation only after the attack has ceased. The effects can be observed by observing the usual output variables (Figures 6a, 6b and 6c).

V. Conclusion

A DDOS-like attack was simulated on a plant and controller system and the effects of attacks on different routers were observed. While the effects of the total communication disruption might have been estimated, the effects of the other two attacks are a harder to predict. The same attack on different routers causes no change in one case and severe problems in another. If the system were more complicated, then obtaining the effects would require intensive analytical computations, or indeed, could very well be intractable. In such a case, a simulation is the best way to estimate the effects, and to implement and compare different network configurations and redundancies.

The chemical plant was thus a proof-of-concept implementation of a simulation system composed of models in different domains and environments. The use of C2WindTunnel facilitated the interaction and data transfer between the environments, and in setting up the attacks and monitoring the response.

VI. Future Work

The results of the simulation can be used to analyze the current network and controller, and develop more robust control algorithms and improve the network, for example by using redundancies. The SCADA system itself might be expanded to employ a Fault Detection and Isolation and/or an Intrusion Detection System.

The attack that was simulated is one attack on the availability of a system. Future work involves observing the effect of other common network security attacks on integrity and confidentiality of the data as well, like eavesdropping, misdirection and spoofing.

Another direction for future work involves simulation of systems including hardware-in-the-loop.

Acknowledgment

This work was supported in part by TRUST (Team for Research in Ubiquitous Secure Technology), which receives support from the National Science Foundation (NSF award number CCF-0424422) and the following organizations: AFOSR (#FA9550-06-1-0244), BT, Cisco, DoCoMo USA Labs, EADS, ESCHER, HP, IBM, iCAST, Intel, Microsoft, ORNL, Pirelli, Qualcomm, Sun, Symantec, TCS, Telecom Italia and United Technologies.

References

[1] N. Lawrence Ricker, *Model predictive control of a continuous, nonlinear, two-phase reactor*. Journal of Process Control, Volume 3, Issue 2, May 1993, Pages 109-123.

[2] J. O. Calvin, R. Weatherly, *An introduction to the high level architecture (HLA) runtime infrastructure (RTI)*. Proceedings of the 14th Workshop on Standards for the Interoperability of Defence Simulations, Orlando, FL, March 1996, pp. 705-715.

[3] G. Hemingway, H. Neema, H. Nine, J. Sztipanovits, G. Karsai, *Rapid Synthesis of HLA-Based Heterogeneous Simulation: A Model-Based Integration Approach*. in review for Simulation.

[4] R. Crosbie, J. Zenor, *High Level Architecture*. `http://www.ecst.csuchico.edu/~hla/`.

[5] *HLA standard - IEEE standard for modeling and simulation (M&S) high-level architecture (HLA) — framework and rules*. IEEE Std. 1516-2000, pp.i-22, 2000

[6] *OMNeT++ Simulation Package*. `http://www.omnetpp.org/`

A K/N Attack-Resilient ICT Shield for SCADA Systems, with State Based Attack Detection

I. Nai Fovino, A. Carcano, M. Guglielmi, M. Masera
Institute for the Protection and Security of the Citizen
Joint Research Centre, EU Commission
Via E. Fermi 1, 21027 Ispra, Italy
igor.nai@jrc.ec.europa.eu

***Abstract*—The security of Critical Infrastructures has become a prominent problem with the advent of modern ICT technologies used to improve the performance and the features of Process Control Systems. Several scientific works have showed how Supervisory Control And Data Acquisition Systems (SCADA), i.e. the systems that control industrial installations, are exposed to cyber-attacks. Traditional ICT security countermeasures (e.g. classic Firewalls, Antiviruses and IDS) fail in providing a complete protection to these systems since the needs of SCADA systems are different from those of traditional ICT for which security tools have been developed (Office PCs, TCP/IP communication protocols etc.). In this paper we present an innovative approach to the protection of SCADA systems based on three key concepts: Critical State based event correlation, SCADA protocols filtering and K-survivability.**

Keywords: SCADA systems, security

I. Introduction

Modern *Industrial systems* (e.g. power plants, water plants, smart grids, chemical installation etc.) make large use of ICT technologies. In the last years, those systems started to use public networks (i.e. Internet) for system-to-system interconnection. As a result, thanks to this architectural advance, it has been possible to provide new services and features (implementation of the Energy Market, Energy Smart Grids, remote maintenance and optimization, self orchestrating distributed industrial systems etc.). However, this connectivity has exposed industrial installations to new sources of possible threats. As described by Nai et. all [5] [1], such infrastructures are exposed to ad-hoc created attacks aiming at interfering with, and in some case, taking the control of the process network of the industrial installation. The core of industrial installations is traditionally the "so called" SCADA (System Control and Data Acquisition). SCADA protocols and architectures are dedicated to very specific functions: those useful for controlling the operation of technical systems. Due to their characteristics they can be differentiated from classical ICT devices, protocols and systems. For that reason, at the present time, traditional ICT security technologies are not able to protect industrial installations in an adequate way against ad-hoc SCADA-tailored attacks. We defend that a new set of dedicated ICT security technologies needs to be designed in order to protect industrial critical installations. In this work, we focus our attention on a novel ICT security architecture which put together protocol filtering techniques, signature based verification and intrusion detection. In particular the intrusion detection technique used is the result of our recent studies in this field [13]. The paper is organized as follows: in section II we provide a brief overview of the state of the art in the field of SCADA security, while in section III we provide an overview of the vulnerabilities of SCADA systems that we want to address with the proposed work. In section IV we present the details of our *ICT shield*; in section V after describing our testbed, we present and discuss the experimental results we have obtained when testing an implementation of the proposed approach.

II. Related Works

As claimed in the introduction, only recently the security of SCADA systems assumed an ICT perspective. Adam and Byres [2] presented an interesting high level analysis of the possible threats affecting a power plant system, a categorization of the typical hardware devices involved and some high level discussion about intrinsic vulnerabilities of the common power plant architectures. A more detailed work on the topic of SCADA security, is presented by Chandia, Gonzalez, Kilpatrick, Papa and Shenoi [6]. In this work, the authors describe two possible strategies for securing SCADA networks, underlying that several aspects have to be improved in order to secure that kind of architecture. A relevant part of the vulnerabilities of SCADA systems is due to the specialized communication protocols they use to communicate with the field devices (e.g. Modbus, DNP3, Fieldbus etc.). Some work has been done about the security of such specialized communication protocols: for example, Majdalawieh, Parisi-Presicce and Wijesekera [7] presented an extension of the DNP3 protocol, called DNPsec, which tries to address some of the known security problems of such Master-slave control protocols (i.e. integrity of the commands, authentication, non repudiation etc.). at the same way, the DNP3 User group proposed a "Secure DNP3" implementing authentication mechanisms for certain type of commands and packets. This approach is extremely close to the one adopted in the IEC 62351-5 standard. Nai et al. [3] presented a secure implementation of the Modbus protocol aimed at introducing integrity, authentication and anti-replay mechanisms in the control flows based on the Modbus protocol. Similar approaches have been presented also by Heo, Hong, Ju, Lim, Lee and Hyun [8] while Mander,

Navhani and Cheung [9] presented a proxy filtering solution aiming at identifying and avoiding anomalous control traffic. The proposed solution is extremely interesting, however it does not protect the system against two particular scenarios: (1) The scenario in which an attacker is able to inject malicious packets directly in the network segment between the proxy and the RTUs, and (2) The scenario in which both the proxy and the master have been corrupted and collaborate in order to damage the process network. Finally, Nai et al. in [14], presented an architecture integrating a mesh of distributed packet filtering mechanisms based on signatures, with cryptography based integrity mechanisms. While on the one hand this architecture constitutes a first, significant improvement in the security of control systems, since it introduces the use of specialized firewalls able to analyze the Modbus protocol, on the other hand the filtering feature is only capable to block what we can define as "atomic attacks" constituted by a single not licit packet. The present work extends this architecture, introducing the concepts of Critical State based Filtering analysis and Multilayer Critical State Based and Proactive Monitoring.

III. SCADA VULNERABILITIES OVERVIEW

SCADA systems are widely used in industrial installations to control and maintain field sensors and actuators. The basic bricks of a SCADA system are:

- **Master Terminal Unit (MTU)**: The MTU presents data to the operator, gathers data form the remote PLCs and actuators site, and transmits control signals. It contains the high level logic of the industrial system under control.
- **Remote Terminal Unit (RTU)**: it acts as a slave in the master/slave architecture. Sends control signals to the device under control, acquires data from these devices, receives commands from the MTU and transmits the data gathered to the MTU. An RTU may be a PLC.

The core of the *control flow* of every SCADA system is the communication protocol (e.g. Modbus, Profibus, DNP3 etc.). By using these protocols it is possible, for example, to force the opening of a valve, etc. In this paper we concentrate our attention on two different industrial protocols: *Modbus* and *DNP3*. More specifically, we consider their most recent evolution, i.e. their TCP/IP version. In what follows we provide an overview of the typical vulnerabilities of these two protocols, which are, however, quite similar to the vulnerabilities affecting the other commonly used SCADA protocols (Profibus, Fieldbus etc.). The porting of Modbus and DNP3 over TCP/IP has introduced new layers of complexity for managing the reliable delivery of control packets in an environment with strong real time constraints. In addition, it has opened new possibilities to attackers motivated to cause damages to target industrial systems. In particular, those protocols:

1) Do not apply any mechanism for checking the integrity of the command packets sent by a Master to a Slave and vice-versa.
2) Do not perform any authentication mechanism between Master and Slaves, i.e. everyone could claim to be the "Master" and send commands to the slaves.
3) Do not apply any anti-repudiation or anti-replay mechanisms.

These security shortcomings can be used by malicious users for performing different kind of attacks :

1) **Unauthorized Command Execution:** The lack of authentication between Master and Slave can be used by attackers to forge packets which can be directly sent to a pool of slaves.
2) **SCADA-DOS:** On the basis of the same principle, an attacker can also forge meaningless Modbus/DNP3 packets, always impersonating the Master, and consume the resources of the RTU
3) **Man-in-the-Middle attacks:** The lack of integrity checks allows attackers to access the production network for implementing typical Man-in-the-Middle (MITM) attacks, modifying the legal packets sent by the master.
4) **Replay-Attacks:** The lack of anti-replying mechanisms allows attackers to re-use captured legitimate Modbus/DNP3 packets.

Finally, on top of these classes of attacks, since anti-repudiation mechanisms are not implemented, it is hard to proof the trustworthiness of malicious Masters, which could have been compromised.

IV. A SECURE SURVIVABLE ARCHITECTURE

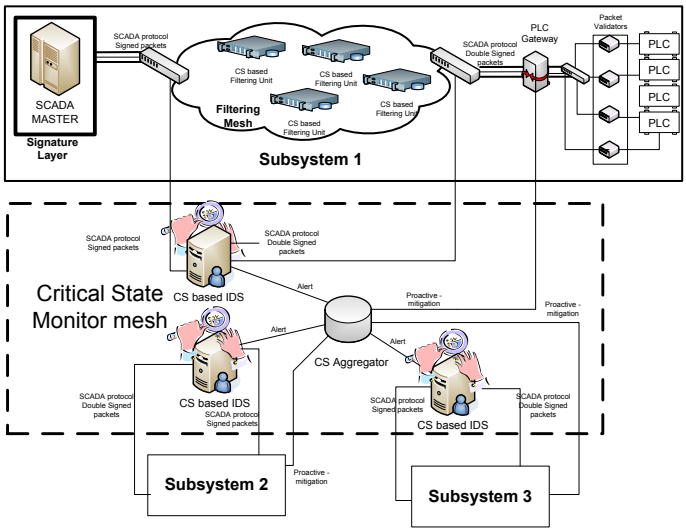


Fig. 1: High level architectural schema

As stressed in the previous section, the weaker point of a SCADA architecture is the communication channel between SCADA servers and PLCs. As showed for example in [1], by taking advantage of those vulnerabilities, a motivated attacker could be able to gain the control of the process network of an industrial installation. To mitigate this threat there are two possibilities: (a) to completely redesign the communication protocols, or (b) to wrap the current architecture into a sort of security shield, limiting as much as possible the impact of this new layer of the performance of the system.

In this work we decided to follow the second option. In the following we briefly describe how the new proposed architecture (showed in figure 1) works, leaving to the next sections the detailed description of each element. When the master needs to send a command to a slave, instead of sending the normal SCADA protocol packet, it builds a new secure SCADA packet, containing a Time Stamp (to avoid replay-attacks) on top of the original SCADA packet, and signed with the private key of the Master. Since, on the other side of the communication channel, the slave will execute the command only if the signature of the master and the time stamp are valid, this mechanism prevents the risk of malicious packets injection by an attacker having access to the process network. However, if an attacker is able to directly corrupt a master, the authentication mechanism we have introduced will be easily circumvented. For that reason, we introduce a "filtering gap" between the Master and the slaves. This gap hosts a set of special filtering units (FU) we have designed and developed on the basis of a new concept of "Critical State based filtering". Differently from the work presented in [14] by Nai et al., here the filtering units operates at two different levels: (1) signature based single packet filtering (to block single malicious packets), (2) Subsystem Level Critical State detection (which allows detecting complex attack patterns based on the use of chains of licit commands). As it is possible to see in figure 1, the filtering gap contains a set of different FU. In our architecture in fact, to mitigate the risk of collusion between a corrupted master and a corrupted FU, we introduce the concepts of system diversity and $\frac{K}{N}$ resilience. Roughly speaking, instead of having a single FU, in the filtering gap we introduce N different FU, where for different we intend that they work on different platforms, operating systems etc. The Master broadcasts the signed SCADA packets to all the FUs, which independently validate the signature of the Master, the time stamp, and then filter the packets according to their internal "single packet" and "Critical State" filters. If the packet passes all the checks, each FUs signs the original packet also with its own private key and forwards the packet to the proper destination. The slave verifies the signature of the FU, verifies the signature of the Master, and executes the command only if at least K/N FUs agree on the safeness of the packet. In this way an attacker, in order to make the system execute a malicious set of packets, has to corrupt the Master and at least K+1 filtering units.

Unfortunately, real industrial installations are composed of several subsystems, with different Masters and several slaves (up to thousands). Each subsystem might be in a safe state, but some of its configurations, as side effect, could put another subsystem into a critical state. In that scenario, the presented approach might fail. For that reason, taking advantage of our recent work in the field of multi-layer Intrusion Detection for indutrial systems [13] [15], we have introduced in our architecture a specialized set of Critical State based IDS, one for each subsystem. They are able to monitor the state of each subsystem, share information and aggregate the local events, in order to predict whether a set of commands, which are safe for a target subsystem, can have a negative impact, as side effect, on another subsystem. If this is the case, the central aggregator, is able to act directly on a dedicated gateway to interrupt the potentially malicious flow.

In what follows we provide some details about the key elements of the proposed architecture.

A. Integrity-Authentication Layer

As underlined in the previous section, one of the most relevant weaknesses of the SCADA protocols is the lack of authentication and integrity mechanisms. Only recently, for example with the introduction of Secure DNP3, there is some improvement in this field. However, solutions such as Secure DNP3 modify the specification of the protocol, posing serious questions about their applicability on existing installations. On the other hand the use of typical ICT tunneling mechanisms (ipsec, security features of IPv6, SSL, TLS) could introduce latencies and interferences in the process flow, which might not be acceptable. For that reason we created a lightweight authentication mechanism which embeds the original SCADA packets into a minimal security envelop, which from one side make use of the classic authentication schema, and on the other uses only the minimum number of security features needed for our purposes. To explain our approach, we present the case of Modbus, but the same principle has also been applied to DNP3.

A SCADA system using Modbus TCP/IP embeds a standard Modbus data frame into a TCP frame, without the Modbus checksum. On top of the standard Modbus packet it is introduced a dedicated 7-byte header called MBAP (Modbus Application Protocol header), containing a transaction identifier, a protocol identifier, the packet length and a unit identifier. In the following we describe how we have introduced a "security layer" in this protocol, taking as entry point the security features we consider relevant to protect the process control flow.

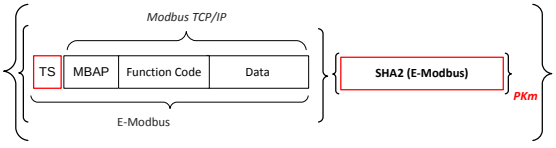


Fig. 2: Secure Modbus Application Data Unit

- **Integrity**: the integrity is guaranteed by using a secure hashing function SHA2. The digest obtained is used to verify the integrity of the packet. For being able to verify it, the digest is sent with the original packet.
- **Authentication**: by adopting a signature scheme (e.g. a public/private key signature scheme), authentication will be enforced, since only the owner of the private key will be able to sign the digest. Of course, this assumption is weak when the owner of the private key does not adequately protect it.

- **Non-repudiation**: a side-effect of the use of a signature scheme in a communication protocol is the introduction of a non-repudiation mechanism. In theory only the owner of a private key can send a target message/command. Since this message will be processed only if the signature is valid, as a consequence a command will be executed only when its legitimate origin can be validated.
- **Anti-replay protection**: In order to avoid the scenario in which an attacker re-uses a "pre-captured" Modbus packet signed by an authorized actor of the communication, the protocol needs a method to discriminate between a "new packet" and an "used packet". The lightest way to achieve this goal is the use of a *time-stamp* incorporated into the ADU, as shown in figure 2. The time-stamp will be used by the receiver of a target packet in combination with an internal "time-window" in order to check the validity of the packets.

The presented schema (figure 2) is able to guarantee a high level of security without impacting on the normal process control flow. Moreover, as it is possible to see in figure 1, the security envelope can be seamlessly applied by using user-space applications intercepting the Modbus packet (master-side) or by adopting a validation gateway connected transparently between the PLC and the rest of the network. This solution can be therefore applied on all the existing architectures without any significant impact.

B. Critical State Filtering System

The introduction of the authentication and integrity layer is completely useless when one of the actors at the edges of the communication flow is corrupted; if an attacker takes the control of a master for example, he will be able to sign in a licit way malicious packets. For that reason we have introduced a new entity in the architecture that independently checks the safeness of the packets exchanged - which is based on the identification of the Critical States of the system. Traditional firewalls generally fail in fulfilling this task in the SCADA context for the following reasons:

- The SCADA protocols (in our case Modbus) are dedicated application level protocols. Traditional firewalls do not usually implement any analyzing function for the SCADA protocol payload.
- The heuristic engines of firewalls and IDSs have not been developed to identify malicious behaviors of SCADA_over_ TCP protocols.

In any case even the development of a firewall able to analyze SCADA protocols might not be enough to protect process control systems. The following example clarifies this statement: consider a system with a pipe P_1 in which flows high pressure steam. The pressure is regulated by two valves V_1 and V_2. If an attacker is able to inject command packets in the process network, it could, for example, send a packet to the PLC controlling the valve V_2 to force its complete closure and a command to the PLC controlling the valve V_1 in order to maximize the incoming steam. These two operations, taken separately, are perfectly licit. However, if sent in sequence, they are able to put the system into a critical state since the pressure in the pipe P_1 will became soon too high and the pipe could explode. To solve this problem we developed an innovative filtering technique along the following lines:

1) The core of every industrial system is the *process network*, and the core of each industrial process network is the *SCADA system*. The SCADA system controls the process running inside the industrial system. In this way, by monitoring its activity it is possible – at least in principle – to control the activity of the entire industrial system.
2) Every industrial system is, when designed and deployed, well analyzed and all the possible "unwanted" states are usually identified. These unwanted states are what we identify as *critical states* i.e. system states which can be dangerous for the industrial system.
3) The data flowing among masters and slaves of a SCADA system can be used to reconstruct the *virtual image* of the state of the monitored system. We can thus compare such "virtual state" with the critical states to be avoided. Furthermore, upon tracing the evolution of the virtual state, we can predict whether the system is evolving into a critical state.
4) We model the industrial system in the following way: we identify a set of critical states for each subsystem composing the industrial system and we describe the dependencies among the different subsystems in such a way that we can then efficiently monitor the state of a (possibly very complex) system. In this way, we are able to detect many types of attacks. The effectiveness of this approach depends upon the granularity used in the representation of the virtual state and on the effects that such attacks can have on the evolution of those states.

Technically speaking, a CS-Filtering Unit is composed of the following elements:

- **Integrity-Authentication checker**: it verifies time-stamps and signatures of the analyzed packets.
- **A Virtual System**: a virtual system is a collection of software objects which simulate the active elements of the system monitored by the FU. It is built automatically by the FU on the basis of a system description (written using a formal language we have created for the purpose [15]). The Virtual System is kept alive using the data flowing between master and slaves.
- **Master Emulator**: To avoid the risk of divergence between the real system and the virtual system, the FU embeds a Master emulator able to periodically query the field network about its own state.
- **CS-State Checker**: it monitors the evolution of the virtual system in search for the occurrence of critical states triggered by a certain chain of packets.
- **Firewall**: it blocks data flows which have been detected as malicious by the CS-State Checker

More details about this technique and the performance obtained can be found in [15]. As previously described, the use of a single filtering unit exposes the system to the risk of collusion between a corrupted master and a corrupted FU. For that reason in our architecture we have introduced the concept of diversity and $\frac{K}{N}$ resilience. Instead of using a single filter, we use a mesh of independent filters, installed on machines using different operating systems and setups, in order to diminish the possibility of having an attacker able to corrupt simultaneously all the FUs. Moreover, the Master sends the request to all the FUs, and only if K of them agree on the safety of a packet, the slave will execute the command contained in the packet.

Here below we describe the different steps required by the introduction of the multiple FUs architecture:

1) The master (according to the new protocol specifications) composes the Modbus request(M_{req}) with the time stamp and the slave's address
2) The master calculates the digest through the hash function SHA2.
3) The Master signs the Modbus request/response digest with his private key PK_m and sends it to the N Filtering Unit

$$Mrd = (TS|Modbus), Enc\{SHA2, PKm\} \quad (1)$$

$$= (TS|Modbus), Enc\{C, PKm\} \quad (2)$$

4) Each FU, independently validates the Modbus request using the Master's public key (SK_m)

$$M_r = ((TS|Modbus), Dec\{C, SK_m\}) \quad (3)$$

5) Each Filtering Unit (FU) analyzes the Modbus request destination and function; if the packets contains a forbidden address or a forbidden instruction, the FU adds it into a dedicated stack of "malformed packets".
6) Each FU checks whether the command brings the virtual image of the system into a critical state. In this case it blocks the packet and sends an alert to the Critical State Correlation System
7) Each FU, taking as feed the analyzed traffic and by querying periodically the field network, keeps updated a digital representation of the system physical state.
8) If the packet is considered safe, each FU signs it with his private key PK_f and forwards the packet to the slave.

$$M_{rF} = (Mr, Enc\{SHA2(Mrd), PK_f\}) \quad (4)$$

9) The slave(PLC) validates the Modbus request filtered (MrF) using the Filtering Unit Public Key (SK_f)

$$Mr = Dec\{SHA2, SK_f\} \quad (5)$$

10) The slave(PLC) validates the Modbus request/response (M_r) using the Master's Public Key

$$M_{req} = Dec\{SHA2, SK_m\} \quad (6)$$

11) The slave stores the messages in a special stack for then executing the command if and only if it receives the same packet from K filtering units; otherwise after a predefined time it trashes the messages.

A key point for the robustness of this architecture is the minimum value K of agreeing responses which trigger the execution of a command. This value has to be tailored to the requirements of the different installations in which the architecture is inserted.

C. Critical State IDS proactive monitoring

An Intrusion Detection Systems (IDS) is generally composed of a set of distributed sensors, analyzing on the fly certain characteristics of the system being monitored, in search for evidence of attacks. The scientific literature of IDS is extremely prolific, and all the existing solutions can be grouped into two classes: (a) Anomaly Based IDS and (b)Signature Based IDS. The work of Vollmer and Manic [16] can be considered an example of the class (a), while Snort [11] is an example of the class (b); however, in the field of Industrial Control Processes and SCADA, it is practically non existent. Only recently, Digitalbond [12] released the first set of signatures for Snort [11] to analyze SCADA traffic on the basis of single packets. Nai et al. in [13] and [15] presented an innovative approach to intrusion detection, based on the same concept of the Critical State Based Filtering Units. As described before, by providing an Network IDS sensor with a "virtual image" of the subsystem under control which is fed by the flow between SCADA master and slaves, it can directly monitor the state of an entire subsystem identifying possible complex attacks aiming at driving the sub-system into a critical state. In our case, we adopted this approach in a multi-layered configuration. In other words, each CS-IDS sensor monitors a subsystem controlled by a single Master, using as feed the traffic flowing from the Master to the filtering gap, from the filtering gap to the slaves and vice-versa. Each sensor, when detecting an operation that potentially might have an effect on other subsystems, sends an alert to a Critical State Correlation System (CSCS). This can be considered as a CS-IDS sensor containing a more abstract virtual-system that "captures" the causal relations between different subsystems. The CSCS, analyzing the evolution of the sub-system critical states, is able to detect Global Critical States, and on the basis of a set of Critical-State recovery rules, can directly take action to mitigate the effects of the attack. For instance, under some conditions, the CSCS can for example communicate with the PLC-Gateway, asking to interrupt a certain commands flow, or can directly send commands to the field network (emulating a SCADA master), to mitigate the effects of the attack. Moreover, since the CS-IDS sensor receives as input both the traffic entering into the filtering gap and coming out from it, by comparing the two flows it is able to detect whether a FU is corrupted or collaborating with a corrupted Master. The CS-IDS sensor is described more in detail in [13] while the formal language used to describe the high level critical states is described in [15].

D. CS Rules

The language we have design to represent the critical states of the system (i.e. the rules of the IDS/Filtering unit) has the form $condition \rightarrow action$, where $action$ represents an alert. The remaining part of the rule, $condition$, is a boolean formula, composed by conjunctions and/or disjunctions of predicates describing what values can be assumed by the different PLCs' components.

The PLC's elements taken into account by our rule language are coils C, registers R, digital inputs DI, digital outputs DO, analog inputs AI and analog outputs AO.

The difference between traditional IDS rule languages and our language, is that in the last case, the predicates (upon which the condition that triggers the corresponding rule is formed) are defined over the states of the *PLCs* while in the traditional signature based rule languages, the content of a packet is represented.

In fact, upon interception of a packet, the IDS updates the SCADA system state, changing the parameters of the PLC (or PLCs) to which the packet is addressed, according to the information contained in the packet payload.

Then, the IDS checks whether some PLCs' configuration as represented in the resulting state triggers a rule. If this is the case, the corresponding action – prescribed by the rule – is performed. In the following we describe the rule language using the standard BNF notation:

$$\langle rule \rangle := \langle condition \rangle \rightarrow \langle action \rangle \ \langle condition \rangle := \langle predicate \rangle \mid \langle predicate \rangle \langle conn \rangle \langle predicate \rangle$$
$$\langle predicate \rangle := \langle term \rangle \langle relation \rangle \langle term \rangle$$
$$\langle term \rangle := \langle PLCName \rangle \mid \langle value \rangle$$
$$\langle PLCName \rangle := PLC \langle number \rangle . \langle comp \rangle \langle number \rangle$$
$$\langle action \rangle := Alert \mid Log \mid Look \langle rule \rangle$$
$$\langle conn \rangle := and \mid or$$
$$\langle relation \rangle := \leq \mid \geq \mid < \mid > \mid =$$
$$\langle comp \rangle := C \mid R \mid DI \mid DO \mid AI \mid AO$$
$$\langle value \rangle := 0 | ... | 2^{16} - 1$$

Consider the following rule stating that if coil $C23$ of $PLC1$ has value 0 and coil $C17$ of $PLC2$ has value 1 (corresponding, respectively, to open Valve 1 and close Valve 2), then the IDS performs an *Alert* action of the last packet that changed the state $PLC1.C23 = 0 \, and \, PLC2.C17 = 1 \rightarrow Alert$ Thus, a packet addressed to $PLC1$ containing a command for switching coil $C23$ to 0 – given that the other condition stated in the rule is satisfied – will trigger an *Alert* action. Again, note that the switching request (local to $PLC1$) could be a perfectly legal one, but it could become critical when other, non-local conditions are tested.

V. Experimental Tests

The described architectural approach has been fully implemented in a working prototype. We have developed the CS-filtering units, the CS-IDS sensors, the CS-correlation system. Moreover we have modified the TCP/IP stack to support the secure Modbus protocol. In what follows we describe the experimental facility used to test this architecture. Additionally, in the following subsection we present the first results obtained.

A. Experimental Platform

The architecture presented was tested in the SCADA security laboratory of our research institute. This laboratory has a protected environment that reproduces the dynamics and the configurations of different industrial systems. In the configuration used for our tests, we reproduced the typical architecture of a turbo-gas power plant. Figure 3 shows the high level schema of the facility. Thanks to a large campaign of analysis of an existing turbo-gas power plant [4] the elements composing the experimental environment are very close the reality. The field network, i.e. the most physical part of the emulated plant, is composed of a mix of real PLCs (ABB AC800) and virtual PLCs (software PLCs which we have developed in order to increase the number of possible scenarios and system configurations). The emulated system is monitored by a set of sensors (host based and network based), for gathering as much information as possible during each experiment.

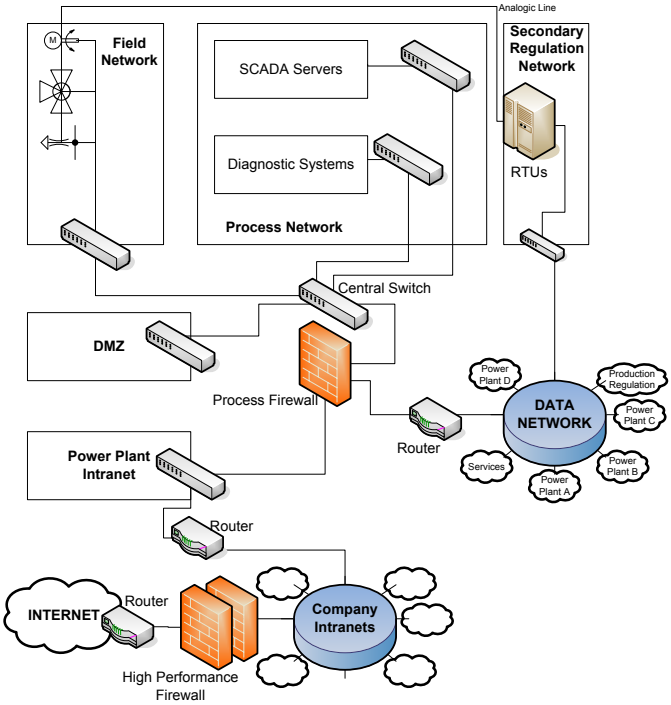


Fig. 3: Experimental Facility

B. Experimental Results

The purpose of the experiments was to measure the impact of our architecture on the communications performance between a Master and a Slave when executing a Modbus transaction. A Modbus transaction is a request/response message exchange. The duration of a transaction is the time measured between the request sent by the master and the receipt of the slave response. The part of our architecture that affect more

the Master/Slave transaction is the "*Integrity-Authentication Layer*". For this reason we made the experiment described in Figure 4. The Master station sends a request message to the

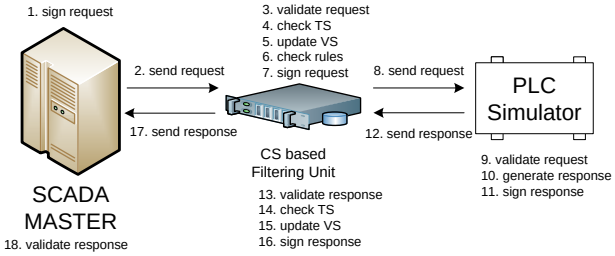


Fig. 4: Test schema

Slave station which responds with the appropriate response message. In the middle there is one filtering unit that performs the authentication and integrity checks as described before. We used two simulators written in C# under Windows XP SP3 to simulate the master and the slave; all the machines involved mount an AMD Athlon(tm) 64 X2 Dual Core Processor 5800+ 3.01 Ghz and 3.25 GB of RAM. We measured the time elapsed when executing the steps described in section IV, those required to complete our secure transaction. The measured times are shown in Table I.

N.	Task	Calculation	Time
1	M: Sign Req	$M_{rd} = (TS\|Modbus)$, $Enc\{SHA2, PKm\}$	10 ms
2	M: Send Req	Send M_{rd}	1 ms
3	FU: Validate Req	$M_r = ((TS\|Modbus)$, $Dec\{SHA2, SK_m\})$	1 ms
4	FU: Check TS	Anti-Replay check	0.5 ms
5	FU: Update VS	Update Virtual System	1 ms
6	FU: Check Rules	Check signature based and CS rules	1 ms
7	FU: Sign Req	$M_{rF} = (Mr,$ $Enc(\{SHA2(Mrd), PK_f\})$	7 ms
8	FU: Send Req	Send M_{rF}	1 ms
9	S: Validate Req	$M_{req} = Dec(\{SHA2, SK_m\})$	1*K ms

TABLE I: Time elapsed for each step in the "*Integrity-Authentication Layer*"

M = Master
FU = Filtering unit
S = Slave
Req = Request Message

The introduction of authentication and integrity affects the communications performance in terms of:

- **Packet size.**
- **Modbus transaction duration.**

Table II shows the packet size overhead introduced by the use of Secure Modbus.

Transaction	Modbus size	Secure Modbus size
M → FU	258 bytes	258 + 1 (ts) + 128 (sign M) = 387
FU → S	258 bytes	258 + 1 (ts) + 128 (sign M) + 128 (sign FU) = 515

TABLE II: Modbus and Secure Modbus Packet Size

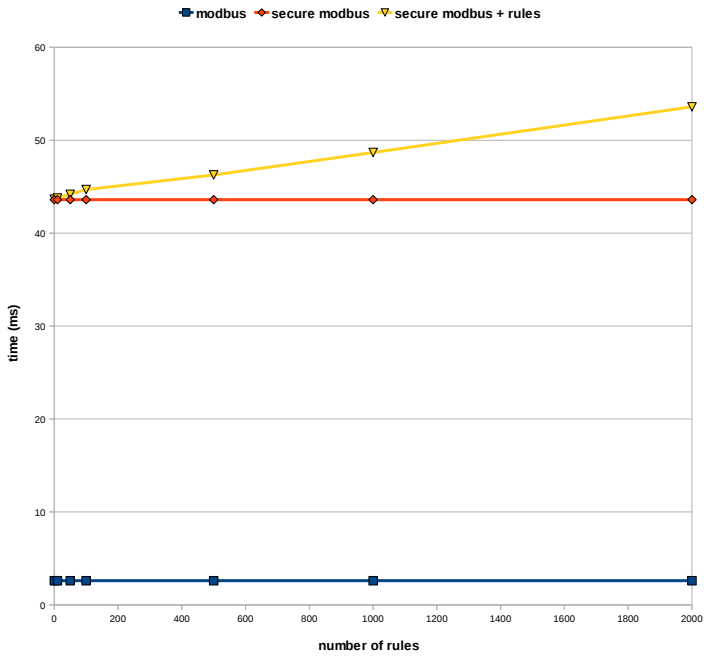


Fig. 5: Experimental Tests

The original Modbus packet size in Table II is 258 bytes because the packet sent contains the Function Code *15 Write Multiple Coils* and the number of coils to read is 1968 (0x07B0 hex is the maximum value allowed according to the Modbus specification [10]). We chose this Function Code for two reasons:

1) It is one of the largest packets possible to build with the Modbus protocol (the maximum is 260 bytes according to the Modbus specification [10]), i.e. this is one of the worst cases in terms of packet size.
2) It involves many coils, so the Filtering Unit has to update the values of 1968 coils and this is also the worst case for the "*Virtual System Update* ".

As we claimed before, the introduction of authentication and integrity affects also the Modbus transaction duration. The time spent to complete a Modbus transaction is about 2 milliseconds in a network without background traffic. Using our architecture some delays are introduced:

- **Integrity-Authentication Delay:** it is the time spent to calculate the digest and to sign and verify the packet by the Master, Slave and Filtering Units.
- **Virtual System Update Delay:** it is the time spent to update the Virtual System used by each Filtering Unit.
- **Check Rules Delay:** it is the time spent to check the "*Signature-Based*" and "*Critical State-Based*" rules by each Filtering Unit.

The "*Integrity-Authentication Time*" introduces a very large delay, about 42 milliseconds. The "*Virtual System Update Time*" is a relatively small time and even in the worst case used

in our tests (1968 coils to update); it does not have significant effects upon the transaction time. Regarding the "*Check Rules Time*" we repeated the test using 6 different sets of rules (10, 50, 100, 500, 1000, 2000). The results of these experiments

Rules	Modbus	Secure Modbus	Secure Modbus + Rules
0	2.6	43.6	43.6 + **0.007** = 43.67
10	2.6	43.6	43.6 + **0.016** = 43.76
50	2.6	43.6	43.6 + **0.058** = 44.18
100	2.6	43.6	43.6 + **1.09** = 44.69
500	2.6	43.6	43.6 + **2.66** = 46.26
1000	2.6	43.6	43.6 + **5.07** = 48.67
2000	2.6	43.6	43.6 + **9.99** = 53.59

TABLE III: Time elapsed

are shown in the Table III: the first column shows the time for a normal Modbus transaction, the second column shows the time for a secure Modbus transaction (with authentication and integrity checks), and the third column shows the time for a secure Modbus transaction with the filtering rules checks. The last column shows how the elapsed time increases with the number rules. In the Chart (Figure 5) we compare the performance of the three architectures (only Modbus, Secure Modbus, Secure Modubus + filtering).

The line at the bottom represents the normal Modbus transaction time; the straight line at the top represents the secure Modbus transaction time and the line with a linear growth represents the time for a secure Modbus transaction with the rules checking. It is not possible to say whether the delay introduced by our architecture can actually affect or not the performance of SCADA network, because this depends on the type of process that the Master is controlling. We plan to improve the performance of the presented architecture by using more advanced lightweight cryptographic protocols.

VI. Conclusions

The security of SCADA systems requires new approaches that could afford a workable answer to the more urgent problems. In this paper we propose an architecture combining different security solutions, that altogether are able to respond to the main cyber vulnerabilities. The principle we followed is that it is not realistic to expect dramatic changes in the topological and functional part of the SCADA implementing the control functions. Therefore, any improvement of the security of existing SCADA should be attained by combining elements that make use of existing features and that can be easily integrated. The architecture we propose supplies the following capabilities: authentication and non-repudiation by means of a signature scheme, integrity protection guaranteed by a secure hash function, anti-replay protection from the use of time-stamps, an innovative filtering of potentially dangerous packets, and finally a state-based attack detection mechanism (that acts as an ad-hoc IDS), based on the proactive monitoring of the critical states of the system. The composition of all these security functions result in a security shield providing a complete defence of the SCADA system. For trying the solution, we implemented it and tested it in our experimental platform simulating a power plant. The paper presents in a brief manner some of the results of the tests, which clearly indicate that the proposed solution, while protecting the SCADA system, doesn't introduce significant delays and therefore doesn't affect the control functions. In the future we intend to improve the implementation of the architecture for reaching still better performance, and to extend the testing campaign for verifying its robustness against a wide set of attacks.

References

[1] I. Nai Fovino, M. Masera, R. Leszczyna: ICT Security Assessment of a Power Plant, a Case Study In Proceeding of the Second Int. Conference on Critical Infrastructure Protection, Arlington, USA, March 2008

[2] A. A. Creery, E. J. Byres: Industrial Cybersecurity for power system and SCADA networks IEE Industry Apllication Magazine, July-August 2007

[3] I. Nai Fovino, A. Carcano and M. Masera: Secure Modbus Protocol, a proof of concept. In Proc. of the 3rd IFIP Int. Conf. on Critical Infrastructure Protection, Hanover, NH., USA, 2009.

[4] I. Nai Fovino, M. Masera, R. Leszczyna: ICT Security Assessment of a Power Plant, a Case Study In Proceeding of the Second Int. Conference on Critical Infrastructure Protection, Arlington, USA, March 2008

[5] A. Carcano, I. Nai Fovino, M. Masera, A. Trombetta: Scada Malware, a proof of Concept. In proceeding of the 3rd International Workshop on Critical Information Infrastructures Security, Rome, October 13-15, 2008

[6] R. Chandia, J. Gonzalez, T. Kilpatrick, M. Papa and S. Shenoi: Security Strategies for Scada Networks In Proceeding of the First Int. Conference on Critical Infrastructure Protection, Hanover, NH., USA, March 19 - 21, 2007.

[7] M. Majdalawieh, F. Parisi-Presicce, D. Wijesekera: Distributed Network Protocol Security (DNPSec) security framework. In Proceedings of the 21st Annual Computer Security Applications Conference, December 5-9,2005, Tucson, Arizona.

[8] J. H. C. S. Hong, S. Ho Ju, Y. H. Lim, B. S. Lee, D. H. Hyun: A Security Mechanism for Automation Control in PLC-based Networks In Proceedings of the ISPLC '07. IEEE International Symposium on Power Line Communications and Its Applications 26-28 March 2007, pp 466-470, Pisa, Italy

[9] T. Mander, F. Nabhani, L. Wang, R. Cheung: Data Object Based Security for DNP3 Over TCP/IP for Increased Utility Commercial Aspects Security In Proceedings of the Power Engineering Society General Meeting, 2007. IEEE 24-28 June 2007, pp 1-8, Tampa, FL, USA.

[10] http://www.modbus.org/

[11] M. Roesch: Snort -Lightweight Intrusion Detection for Networks. Proceedings of LISA '99: 13th Systems Administration Conference, Seattle, Washington, USA, November 712, 1999

[12] http://www.digitalbond.com/index.php/research/ids-signatures/modbus-tcp-ids-signatures/ , last access 9/04/2009

[13] I. Nai Fovino, A. Carcano, M. Masera, A. Trombetta, T. Delacheze-Murel: Modbus/DNP3 State-based Intrusion Detection System". In Proceedings of the 24th International Conference on Advanced Information Networking and Applications, Perth, Australia, 20-23 April 2010.

[14] I. Nai Fovino, A. Carcano, M. Masera: A secure and Survivable Architecture for SCADA Systems. In Proceedings of The Second International Conference on Dependability, DEPEND 2009, June 18-23, 2009 - Athens/Vouliagmeni, Greece.

[15] I. Nai Fovino, M. Guglielmi, M. Masera, A. Carcano, A. Trombetta. Distributed Critical State Detection System for Industrial Protocols. In Proceeding of the 4th Annual IFIP International Conference on Critical Infrastructure Protection, Washington DC, USA, March 14 - 17, 2010.

[16] Vollmer, T.; Manic, M.. Computationally efficient Neural Network Intrusion Security Awareness. Resilient Control Systems, 2009. ISRCS '09. 2nd International Symposium on , vol., no., pp.25-30, 11-13 Aug. 2009

A Trust Based Distributed Kalman Filtering Approach for Mode Estimation in Power Systems

Tao Jiang Ion Matei John S. Baras
Institute for Systems Research
and Department of Electrical and Computer Engineering
University of Maryland
College Park, MD
{tjiang,imatei,baras}@umd.edu

***Abstract*—We consider distributed mode estimation in power systems. The measurements are observed by PMUs (Power Management Units). We introduce a novel model of trust, using weights on the graph links and nodes that represent the networked PMUs. We describe two algorithms that integrate distributed Kalman filtering with these trust weights. We consider two interpretations of these trust weights as information accuracy and reliability. We show that by appropriate use of these weights the distributed estimation algorithm avoids using information from untrusted PMUs. Simulation experiments further demonstrate the behavior of these algorithms.**

I. Introduction

The digital control and protection of power systems require the collection of huge amounts of data to estimate various parameters in real-time. For instance, when a short circuit occurs in a power transmission line, the steady state values of the post-fault currents and voltages must be estimated to locate the fault location. Furthermore, next generation power grids involve large interconnected power networks, resulting in greater emphasis on reliable and secure operations [1]. The large scale communication networks underlying the power grids make it impossible to collect data and control power systems in a centralized manner. The new power systems must have a *distributed* communication and control system in the face of an ever changing environment such as equipment failures and even attacks (e.g. cyber-attacks).

Because the new communication and control system enables many more interactions between many more participants, it has security requirements beyond the conventional Confidentiality, Integrity and Availability properties provided by conventional security systems. For example, integrity and confidentiality have nothing to say about the quality of the data obtained from various substations. Nor does confidentiality protect against disclosure of a measurement by an intended recipient. As the community of participants in the power grids operations grows, properties that involve the behavior of participants become increasingly critical for reliable operations and difficult to deal with.

One crucial question is: how the control system can trust the data provided by the communication network? Our research efforts are motivated by two key observations. First, due to the distributed and dynamic nature of the power systems, the uncertainty of data accuracy has to be taken into consideration. Second, PMUs in the power grids often operate unattended in physically insecure environments, and are designed with an emphasis on numbers and low cost which makes security measures such as tamper-proof hardware not cost effective. Therefore, we cannot only resort to costly cryptography to guarantee reliable operations. In this paper, the concept of trust is used in a specific problem of power systems: mode estimation. We propose a trust based distributed Kalman filtering approach to estimate the modes of power systems. We show that by establishing appropriate trust relations, the estimation is more resilient to attacks.

II. Problem formulation

Large interconnected power networks are often associated with inter-area oscillations between clusters of generators. These inter-area oscillations are of critical importance in system stability and require on-line observation and control [2]. The inter-area oscillations (often referred to as modes) are damped sinusoids which all have a particular frequency and damping factor. The damping factor determines the transient ability of the system to stabilize post disturbance. Therefore, it is critical to have a rapid and good estimation of the damping factor in large distributed power systems.

This work addresses automatic detection of oscillations in power systems using dynamic data such as currents, voltages and angle differences measured across transmission lines given that some measurements are false. The measurements are provided on-line by the PMUs distributed throughout the large-area power system. The power system is assumed being driven by disturbances around nominal operating points ([3]), therefore linear models can be used to linearize the system and to model oscillations.

The linearizaton method used in this paper is based on the work by Lee and Poon [4]. Disturbance inputs in a power system (such as load changes) consist of M frequency modes and, with the initial steady-state value eliminated, can be generalized over a specific time period as

$$f(t) \quad = \quad a_1 \exp(\sigma_1 t) \cos(\omega_1 t)$$

$$+\sum_{j=2}^{M} a_j \exp(\sigma_j t)\cos(\omega_j t + \phi_i) \quad (1)$$

where a_i are oscillation amplitudes, σ_i are damping constants, ω_i are the oscillation frequencies and ϕ_i are phase angles of the oscillations. Without loss of generality, we consider two modes in Eqn. (1), given by

$$\begin{aligned} f(t) &= a_1 \exp(\sigma_1 t)\cos(\omega_1 t) + \\ & \quad a_2 \exp(\sigma_2 t)\cos(\omega_2 t + \phi_2), \end{aligned} \quad (2)$$

which is a nonlinear function of the parameters a_i, σ_i and ϕ_i. Using the first two terms in the Taylor series expansion of the exponential function and expanding the trigonometric functions, we have that

$$\begin{aligned} f(t) &= a_1(1+\sigma_1 t)\cos(\omega_1 t) + \\ & \quad a_2(1+\sigma_2 t)[\cos\phi_2 \cos(\omega_2 t) - \sin\phi_2 \sin(\omega_2 t)]. \end{aligned} \quad (3)$$

We introduce the notation:

$$\begin{aligned} x_1 &= a_1 & x_2 &= a_1\sigma_1 \\ x_2 &= a_2\cos\phi_2 & x_4 &= a_2\sigma_2\cos\phi_2 \\ x_5 &= a_2\sin\phi_2 & x_6 &= a_2\sigma_2\sin\phi_2 \end{aligned}$$

and

$$\begin{aligned} c_{11} &= \cos(\omega_1 t) & c_{12} &= t\cos(\omega_1 t) \\ c_{13} &= \cos(\omega_2 t) & c_{14} &= t\cos(\omega_2 t) \\ c_{15} &= -\sin(\omega_2 t) & c_{16} &= -t\sin(\omega_2 t) \end{aligned}$$

Then we have

$$f(t) = \sum_{i=1}^{6} c_{1i}(t) x_i(t). \quad (4)$$

The power system is sampled at a preselected rate, say every Δt seconds. Eqn. (4) can be written in discrete time k, $k = 1, \ldots, K$. We have the linear measurement model as the following:

$$y_i(k) = C_i x(k) + v_i(k), \quad (5)$$

where $y_i(k)$ is the measurement of the state $x(k)$ made by PMU i, and $v_i(k)$ is the measurement noise assumed Gaussian with zero mean and covariance matrix R_i.

For N measurements, Eqn. (5) can be written in vector form as

$$y(k) = Cy(k) + v(k). \quad (6)$$

The state transition matrix $A(k)$, which relates the state $x(k)$ to $x(k-1)$ is the identity matrix. The state space equation is given by

$$x(k+1) = A(k)x(k) + w(k), \quad (7)$$

where $w(k) \in \mathbb{R}^n$ is the state noise, assumed Gaussian with zero mean and covariance matrix Q. The initial state x_0 has a Gaussian distribution, with mean μ_0 and covariance matrix P_0. Eqn. (6) and (7) form a linear random process that can be estimated using the Kalman filter algorithm.

Having estimated the parameter vector $x(k)$, the amplitude, damping constant, and phase angle can be calculated at any time step k using the following equations:

$$a_1(k) = x_1(k) \quad (8)$$

$$\sigma_1(k) = \frac{x_2(k)}{x_1(k)} \quad (9)$$

$$a_2(k) = [x_3^2(k) + x_5^2(k)]^{1/2} \quad (10)$$

$$\sigma_2(k) = \left[\frac{x_4^2(k) + x_6^2(k)}{x_3^2(k) + x_5^2(k)}\right]^{1/2} \quad (11)$$

$$\phi_2(k) = \tan^{-1}\left[\frac{x_6(k)}{x_4(k)}\right] = \tan^{-1}\left[\frac{x_5(k)}{x_3(k)}\right]. \quad (12)$$

Fig. 1 shows a power system with several PMUs. Measurements from the entire grid are synchronized via a satellite. As

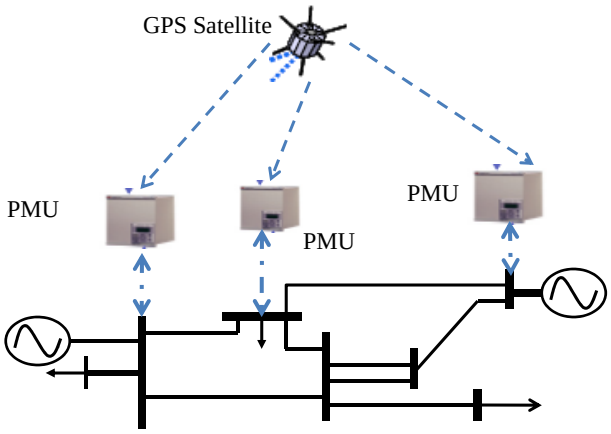


Fig. 1. An Overview of the Monitoring System

we discussed in Section I, distributed computation and communication are needed given the large scale communication networks underlying the power grid. We consider a power system with N multiple recording sites (PMUs) to measure the output signals, indexed by i. The goal of each PMU i is to compute an accurate estimation of the state $x(k)$, using: the local measurements $y_i(k)$; the information received from the PMUs in its communication neighborhood (e.g. measurements and estimates); and the confidence in the information received from other PMUs provided by the trust model described in the following sections.

Each PMU i has a communication neighborhood containing PMUs with whom the PMU can exchange information. Let $\mathcal{N}_i$ denote such a communication neighborhood:

$$\mathcal{N}_i = \{j \mid i \text{ exchanges information with } j\}.$$

The communication neighborhoods of the PMUs determine a communication graph with N vertices, such that a link from i to j exists if PMU i sends information to PMU j.

We attach a positive value T_{ij} to each link (j, i) which represents the confidence value that PMU i places on the information coming from PMU j. The value T_{ij} represents a measure of the trust PMU i has in the information received from PMU j.

There are many different definitions of "trust" depending on the particular domains. An operational definition of "trust" for information, mainly considers two aspects: information *accuracy* and *reliability*. Accuracy reflects the deviation of the information from truth, and reliability is confidence in the assessment of accuracy. In this paper, we apply trust weights to the distributed estimation problem where these two aspects of trust are investigated separately.

III. Distributed Kalman filtering

The main idea behind distributed estimation, found in most of the papers addressing this problem, consists of using a standard Kalman filter locally, together with a consensus step in order to ensure that the local estimates agree [5]. In what follows, we use a simplified version of the algorithm proposed in [5].

Algorithm 1: Distributed Kalman Filtering algorithm with consensus step on estimates [5]

Input: μ_0, P_0

1 Initialization: $\xi_i = \mu_0$, $P_i = P_0$

2 **while** new data exists

3 Compute the intermediate Kalman estimate of the target state:

$$M_i = P_i^{-1} + C_i' R_i^{-1} C_i$$
$$L_i = M_i C_i R_i^{-1}$$
$$\varphi_i = \xi_i + L_i(y_i - C_i \xi_i)$$

4 Estimate the state after a consensus step:

$$\hat{x}_i = \varphi_i + \epsilon \sum_{N_i \cup \{i\}} (\varphi_j - \varphi_i)$$

5 Update the state of the local Kalman filter:

$$P_i = A M_i A' + Q$$
$$\xi_i = A \hat{x}_i$$

For simplicity we omitted the time index in Algorithm 1. Notice that with the exception of line 4, the above algorithm is the standard linear Kalman filter. In line 4, the local information is linearly combined with information received from neighbors. We will refer to line 4 as either the *information fusion step* or the *consensus step*. We will focus our analysis on the values of the weights w_{ij} . In fact they will play the role of the confidence values introduced in the previous section. Unlike the original algorithm [5], we assume that only local estimates are exchanged and not output measurements as well.

IV. Distributed Kalman filtering with trust dependent weights in the consensus step

In this section we develop the distributed filtering equations that take into account the confidence (trust) of the PMUs. We address two cases reflecting what the confidence values represent. In the first case, we assume that the weights w_{ij} are a measure of the *information accuracy*, i.e. the larger the value of w_{ij} is, the more accurate the information received by i from j is. In the second case, the weights w_{ij} are a measure of the *trustworthiness* of the data received by PMU i from PMU j. It may be the case that either a PMU or a link were compromised, so that the information received from the respective PMU or through the respective link is not trustworthy.

A. Distributed Kalman Filtering with accuracy dependent consensus step

We attach to each PMU a trust value. In this subsection, the trust refers to the accuracy of information. The larger the trust value is, the more accurate the information received from the respective PMU is. The information exchanged between PMUs is represented by estimates. As previously mentioned, we denote by T_{ij} the trust PMU i has in information received from PMU j. We propose to choose the trust values to be inversely proportional to the estimation error, according to the formula:

$$T_{ij} = \frac{1}{trace(M_j)},\ j \in \mathcal{N}_i, \tag{13}$$

where M_j represents the covariance matrix of the estimation error from the standard Kalman filter step. The properties of this matrix will be affected by how *observable* the state is from PMU j, (such as the rank of matrix C_j) and how noisy the measurements are, i.e. the variance of the measurements' noise R_j. We can expect the variance of the estimation error, given by the trace of M_j, to be small for highly observable measurements with low noise. Therefore, we computed the weight values in the information fusion step, by normalizing the trust values T_{ij}:

$$w_{ij} = \frac{T_{ij}}{\sum_k T_{ik}}. \tag{14}$$

This way, we assign a larger influence to the more accurate estimates, directing the resulting average towards estimates with high accuracy. Note however that the matrix M_j is not the actual covariance matrix of the estimation error for the current estimate $\hat{x}_j$, but the covariance error given by the standard Kalman filter. In does however reflect the observability properties of the PMU, making it a good candidate for constructing the weight values. We summarize the idea introduced above in Algorithm 2.

B. Distributed estimation with reliability dependent consensus step

In this subsection we propose a distributed estimation scheme where the averaging operation depends on the reliability of the PMUs. We assume that PMUs may be compromised and may send data aimed at modifying the result of the estimation process. The update mechanism for the trust values T_{ij} is based on the notion of *belief divergence* [6]:

$$d_i = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \|\hat{x}_i - \hat{x}_j\|^2, \tag{15}$$

where we denoted by $\hat{x}_i$ the current estimates.

Algorithm 2: Distributed Kalman Filtering Algorithm with accuracy dependent consensus step on estimates

Input: μ_0, P_0

Initialization: $\xi_i = \mu_0$, $P_i = P_0$

while new data exists

Compute the intermediate Kalman estimate of the target state:

$$M_i = P_i^{-1} + C_i' R_i^{-1} C_i$$
$$L_i = M_i C_i R_i^{-1}$$
$$\varphi_i = \xi_i + L_i(y_i - C_i \xi_i)$$

Compute the consensus weight values:

$$T_{ij} = \frac{1}{trace(M_j)}$$
$$w_{ij} = \frac{\bar{w}_{ij}}{\sum_k \bar{w}_{ik}}$$

Estimate the state after a consensus step:

$$\hat{x}_i = \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} \varphi_j$$

Update the state of the local Kalman filter:

$$P_i = A M_i A' + Q$$
$$\xi_i = A \hat{x}_i$$

The belief divergence d_i, gives to PMU i a measure of how different its own estimate is with respect to the estimates of the other PMUs within its communication neighborhood.

Since the PMUs exchange only state estimates, every PMU will compute a belief divergence, d_{ij}, for each PMU in his neighborhood, according to the formula:

$$d_{ij} = \frac{1}{\mathcal{N}_i - 1} \sum_{k \in \mathcal{N}_i} \|\hat{x}_j - \hat{x}_k\|^2. \tag{16}$$

This metric shows how far a received estimate is from the other received estimates in some neighborhood. Note that in the fusion step, estimates far from their real values are prone to hurt more. However, if enough neighbors provide reliable information, the belief divergence for a PMU sending false information is going to by high. We use the locally computed belief divergence metric, to update the trust values T_{ij}. We first choose a positive constant c_i, satisfying:

$$c_i > \max\{d_{ij} \mid j \in \mathcal{N}_i\}.$$

We use the constant c_i in the following formula for updating the trust values:

$$T_{ij} = c_i - d_j, \ j \in \mathcal{N}_i \tag{17}$$

Notice that the parameters c_i were chosen so that the trust value T_{ij} are nonnegative. Moreover, c_i are discriminating in the sense that they influence the ratios T_{ij}/T_{ik}. Typically, the smaller c_i is, the more PMUs with large values of the belief divergence are penalized. From (17) we note that we favor the PMU whose estimate is close to the other estimates in its neighborhood, in a sense 'accelerating convergence' to consensus. We denote by p_{ij} the normalized versions of the trust values T_{ij}, computed according to the formula:

$$p_{ij} = \frac{T_{ij}}{\sum_{k \in \mathcal{N}_i} T_{ik}}, \tag{18}$$

which may be interpreted as the "probability the data received by PMU i from j are accurate". Note from the above formulas that, although small, the normalized trust values are not necessarily zero for PMUs with large belief divergence. Therefore if the value of a false estimate is large compared with the others, it will still influence negatively the information fusion step. That is why we introduce a thresholding scheme on the normalized trust values. Let p_i^{min} be the minimum value accepted for p_{ij}. If $p_{ij} < p_i^{min}$ the trust value T_{ij} will be set to zero, hence filtering out information that is not considered sufficiently trustworthy. The lower bound p_i^{min} should be chosen to be inversely proportional to the size (cardinality) of the neighborhood.

The updated trust values are next used to compute the weights in the consensus step:

$$w_{ij} = \frac{T_{ij}}{\sum_{k \in \mathcal{N}_i} T_{ik}}. \tag{19}$$

The distributed estimation algorithm with a reliability dependent averaging scheme is presented in Algorithm 3 below. The intuition behind our proposed algorithm is that if a node j sends false data, the other nodes will compute large belief divergence values, and hence low trust values, which together with the thresholding scheme will eliminate the node from the information flow. The consensus step has the role of producing a new state estimate by averaging the estimates on neighborhoods. If an estimate is not accurate enough, it may drag the updated estimate towards the wrong direction. By computing the consensus weight values using a trust dependent mechanism, we try to minimize the possibility of an estimate update moving in the wrong direction. By adjusting the minimum accepted value for the normalized trust values, p_i^{min} , the PMUs can control their sensibility with respect to the received data.

V. Simulations

In this section, we report results on simulations and test of our implementation of the disitributed Kalman filter algorithm to estimate the oscillation amplitudes and the damping coefficients of a practical example, given in [4]. It is noted that it has two modes at $\omega_1 = 0.4Hz$ and $\omega_2 = 0.5Hz$. A model of power system was used as shown in Figure 2. The model employs five measurements, where each PMU is installed over a line connected to one generator.

We first test Algorithm 2 against Algorithm 1, where independent white noise with different SNR was added to each measurement before feeding them into the estimation procedure. For computing the weights w_{ij} in Algorithm 1 we used the original scheme proposed in [5], the value for

Algorithm 3: Distributed Kalman Filtering Algorithm with a reliability dependent consensus step on estimates

Input: μ_0, P_0

Initialization: $\xi_i = \mu_0$, $P_i = P_0$

while new data exists

Compute the intermediate Kalman estimate of the target state:

$$M_i = P_i^{-1} + C_i' R_i^{-1} C_i$$
$$L_i = M_i C_i R_i^{-1}$$
$$\varphi_i = \xi_i + L_i(y_i - C_i \xi_i)$$

Compute locally the belief divergence:

$$d_{ij} = \frac{1}{\mathcal{N}_i - 1} \sum_{k \in \mathcal{N}_i} \|\varphi_j - \varphi_k\|^2$$

Compute the trust values:

$$T_{ij} = c_i - \bar{d}_{ij}, \ j \in \mathcal{N}_i$$

Compute the normalized trust values:

$$p_{ij} = \frac{T_{ij}}{\sum_k T_{ik}}$$

Eliminate insufficiently accurate data by setting T_{ij} to zero if $p_{ij} < p_i^{min}$

Compute the consensus weight values:

$$w_{ij} = \frac{T_{ij}}{\sum_k T_{ik}}$$

Estimate the state after a consensus step:

$$\hat{x}_i = \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} \varphi_j$$

Update the state of the local Kalman filter:

$$P_i = A M_i A' + Q$$
$$\xi_i = A \hat{x}_i$$

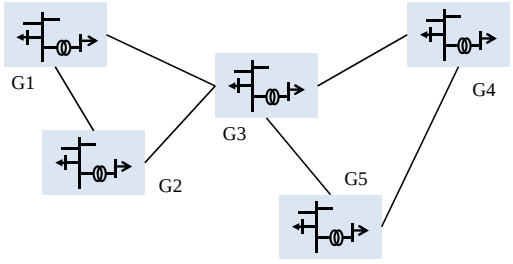


Fig. 2. Power System for Simulations

ϵ being chosen such that the average estimation error per node was as small as posssible. More precisely we want to compare the average estimation errors per node, given by the two algorithms. Since the trust weights are computed so that more weight is given to information coming from PMUs with smaller variance of the estimation error, we would expect Algorithm 2 to perform better, in the sense that the average estimation error per node should converge to a smaller value.

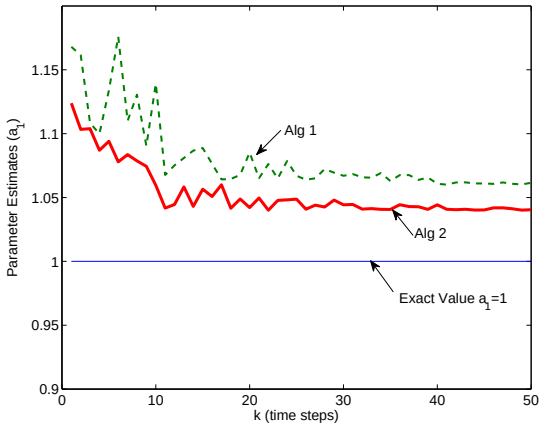


Fig. 3. Comparison of estimating parameter a_1 given by Alg 1 and Alg 2 respectively

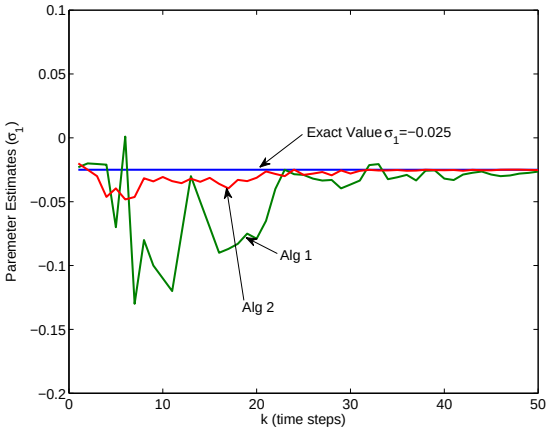


Fig. 4. Comparison of estimating parameter σ_1 given by Alg 1 and Alg 2 respectively

The comparison results for estimating parameters a_1 and σ_1 are shown in Fig. 3 and 4. The results for a_2 and σ_2 are similar. We observe that Algorithm 2, as expected, performs better. This is mainly due to the fact that in the estimation fusion step, we move the estimate updates closer to the local estimate with better observability and lower measurement noise.

For testing Algorithm 3, we assume that the measurements from the PMU connecting G3 were compromised and send false information to all the other PMUs. The goal of the PMU in G3 is to shift the estimates of other nodes away from their true values. We consider the case when the PMU connecting G3 sends to its neighbors a white noise with standard deviation equal to 0.1. The PMU connecting G3 is chosen because it is centered and has potential to do a lot of damage since

it is connected to all other PMUs. We compare the results using Algorithm 1 and Algorithm 3. The results for estimating parameter a_1 and σ_1 are shown in Figure 5 and Figure 6 respectively.

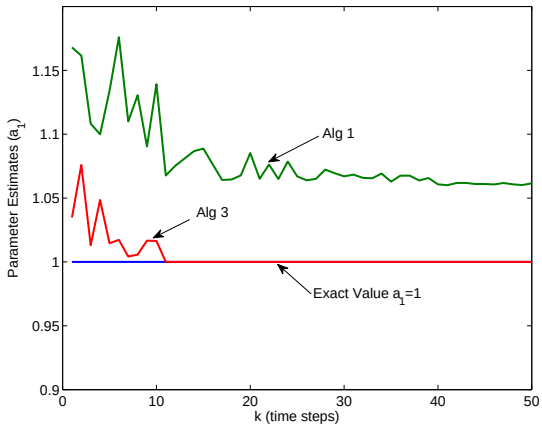


Fig. 5. Distributed Kalmann filtering with constant false information, estimating a_1

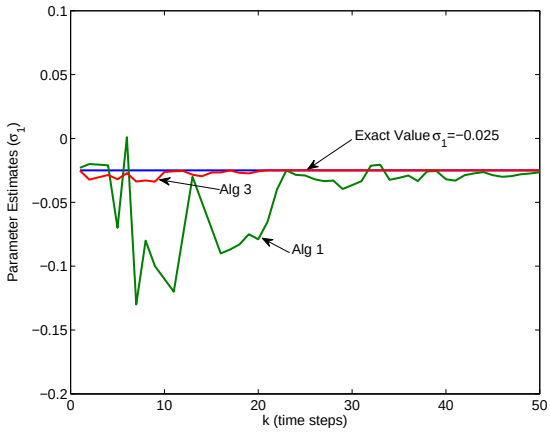


Fig. 6. Distributed Kalmann filtering with constant false information, estimating σ_1

We observe that Algorithm 3 is able to detect the false data provided by the PMU connecting G3 and eliminate it from further participation in the processing. The other PMUs are able to estimate closely the parameters. However, the false data does have influence on how fast the estmates converge to the real value at the beginning, since the false data are not immediately detected and rejected, the PMUs are able to compute parameter estimates that are close to the state values.

VI. Conclusion

In this paper, a distributed Kalman filtering approach is used to estimate oscillation modes in power systems that have false measurements and even under attacks. We proposed two modified distributed Kalman filtering algorithms, which incorporate the notion of trust. The first algorithm uses the trust notion to quantify the estimation errors in terms of observation and measurement noise. The second algorithm interpreted trust in terms of security. The low trusted PMUs are excluded from the estimation procedure. Via simulations, we compared our trust based algorithms with the original distributed Kalman filtering algorithm and showed that our modified algorithms perform better when there are large noises in the system and are able to detect malicious data.

Acknowledgement

Research partially supported by the Defense Advanced Research Projects Agency (DARPA) under award number 013641-001 for the Multi-Scale Systems Center (MuSyC), through the FRCP of SRC and DARPA. The authors acknowledge useful discussions and suggestions received through their participation in the EU project VIKING.

References

[1] V. Vittal, "Consequence and Impact of Electric Utility Industry Restructuring on Transient Stability and Small-signal Stability Analysis", *Proc. IEEE*, vol. 88, no. 2, pp. 196-207, Feb. 2000.

[2] M. Klein, G. J. Rogers and P. Kundur, "A Fundamental Study of Inter-area Oscillations in Power Systems", *IEEE Trans. Power Syst.*, vol. 6, no. 3, pp. 914-921, Aug. 1991.

[3] G. Ledwich and E. Palmer, "Modal estimates from normal operation of power systems", *2000 IEEE Power Eng. Soc. Winter Meeting. Conf. Proc.*, Singapore, 2000, vol. 2, pp. 15271531.

[4] K. C. Lee and K.P. Poon, "Analysis of power system dynamic oscillation with beat phenomenon by Fourier transformation", *IEEE Trans. Power Syst.*, vol. 5, no. 1, pages 148-153, 1990.

[5] R. Olfati-Saber, "Distributed Kalman Filtering for Sensor Networks", *Proceedings of the* 46^{th} *IEEE Conference on Decision and Control*, pages 5492-5498, 2007

[6] C. De Kerchove and P. Van Doren, "Iterative filtering for a dynamical reputation system", arXiv, 2007.

An Approach to Network Security Assessment based on Probalistic Relational Models

Fredrik Löf, Johan Stomberg,
Teodor Sommestad, Mathias Ekstedt
Industrial Information and Control Systems
Royal Institute of Technology (KTH)
Stockholm, Sweden

Jonas Hallberg, Johan Bengtsson
Swedish Defence Research Agency (FOI)
Linköping, Sweden

***Abstract*—To assist rational decision making regarding network security improvements, decision makers need to be able to assess weaknesses in existing or potential new systems. This paper presents a model based assessment framework for analyzing the network security provided by different architectural scenarios. The framework uses a probabilistic relational model to express attack paths and related countermeasures. In this paper, it is demonstrated that this method can be used to support analysis based on architectural models. The approach allows calculating the probability that attacks will succeed given the instantiated architectural scenario. Moreover, the framework is scalable and can handle the uncertainties that accompany an analysis. The method has been applied in a case study of a military network.**

***Keywords* – Probabilistic Relational Model, Network Security, Security Assessment, Attack Graph, Architecture Model**

I. INTRODUCTION

Many modern organizations depend on different types of information systems for business activities. An important aspect in these systems is network security as the consequences of a breach in security can be very damaging for the organization; from loss of trade secrets to theft and sabotage of critical infrastructure and services. With the importance of network security, it is natural to assess this aspect of a network. A security assessment might reveal unknown system weaknesses and show possible improvements, as well as work as a foundation for management and configuration decisions to find the most efficient application of resources when improving security.

A number of security assessment methods have been developed with different approaches to how security is evaluated. An overview of security measurement methods can be found in [32]. Attack graphs provide the foundation for several of these methods. An attack graph is a way to represent how an attacker can reach a goal in a system by defining what sub-goals the attacker must accomplish and plotting these different sub-goals as different paths or barriers the attacker must overcome [2,3,4,5]. In this way a security analyst can for instance find key points that must be protected or analyze a possible breach after the fact.

Scalability is an issue when constructing attack graphs. Recent results have decreased the complexity of the computations required to construct attack graphs [33,7,34]. However, the amount of input required to produce realistic attack graphs is considerable [35]. Also, it is often recommended that automated network scans should be avoided in operational industrial control system environments as these can disrupt operations [38,39]. To manually collect and update detailed network configuration data in this environment appears prohibitively expensive. Moreover, from a human point of view, attack graphs quickly become ungraspable due to their size and complexity. It is thus difficult to use the attack graphs to try out new, alternative solutions by manually changing some parameters in the graphs.

Probabilistic treatment of the relationship between different attack steps is an alternative solution to this challenge. This solution can also reduce the amount of input needed to model attacks. In [36] Bayesian networks are used to represent possible attacks more compactly and to calculate the probability that a network attack succeeds, as opposed to using deterministic attack graphs.

This paper describes a method to assess network security based on the Probabilistic Relational Model (PRM) formalism [1], which is a combination of Bayesian networks, attack graphs, and architectural models. The probabilistic approach can make predictions without all the details included in traditional attack graphs [35]. The drawback of this approach is the precision in the assessment since it by nature makes probabilistic estimations.

The basis for an analysis in the proposed method is a metamodel describing the system architecture, in terms of its components and their attributes, as well as possible attacks, in the form of conceptual attack graphs. With such a metamodel as a basis an analyst creates an instance model representing the architecture of the network. This instance model is used to calculate probabilities that an attacker might succeed with different potential attacks on the system architecture, thus providing decision-makers with information regarding network security.

The purpose of this article is to test whether PRMs can be used to assess network security. Thus, while the approach in general is intended for the security analysis of architecture models, this paper focuses on a PRM for communication networks.

The method has been applied in a study to assess the network security of a military network from the international

interoperability exercise Combined Endeavour 07. A short overview of this case is also presented in this paper.

II. Probabilistic Relational Models

A Probabilistic Relational Model [1] specifies the metamodel for the architecture models and the probabilistic dependencies between attributes of the architecture objects. A PRM defines a probability distribution over the attributes of the objects in an instantiated architecture model. The probability distribution can be used to infer the values of unknown attributes. This inference can also take into account evidence on the state of observed attributes.

A. Architecture metamodel

An architecture metamodel, M, describes a set of classes. Each class X is associated with a set of descriptive attributes and a set of relationships between attributes and classes. For example, a class Firewall might have the descriptive attribute Bypass Packet Filtering, with the domain {True, False} and the relationship Perimeter Defense to the class Host (cf. Figure 1). A relationship between attributes from these classes can then be defined through the class relationship. Every attribute has a conditional probability table (CPT) describing the probability distribution for the values of the attribute.

B. Architecture instance models

An architecture instantiation I (or an architecture model) specifies the set of objects in each class X, and the values for attributes, X.A, and relationships, X.r, of each object. For example, Figure 2 presents an instantiation of the metamodel described in Figure 1. It specifies a particular Firewall (Färist, [37]), two Authentication Services, two NIDS (Network Intrusion Detection System), one HIDS (Host-based Intrusion Detection System) and one Host.

C. Probablistic model over attributes

A PRM specifies a probability distribution over all instantiations I of the metamodel M. Like a Bayesian network [8] it consists of a qualitative dependency structure and associated quantitative parameters. The qualitative dependency structure is defined by associating attributes X.A with a set of parents Pa(X.A). This is done by specifying a search path through relationships in the instance model that describes the parents. For example, the attribute Host.MaliciousCodeAttack has a parent Host.*PerimeterDefense*.BypassContentFiltering, meaning that the possibility of an attack with malicious code against a host depends on the probability that the attack bypasses the content filtering of the host's perimeter defense. It is the class relationship between Firewall and Host – called Perimeter Defense – that makes the attribute relationship possible.

We can now define a PRM for a metamodel M as follows. For each class X and each descriptive attribute $A \in At(X)$, we have a set of parents Pa(X.A), and a conditional probability distribution that represents P(X.A|Pa(X.A)).

An attribute A only has one CPT but can have multiple parents of the same type. This is solved with an aggregation function such as MAX or MIN on the relationships between the attribute and its' parents. The aggregation functions work over all the parents of the same type and find one value to use for the instance attribute CPT.

A PRM thus enables the calculation of the probabilities of various architecture instantiations. This makes it possible to infer the probability that a certain attribute assumes a specific value, given some – possibly incomplete – evidence about the rest of the architecture instantiation.

III. Architectural Metamodel

A PRM can have many perspectives, this PRM consider the probability of successful attacks. In the study that was performed on the Combined Endeavour 07, the main target, and the focus of the analysis of the PRM was to investigate the difficulty of acquiring administrator level rights on a host. The metamodel with attribute relationships is presented in Figure 1.

A. Development of the metamodel

The metamodel was developed in two steps. First, a qualitative core structure describing different system components and their interconnected relationships was developed. Second, quantitative values were added to populate the conditional probability tables. The model was validated by consulting multiple domain experts.

The qualitative structure of the PRM was defined from a literature study that addressed common security components – how they can affect possible attacks and protect other network components. Literature was drawn from NIST [9, 10, 11, 12], NISCC [13, 14] and papers [15,16,17,18,19,20,21,22,24,25,26]. Five classes representing components and a number of component attributes were selected as the most relevant for the model. The selections were deemed to be representative for common protection technologies and possible attack vectors, with a high level of abstraction. With the classes and attributes in place, their relationships with other classes and attributes were addressed. The resulting core model was validated by multiple domain experts in a number of interviews where they confirmed the attribute selections and the definition of the qualitative structure as relevant. The experts consulted were a senior security consultant from a leading European network security firm, as well as multiple senior members and alumni of the student computer association at the Royal Institute of Technology – some of them with many years of experience working in the field of network administration and security.

In the second step of development, quantitative data was added to the model in the construction of the CPTs. The CPTs require a probability distribution to be defined for the conditions introduced by the qualitative dependency structure.

A few statistical studies were found regarding the efficiency of certain security functions given certain conditions. For instance, [27] was used to specify the conditional probability for *Bypass Anomaly Based Detection* in the NIDS class. To complement the literature, the network experts mentioned above were consulted in further interviews. They provided approximations for the probability of certain types of attacks to succeed different conditions, as well as participated in discussions regarding the validity of other numbers drawn from the literature. This way most CPTs were discussed and validated. See part III.D for two examples of attributes.

Table 1 list all attributes in the PRM. This table also shows the literature used to define the qualitative structure and the quantitative parameters associated with each parent. The qualitative structure is the parents defined in the PRM; the

quantitative structure is the CPT for the attribute. The table also lists an uncertainty level for every CPT that is further described in the list of the attributes. The attributes have support in an average of three references.

B. The resulting PRM

Figure 1 shows the final PRM metamodel. The model consists of five *classes* representing common security components in a network: Firewall, Authentication, NIDS, HIDS and Host. The classes have *class relationships* between them, such as Perimeter Defense that represents that a firewall can provide perimeter defense for other components. Every class has a number of *attributes* representing attack steps that can be directed towards the component, or security functions that the component can provide. The latter case represents functions that an intruder must bypass. For instance, the attribute *Exploit Remote Access* in the class *Firewall* represents how an attacker might attempt to gain control over a firewall through remote configuration. Another example is the attribute Bypass Content Filtering representing the chances of bypassing the content filtering of a firewall during an attack. Between the attributes there are arrows showing *parent relationships*. In the class Firewall there is an arrow from Exploit Remote Access to Bypass Content Filtering showing that the probability of bypassing content filtering is influenced by the probability of successfully exploiting the remote configuration, e.g. an attacker might disable the content filtering through remote configuration.

C. List of attributes

Table 1 presents all the attributes in the metamodel along with related references, sorted under their respective classes. In the first column is the name of the attribute. The second column lists the qualitative references that were used to define the parents of the attribute. The third column lists the quantitative references for the CPT of attributes. The asterisks indicate that the corresponding CPTs are defined through the logic of the model structure. For instance: Pa(*SpoofAttack*)= *BypassSpoofCountermeasure* as the only parent. If there is a countermeasure and the bypass attack succeeds then the spoof attack will also succeed. The asterisk references primarily function as logic OR. Double asterisks show tables that are based solely on expert approximation.

The fourth column represents the uncertainty level of the CPT, graded as Low (L) or High (H). Low means that the CPT has multiple sources and that the experts were more certain in their estimates. High means that the CPT should be prioritized for further research as it is primarily based on expert approximation and can be improved with more systematic case studies.

D. Sample attributes

To show the development of the attributes and what they represent in a modeling context, two attributes will be described in detail. The first example is *Bypass Signature-based Detection* from the class NIDS. Signature-based detection is described in [9, 15, 16] and is a core security

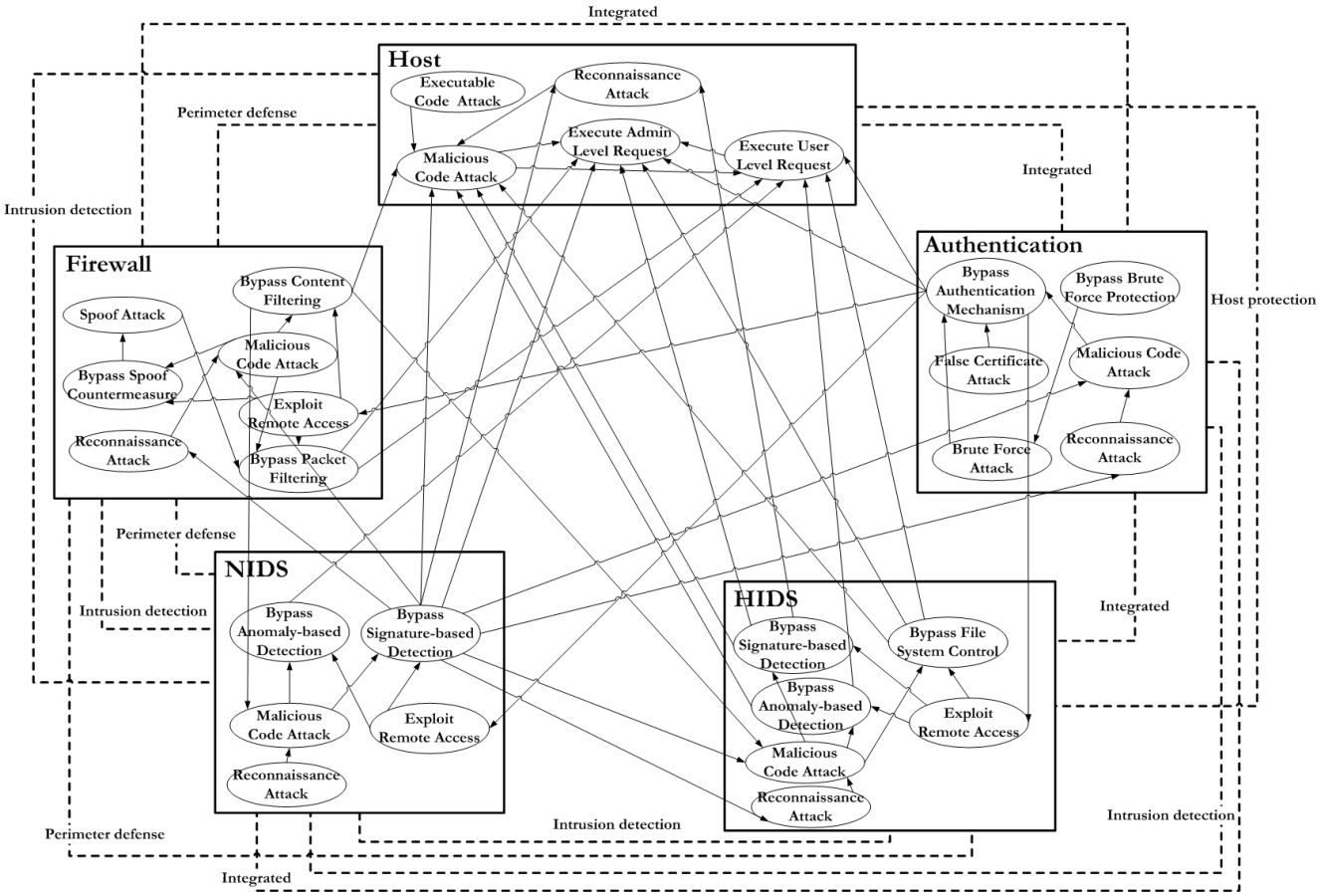


Figure 1: Metamodel for analysis of network security

principle used by different NIDSs. The attribute represents the attack step of evading this detection, which in this paper is described as a bypass attack.

Table 1: List of attributes in metamodel

Classes and attributes	Qualitative	Quantitative	Uncertainty
Firewall Class			
Bypass Packet Filtering	[10,13, 17, 18]	**	H
Spoof Attack	[10, 13,19]	*	L
Bypass Spoof Countermeasure	[10,13,19,11]	**	H
Reconnaissance Attack	[9,15,20,16,21]	**	L
Bypass Content Filtering	[10,17,19]	**	H
Malicious Code Attack	[18,19,22,17]	[23,24]	H
Exploit Remote Access	[10,18]	*	L
Authentication Service Class			
Bypass Authentication mechanism	[17,22,19,25]	*	L
False Certificate Attack	[17,11]	**	H
Brute Force Attack	[11,25]	*	L
Bypass Brute Force Protection	[11,17,25]	[11,17]	H
Reconnaissance Attack	[26]	**	L
Malicious Code Attack	[19,21,26,12]	[23,24]	H
NIDS Class			
Bypass Signature Based Detection	[9,15,16]	[27]	L
Bypass Anomaly Based Detection	[9,15,16,19]	[27]	L
Reconnaissance Attack	[26]	**	L
Malicious Code Attack	[19,28,17,25]	[23,24]	H
Exploit Remote Access	[22]	*	L
HIDS Class			
Bypass Signature Based Detection	[9,15,16]	[27]	L
Bypass Anomaly Based Detection	[9,15,16]	[27]	L
Bypass File System Control	[9,15,20,14]	[29]	L
Exploit Remote Access	[22]	*	L
Reconnaissance Attack	[26]	**	L
Malicious Code Attack	[19,28,17,25]	[24,23]	H
Host Class			
Admin Level Request	[26,14]	**	H
User Level Request	[26,14]	**	H
Malicious Code Attack	[26,22,21]	[23,24]	H
Reconnaissance Attack	[26]	**	L
Executable Code Attack	[19,12]	[30]	L

The attribute has two states, either the attack step succeed in bypassing the detection or it does not: True (T) or False (F) respectively. For this attribute there are two parents: *Malicious Code Attack* and *Exploit Remote Access*. A malicious code attack is an attack where the goal is to exploit a vulnerability in the target through remote code execution. Exploiting remote access refers to an attack against a remote configuration interface on the NIDS. Remote configuration over the network facilitates the work of system administrators, but also represents a significant weakness as an attacker might gain unauthorized access and shut down security measures.

The CPT for the attribute Bypass Signature-based Detection is captured in Table 2. If it is possible to do a malicious code attack against the NIDS, then the signature-based detection will be disabled (the third and fourth columns in Table 2), i.e. *Bypass Signature-based Detection* is true. Analogously, the detection can also be disabled if it is possible to exploit the remote access of the NIDS (column one and three in Table 2). If none of these attacks succeed, then the detection rate of the NIDS defines how often the intrusion will fail, in this case 38 % (the last column in table 2). This value is drawn from a study of Snort without customized signatures or any system-specific training: *"The Snort has a flat low detection rate of 38% with any rate of false alarms."* [27]. The results of the study are generalized and assumed representative for many IDSs, while the other numbers are derived from the modeling logic. Finally, the CPT is assumed to have low uncertainty as the IDS study examines a common version of Snort under given conditions, giving exactly the type of attack statistics preferable for the model.

Table 2: CPT for the attribute Bypass Signature-based Detection

NIDS.MaliciousCodeAttack		T		F	
NIDS.ExploitRemoteAccess		T	F	T	F
NIDS.BypassSignatureBasedDetection	T	1	1	1	0.62
	F	0	0	0	0.38

The second example is the attribute *Bypass Spoof Countermeasure* from the Firewall class. This attribute is described in [10,19,13,17] and represents the ability of the firewall to detect or prevent address spoofing. As in the first example, the attribute has two states in this case signifying whether an attack with a spoofed address bypasses the countermeasure. The parents are *Malicious Code Attack* and *Exploit Remote Access* that both function basically the same as in the first example.

The CPT for *Bypass Spoof Countermeasure* is shown in table 3. If either of the two attacks directed towards the spoof countermeasures succeed, the countermeasures will be disabled and thus bypassed (columns three through five). If both types of attacks fail, then the efficiency of the countermeasures will determine whether the spoofing succeeds. In this case there was no appropriate literature with relevant statistics so the experts were consulted to find the needed numbers. An interview was conducted with three experts, with the first question being a discussion about spoofing and spoof countermeasures in general. Different types of

countermeasures were discussed, such as ingress and egress filtering. The experts were asked to judge the efficiency of such countermeasures for a properly configured network with a security level that could be described as “awareness”, e.g. not the chief concern and neither an ignored subject in the administration of the network.

Table 3: CPT for attribute Bypass Spoof Countermeasure

Firewall.MaliciousCodeAttack		T		F	
Firewall.ExploitRemoteAccess		T	F	T	F
Firewall.BypassSpoofCountermeasure	T	1	1	1	0.05
	F	0	0	0	0.95

According to the experts, spoofing is straightforward to perform and something that an attacker can be assumed to succeed with in nearly every case if there is no spoof countermeasure. If the attacker is unable to exploit remote access or perform a malicious code attack to disable the spoof countermeasures, the attacker might be able to bypass the countermeasures in a small number of cases. The domain experts suggested that approximately 5% of installed firewalls would allow spoof countermeasures to be bypassed even if both parents are false. This is represented in the rightmost column.

IV. Instance Model – CE07

The metamodel was applied in an assessment of network security in the military network Combined Endeavor 07. Combined Endeavor is an annual international interoperability exercise between the defense forces of NATO and Partnership for Peace, including more than 40 nations in 2007. The participants were organized in different regions and every region built an IP-backbone for interconnection of all subnets, transmission links and network components. In the study one region was evaluated through two different scenarios wherein attacks against two targets were modeled. Instance models were created by finding possible attack paths and adding instances of the network components that could affect the hypothetical attack. With the metamodel as a basis, the user needed only provide a small amount of information to perform this analysis. One instance model is shown in Figure 2. Attribute relationships are not presented in the figure due to lack of space, but they of course follow the metamodel.

An attack towards a host behind a Färist firewall, multiple IDSs and with authentication systems is described with an instance model. This gives seven objects that are connected as shown in Figure 2. Evidence is added in the instance model where the attributes of the actual components are known to deviate from the default values of the metamodel. One example is the NIDSs which lack anomaly-based detection, and this is described by setting the attribute *Bypass Anomaly-based Detection* to True.

To find the results, the inferred values of all attributes are calculated in the software Enterprise Architecture Tool (EAT) which is being developed for this type of system evaluation [31]. In this scenario the result is defined as the probability of executing a request in the target, i.e. the value of the attributes *Execute User Level Request* and *Execute Admin Level Request* in the object *Target host*. The result was two percentage values giving an estimate of system security, in this case for both attributes a 0.00702 probability of an attacker successfully executing a request. An alternate scenario was also tested where anomaly-based detection was added to the IDSs, and this gave a value of 0.00364.

V. Discussion

There are some issues regarding scalability, model scope, and data collection that should be noted. The scalability problem of attack graphs is partly solved by this approach as the metamodel and classes compartmentalize many factors

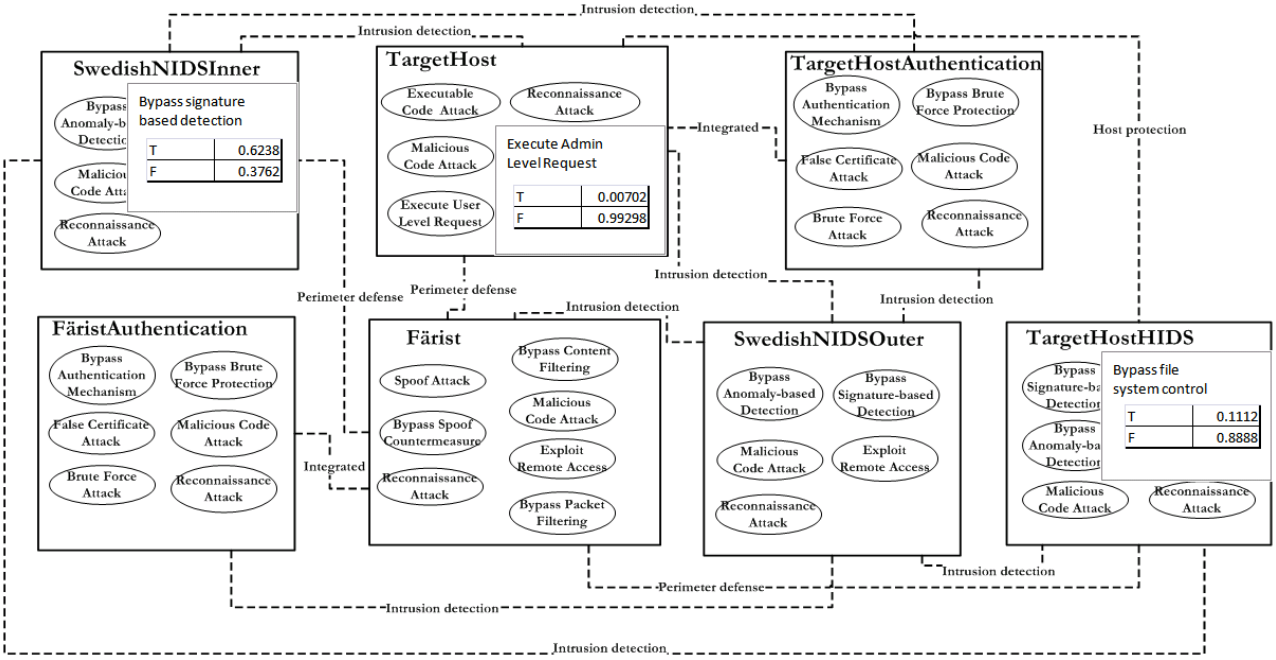


Figure 2: Instance model from security assessment scenario

regarding attacks. While a large network might result in a complex instance model with many classes and class relationships, it is still a relatively low number of entities to handle for the analyst.

Another aspect to consider is the scope of the model. The metamodel constructed for this study has a high level of abstraction rather than extensive technical details, which naturally means that some factors that could be argued to be relevant have been excluded or combined in abstract attributes.

By using probabilistic methods the model considers many factors indirectly. Compared to deterministic methods the probabilistic method does not need to explicitly include every aspect and the abstraction level can be higher. It must be acknowledged though, that the model can be improved upon in this regard. Statistics and averages are easier to apply with a probabilistic method and this is sufficient information for practical use of the PRM. All information does not have to be exact and correct to facilitate a decision when relative comparisons are done.

An additional benefit with this approach is that the instantiated models can be quite easily comprehended by industrial security analysts as well as other system engineers and analysts since the models are not overwhelmingly large. The models are aligned with many commonly used metamodels for specifying system architectures.

When performing the study, the main problem was finding good sources for the large number of probability tables. While some data is based on the consensus of multiple experts' opinions, the model would certainly benefit from more objective data. Almost all the experts found it hard to provide the requested data before they got a rigorous explanation of the method. One expert whom had worked with attack graphs earlier found it much easier than the others. Thus, it is possible to improve the metamodel regarding the level of abstraction and numbers in the probability tables. A further refinement could be done with a deeper level of detail and future research regarding intrusion statistics. With comprehensive component data this might give very accurate results, even though the method aims for simplicity rather than describing the whole truth of security matters.

VI. Conclusions

After applying this method in a study it can be concluded that it is possible to use PRM methodology for security assessments. The development of the metamodel takes time and requires a large amount of data to populate the CPTs with numbers, but it is also a very important step in the process. This is the main strength of the method, as a community of experts can collect and define the necessary theory in a metamodel that an analyst then can apply quickly and easily without the same level of knowledge in security theory.

Another advantage is the flexibility and abstraction in theory construction, as the metamodel can be further customized depending on the exact subject and focus of study. Depending on requirements, organizational policies and the type of network to study, the theory model can be defined to cover the precise aspects needed for a study or an organization. This flexibility also means that the PRM can be extended to cover more classes and attributes.

The developed network PRM and CPTs give a good basis for a basic intrusion and security analysis. While the results from the study give a good guideline and approximation of security, the method does not claim to describe the complete truth. The metamodel focuses on a range of technical aspects with a high level of abstraction regarding subjects such as malicious code and user behavior. Areas such as on-site security, attacker profiles or administrator action are not directly covered and could be added in further refinement of the metamodel.

VII. References

[1] Getoor L, Taskar B. *Introduction to statistical relational learning.* s.l. : MIT Press, 2007.

[2] C. Ramakrishnan, R. Sekar. Model-Based Analysis of Configuration Vulnerabilities. *Proceedings of the 7:th ACM Conference on Computer and Communication Security.* November 2000.

[3] R. Ritchey, P. Ammann. Using Model Checking to Analyze Network Vulnerabilities. *Proceedings of the IEEE Symposium on Security and Privacy.* 2000.

[4] O. Sheyner, J. Haines, S. Jha, R. Lippmann, J.Wing. Automated Generation and analysis of Attack Graphs. *Proceedings of the IEEE Symposium on Security and Privacy.* 2000.

[5] C.Phillips, L.Swiler. A Graph-Based System for Network-Vulnerability Analysis. *Proceedings of the New Security Paradigms Workshop.* 1998.

[6] P. Ammann, D. Wijesekera, S.Kaushik. Scalable, Graph-Based Network Vulnerability Analysis. *Proceedings of the 9:th ACM Conference on Computer and Communications Security.* November 2000.

[7] X. Ou, W. Boyer, M. McQueen. A Scalable Approach to Attack Graph Generation. *Proceedings of the 13:th ACM conference on Computer and Communications security.* 2006.

[8] Jensen, F V. *Bayesian Networks and Decision Graphs.* Secaucus, NJ : Springer New York, 2001.

[9] *Guide to Intrusion Detection and Prevention Systems.* Scarfone K, Mell P. 2007, NIST SP800-94.

[10] *Guidelines on Firewall and Firewall Policy.* Scarfone K, Hoffman P. 2008, NIST SP800-41.

[11] *Recommendation for Key Management.* et al, Barker. 2007, NIST SP800-57.

[12] *An Introduction to Computer Security: The NIST Handbook.* Sterne, et al. 1995, NIST SP800-12.

[13] *Understanding Firewalls.* NISCC. 2005, NISCC Technical Note 10/04.

[14] *Understanding Intrusion Detection Systems.* NISCC. 2003, NISCC Technical Note 09/03.

[15] *Study of Intrusion Detection Systems (IDSs) in Network Security.* Wu J, Hu Z. 2008, Conference on Wireless.

[16] *Intrusion Techniques: Comparative Study of Network Intrusion Detection Systems.* Garuba, et al. 2008, Fifth International Conference on Information Technology: New Generations.

[17] Orrey, et. al. Penetration Testing Framework 0.54. *vulnerabilityassessment.co.uk.* [Online] 10 08, 2009. [Cited: 10 08, 2009.] http:www.vulnerabilityassessment.co.uk.

[18] *FireCracker: A Framework for Inferring Firewall Policies using Smart Probing.* Samak, et al. 2007, IEEE International Conference on Network Protocols.

[19] *Just How Secure Are Security Products?* Geer, D. 2004, Computer June.

[20] *Network Security on the Intrusion Detection System Level.* Vokorokos, et al. 2006, INES.

[21] *Network Intrusion Detection – Automated and Manual Methods Prone to Attack and Evasion.* Chaboya, et al. 2006, IEEE Privacy & Security, Volume 4 Issue 6.

[22] *On the Use of Security Metrics Based on Intrusion Prevention System Event Data: An Empirical Analysis.* Chrun, et al. 2008, 11th IEEE High Assurance Systems Engineering Symposium.

[23] *Common Vulnerability Scoring System.* Scarfone, et al. 2006, Security & Privacy, Nov-Dec.

[24] *Estimating Software Vulnerabilities.* J, Jones. 2007, Security & Privacy, July-Aug.

[25] *Packet Saga, Using Strategic Hacking To Terrorize Commercial And Governmental Entities On The Internet.* Nassar K, Ali W. 2005, 3rd International Conference on Information & Communication Technology.

[26] *Policy Management for Network-based Intrusion Detection and Prevention.* Chen Y, Yang Y. 2004, Network Operations and Management Symposium.

[27] *Defending Distributed Systems Against Malicious Intrusions and Network Anomalies.* Hwang, Chen, Liu. 2005, 19th IEEE International Parallel and Distributed Processing Symposium.

[28] *Towards Survivable Intrusion Detection System.* Yu D, Frincke D. 2004, Proceedings of the 37th Hawaii International Conference on System Sciences.

[29] McAfee. *Anti-Malware Detection Rates Comparative Testing.* s.l. : West Coast Labs , 2008.

[30] Cisco Systems. *Understanding Remote Worker Security: A Survey of User Awareness vs. Behavior.* s.l. : Cisco Systems White Paper, 2006.

[31] Department for industrial information and control systems, KTH. *Project page for Enterprise Architecture Tool.* [Online] http://www.ics.kth.se/eat.

[32] V. Verendel, "Quantified security is a weak hypothesis: a critical survey of results and assumptions," *New Security Paradigms Workshop*, 2009.

[33] Zhang, B., Lu, K., Pan, X., & Wu, Z. (2009). Reverse Search Based Network Attack Graph Generation. In *2009 International Conference on Computational Intelligence and Software Engineering* (pp. 1-4). IEEE. doi: 10.1109/CISE.2009.5365235.

[34] Anming Xie, Guodong Chen, Yonggang Wang, Zhong Chen, Jianbin Hu, "A New Method to Generate Attack Graphs,", pp.401-406, 2009 Third IEEE International Conference on Secure Software Integration and Reliability Improvement, 2009

[35] Roschke, S., Cheng, F., Schuppenies, R., & Meinel, C. (2009). Towards Unifying Vulnerability Information for Attack Graph Construction. In *Proceedings of the 12th International Conference on Information Security* (p. 233). Springer.

[36] Y. Liu, M. Hong, Network vulnerability assessment using Bayesian networks, in proceedings of SPIE, , pp. 61-71, Orlando, Florida, USA, 2005.

[37] Tutus Digital Gatekeepers, Retrieved at: http://www.tutus.se/farist-fw.html, 2010-03-19.

[38] Stouffer, K., Falco, J., & Kent, K. (2008). Guide to Industrial Control Systems (ICS) Security Recommendations of the National Institute of Standards and Technology. *NIST Special Publication*, *800*-82.

[39] Finco, G., & others. (2007). Cyber Security Procurement Language for Control Systems. *Idaho National Labs*, (August).

False Data Injection Attacks in Control Systems

Yilin Mo, Bruno Sinopoli *†

Abstract

This paper analyzes the effects of false data injection attacks on Control System. We assume that the system, equipped with a Kalman filter and LQG controller, is used to monitor and control a discrete linear time invariant Gaussian system. We further assume that the system is equipped with a failure detector. An attacker wishes to destabilize the system by compromising a subset of sensors and sending corrupted readings to the state estimator. In order to inject fake sensor measurements without being detected the attacker needs to carefully design its inputs to fool the failure detector, since abnormal sensor measurements usually trigger an alarm from the failure detector. We will provide a necessary and sufficient condition under which the attacker could destabilize the system while successfully bypassing the failure detector. A design method for the defender to improve the resilience of the CPS against such kind of false data injection attacks is also provided.

1. Introduction

Cyber Physical Systems (CPS) refer to the embedding of widespread sensing, computation, communication and control into physical spaces [1]. Application areas are as diverse as aerospace, chemical processes, civil infrastructure, energy, manufacturing and transportation, most of which are safety-critical. The availability of cheap communication technologies such as the internet makes such infrastructures susceptible to cyber security threats, which may affect national security as some of them, such as the power grid, are vital to the normal operation of our society. Any successful attack may significantly hamper the economy, the environment or may even lead to loss of human life. As a result, security is of primary importance to guarantee safe operation of CPS. The research community has acknowledged the importance of addressing the challenge of designing secure CPS [2] [3].

The impact of attacks on the control systems is addressed in [4]. The authors consider two possible classes of attacks on the CPS: Denial of Service (DoS) attacks and deception attacks (or false data injection attacks). The DoS attack prevents the exchange of information, usually either sensor readings or control inputs between subsystems, while false data injection attack affects the data integrity of packets by modifying their payloads. A robust feedback control design against DoS attacks is further discussed in [5]. We feel that false data injection attacks can be subtler than DoS attacks as they are in principle more difficult to detect and have not been thoroughly investigated. In this paper, we want to analyze the impact of false data injection attacks on control systems.

A significant amount of research effort has been carried out to analyze, detect and handle failures in control systems. Sinopoli et al. study the impact of random packet drops on controller and estimator performance [6]. In [7], the author reviews several failure detection algorithms in dynamic systems. Results from robust control and estimation [8], a discipline that aims at designing controllers and estimators that function properly under uncertain parameters or unknown disturbances, is also applicable to some control system failures. However, a large proportion of the literature assumes that the failure is either random or benign. On the other hand, a cunning attacker can carefully design its attack strategy and deceive both detectors and robust estimators. Hence, the applicability of failure detection algorithms is questionable in the presence of a smart attacker.

Before describing our problem setup we wiah to review some of the existing literature concerning secure data aggregation over networks in the presence of compromised sensors. In [9], the author provides a general framework to evaluate how resilient the aggregation scheme is against compromised sensor data. Liu et al. study the estimation scheme in power grids and show

*Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA. Email: ymo@andrew.cmu.edu, brunos@ece.cmu.edu

†This research is supported in part by CyLab at Carnegie Mellon under grant DAAD19-02-1-0389 from the Army Research Office Foundation. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of ARO, CMU, or the U.S. Government or any of its agencies.

that under some assumptions the attacker can modify the state estimate undetected [10]. However, in both studies, the authors only consider static systems with estimators that rely exclusively upon current sensor measurements. In reality, in a control system the actions taken by the attacker will not only affect the current states but also the future ones. An attacker could potentially use this fact to perform its attack over time and destabilize the system. On the other hand, the dynamics of the system could be used by the failure detector since the attack may be detected in the near future even if it results undetectable when it first occurs.

In this paper, we study the effects of false data injection attacks on control systems. We assume that the control system, which is equipped with a Kalman filter, LQG controller and failure detector, is monitoring and controlling a linear time-invariant system. The attacker's goal is to destabilize the system by compromising a subset of sensors and sending altered readings to the state estimator. The attacker also wants to guarantee that its action can bypass the failure detector. Under these assumptions, we will give a necessary and sufficient condition under which the attacker could destabilize the system without being detected.

The rest of the paper is organized as follows: In Section 2 we formulate the problem by revisiting and adapting Kalman filter, LQG controller and failure detector to our scenario. In Section 3, we define the threat model of false data injection attacks. In Section 4 we prove a necessary and sufficient condition under which the attacker could destabilize the system. We will also give some design criteria to improve the resilience of the CPS against false data injection attacks. A numerical example is provided in Section 5 to illustrate the effects of false data injection attacks on the CPS. Finally Section 6 concludes the paper.

2. Problem Formulation

In this section we model the CPS as a linear control system, which is equipped with a Kalman filter, LQG controller and failure detector.

2.1. Physical System

We assume that the physical system has Linear Time Invariant (LTI) dynamics, which take the following form:

$$x_{k+1} = Ax_k + Bu_k + w_k, \tag{1}$$

where $x_k \in \mathbb{R}^n$ is the vector of state variables at time k, $u_k \in \mathbb{R}^p$ is the control input, $w_k \in \mathbb{R}^n$ is the process noise at time k and x_0 is the initial state. w_k, x_0 are independent Gaussian random variables, and $x_0 \sim \mathcal{N}(0,\, \Sigma)$, $w_k \sim \mathcal{N}(0,\, Q)$.

2.2. Kalman filter

A sensor network is deployed to monitor the system described in (1). At each step all the sensor readings are collected and sent to a centralized estimator. The observation equation can be written as

$$y_k = Cx_k + v_k, \tag{2}$$

where $y_k = [y_{k,1}, \ldots, y_{k,m}]^T \in \mathbb{R}^m$ is a vector of measurements from the sensors, and $y_{k,i}$ is the measurement made by sensor i at time k. $v_k \sim \mathcal{N}(0,\, R)$ is the measurement noise independent of x_0 and w_k.

A Kalman filter is used to compute state estimation $\hat{x}_k$ from observations y_ks:

$$\hat{x}_{0|-1} = 0,\, P_{0|-1} = \Sigma, \tag{3}$$

$$\begin{aligned} \hat{x}_{k+1|k} &= A\hat{x}_k + Bu_k,\, P_{k+1|k} = AP_kA^T + Q, \\ K_k &= P_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1}, \\ \hat{x}_k &= \hat{x}_{k|k-1} + K_k(y_k - C\hat{x}_{k|k-1}), \\ P_k &= P_{k|k-1} - K_kCP_{k|k-1}. \end{aligned} \tag{4}$$

Although the Kalman filter uses a time varying gain K_k, it is well known that this gain will converge if the system is detectable. In practice the Kalman gain usually converges in a few steps. We can safely assume the Kalman filter to be already in steady state. Let us define

$$P \triangleq \lim_{k\to\infty} P_{k|k-1},\, K \triangleq PC^T(CPC^T + R)^{-1}. \tag{5}$$

The update equations of Kalman filter are as follows:

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + K\left[y_{k+1} - C(A\hat{x}_k + Bu_k)\right], \tag{6}$$

For future analysis, let us define the residue z_{k+1} at time $k+1$ to be

$$z_{k+1} \triangleq y_{k+1} - C(A\hat{x}_k + Bu_k). \tag{7}$$

(6) can be simplified as

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + Kz_{k+1}. \tag{8}$$

The estimation error e_k at time k is defined as

$$e_k \triangleq x_k - \hat{x}_k. \tag{9}$$

Manipulating (6), (7), we get the following recursive equation:

$$e_{k+1} = (A - KCA)e_k + (I - KC)w_k - Kv_k. \tag{10}$$

2.3. LQG Controller

An LQG controller is used to stabilize the system by minimizing the following objective function[1]:

$$J = \lim_{T\to\infty} \min_{u_0,\ldots,u_T} E\frac{1}{T}\left[\sum_{k=0}^{T-1}(x_k^T W x_k + u_k^T U u_k)\right], \quad (11)$$

where W, U are positive semidefinite matrices and u_k is measurable with respect to $y_0,\ldots,y_k$, i.e. u_k is a function of previous observations. It is well known that the optimal controller of the above minimization problem is a fixed gain controller, which takes the following form:

$$u_k = -(B^T S B + U)^{-1} B^T S A \hat{x}_k, \quad (12)$$

where u_k is the optimal control input and S satisfies the following Riccati equation

$$S = A^T S A + W - A^T S B (B^T S B + U)^{-1} B^T S A. \quad (13)$$

Let us define $L \triangleq -(B^T S B + U)^{-1} B^T S A$, then $u_k = L x_{k|k}$.

The systems is stable if and only if $Cov(e_k)$ and J are both bounded. In particular that implies both matrices $A - KCA$ and $A + BL$ are stable. In the rest of the paper, we will only consider stable systems. Further, we assume to be already in steady state, which means $\{x_k, y_k, \hat{x}_k\}$are stationary random processes.

2.4. Failure Detector

A failure detector is often used in control system. For example, a χ^2 failure detector computes the following quantity

$$g_k = z_k^T \mathscr{P}^{-1} z_k, \quad (14)$$

where $\mathscr{P}$ is the covariance matrix of the residue z_k. Since z_k is Gaussian distributed, g_k is χ^2 distributed with m degrees of freedom. As a result, g_k cannot be far away from 0. The χ^2 failure detector will compare g_k with a certain threshold. If g_k is greater than the threshold, then an alarm will be triggered.

Other types of failure detectors have also been considered by many researchers. In [11] [12], the authors design a linear filter other than the Kalman filter to detect sensor shift or shift in matrices A and B. The gain of such filter is chosen to make the residue of the filter more sensitive to certain shift, which helps to detect a particular failure. Willsky et al. A generalized likelihood ratio test to detect dynamics or sensor jump is also proposed by Willsky et al. in [13].

[1] We assume an infinite horizon LQG controller is implemented.

To make the discussion more general, we assume the detector implemented in the CPS triggers an alarm based on following event:

$$g_k > threshold, \quad (15)$$

where g_k is defined as

$$g_k \triangleq g(z_k, y_k, \hat{x}_k, \ldots, z_{k-\mathscr{T}+1}, y_{k-\mathscr{T}+1}, \hat{x}_{k-\mathscr{T}+1}). \quad (16)$$

The function g is continuous and $\mathscr{T} \in \mathbb{N}$ is the window size of the detector. It is easy to see for χ^2 detector, $g_k = z_k^T \mathscr{P}^{-1} z_k$. We further define the probability of alarm for the failure detector to be

$$\beta_k = P(g_k > threshold). \quad (17)$$

At a first glance, it seems that certain choice of g function will affect detection differently. However, since the χ^2 detector along with many other detectors performs detection by computing a certain function of $\hat{x}_k, y_k, z_k$, then none of these detectors will be able to distinguish the healthy system from the partial compromised system if, under the malicious attack, the vectors $\hat{x}_k, y_k, z_k$ have the same statistical properties as those of healthy system. In Section 4, we show how the attacker can systematically attack the system without being noticed by the failure detector if a particular algebraic condition holds.

3. False Data Injection Attacks

In this section, we assume that a malicious third party wants to compromise the integrity of the system described in Section 2. The attacker is assumed to have the following capabilities:

1. It knows the system model: We assume that the attacker knows matrices A, B, C, Q, R as described in Section 2 and the observation gain and control gain K, L.

2. It can control the readings of a subset of the sensors, denoted by S_{bad}. As a result, (2) now becomes
$$y'_k = Cx'_k + v_k + \Gamma y^a_k, \quad (18)$$
where $\Gamma = diag(\gamma_1,\ldots,\gamma_m)$ is the sensor selection matrix. γ_i is a binary variable and $\gamma_i = 1$ if and only if $i \in S_{bad}$. y^a_k is the malicious input from the attacker. Here we write the observations and states as y'_k and x'_k since they are in general different from those of the healthy system due to the malicious attack.

3. The intrusion begins at time 0. As a result, the initial conditions for the partial compromised system will be $\hat{x}'_{-1} = 0$, $Ex_0 = 0$.

Figure 1 shows the diagram of the partial compromised system.

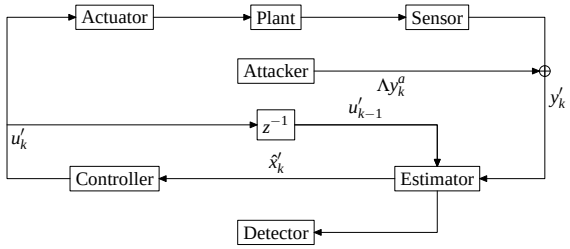


Figure 1. System Diagram

Definition 1. *An attack sequence $\mathscr{Y}$ is defined as an infinite sequence which takes the following form $y_0^a, y_1^a, \ldots$.*

It is easy to see that all the states of the partially compromised system are a function of $\mathscr{Y}$. For example, x_k' can be written as $x_k'(\mathscr{Y})$. However, in order to simplify the notation, we will use x_k' when there is no confusion. Under the previous assumptions, the new system dynamics can be written as

$$\begin{aligned} x_{k+1}' &= Ax_k' + Bu_k' + w_k, \\ y_k' &= Cx_k' + v_k + \Gamma y_k^a, \\ \hat{x}_{k+1}' &= A\hat{x}_k' + Bu_k' + K\left[y_{k+1}' - C(A\hat{x}_k' + Bu_k')\right], \\ u_k' &= L\hat{x}_k'. \end{aligned} \tag{19}$$

We can also define the new residue and estimation error respectively as

$$z_{k+1}' \triangleq y_{k+1}' - C(A\hat{x}_k' + Bu_k'),\ e_k' \triangleq x_k' - \hat{x}_k'. \tag{20}$$

Finally, the new probability of alarm is defined as

$$\beta_k' = P(g_k' > threshold), \tag{21}$$

where

$$g_k' \triangleq g(z_k', y_k', \hat{x}_k', \ldots, z_{k-\mathscr{T}+1}', y_{k-\mathscr{T}+1}', \hat{x}_{k-\mathscr{T}+1}'). \tag{22}$$

The differences between the two systems are defined as

$$\begin{aligned} \Delta x_k \triangleq\ & x_k' - x_k, \Delta\hat{x}_k \triangleq && \hat{x}_k' - \hat{x}_k, \\ \Delta u_k \triangleq\ & u_k' - u_k, \Delta y_k \triangleq && y_k' - y_k, \\ \Delta z_k \triangleq\ & z_k' - z_k, \Delta e_k \triangleq && e_k' - e_k, \Delta\beta_k = \beta_k' - \beta_k, \end{aligned} \tag{23}$$

where $x_k, \hat{x}_k, y_k, u_k, \beta_k$ are given by equations (1), (2), (6), (12), (17). $\Delta x_k, \Delta\hat{x}_k, \Delta u_k, \Delta y_k, \Delta z_k, \Delta e_k, \Delta\beta_k$ represent the differences between the partially compromised system and the healthy system.

The following definition defines what constitutes a "successful" attack.

Definition 2. *An attack sequence $\mathscr{Y}$ is (ε,α)-successful if there exists $T \in \mathbb{N}$, such that the following holds:*

$$\|\Delta x_T(\mathscr{Y})\| \geq \alpha, \Delta\beta_k(\mathscr{Y}) \leq \varepsilon, \forall k = 0, 1, \ldots, T-1.$$

The system is called (ε,α)-attackable if there exists a (ε,α)-successful attack sequence $\mathscr{Y}$ on the CPS.

Remark 1. *It is worth noticing that simply injecting a large y_k^a will result in a large Δz_k which, in turn, will induce the failure detector to trigger an alarm immediately.*

Although the definition of (ε,α)-attackable is simple, it is not so easy to verify whether a system is (ε,α)-attackable, especially when the form of g is complex. As a result, we will consider a limit case of (ε,α)-attackability.

Definition 3. *A control system is perfectly attackable if there exists an attack sequence $\mathscr{Y}$ such that the following holds:*

$$\limsup_{k\to\infty} \|\Delta x_k(\mathscr{Y})\| = \infty, \|\Delta z_k(\mathscr{Y})\| \leq 1, \forall k = 0, 1, \ldots, .$$

The next theorem shows that perfect attackability implies (ε,α)-attackability.

Theorem 1. *If a control system is perfectly attackable, then it is also (ε,α)-attackable for any $\varepsilon,\alpha > 0$,*

Proof. Since the system is perfectly attackable, there exists an attack sequence $\mathscr{Y}$, such that

$$\limsup_{k\to\infty} \|\Delta x_k(\mathscr{Y})\| = \infty, \|\Delta z_k(\mathscr{Y})\| \leq 1, k = 0, 1, \ldots \tag{24}$$

Manipulating equations (6) (12) (19), we can prove that:

$$\begin{aligned} \Delta\hat{x}_{k+1} &= (A+BL)\Delta\hat{x}_k + K\Delta z_{k+1}, \\ \Delta y_{k+1} &= \Delta z_{k+1} + C(A+BL)\Delta\hat{x}_k. \end{aligned} \tag{25}$$

Stability of $A+BL$ is guaranteed by the stability of the original system. Therefore, if $\|\Delta z_k(\mathscr{Y})\| \leq 1$ for all $k = 0, 1, \ldots$, then $\Delta\hat{x}_k(\mathscr{Y})$ and $\Delta y_k(\mathscr{Y})$ will be uniformly bounded for all k. Define the bounds to be

$$M_1 = \sup_k \|\Delta\hat{x}_k(\mathscr{Y})\|, M_2 = \sup_k \|\Delta y_k(\mathscr{Y})\|, \tag{26}$$

where $M_1, M_2 < \infty$ are constants. Due to the continuity of g, there exists $\varepsilon' > 0$ such that if $\|\Delta z_k\| \leq \varepsilon'$, $\|\Delta\hat{x}_k\| \leq \varepsilon'$, $\|\Delta y_k\| \leq \varepsilon'$, then

$$|P(g_k' > threshold) - P(g_k > threshold)| \leq \varepsilon,$$

Since $\Delta z_k(\mathscr{Y}), \Delta\hat{x}_k(\mathscr{Y}), \Delta y_k(\mathscr{Y})$ are uniformly bounded, by linearity, we can find $\delta > 0$, such that

$$\|\Delta z_k(\delta\mathscr{Y})\| \le \varepsilon', \|\Delta\hat{x}_k(\delta\mathscr{Y})\| \le \varepsilon', \|\Delta y_k(\delta\mathscr{Y})\| \le \varepsilon', \forall k.$$

By the stationarity of the random process $\{x_k, y_k, \hat{x}_k\}$, we know that

$$|P(g'_k > threshold) - P(g_k > threshold)| \le \varepsilon, \forall k.$$

Finally by linearity,

$$\limsup_{k\to\infty} \Delta x_k(\delta\mathscr{Y}) = \delta \limsup_{k\to\infty} \Delta x_k(\mathscr{Y}) = \infty.$$

Hence, $\delta\mathscr{Y}$ is an (ε,α)-successful attack sequence and the system is (ε,α)-attackable. □

In the next section, we will give a necessary and sufficient condition for a system to be perfectly attackable.

4. Main Result

In this section, we will provide an algebraic condition to identify perfectly attackable system, which is given by the following theorem:

Theorem 2. *The control system (1) is perfectly attackable if and only if A has an unstable eigenvalue and the corresponding eigenvector v satisfies:*

1. $Cv \in span(\Gamma)$, *where* $span(\Gamma)$ *is the column space of* Γ.
2. *v is a reachable state of the dynamic system* $\Delta e_{k+1} = (A - KCA)\Delta e_k - K\Gamma y^a_{k+1}$.

Before proving the theorem, we need the following lemmas:

Lemma 1. *The CPS is perfectly attackable if and only if there exists an attack sequence* $\mathscr{Y}$ *such that*

$$\limsup_{k\to\infty} \|\Delta e_k\| = \infty, \|\Delta z_k\| \le 1, k = -1, 0, \ldots \quad (27)$$

Proof. The proof follows from the boundedness of $\Delta\hat{x}_k$ and the fact that $\Delta x_k = \Delta\hat{x}_k + \Delta e_k$. Due to space limitation the complete proof will be omitted. □

Using Lemma 1, we can use Δe_k to prove that the system is perfectly attackable. The main advantages of substituting Δx_k with Δe_k is that Δe_k follows a simpler recursive equation:

$$\Delta e_{k+1} = (A - KCA)\Delta e_k - K\Gamma y^a_{k+1}. \quad (28)$$

Moreover,

$$\Delta z_{k+1} = CA\Delta e_k + \Gamma y^a_{k+1}. \quad (29)$$

Before proving Theorem 2, we need an additional lemma:

Lemma 2. *Let* $p \in \mathbb{R}^n$ *be a vector, and* $\lim_{k\to\infty} A^k p \neq 0$, *then there exists an unstable eigenvector v of matrix A, such that* $p \in span(p, A^2 p, \ldots, A^{n-1} p)$.

The proof is based on the Jordan decomposition of the A matrix and on Carley-Hamilton Theorem. The complete proof is omitted due to space limits. Now we are ready to prove Theorem 2.

Proof of Theorem 2. First we will prove the necessity. Suppose that CPS is perfectly attackable, then by Lemma 1, there exists an successful attack sequence $\mathscr{Y}$ such that

$$\limsup_{k\to\infty} \|\Delta e_k\| = \infty, \|\Delta z_k\| \le 1, k = 0, 1, \ldots.$$

A peak subsequence $\{\Delta e_{i_k}\}$ from Δe_i is defined as

$$\Delta e_{i_0} = \Delta e_0, \Delta e_{i_k} = \min\{j : \|\Delta e_j\| > \|\Delta e_{i_{k-1}}\|\}, \quad (30)$$

which means that the norm $\|\Delta e_{i_k}\|$ is larger than the norm of any preceding term in the original sequence. Since Δe_k is unbounded, there always exists such a subsequence and $\lim_{k\to\infty} \Delta e_{i_k} = \infty$. Now consider the normalized vectors defined as

$$p_k \triangleq \frac{1}{\|\Delta e_k\|} \Delta e_k. \quad (31)$$

It is trivial to see $\|p_k\|$ is bounded. As a result, there exists an index set $\{j_k\} \subset \{i_k\}$ such that all of the subsequences $\{p_{j_k}\}, \{p_{j_k-1}\}, \ldots, \{p_{j_k-n+1}\}$ converge as k goes to infinity, due to Bolzano-Weierstrass theorem. Let us define

$$q_l \triangleq \lim_{k\to\infty} p_{j_k-l}, l = 0, 1, \ldots, n-1. \quad (32)$$

In addition, since

$$\|\Delta e_{k+1}\| = \|A\Delta e_k - K\Delta z_{k+1}\| \le \|A\|\|\Delta e_k\| + \|K\|,$$

and Δe_{j_k} is unbounded, $\lim_{k\to\infty} \Delta e_{j_k-l} = \infty$ for all l from 0 to $n-1$. As a result

$$\lim_{k\to\infty} \frac{\Delta e_{j_k}}{\|\Delta e_{j_k-1}\|} = \lim_{k\to\infty} \frac{A\Delta e_{j_k-1} - K\Delta z_{j_k}}{\|\Delta e_{j_k-1}\|}$$
$$= A \lim_{k\to\infty} \frac{\Delta e_{j_k-1}}{\|\Delta e_{j_k-1}\|} = Aq_1.$$

Therefore

$$q_0 = \lim_{k\to\infty} \frac{\|\Delta e_{j_k-1}\|}{\|\Delta e_{j_k}\|} \lim_{k\to\infty} \frac{\Delta e_{j_k}}{\|\Delta e_{j_k-1}\|} = Aq_1/\|Aq_1\|.$$

Similarly, it is easy to show that $q_l = Aq_{l+1}/\|Aq_{l+1}\|$. Hence,

$$span(q_0, \ldots, q_{n-1}) = span(A^{n-1}q_{n-1}, \ldots, Aq_{n-1}, q_{n-1}).$$

By definition of $\{\Delta e_{i_k}\}$, $\|\Delta e_{j_k}\| \geq \|\Delta e_{j_k-1}\|$. Thus, $\|Aq_1\| \geq \|q_1\|$, which implies that $\lim_{k\to\infty} A^k q_{n-1} \neq 0$. From Lemma 2 it follows that there exists an unstable eigenvector v in the span of $q_0,\ldots,q_{n-1}$. Since

$$\|\frac{\Delta z_{j_k+1}}{\|\Delta e_{j_k}\|}\| = \|Cp_{j_k} + \Gamma\frac{y^a_{j_k+1}}{\|\Delta e_{j_k}\|}\| \leq \frac{1}{\|\Delta e_{j_k}\|},$$

$$Cp_{j_k} \in span(\Gamma) + B(0,(\|\Delta e_{j_k}\|)^{-1}),$$

where $B(0,(\|\Delta e_{j_k}\|)^{-1})$ is a ball center at 0 with radius $(\|\Delta e_{j_k}\|)^{-1}$. As a result

$$Cq_0 \in \bigcap_{l=1}^{\infty}\left[span(\Gamma) + B(0,(\|\Delta e_{j_k}\|)^{-1})\right] = span(\Gamma).$$

Similarly, CAq_l belongs to $span(\Gamma)$ for all l from 0 to $n-1$. As a result, $CAv \in span(CAq_0,\ldots,CAq_{n-1}) \subset span(\Gamma)$, which implies $Cv \in span(\Gamma)$.

For reachability, since Δe_k is reachable, $\alpha\Delta e_k$ is reachable for any $\alpha \in \mathbb{R}$. In particular, p_k is reachable for all k. Since the reachable subspace is closed, the limit q_l is reachable, which implies v is reachable, thus proving the necessary condition.

We now want to prove sufficiency. Since $Cv \in span(\Gamma)$, there exists y^* such that $\Gamma y^* = Cv$. Furthermore, since v is reachable, there exist $y^a_0,\ldots,y^a_{n-1}$, where n is the dimension of state space, such that $\Delta e_{n-1} = v$. Define

$$M = \max_{k=0,\ldots,n-1}\|\Delta z_k\|. \tag{33}$$

By linearity, if the attacker injects $y^a_0/M,\ldots,y^a_{n-1}/M$, then $\Delta e_{n-1} = v/M$ and $\|\Delta z_k\| \leq 1$ for $k = 0,\ldots,n-1$. As a result, the attacker could choose the attack sequence to be

$$y^a_{n+i} = y^a_i - \frac{\lambda^{i+1}}{M}y^*, i = 0,1,\ldots \tag{34}$$

One can prove that with the above attack sequence, the following equality and inequality hold :

$$\Delta e_{n+i} = \Delta e_i + \frac{\lambda^{i+1}}{M}v, i = 0,1,\ldots, \tag{35}$$

$$\|\Delta z_{n+i}\| = \|\Delta z_i\| \leq 1, i = 0,1,\ldots \tag{36}$$

Since $|\lambda| \geq 1$, $\Delta e_k \to \infty$, which implies that the system is perfectly attackable. □

Remark 2. *The attacker could use the results of Theorem 2 to design an attack sequence $\mathscr{Y}$ based on the eigendecomposition of A and the Γ matrix.*

On the other hand, the defender could also perform an eigendecomposition on A matrix, find all the unstable eigenvector v and then compute Cv. For each Cv, the non-zero elements will indicate the sensors needed by the attacker to perform a successful attack along direction v. Therefore if Cv is a sparse vector, an attacker could initiate an attack on the direction of v by compromising only a few sensors. As a result, the defender could increase the resilience of the system by installing redundant sensors to measure mode v.

5. Illustrative Examples

In this section, we will provide a numerical example to illustrate the effects of false data injection attacks.

Consider a vehicle moving along the x-axis. The state space includes position x and velocity $\dot{x}$ of the vehicle. An actuator is used to control the speed of the vehicle. As a result, the system dynamics is as follows:

$$\begin{aligned}\dot{x}_{k+1} &= \dot{x}_k + u_k + w_{k,1},\\ x_{k+1} &= x_k + (\dot{x}_{k+1} + \dot{x}_k)/2 + w_{k,2}\\ &= x_k + \dot{x}_k + u_k/2 + w_{k,1}/2 + w_{k,2},\end{aligned} \tag{37}$$

which can be written in the matrix form as

$$X_{k+1} = \begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}X_k + \begin{bmatrix}1\\ 0.5\end{bmatrix}u_k + w_k, \tag{38}$$

where

$$X_k = \begin{bmatrix}\dot{x}\\ x\end{bmatrix}, w_k = \begin{bmatrix}w_{k,1}\\ w_{k,2} + 0.5w_{k,1}\end{bmatrix}. \tag{39}$$

Suppose two sensors are measuring the velocity and position respectively. Hence

$$y_k = X_k + v_k. \tag{40}$$

We assume the position sensor is compromised, i.e. $\Gamma = diag(0,1)$. We further impose the following parameters on the system

$$Q = R = W = I_2, U = 1.$$

The steady state Kalman gain and the LQG control gain under the previous assumptions are respectively

$$K = \begin{bmatrix}0.5939 & 0.0793\\ 0.0793 & 0.6944\end{bmatrix}, L = \begin{bmatrix}-1.0285 & -0.4345\end{bmatrix}.$$

Since $[01]'$ is an unstable eigenvector and is in the span of Γ and reachable, by Theorem 2, the system is perfectly attackable. Using the result we derived in Section 4, we design the attack sequence $\mathscr{Y}$ to be

$$\begin{aligned}&y^a_0 = [0,-1.000]', y^a_1 = [0,-0.367]',\\ &y^a_k = y^a_{k-2} - [0,-0.485]', k \geq 2.\end{aligned} \tag{41}$$

Figure 2 shows the evolution of the ΔX_k and Δz_k. It is easy to see that $\|\Delta z_k\|$ is always less than 1 and Δx_k goes to infinity, showing that the system is perfectly attackable.

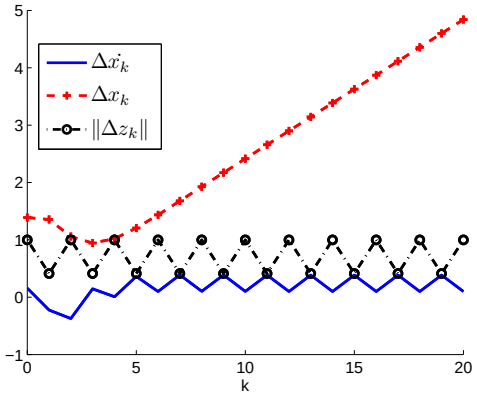


Figure 2. Evolution of $\Delta\dot{x}_k$, Δx_k, $\|\Delta z_k\|$

6. Conclusion and Future Work

This paper proposes a false data injection attack model and analyze the effects of such kind of attacks on a linear time-invariant Gaussian control system. We prove the existence of a necessary and sufficient condition under which the attack could destabilize the system while successfully bypassing a large set of possible failure detectors. We also provide a design criterion to improve the resilience of the system to false data injection attacks.

Future work will be directed toward deriving conditions under which the system is (ε, α)-attackable. We also plan to combine both the false data injection attacks and DoS attacks and study their effects on control systems.

References

[1] E. A. Lee, "Cyber physical systems: Design challenges," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2008-8, Jan 2008. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-8.html

[2] E. Byres and J. Lowe, "The myths and facts behind cyber security risks for industrial control systems," in *Proceedings of the VDE Kongress*. VDE Congress, 2004.

[3] A. A. Cárdenas, S. Amin, and S. Sastry, "Research challenges for the security of control systems," in *HOTSEC'08: Proceedings of the 3rd conference on Hot topics in security*. Berkeley, CA, USA: USENIX Association, 2008, pp. 1–6.

[4] ——, "Secure control: Towards survivable cyber-physical systems," in *Distributed Computing Systems Workshops, 2008. ICDCS '08. 28th International Conference on*, June 2008, pp. 495–500.

[5] S. Amin, A. Cardenas, and S. S. Sastry, "Safe and secure networked control systems under denial-of-service attacks." in *Hybrid Systems: Computation and Control*. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, April 2009, pp. 31–45. [Online]. Available: http://chess.eecs.berkeley.edu/pubs/597.html

[6] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. Sastry, "Foundations of control and estimation over lossy networks," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 163–187, Jan. 2007.

[7] A. Willsky, "A survey of design methods for failure detection in dynamic systems," *Automatica*, vol. 12, pp. 601–611, Nov 1976.

[8] R. Stengel and L. Ryan, "Stochastic robustness of linear time-invariant control systems," *Automatic Control, IEEE Transactions on*, vol. 36, no. 1, pp. 82–87, Jan 1991.

[9] D. Wagner, "Resilient aggreagion in sensor networks," in *ACM Workshop on Security of Ad Hoc and Sensor Networks*, Oct 25 2004.

[10] Y. Liu, P. Ning, and M. Reiter, "False data injection attacks against state estimation in electric power grids," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, November 2009.

[11] R. V. Beard, "Failure accommodation in linear systems through self-reorganization," Man Vehicle Laborotory, Cambrideg, Massachusetts, Tech. Rep. MVT-71-1, February 1971.

[12] H. L. Jones, "Failure detection in linear systems," Ph.D. dissertation, M.I.T., Cambridge, Massachusetts, 1973.

[13] A. S. Willsky and H. L. Jones, "A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems," *IEEE Transactions on Automatic Control*, vol. 21, pp. 108–112, Feburary 1976.

On Security Indices for State Estimators in Power Networks

Henrik Sandberg, André Teixeira, and Karl H. Johansson

***Abstract*— In this paper, we study stealthy false-data attacks against state estimators in power networks. The focus is on applications in SCADA (Supervisory Control and Data Acquisition) systems where measurement data is corrupted by a malicious attacker. We introduce two security indices for the state estimators. The indices quantify the least effort needed to achieve attack goals while avoiding bad-data alarms in the power network control center (stealthy attacks). The indices depend on the physical topology of the power network and the available measurements, and can help the system operator to identify sparse data manipulation patterns. This information can be used to strengthen the security by allocating encryption devices, for example. The analysis is also complemented with a convex optimization framework that can be used to evaluate more complex attacks taking model deviations and multiple attack goals into account. The security indices are finally computed in an example. It is seen that a large measurement redundancy forces the attacker to use large magnitudes in the data manipulation pattern, but that the pattern still can be relatively sparse.**

I. Introduction

In Fig. 1, a schematic block diagram of a modern power network control sytstem is shown. The power network models we consider are on the transmission level. They should be thought of as large and consisting of up to hundreds of buses that are spread out over a large geographic area (a region in a country, for example). To monitor and control the behavior of such large-scale systems, SCADA (Supervisory Control and Data Acquisition) systems are used to transmit measurements, status information, and circuit-breaker signals to and from Remote Terminal Units (RTUs) that are connected to substations, see [1]–[3]. For such large-scale systems, lost data and failing sensors are common. The incoming data is therefore often fed to a so-called *state estimator* which provides Energy Management Systems (EMS) and the human operator in the control center with hopefully accurate information at all times.

The technology and the use of the SCADA systems have evolved quite a lot since the 1970s when they were introduced. The early systems were mainly used for logging data from the power network. Today a modern system is supported by EMS such as automatic generation control (AGC), optimal power flow analysis, and contingency analysis (CA), as is indicated in Fig 1. With the advent of new sensors such as PMUs (Phasor Measurement Units), so-called Wide-Area Monitoring and Control Systems (WAMS/WAMC) will also

This work is supported in part by the European Commission through the VIKING project, and the ACCESS Linnaeus Center at KTH.

H. Sandberg, A. Teixeira, and K. H. Johansson are with the Automatic Control Lab, School of Electrical Engineering, Royal Institute of Technology, 100 44 Stockholm, Sweden. {hsan,andretei,kallej}@ee.kth.se

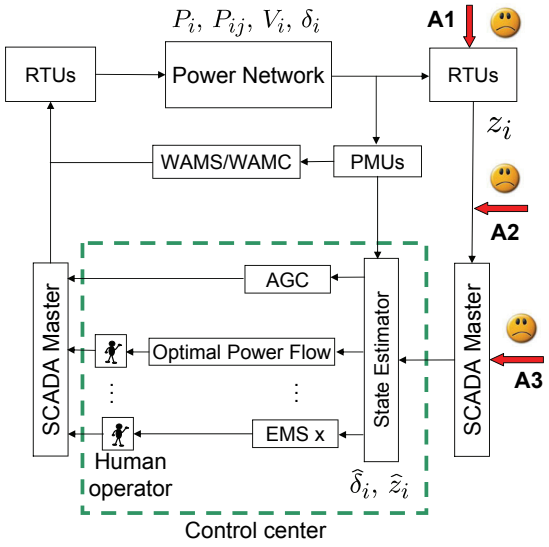


Fig. 1. A schematic block diagram of a power network, a SCADA system, and a control center. Noisy measurements (z_i) of power flows (P_i, P_{ij}) are sent over the SCADA system to the state estimator where estimates of for example the bus phase angels ($\hat{\delta}_i$) are computed. The effect of manipulations on the measurement data z_i are considered in this paper. The manipulations can arise from attacks at various levels A1–A3 in the system. Figure adapted from [4].

be introduced. This provides yet another layer of control in the modern power network control systems. One motivation for this paper is that SCADA/EMS systems are increasingly more connected to office LANs in the control center. Thus these critical infrastructure systems are potentially accessible from the internet. The SCADA communication network is also heterogeneous and consists of fibre optics, satellite, and microwave connections. Data is often sent without encryption. Therefore many potential security threats exist for modern power control systems, as has been pointed out in for example [4].

The focus of this work is on the state estimator and its so-called Bad Data Detection (BDD) system that is used to remove faulty data, see [2], [3], [5]. The BDD system works by checking that the received data (z_i in Fig. 1) reasonably well matches a physical model of the power network. In the recent paper [6], it was shown how an attacker can avoid triggering the BDD system by coordinated attacks on the measurement data z_i. The attacker can corrupt these data by attacking the RTUs (A1), by tampering with the heterogeneous communication network (A2), or by breaking into the SCADA system through the control center office LAN (A3). In this paper, we further analyze this problem and quantify how sensitive the state estimator is to these

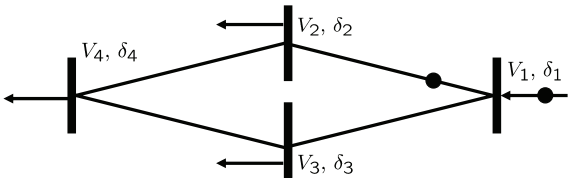


Fig. 2. A simple 4-bus power network. Each bus has a voltage (V_i) and phase angle (δ_i) associated to it. The dots indicate available active power flow measurements.

attacks.

A. Related Work and Contribution of This Paper

False-data injection attacks in power networks were first studied in [6], to the authors' best knowledge. In [6], it was shown that an attacker can manipulate the state estimate while avoiding bad-data alarms. It was also shown that rather simple false-data attacks often can be constructed by an attacker with access to the power network model. The attacker's goal in [6] was either random or targeted false-data attacks. In the targeted attacks, the goal was to change the state estimate into a specific target value.

In this paper, we study a different targeted attack scenario. Here the goal is to manipulate one power flow measurement and to change related measurements in a consistent manner so that no alarms are triggered. Or more accurately: so that the risk of alarms is not increased. At the same time, this shall be done using as small effort as possible. These targeted attacks require less knowledge about the system than the targeted attacks in [6], since the state vector is not necessarily involved. By "small-effort attacks" we here mean either to corrupt as few measurements as possible, or to corrupt the magnitude of the measurement vector as little as possible. The least efforts are then used to define security indices for each targeted measurement. The indices are bounded or computed using simple matrix search techniques or convex optimization. Our study shows that large measurement redundancy gives large magnitude attacks, but that they can still be sparse. Finally, we develop a convex optimization framework that can be used to evaluate false-data attacks which deviate from the model in order to decrease the attack effort and still only marginally increase the risk of a bad-data alarm. Multiple attack goals can also be included in this framework.

II. Power Network Modeling and State Estimation

In this section, we review basic steady-state power network modeling and state-estimation techniques.

A. Active Power Flow Models

It is assumed the power system has $n+1$ buses. Here we will only consider models of the active power flows P_{ij}, active power injections P_i, and bus phase angles δ_i, where $i, j = 1, \ldots, n+1$. It is also of interest to study reactive power flows and the voltage levels, but we leave this for future work.

Consider the simple 4-bus power network in Fig. 2. We assume throughout that the power network has reached a steady state. Since measurements are only sent at a low frequency in the SCADA systems, transients cannot be seen in the state estimator. Assuming that the resistance in the transmission line connecting buses i and j is small compared to its reactance, we have that the active power flow from bus i to bus j is [2],

$$P_{ij} = \frac{V_i V_j}{X_{ij}} \sin(\delta_i - \delta_j). \tag{1}$$

At each bus i, active power can also be injected through a generator. Denote this quantity with P_i. A negative P_i indicates a power load. Assuming that there are no losses, conservation of energy yields that for all buses it holds that

$$P_i = \sum_{k \in \mathcal{N}_i} P_{ik}, \tag{2}$$

where $\mathcal{N}_i$ is the set of all buses connected to bus i. The models we use are based on application of (1) and (2) on each bus in the network.

Remark 1: It is possible to include resistive losses in (1) and shunt loads in (2), see [2], but to simplify notation we leave this out.

B. State Estimation

The state-estimation problem we consider consists of estimating n phase angles δ_i given a set of active power flow measurements. One has to fix one (arbitrary) bus phase angle as reference angle, for example $\delta_1 := 0$, and therefore only n angles have to be estimated. The voltage level of each bus is assumed to be known, as well as the reactance of each transmission line.

The m active power flow measurements are denoted by z_i, and are equal to the actual power flow plus independent random measurement noise e_i, which we assume has a Gaussian distribution of zero mean,

$$e = \begin{pmatrix} e_1 \\ \vdots \\ e_m \end{pmatrix} \in \mathcal{N}(0, R),$$

where $R := \mathbf{E} e e^T$ is the diagonal measurement covariance matrix. For the example in Fig. 2 using the indicated measurements of P_1 and P_{12}, we obtain

$$\begin{aligned} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} &= \begin{pmatrix} P_1 \\ P_{12} \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \end{pmatrix} \\ &= \begin{pmatrix} \frac{V_1 V_2}{X_{12}} \sin(\delta_1 - \delta_2) + \frac{V_1 V_3}{X_{13}} \sin(\delta_1 - \delta_3) \\ \frac{V_1 V_2}{X_{12}} \sin(\delta_1 - \delta_2) \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \end{pmatrix}. \end{aligned}$$

In general, we denote such models by

$$z = P + e = h(x) + e \in \mathbb{R}^m, \tag{3}$$

where $h(x)$ is the power-flow model derived using (1)–(2), and $x \in \mathbb{R}^n$ is a vector of n bus phase angles. Note that here we only analyze the dependence on the phase angles δ_i, and everything else is assumed fixed and known to the

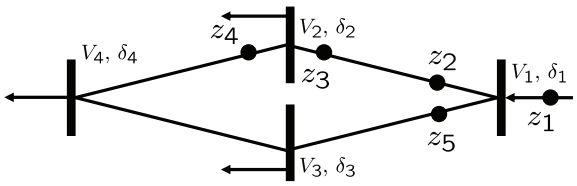


Fig. 3. Same example as in Fig. 2, but with five measurements $z_1 - z_5$ (indicated by dots). This system is observable.

state estimator. This decoupling assumption is common in the literature, see [2], but can be relaxed to include reactive power-flow measurements and bus voltage estimates.

The Gauss-Newton method is often used [2] to estimate the n unknown bus phase angles from power flows measurements z,

$$\hat{x}^{k+1} = \hat{x}^k + (H_k^T R^{-1} H_k)^{-1} H_k^T R^{-1}(z - h(\hat{x}^k)), \quad (4)$$

where $\hat{x}^k \in \mathbb{R}^n$, k denotes iteration number, and H_k is the Jacobian evaluated at $\hat{x}^k$,

$$H_k := \frac{\partial h}{\partial x}(\hat{x}_k) \in \mathbb{R}^{m \times n}.$$

We will assume the phase differences $\delta_i - \delta_j$ in the power network are all small. Then a linear approximation of (3) is accurate, and we obtain

$$z = Hx + e, \quad (5)$$

where $H \in \mathbb{R}^{m \times n}$ is a constant Jacobian matrix. The estimation problem (4) can then be solved in one step,

$$\hat{x} = (H^T R^{-1} H)^{-1} H^T R^{-1} z. \quad (6)$$

The phase-angle estimate $\hat{x}$ can be used to estimate the active power flows by

$$\hat{z} = H\hat{x} = H(H^T R^{-1} H)^{-1} H^T R^{-1} z =: Kz, \quad (7)$$

where K is the so-called "hat matrix" [2]. The BDD system uses such estimates to identify faulty sensors and bad data by comparing the estimate $\hat{z}$ with z, see below.

As an example, assuming the voltages $V_i = 1$ and reactances $X_{ij} = 1$ for the network in Fig. 2, we obtain the model

$$H = \begin{pmatrix} -1 & -1 & 0 \\ -1 & 0 & 0 \end{pmatrix},$$

where $x = \begin{pmatrix} \delta_2 & \delta_3 & \delta_4 \end{pmatrix}^T$, and $\delta_1 = 0$ is the reference bus. However, $H^T H$ is not invertible and it is not possible to use (6) to obtain a unique estimate $\hat{x}$. This network is therefore called *unobservable* [2]. If we add more measurements, such as in the network in Fig. 3, the model becomes

$$H = \begin{pmatrix} -1 & -1 & 0 \\ -1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix}, \quad (8)$$

where $P = \begin{pmatrix} P_1 & P_{12} & P_{21} & P_{24} & P_{13} \end{pmatrix}^T$. Here $H^T H$ is invertible and it is possible to estimate the phase angles in the system. Assuming the measurement error covariance $R = I$, the hat matrix becomes

$$K = \begin{pmatrix} 0.60 & 0.20 & -0.20 & 0 & 0.40 \\ 0.20 & 0.40 & -0.40 & 0 & -0.20 \\ -0.20 & -0.40 & 0.40 & 0 & 0.20 \\ 0 & 0 & 0 & 1.00 & 0 \\ 0.40 & -0.20 & 0.20 & 0 & 0.60 \end{pmatrix}. \quad (9)$$

The hat matrix shows how the power flow measurements z are weighted together to form a power flow estimate $\hat{z}$. The rows of the hat matrix can be used to study the measurement redundancy in the system [2]. Typically a large degree of redundancy (many non-zero entries in each row) is desirable to compensate for noisy or missing measurements. In (9), it is seen that all measurements are redundant except the measurement of P_{24} which is called a *critical measurement*. Without the critical measurement observability is lost. From the hat matrix one is lead to believe that the critical measurement is sensitive to attacks. This is indeed the case as we shall see, but also some of the other measurements are sensitive to attacks. This is however not as easy to see from the hat matrix and we therefore take a different approach to quantify the security here.

III. Problem Formulation

The scenario we consider is that an attacker gains access to the measurements through attacks A1–A3, and is able to change some, or all, of the measurements from z into $z_a := z + a$. The *attack vector* a is the corruption added to the real measurement z. The attacker's goal is to fool the EMS and the human operator that a particular power flow measurement is $z_{k,a} := z_k + a_k$ and not z_k, for some k and fixed scalar a_k. A necessary condition for a stealthy attack is that the BDD system is not triggered (or more accurately, that the alarm risk is not increased). To just corrupt the corresponding measurement z_k into $z_k + a_k$ will typically trigger a bad-data alarm, as seen in the next section. We will consider how many, and by how much, other measurements z_i, $i \neq k$, need to be corrupted in coordination with z_k to avoid triggering alarms. A power flow measurement z_k that requires more and larger corruptions to be altered in stealth is here considered more secure, and will obtain larger security indices, as defined below.

Remark 2: An optimal solution to the above problem in terms of the 2-norm of the attack vector a has recently been presented in [7]. The stealthy attack vector a of minimal 2-norm, $\|a\|_2 = \sqrt{a^T a}$, that achieves $z_{k,a} := z_k + a_k$ is given by $a = \frac{a_k}{K_{kk}} K_{\cdot,k}$, where $K_{\cdot,k}$ is the k-th column of the hat matrix (7) using $R = I$. Generally these attack vectors are not *sparse* (except for critical measurements), however. This can be seen in the example (9). The present study is motivated by the fact that an attacker most likely would use sparse attack vectors, and corrupt as few measurement devices as possible.

IV. Sparse Attacks and the Security Index α_k

In the control center, the measurement residual r,

$$r := z - \hat{z} = P + e - H\hat{x} = (I - K)z, \tag{10}$$

is computed and analyzed in the BDD system. The phase angle estimate $\hat{x}$ is given by (6). If the residual r is larger than expected (measurement errors e will typically make $r \neq 0$), then an alarm is triggered and bad measurements z_i are identified and removed [2], [5], [8]. A key observation in [6] is that an attacker that manipulates the measurements from z into $z_a := z+a$, where $a = Hc \in \mathcal{R}(H)$ and c is an arbitrary vector, is *undetectable* since the residual r is not affected. That certain errors are undetectable by residual analysis has been know for a long time in the power systems community, see for example [5], [8]. It is easy to show that such a lies in the nullspace of $I - K$ in (10). Intuitively this is clear since z_a corresponds to an actual physical state in the power network (minus the measurement error e). The BDD system only triggers when the measurements deviate too much from a possible physical state, at least as long as the linear model is valid.

In light of this, and the problem introduced in Section III, it is natural to consider the following problem:

$$\begin{aligned} \alpha_k &:= \min_c \|Hc\|_0 \\ &\text{such that } 1 = \sum_i H_{ki} c_i, \end{aligned} \tag{11}$$

where $\|Hc\|_0$ denotes the number of non-zero elements in the vector $a = Hc$, and H_{ki} is the element (k, i) of H. In (11), we optimize over all corruptions $a = Hc \in \mathcal{R}(H)$ that do not trigger bad-data alarms. A solution c^* to (11) can be re-scaled to obtain $a^* = a_k Hc^*$ such that the measurement attack $z_a = z + a^*$ achieves the attacker's goal $z_{k,a} = z_k + a_k$, and at the same time corrupts as few measurements as possible. In total, $\alpha_k = \|a^*\|_0$ measurements have to be corrupted to manipulate the measurement z_k. Unfortunately, the problem (11) is non-convex and is generally hard to solve for large problems. However, it is easy to get bounds on α_k even for large models, as shown next.

It is clear that the lower bound $\alpha_k \geq 1$ holds, since at least one measurement (z_k) is corrupted. One can also show that if measurement z_k is a critical measurement, then $\alpha_k = 1$. A simple upper bound can be achieved by looking at the k-th row of H: Every column of H with a non-zero entry in the k-th row can be used to construct a false-data attack vector a that achieves the attack goal. Assume that H_{ki} is non zero. Then the attack vector

$$a_k^i := \frac{a_k}{H_{ki}} H_{\cdot,i},$$

where $H_{\cdot,i}$ denotes the i-th column of H, achieves the attack goal. By selecting the sparsest vector among all a_k^i, we obtain an upper bound $\bar{\alpha}_k^1$ on α_k. Formally we have,

$$\bar{\alpha}_k^1 := \min_{i:H_{ki}\neq 0} \|H_{\cdot,i}\|_0.$$

Since H is typically sparse for power networks, this bound seems many times to be pretty good and is also very fast to compute. A second upper bound, $\bar{\alpha}_k^2$, is discussed in the next section, and the best of them can be used as an upper bound of α_k

$$\bar{\alpha}_k := \min\{\bar{\alpha}_k^1, \bar{\alpha}_k^2\}. \tag{12}$$

Obtaining better easily computed bounds, or even to characterize the exact solution of (11) is an interesting problem for future work.

Remark 3: To obtain a better bound $\bar{\alpha}_k^1$, one can include a column in H that corresponds to the reference bus ($\partial h/\partial \delta_1$).

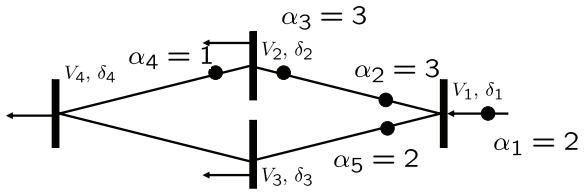


Fig. 4. A power network and its security indices α_k. The flow P_{24} with $\alpha_4 = 1$ is easiest to attack. Only one measurement has to be corrupted. The flows P_{21} and P_{12} with index $\alpha_2 = \alpha_3 = 3$ are hardest to attack, and require a coordinated attack involving three sensors.

TABLE I

The security index α_k, the bound $\bar{\alpha}_k^1$, and the sparsest attack vectors for the power network in Fig. 4

Measurement	Power flow	α_k	$\bar{\alpha}_k^1$	a^*
z_1	P_1	2	2	$\begin{pmatrix}1 & 0 & 0 & 0 & 1\end{pmatrix}^T$
z_2	P_{12}	3	4	$\begin{pmatrix}1 & 1 & -1 & 0 & 0\end{pmatrix}^T$
z_3	P_{21}	3	4	$\begin{pmatrix}-1 & -1 & 1 & 0 & 0\end{pmatrix}^T$
z_4	P_{24}	1	1	$\begin{pmatrix}0 & 0 & 0 & 1 & 0\end{pmatrix}^T$
z_5	P_{13}	2	2	$\begin{pmatrix}1 & 0 & 0 & 0 & 1\end{pmatrix}^T$

In Fig. 4 and in Table I, the security indices α_k and sparse attack vectors for the model (8) are shown. The index makes it easy to locate flows whose measurements are relatively easy to attack without triggering bad-data alarms. In this example, the critical measurement of P_{24} with $\alpha_4 = 1$ is easiest to attack, and P_{21} and P_{12} with index $\alpha_2 = \alpha_3 = 3$ are hardest to attack. It is also seen that the upper bound $\bar{\alpha}_k^1$ is tight in most cases.

Comparing with the hat matrix (9), it is seen that the number of non-zero elements in each row of the hat matrix is not correlated to the number of sensors that has to be involved in a stealthy attack, except in the case of critical measurements (z_4). For example, the measurement z_1 is quite redundant since the estimate $\hat{z}_1$ depends on z_1, z_2, z_3, z_5. But in fact only two measurements (z_1, z_5) have to be manipulated when z_1 is attacked. A large diagonal entry in the hat matrix K seems correlated with a smaller security index, however. Nevertheless, it is not clear from the hat matrix how many, and which, measurements that can be involved in a false-data attack. Hence it seems that measurement redundancy analysis as commonly performed in power systems is not appropriate to evaluate the system's security, and the introduction of other metrics is appropriate.

V. Small Magnitude Attack Vectors and the Security Index β_k

Next we consider a different security index which we denote by β_k. The security index α_k is appropriate to measure resistance against an attacker with limited access to the number of measurements. However, the magnitude of the elements in a sparse attack vector a can be large, and this can be an issue since the power system is nonlinear. An attack vector a with large elements may push the estimator into the nonlinear regime which may lead to bad-data alarms even if $a \in \mathcal{R}(H)$, or non-convergence of the Gauss-Newton method (4). Thus an attacker may want to construct small magnitude attack vectors while achieving his goals. It is also well known that the minimization of the 1-norm that we use below often gives rise to sparse solutions, see for example [9]. Therefore it seems that β_k is a good compromise between a sparse and a small attack vector. The method we introduce below is also based on convex optimization tools, and it is relatively easy to extend this framework to include multiple attack goals and model deviations etc.

The 1-norm of an attack vector a is $\|a\|_1 := \sum_i |a_i|$. This is a measure of the total amount of changes added to the measurement vector z. Let us next study the convex optimization problem

$$\begin{aligned} &\beta_k := \min_c \|Hc\|_1 \\ &\text{such that } 1 = \sum_i H_{ki} c_i, \end{aligned} \tag{13}$$

which can be re-cast into a linear program. A solution c^* to (13) can be re-scaled to obtain $a^* = a_k H c^*$ such that the measurement attack $z_a = z + a^*$ achieves $z_{k,a} = z_k + a_k$, and at the same time the minimal amount of additional power, $\|a^*\|_1$, is added to the measurement vector z. We can interpret the dimensionless quantity β_k as the minimal possible amplification of the attack a_k: The attacker wants to add a_k MWs to the power-flow measurement z_k, but must in the process of doing so add a total change of $\beta_k a_k$ MWs to z in order to avoid triggering alarms.

Remark 4: Since the 1-norm optimal solutions a^* often are sparse, a natural upper bound of α_k is

$$\bar{\alpha}_k^2 := \|a^*\|_0,$$

to be used in (12). One could consider to possibly further improve the bound by using reweighted 1-norm minimization [9].

Remark 5: It is clear that the lower bound $\beta_k \geq 1$ holds. We also have the upper bound $\beta_k \leq \min_{j:H_{kj}\neq 0} \sum_i |H_{ij}/H_{kj}|$. But since β_k can be computed exactly using tools such as CVX [10], these bounds do not seem as important as the bounds on α_k.

It is possible to refine the index β_k to take more complex attack scenarios into account, as long as the constraints are convex. For example, the attacker may be willing to take risks and slightly increase the chance of bad-data alarms. By adding a bias $d \notin \mathcal{R}(H)$ to the attack vector, $a = Hc + d$, it no longer lies in the nullspace of $I - K$, and the risk of a bad-data alarm is increased. The benefit of introducing a bias (from the attacker's point of view) is that it may decrease the size of a and increase its sparsity. It would also be possible to interpret d as an error in the attacker's model.

The measurement residual r (10) in the BDD system is distributed according to

$$r \in \mathcal{N}(Sd, \Omega), \quad \Omega := SR,$$

where $\mathcal{N}$ is the Gaussian distribution, Ω the covariance, and Sd the expected value of the residual. $S := I - K$ is the so-called *residual sensitivity matrix* [2] (remember that K is the hat matrix (7)). Hence $d \neq 0$ changes the expected value of the residual. But it should be clear that if the normalized residual $\|\text{diag}(\Omega)^{-1/2} Sd\|_p$ is small, the risk of a bad-data alarm is still small. Hence, one can introduce a security index β_k^ϵ by

$$\begin{aligned} &\beta_k^\epsilon := \min_a \|a\|_1 \\ &\text{such that } 1 = a_k, \quad \|\text{diag}(\Omega)^{-1/2} Sa\|_p \leq \epsilon, \end{aligned} \tag{14}$$

where we have used that $Sa = S(Hc + d) = Sd$. Depending on the exact BDD system that is being used by the SCADA-system operator and the choice of integer p, the size of ϵ can be related to an increase in probability of a bad-data alarm, see [7]. Common BDD-methods include chi-squares tests and normalized residual tests [2]. Note that the attacker needs to be more informed to solve (14) than to solve (13) since R is needed.

It is also clear that the above framework can be generalized to study attacks with coordinated goals. The optimization problem

$$\begin{aligned} &\min_a \|a\|_1 \\ &\text{such that } a \in \mathcal{G}, \quad \|\text{diag}(\Omega)^{-1/2} Sa\|_p \leq \epsilon, \end{aligned} \tag{15}$$

where $\mathcal{G}$ is a convex set of attack goals, possibly involving more than one measurement, is one such generalization. For example, $\mathcal{G}$ could be intervals such as $\mathcal{G} = \{0.9 \leq a_1 \leq 1.1, -1.1 \leq a_2 \leq -0.9\}$. By solving (15) for various scenarios it is possible for the SCADA-system operator to test the security of the state estimator.

VI. Example: The IEEE 14-bus Power Network

Here we consider the IEEE 14-bus benchmark power network that was also analyzed in [6]. A different perspective is taken here and we compute its security indices and compare with two heuristic redundancy measures. For the computations, the MATLAB package MATPOWER [11] and the optimization toolbox CVX [10] are used. Power flow measurements are added at each bus, and at every end of every interconnecting transmission line. In total there are $m = 54$ measurements, all assumed equally good $R = I$, and the matrix H has size 54×13. This considered system has more measurements than is normal in a power system, and should therefore have large measurement redundancy. The question is: Does this imply security against false-data attacks?

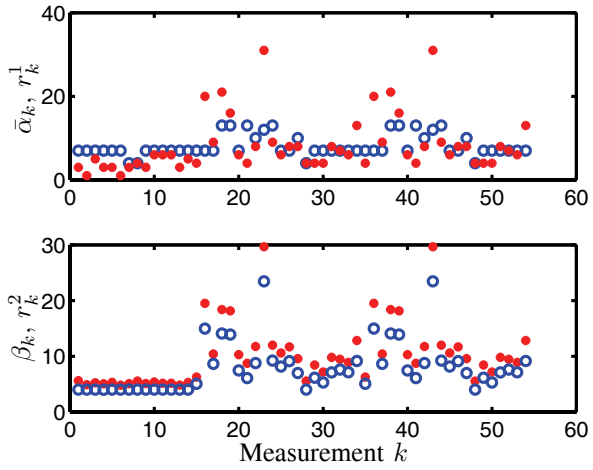


Fig. 5. In the upper plot, the security index bound $\bar{\alpha}_k$ (blue rings) and the redundancy measure r_k^1 (red full circles) are plotted versus measurement number. In the lower plot, the security index β_k (blue rings) and the redundancy measure r_k^2 (red full circles) are plotted. There is no simple connection between $\bar{\alpha}_k$ and r_k^1, whereas the variations in β_k and r_k^2 correlate very well.

In Fig. 5, the security indices $\bar{\alpha}_k$ (bound) and β_k are plotted versus measurement number. For comparison, two heuristic measurement redundancy quantities are also plotted. These are defined by

$$r_k^1 := \#\{|K_{ik}/K_{kk}| \geq 0.33;\, i = 1, \ldots, m\} \geq 1,$$
$$r_k^2 := \sum_i |K_{ik}/K_{kk}| \geq 1,$$

where K is the hat matrix (7). The scaled columns of K are minimal stealthy 2-norm attacks, see Remark 2. Hence these are valid attack vectors, and $\beta_k \leq r_k^2$ with equality for critical measurements. The quantity r_k^1 counts the number of elements in such an attack vector whose magnitude is at least 33% of the attacked measurement. One could expect that those large elements are involved in a sparse attack, and would give a good estimate of α_k. The number 33% is chosen somewhat arbitrarily. However, in these numerical experiments r_k^1 always failed to give accurate predictions of α_k no matter this choice.

As seen in the upper plot of Fig. 5, there is no simple connection between the sparsity of possible attacks (or at least with the bound $\bar{\alpha}_k$) and the quantity r_k^1. Sometimes r_k^1 is too large, and sometimes too small, and it is hard to conclude anything other than that this heuristic must be considered as bad. The number of sensors needed for an attack seemingly has little to do with it.

In the lower plot, the index β_k is plotted together with r_k^2. There is clearly strong correlation between variations in β_k and r_k^2. Maybe this is not so surprising given Remark 2. But note that the optimal 1-norm attacks often are much sparser. To summarize: Large measurement redundancy in terms of r_k^2 seems to give larger security with respect to the security measure β_k (attack vector magnitude), but the quantity r_k^1 has little to do with the security measure α_k (attack vector sparsity).

VII. Summary and Future Work

In this paper, we have introduced two security indices for state estimators in power networks. The indices help to locate power flows whose measurements are potentially easy to manipulate. Large indices indicate that a large coordinated attack is needed in order to not trigger an alarm in the control center. We also showed how convex optimization tools can be used to evaluate attacks, taking deviations from the exact power system model and multiple attack goals into account. We have also seen that simple measurement redundancy quantities seem to give security in terms of attack vector magnitude, but not in terms of attack vector sparsity. This was demonstrated on an IEEE 14-bus network with large measurement redundancy.

For future work, we intend to study how one can use these indices and tools to increase the security. It is also interesting to study the influence of model errors in H.

Acknowledgments

The authors would like to thank Mr. Zhu Kun and Dr. György Dán for helpful and stimulating discussions.

References

[1] M. Shahidehpour, W. F. Tinney, and Y. Fu, "Impact of security on power systems operation," *Proceedings of the IEEE*, vol. 93, no. 11, pp. 2013–2025, 2005.

[2] A. Abur and A. G. Exposito, *Power System State Estimation: Theory and Implementation*. Marcel Dekker, Inc., 2004.

[3] A. Monticelli, "Electric power system state estimation," in *Proceedings of the IEEE*, 2000.

[4] A. Giani, S. Sastry, K. H. Johansson, and H. Sandberg, "The VIKING project: An initiative on resilient control of power networks," in *Proceedings of the 2nd International Symposium on Resilient Control Systems*, Idaho Falls, Idaho, 2009.

[5] L. Mili, T. V. Cutsem, and M. Ribbens-Pavella, "Bad data identification methods in power system state estimation - a comparative study," *IEEE Transactions on Power Apparatus and Systems*, vol. 104, no. 11, pp. 3037–3049, Nov. 1985.

[6] Y. Liu, P. Ning, and M. Reiter, "False data injection attacks against state estimation in electric power grids," in *Proceedings of the 16th ACM conference on Computer and communications security*, Chicago, Illinois, 2009, pp. 21–32.

[7] A. Teixeira, S. Amin, H. Sandberg, K. H. Johansson, and S. S. Sastry, "Cyber-security analysis of state estimators in electric power systems," Submitted to IEEE Conference on Decision and Control, March 2010.

[8] F. F. Wu and W.-H. E. Liu, "Detection of topology errors by state estimation," *IEEE Transactions on Power Systems*, vol. 4, no. 1, pp. 176–183, Feb. 1989.

[9] E. Candès, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted l_1 minimization," *J. Fourier Anal. Appl.*, vol. 14, pp. 877–905, 2008.

[10] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming (web page and software)," http://stanford.edu/ boyd/cvx, June 2009.

[11] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER's extensible optimal power flow architecture," in *Power and Energy Society General Meeting*. IEEE, July 2009, pp. 1–7.

Distributed Fault Detection for Interconnected Second-Order Systems with Applications to Power Networks

Iman Shames, André M. H.Teixeira, Henrik Sandberg, Karl H. Johansson

***Abstract*—Observers for distributed fault detection of interconnected second-order linear time invariant systems is studied. Particularly, networked systems under consensus protocols are considered and it is proved that for these systems one can construct a bank of so-called unknown input observers, and use their output to detect and isolate possible faults in the network. The application of this family of fault detectors to power networks is presented.**

I. Introduction

Critical infrastructures such as power grids, water distribution networks, and transport systems are examples of cyber-physical systems. These systems consist of large-scale physical processes monitored and controlled by SCADA (supervisory control and data acquisition) systems running over a heterogeneous set of communication networks and computers. Although the use of such powerful software systems adds flexibility and scalability, it also increases the vulnerability to hackers and other malicious entities who may perform cyber attacks through the IT systems [1], [2]. Several security breaches have been recently announced [3], [4].

A holistic approach to security of SCADA systems is important because of the complex coupling between the physical process and the distributed software system. Unfortunately a theory for such system security lacking. Increasing the security by adding encryption and authentication schemes helps to prevent some cyber attacks by making them harder to succeed but it would be a mistake to rely solely on such methods, as it is well-known that the overall system is not secured because some of its components are. A method to increase security of networked control systems involve the design of control algorithms that are robust to the effects of cyber attacks [5], [6], [7], [8] and monitoring schemes to detect anomalies in the system caused by attacks [9]. This paper focus on the latter and uses fault detection and isolation (FDI) to design a distributed FDI scheme for a network of interconnected second-order linear systems.

There are various ways to detect and isolate a fault in a system [10], [11], [12], [13]. Observer based approaches have been well studied and some of these methods have been proposed for power systems [14], [15]. However, distributed FDI for systems comprised of a network of autonomous nodes still lacks a thorough theory. A relevant and interesting result is presented in [9] where the authors have proposed a discrete time algorithm to detect the misbehaving node in a network of nodes with single integrator dynamics. Another result related to this work is [16] where the possibility of detecting faults by coordinating certain movements in the formation is shown.

In this paper we consider the problem of distributed fault detection and isolation in a network of nodes with double integrator dynamics seeking to reach consensus. To achieve this goal, we design a bank of continuous time unknown input observers (UIO) in each node, which then monitors its own neighborhood. The existence of such observers is established for two different consensus protocols, and some infeasibility results are provided. As an illustrative example, the application of the proposed method to FDI in power networks is presented.

The outline of the paper is as follows. In the next section the problem is formulated. In Section III we introduce the UIO that we use to obtain the main result of this paper. In Section IV, we propose a solution to the problem posed in Section II. In Section V the application of the method on an illustrative 9 bus power grid is studied. Conclusions and future remarks come in the last section.

II. Problem Formulation

Consider a network of N interconnected nodes and let $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{A})$ be the underlying graph of this network, where $\mathcal{V} = \{i\}_1^N$ is the vertex set with $i \in \mathcal{V}$ corresponding to node i, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set of the graph, and $\mathcal{A} \in \mathbb{R}^{N \times N}$ is the weighted adjacency matrix with nonnegative entries. The undirected edge $\{i, j\}$ is incident on vertices i and j if nodes i and j share a communication link, in which case the corresponding entry in the adjacency matrix $[\mathcal{A}]_{ij}$ is positive and reflects the edge weight. The out-degree of node i is $\deg(i) = \sum_{j \in N_i} [\mathcal{A}]_{ij}$, where $N_i = \{j \in \mathcal{V} : \{i, j\} \in \mathcal{E}\}$ is the neighborhood set of i. The degree matrix $\Delta(\mathcal{G}) \in \mathbb{R}^{N \times N}$ is a diagonal matrix defined as

$$[\Delta]_{ij} = \begin{cases} \deg(i) & , \ i = j \\ 0 & , \ i \neq j \end{cases}.$$

The weighted Laplacian of $\mathcal{G}$ is defined as $\mathcal{L}(\mathcal{G}) = \Delta - \mathcal{A}$.

imansh@cecs.anu.edu.au
{andretei,hsan,kallej}@kth.se

This work was supported in part by the European Commission through the VIKING project, the Swedish Research Council, the Swedish Foundation for Strategic Research, and the Knut and Alice Wallenberg Foundation.

Each node i is assumed to have double integrator dynamics

$$\dot{\xi}_i(t) = \zeta_i(t) \tag{1a}$$
$$\dot{\zeta}_i(t) = u_i(t), \tag{1b}$$

where ξ_i, ζ_i, and u_i are scalars and u_i is the control law

$$u_i(t) = \sum_{j \in N_i} \left[w_{ij}(\xi_j(t) - \xi_i(t)) + (\alpha_{ij}\zeta_j(t) - \beta_i\zeta_i(t)) \right]. \tag{2}$$

We say that node k is faulty if for some functions $f_{\xi k}(t)$ and $f_{\zeta k}(t)$ not identical to zero it holds that

$$\begin{aligned} \dot{\xi}_k(t) &= \zeta_k(t) + f_{\xi k}(t) \\ \dot{\zeta}_k(t) &= u_k(t) + f_{\zeta k}(t). \end{aligned} \tag{3}$$

The functions $f_{\xi k}(t)$ and $f_{\zeta k}(t)$ denote fault signals.

Remark 1: The variables ξ_i and ζ_i can be interpreted as position and velocity, respectively, for a mobile system, or as phase and frequency in the context of power networks.

Let $n = 2N$ and consider $x(t) \in \mathbb{R}^n$,the global state of the network, defined as $x(t) = [\xi_1(t), \ \ldots, \ \xi_N(t), \ \zeta_1(t), \ \ldots, \ \zeta_N(t)]^\top$. The closed-loop dynamics of the network in the presence of faults can be written as

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bv(t) + B_f f(t) \\ y(t) &= Cx(t), \end{aligned} \tag{4}$$

where $v(t) \in \mathbb{R}^r$ is a vector of known external control inputs, $f(t) \in \mathbb{R}^m$ is a vector of fault signals, $y(t) \in \mathbb{R}^p$ is the output vector, and A, B, B_f, and C are matrices of appropriate dimensions. Before stating the problem addressed in this work, we define what is meant by fault *detectability* and *isolability*, according to [13].

Definition 1: Given the system (4), a fault $f_k(t) \in \mathbb{R}$ is said to be detectable if

$$\left. \frac{\partial y}{\partial f_k} \right|_{f_k = 0} \not\equiv 0. \tag{5}$$

In general terms, this means that a detectable fault should produce a change in the output.

Lemma 1: Definition 1 is equivalent to say that the system's Rosenbrock matrix

$$\begin{bmatrix} sI - A & b_{f_k} \\ C & 0 \end{bmatrix} \tag{6}$$

has full normal rank, where $b_{f_k} \in \mathbb{R}^m$ is the k-th column of B_f and normal rank is defined as the rank for almost all $s \in \mathbb{C}$.

This means that the transfer function from $f_k(t)$ to $y(t)$ is not identical to zero.

Definition 2: Given the system (4), a vector of faults $f(t) \in \mathbb{R}^m$ is said to be isolable if

$$\left. \frac{\partial y}{\partial f} \right|_{f=0} df \not\equiv 0. \tag{7}$$

In the case of additive faults, this relates to input observability and it loosely means that any simultaneous occurrence of faults would lead to a change in the output. Furthermore, the following can be said for additive faults

Lemma 2: Given the system (4), the m faults in $f(t)$ are isolable if and only if

$$\text{normal rank} \begin{bmatrix} sI - A & B_f \\ C & 0 \end{bmatrix} = n + m \tag{8}$$

In this paper, we solve the following problem:

Problem 1: When and how can each agent of the networked system (1)-(2) detect and isolate a faulty agent?

We propose a solution to this problem for two different consensus protocols (2) in the coming sections. However, first in the next section we introduce the mathematical tool to be used.

III. Preliminaries

A common technique used in model-based fault diagnosis is to generate a set of residuals which indicate the presence of a fault. The residual is a fault indicator computed from the difference between the measurements and their estimates, which should be close to zero if and only if the fault is not present. In this section, we consider the general linear fault-free system under the influence of an unknown input $d(t) \in \mathbb{R}^q$ to be described by

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bv(t) + Ed(t) \\ y(t) &= Cx(t) \end{aligned} \tag{9}$$

whereas the system in presence of faults is given by

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bv(t) + B_f f(t) + Ed(t) \\ y(t) &= Cx(t) \end{aligned} \tag{10}$$

with the assumption that the matrices E and B_f have full column rank.

Remark 2: Note that the condition on B_f being full column rank is not restrictive, since any singular matrix $D \in \mathbb{R}^{n \times l}$ can be decomposed in $D = D_1 D_2$, with D_1 having full column rank. This implies, however, that not all faults are isolable.

The matrix E is called a disturbance distribution matrix, since it contains information on how a vector of unknown input signals, seen as disturbances, affect the states of the dynamical system.

Definition 3: A state observer is an unknown input observer (UIO) if the state estimation error $e = x - \hat{x}$ approaches zero asymptotically, regardless of the presence of an unknown input d.

A full-order observer for the fault-free system in (9) is described by:

$$\begin{aligned} \dot{z}(t) &= Fz(t) + TBv(t) + Ky(t) \\ \hat{x}(t) &= z(t) + Hy(t) \end{aligned} \tag{11}$$

where $\hat{x}(t) \in \mathbb{R}^n$ is the estimated state and $z(t) \in \mathbb{R}^n$ is the observer's state. Note that if we choose $F = A$, $T = I$, and $H = 0$ we have a full-order Luenberger observer. The matrices in the observer's equations must be designed in order to achieve the decoupling from the unknown input and meet

requirements on the stability of the observer. Choosing the matrices F, T, K, H to satisfy the following conditions

$$\begin{aligned} F &= (A - HCA - K_1 C) \\ T &= (I - HC) \\ (HC - I)E &= 0 \\ K_2 &= FH \end{aligned} \tag{12}$$

we have the estimation error dynamics

$$\dot{e}(t) = Fe(t). \tag{13}$$

We conclude that if the equations in(12) are satisfied and F is stable, then the observer (11) is an UIO. The following proposition formalizes this.

Proposition 1 ([11]): The observer (11) is an UIO for (9) if and only if

i) $\text{rank}(CE) = \text{rank}(E)$

ii) (C, A_1) is a detectable pair, where

$$A_1 = A - HCA \tag{14}$$

For a proof and more details the reader is refered to [11], [13].

Now consider the system (10). As suggested in [11], a possible method of detecting and isolating the faults present in the process is to use the so called generalized observer scheme (GOS), where we construct a bank of observers generating a structured set of residuals such that each residual is decoupled from one and only one fault, but being sensitive to all other faults.

Definition 4: A residual $r(t)$ is a fault indicator function, which satisfies the following condition:

$$r(t) = 0 \iff f(t) = 0$$

where $f(t)$ represents the fault signal.

The detection and isolation of a fault in the k-th component,$f_k \neq 0$, is based on that:

$$\begin{aligned} &\|r_k(t)\| < \Theta_{f_k} \\ &\|r_j(t)\| \geq \Theta_{f_j} \ , \forall j \neq k, \end{aligned} \tag{15}$$

where $r_i(t)$ is the residual insensitive to a fault in the i-th component and $\Theta_{f_i} > 0$ is the isolation threshold, which can be constant or time varying. Note that this approach is feasible only if a single additive fault is present. If more faults are present, they can be detected using this method, but they cannot be isolated. To isolate multiple faults, one could enlarge the observer bank with multi-fault detectors.

Suppose there is a single active fault, $f_i(t) \neq 0$. In order to render an observer insensitive to $f_i(t)$, this fault could be regarded as an unknown input and the observer could then be computed using the UIO theory. The system (10) can be rewritten as

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bv(t) + B_f^i f^i(t) + b_{f_i} f_i(t) \\ y(t) &= Cx(t) \end{aligned} \tag{16}$$

where b_{f_i} is the i-th column of B_f, $f_i(t)$ the i-th component of $f(t)$, B_f^i is B_f with the i-th column deleted and $f^i(t)$ the fault vector $f(t)$ with its i-th component removed. The UIO decoupled from b_{f_i} has the same structure as in (11) and is described by

$$\begin{aligned} \dot{z}_i(t) &= F_i z_i(t) + T_i B v(t) + K_i y(t) \\ \hat{x}_i(t) &= z_i(t) + H_i y(t) \end{aligned} . \tag{17}$$

Remark 3: Note that for such a UIO to exist, $f_i(t)$ must be detectable. This follows from that Cond. ii) of Prop. 1 is equivalent to requiring the asymptotic stability of the transmission zeros of the system (A, b_{f_i}, C), which implies that

$$\text{normal rank} \begin{bmatrix} sI - A & b_{f_i} \\ C & 0 \end{bmatrix} = n + 1 \tag{18}$$

It is easy to show that we have the following observer error and residual dynamics

$$\begin{aligned} \dot{e}_i(t) &= F_i e_i(t) - T_i B_f^i f^i(t) \\ r_i(t) &= C e_i(t) \end{aligned} . \tag{19}$$

Note that the residual dynamics are driven by the k-th fault if $T_i b_{f_k} \neq 0, k \neq i$. We can compute similar observers for all the other faults and then use the threshold logic in (15) to isolate the fault.

From Remark 3 we conclude that the existence of UIOs for all additive faults requires such faults to be detectable. Together with the assumption that B_f has full-column rank, we conclude that the existence of a bank of UIOs ensures the isolability of all additive faults.

Next we show that one can construct such UIO for two consensus protocols applied to a networked system.

IV. Fault detection for networked systems

Consider the networked system introduced in Section II with the following consensus protocol

$$m_i u_i(t) = -d_i \zeta_i(t) + \sum_{j \in N_i} w_{ij} \left(\xi_j(t) - \xi_i(t) \right). \tag{20}$$

where $m_i, w_{ij}, d_i > 0 \in \mathbb{R}$ and $\xi_i \in \mathbb{R}$. Recall the networked system (4), with $x(t) = [\xi_1(t), \cdots, \xi_N(t), \zeta_1(t), \cdots, \zeta_N(t)]^\top$ and

$$\begin{aligned} A &= \begin{bmatrix} 0_N & I_N \\ -\bar{M}\mathcal{L} & -\bar{D}\bar{M} \end{bmatrix} \\ B &= \begin{bmatrix} 0_N \\ \bar{M} \end{bmatrix} \\ \bar{M} &= \text{diag}\left(\frac{1}{m_1}, \cdots, \frac{1}{m_N} \right) \\ \bar{D} &= \text{diag}\left(d_1, \cdots, d_N \right). \end{aligned} \tag{21}$$

Assume that ξ_k does not satisfy Equation (1a), but

$$\dot{\xi}_k(t) = \zeta_k(t) + f_k(t) \tag{22}$$

where $f_k(t)$ corresponds to a fault in node k. In the presence of this fault, (4) transforms into

$$\dot{x}(t) = Ax(t) + b_f^k f_k(t) \tag{23}$$

with $b_f^k = [\bar{b}_f^{k\top}\ 0_{1\times N}]^\top$ where $\bar{b}_f^k$ is an N dimensional vector with all zero entries except one that corresponds to the faulty node k. Furthermore, we assume node i has access to

$$y_i(t) = C_i x(t), \quad C_i = \left[\bar{C}_i\ 0_{\tilde{N}_i\times N}\right], \tag{24}$$

with $\bar{C}_i$ being an $|\tilde{N}_i|$ by N matrix with full row rank, where each of the rows have all zero entries except for one entry at the j-th position that corresponds to those nodes that are neighbors of i, where $\tilde{N}_i = N_i \cup i$ and $j \in \tilde{N}_i$.

To tackle Problem 1 we need to show that one can construct a UIO at each node i under the consensus protocol (20) using measurements (24).

Before presenting the main result of this paper we have the following lemma.

Lemma 3: If an undirected graph $\mathcal{G}$ is connected, then any partition of its Laplacian matrix $\mathcal{L}$, induced by a strict subset of nodes $\bar{\mathcal{V}} \subset \mathcal{V}$, is invertible.

Proof: See [17]. ∎

Now we are ready to state the following theorem concerning the existence of a UIO for the consensus protocol (20).

Theorem 1: There exists a UIO for the system (23) with measurements (24) of node i if the graph $\mathcal{G}$ is connected and $k \in N_i$.

Proof: First we have to show that

$$\text{rank}\left(C_i b_f^k\right) = \text{rank}\left(b_f^k\right) = 1.$$

Denote the row of C_i that reads the output of node k, c_i^k. It is obvious that $c_i^k b_f^k = 1$ and $c_i^j b_f^k = 0,\ j \neq k$. Hence, $C_i b_f^k$ is a vector with zero entries except one which is equal to 1, thus the rank is equal to 1. Then we have to show that

$$\text{rank}(\mathcal{D}) = 2N + 1$$

for all $\text{Re(s)} \geq 0$ where

$$\mathcal{D} = \begin{bmatrix} sI_{2N} - A & b_f^k \\ C_i & 0_{\tilde{N}_i\times 1} \end{bmatrix}.$$

We have

$$\text{rank}(\mathcal{D}) = \text{rank}\begin{bmatrix} sI_N & -I_N & \bar{b}_f^k \\ \bar{M}\mathcal{L} & sI_N + \bar{D}\bar{M} & 0_{N\times 1} \\ \bar{C}_i & 0_{|\tilde{N}_i|\times N} & 0_{|\tilde{N}_i|\times 1} \end{bmatrix}$$

Applying some row and column operations we obtain

$$\text{rank}(\mathcal{D}) = \text{rank}\begin{bmatrix} 0_N & -I_N & \bar{b}_f^k \\ a(s) & 0_N & b(s) \\ \bar{C}_i & 0_{|\tilde{N}_i|\times N} & 0_{|\tilde{N}_i|\times 1} \end{bmatrix},$$

with

$$\begin{aligned} a(s) &= s^2 I_N + s\bar{D}\bar{M} + \bar{M}\mathcal{L} \\ b(s) &= (sI_N + \bar{D}\bar{M})\bar{b}_f^k \end{aligned}.$$

We apply a transformation P to the system so that

$$\begin{aligned} \bar{x} = Px = [&\xi_{\tilde{i}_1},\ \cdots,\ \xi_{\tilde{i}_{|\tilde{N}_i|}},\ \xi_{\bar{i}_1},\ \cdots,\ \xi_{\bar{i}_{|\bar{N}_i|}}, \\ &\zeta_{\tilde{i}_1},\ \cdots,\ \zeta_{\tilde{i}_{|\tilde{N}_i|}},\ \zeta_{\bar{i}_1},\ \cdots,\ \zeta_{\bar{i}_{|\bar{N}_i|}}]^\top, \end{aligned}$$

where $\tilde{i}_j \in \tilde{N}_i$, $\bar{i}_j \in \bar{N}_i$, and $\bar{C}_i^* = \bar{C}_i P = [I_{\tilde{N}_i} 0_{\tilde{N}_i\times\bar{N}_i}]$, where $\tilde{N}_i = i \cup N_i$ and $\bar{N}_i = \mathcal{V} \setminus \tilde{N}_i$. After this operation we can write the Laplacian as $\bar{\mathcal{L}} = P^{-1}\mathcal{L}P = \begin{bmatrix} \mathcal{L}_{|\tilde{N}_i|} & l_{|\tilde{N}_i|\times|\bar{N}_i|} \\ l_{|\bar{N}_i|\times|\tilde{N}_i|} & \mathcal{L}_{|\bar{N}_i|} \end{bmatrix}$. Furthermore $P^{-1}\bar{M}P = \begin{bmatrix} \bar{M}_{1|\tilde{N}_i|} & 0_{|\tilde{N}_i|\times|\bar{N}_i|} \\ 0_{|\bar{N}_i|\times|\tilde{N}_i|} & \bar{M}_{2|\bar{N}_i|} \end{bmatrix}$, $P^{-1}\bar{D}P = \begin{bmatrix} \bar{D}_{1|\tilde{N}_i|} & 0_{|\tilde{N}_i|\times|\bar{N}_i|} \\ 0_{|\bar{N}_i|\times|\tilde{N}_i|} & \bar{D}_{2|\bar{N}_i|} \end{bmatrix}$, $\tilde{b}_f^k = P^{-1}\bar{b}_f^k$, and $\tilde{b}_f^{k*} = P^{-1}(sI_N + \bar{D}\bar{M})\bar{b}_f^k$.

After applying the transformation we have

$$\text{rank}(\mathcal{D}) = \text{rank}\begin{bmatrix} 0_{|\bar{N}|\times|\tilde{N}_i|} & 0_{|\bar{N}_i|\times|\bar{N}_i|} & -I_N & \tilde{b}_f^k \\ c(s) & \bar{M}_1 l_{|\tilde{N}_i|\times|\bar{N}_i|} & 0_{|\tilde{N}_i|\times N} & \tilde{b}_f^{k*} \\ \bar{M}_2 l_{|\bar{N}_i|\times|\tilde{N}_i|} & d(s) & 0_{|\bar{N}_i|\times N} & 0_{|\bar{N}_i|\times 1} \\ I_{|\tilde{N}_i|} & 0_{|\tilde{N}_i|\times|\bar{N}_i|} & 0_{|\tilde{N}_i|\times N} & 0_{|\tilde{N}_i|\times 1} \end{bmatrix},$$

with

$$\begin{aligned} c(s) &= \bar{M}_1\mathcal{L}_{|\tilde{N}_i|} + s^2 I_{|\tilde{N}_i|} + s\bar{M}_1\bar{D}_1 \\ d(s) &= \bar{M}_2\mathcal{L}_{|\tilde{N}_i|} + s^2 I_{|\tilde{N}_i|} + s\bar{M}_2\bar{D}_2 \end{aligned}.$$

It is evident that the first and the third columns are independent of the rest, thus

$$\begin{aligned} \text{rank}(\mathcal{D}) = |\tilde{N}_i| + N + \\ + \text{rank}\begin{bmatrix} \bar{M}_1 l_{|\tilde{N}_i|\times|\bar{N}_i|} & \tilde{b}_f^{k*} \\ \bar{M}_2\mathcal{L}_{|\bar{N}_i|} + s^2 I_{|\bar{N}_i|} + s\bar{M}_2\bar{D}_2 & 0_{|\bar{N}_i|\times 1} \end{bmatrix}. \end{aligned}$$

We know from Lemma 3 that any partition of the Laplacian matrix is invertible so the last column is independent of the rest as well so

$$\text{rank}(\mathcal{D}) = |\tilde{N}_i| + N + |\bar{N}_i| + 1 = 2N + 1 \tag{25}$$

∎

Remark 4: Note that if the graph is not connected, the networked system (23) can be decomposed in several decoupled subsystems, each corresponding to a connected subset of the network. Theorem 1 then applies to each subsystem.

Theorem 1 establishes that a UIO can be constructed at node i that can observe node k. The existence of such observer leads to detection of a possible fault at node k by node i using the method described in Section III.

In Theorem 1 we stated that a fault in ξ_k can be isolated with the measurements of the form (24). In the next theorem we identify what type of faults cannot be isolated.

Theorem 2: Consider the system described by (23). For any of the following pairs of C_i and b_f^k no UIO of the form (11) exists:

i) $b_f^k = [\bar{b}_f^{k\top}\ 0_{1\times N}]^\top$, $C_i = \left[0_{\tilde{N}_i\times N}\ \bar{C}_i\right]$
ii) $b_f^k = [0_{1\times N}\ \bar{b}_f^{k\top}]^\top$, $C_i = \left[0_{\tilde{N}_i\times N}\ \bar{C}_i\right]$
iii) $b_f^k = [0_{1\times N}\ \bar{b}_f^{k\top}]^\top$, $C_i = \left[\bar{C}_i\ 0_{\tilde{N}_i\times N}\right]$

Proof: To see that no UIO exists for (i) and (iii) one needs to check that

$$\text{rank}\left(C_i b_f^k\right) = \text{rank}\left(b_f^k\right) = 0.$$

Hence, the first condition of Proposition 1 is not satisfied. For (ii), similar to the calculations in proof of Theorem 1 for the case where $s = 0$, we have

$$\text{rank}(\mathcal{D}) = \text{rank} \begin{bmatrix} 0_N & -I_N & \bar{b}_f^k \\ \bar{M}\mathcal{L} & 0_N & \bar{D}\bar{M}\bar{b}_f^k \\ 0_{\tilde{N}_i \times N} & \bar{C}_i & 0_{|\tilde{N}_i| \times 1} \end{bmatrix}. \quad (26)$$

Considering the first column, it is known that $\mathcal{L}$ is rank deficient, and hence the second condition of Theorem 1 is not satisfied. ■

Cases i) and iii) from Theorem 2 suggest that if there is an unknown input affecting one of the states of one of the nodes in a network, it is not possible to have a UIO without measuring the same state throughout the network as the one affected by the unknown input. For example, if an unknown input(fault) is affecting the velocity of one of the nodes, by measuring positions alone we cannot have a UIO to observe the states of the network. On the other hand, in case ii) we see that the first condition of Proposition 1 is satisfied, but the UIO still does not exist. What happens in this case is that the system is not detectable, as seen by observing the first two columns of (26).

In what comes next we introduce the conditions where a UIO exists for observing ζ_j $j \in N_i$ and consequently detecting a fault in them.

Theorem 3: Consider the system described by (23)–(24). For $C_i = \begin{bmatrix} \bar{C}_i & 0_{|\tilde{N}_i| \times N} \\ 0_{|\tilde{N}_i| \times N} & \bar{C}_i \end{bmatrix}$, where $\bar{C}_i$ is a $|\tilde{N}_i|$ by N matrix, and $b_f^{k\top} = \begin{bmatrix} 0_{1\times N} & \bar{b}_f^{k\top} \end{bmatrix}$ and b_f^k being an N by 1 vector with having k-th entry as its only nonzero entry, a UIO exists to observe ζ_j $j \in N_i$

Proof: The proof for existence of a UIO is similar to the previous case and is omitted ■

For the rest of this section we consider another consensus protocol [18]:

$$u_i(t) = \sum_{j \in N_i} w_{ij} \left[(\xi_j(t) - \xi_i(t)) + \gamma(\zeta_j(t) - \zeta_i(t)) \right], \quad (27)$$

Furthermore, for the whole network with a faulty node k and the same selection of x we have

$$\dot{x}(t) = Ax(t) + b_f^k f_k(t) \quad (28)$$

where

$$A = \begin{bmatrix} 0_N & I_N \\ -\mathcal{L} & -\gamma\mathcal{L} \end{bmatrix}, \quad (29)$$

and $\mathcal{L}$ is the weighted Laplacian matrix with the weight $w_{ij} > 0$, $\gamma > 0$, $b_f^{k\top} = \begin{bmatrix} \bar{b}_f^{k\top} & 0_{1\times N} \end{bmatrix}$ with b_f^k being an N by 1 vector with having k-th entry as its only nonzero entry. We further assume that node i measures $y_i(t)$ at time t which satisfies

$$y(t) = C_i x(t), \quad (30)$$

and $C_i = \begin{bmatrix} \bar{C}_i & 0_{|\tilde{N}_i| \times N} \\ 0_{|\tilde{N}_i| \times N} & \bar{C}_i \end{bmatrix}$, where $\bar{C}_i$ is a $|\tilde{N}_i|$ by N matrix. Now we have the following theorem.

Theorem 4: There exists a UIO for the system (28) if the graph $\mathcal{G}$ is connected, measurements of the form (30) are available and the faulty node k is in the neighborhood of node i, N_i.

Proof: The proof is very similar to that of Theorem 1 and is omitted. ■

For detecting fault in $\zeta_k(t)$, $j \in N_k$ we have the following theorem.

Theorem 5: For $b_f^{k\top} = \begin{bmatrix} 0_{1\times N} & \bar{b}_f^{k\top} \end{bmatrix}$ with b_f^k being an N by 1 vector with having k-th entry as its only nonzero entry, a UIO exists to observe $\zeta_k(t)$, $k \in N_i$.

Proof: The proof for existence of a UIO is similar to the previous case and is omitted ■

So far we have established what type of measurements should be available at node i to be able to detect a fault in $k \in N_i$ using a UIO based fault detection scheme. More specifically we have shown that if a node aims to detect a fault in a state of one of its neighbors using the aforementioned UIO based scheme, it has to measure the same state of all of its neighbors.

V. Power Networks Application

Power systems are an example of very complex systems in which generators and loads are dynamically interconnected. Thus they can be seen as networked systems, where each bus is a node. We will now provide a simple model for the active power flow in a power grid. Such model and additional details of power networks can be found in [19].

The behavior of a bus i can be described by the so-called *swing equation*:

$$m_i\ddot{\delta}_i(t) + d_i\dot{\delta}_i(t) - P_{mi}(t) = -\sum_{j \in N_i} P_{ij}(t), \quad (31)$$

where m_i and d_i are the inertia and damping coefficients, respectively, P_{mi} is the mechanical input power and P_{ij} is the active power flow from bus i to j. Considering that there are no power losses nor ground admittances and letting $V_i = |V_i| e^{j\delta_i}$ and δ_i be, respectively, the complex voltage and the phase angle of bus i, the active power flow between bus i and bus j, P_{ij}, is given by:

$$P_{ij}(t) = k_{ij} \sin(\delta_i(t) - \delta_j(t)), \quad (32)$$

where $k_{ij} = |V_i| |V_j| b_{ij}$ and b_{ij} is the susceptance of the power line connecting buses i and j.

Since the phase angles are close, we can linearize (32), rewriting the dynamics of bus i as:

$$m_i\ddot{\delta}_i(t) + d_i\dot{\delta}_i(t) = u_i(t) + v_i(t), \quad (33)$$

with

$$\begin{aligned} u_i(t) &= -\sum_{j \in N_i} k_{ij}(\delta_i(t) - \delta_j(t)) \\ v_i(t) &= P_{mi}. \end{aligned} \quad (34)$$

Consider a power network with $\mathcal{G}(\mathcal{V}, \mathcal{E})$ as its underlying graph with $N = |\mathcal{V}|$ nodes, where each node corresponds to a bus in the power network. Rewriting (33) and (34) in state-state form and considering $x =$

$\left[\delta_1(t),\ \cdots,\ \delta_N(t), \dot{\delta}_1(t),\ \cdots,\ \dot{\delta}_N(t)\right]^\top$, we can write the network's dynamics as

$$\dot{x}(t) = Ax(t) + Bv(t), \tag{35}$$

where $B = \left[0_N\ \bar{M}^\top\right]^\top$, A and $\bar{M}$ are given by (21) and $v(t) = [P_{m1}\ \cdots\ P_{mN}]^\top$ is the collection of input power at each bus. These are generator's power inputs or load power consumptions, which we assume as known. The dynamics of the power network correspond to (21) with an additional known input $v(t)$ and thus the results from Section IV can be used to detect and isolate faults in power networks.

Remark 5: The stability and convergence properties of the system $\dot{x}(t) = Ax(t)$ where $d_1 = \cdots = d_N$ are studied in [20], and the case where d_i, $i = 1, \cdots, N$ are not necessarily equal is not presented here due to lack of space.

In the example that follows next, we consider that the network is being affected by faults corresponding to unexpected changes in the power generation or consumption. Assume that a fault has occurred at node k. The power network under such conditions can be modeled as

$$\dot{x} = Ax + Bv(t) + b_f^k f_k, \tag{36}$$

where b_f^k is the k-th column of B and therefore it can be written as $b_f^k = \left[0_{1\times N}\ \bar{b}_f^{k\top}\right]^\top$ with $\bar{b}_f^{k\top}$ being a column vector with $\frac{1}{m_k}$ in the k-th entry and zero in all other entries. Thus, from Theorem 3 there exists an UIO for such system at a given node i if $k \in N_i$ and $y_i = C_i x$ with

$$C_i = \begin{bmatrix} \bar{C}_i & 0_{|\tilde{N}_i|\times N} \\ 0_{|\tilde{N}_i|\times N} & \bar{C}_i \end{bmatrix}.$$

Thus we need to measure the phase and frequency of the neighbors to be able to detect the faulty node. Having such measurements, this type of faults can be detected and isolated in a distributed way using UIOs, as we show with the following example.

Consider the power network presented in Fig. 1. The power grid's topological parameters and the generators' dynamic coeficients (m_i and d_i) were taken from [21], while the dynamic coeficients of the rest of the buses were arbitrarily taken from reasonable values. The system matrices used in the simulation can be found in the appendix.

The power network is evolving towards the steady-state when, at time instant $t = 2s$, a fault occurs at node 6, as presented in Fig. 2. By implementing a bank of observers at bus 7, the fault is successfully detected and isolated in the presence of measurement noise, since the residual corresponding to bus 6 became larger than the other residuals, as illustrated in Fig. 3.

Remark 6: Because of Theorem 2 we know that we cannot solve the fault detection problem using UIO with having access to less information than the information available through $y_i = C_i x$, with the above-mentioned C_i.

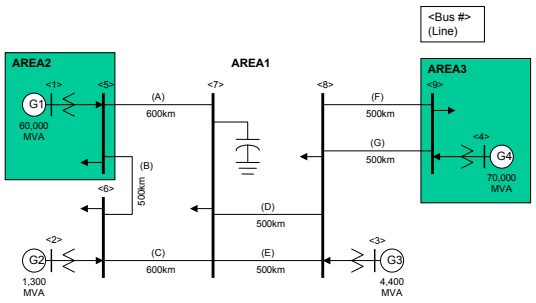


Figure 1. Power network with 9 buses [21].

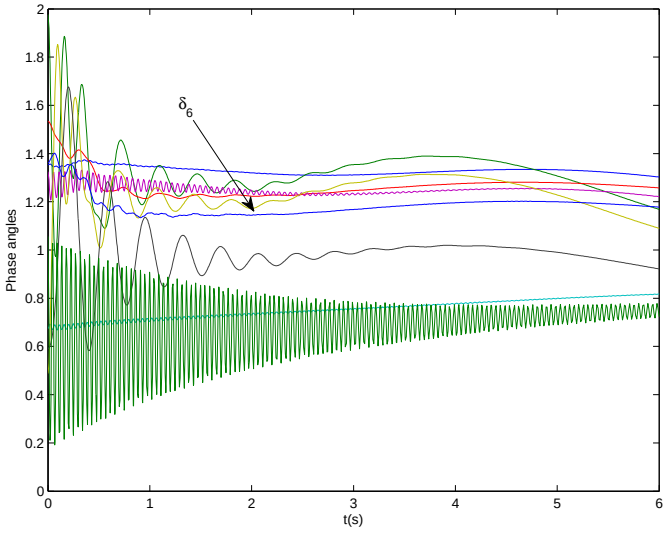


Figure 2. Phase angles of the power network.

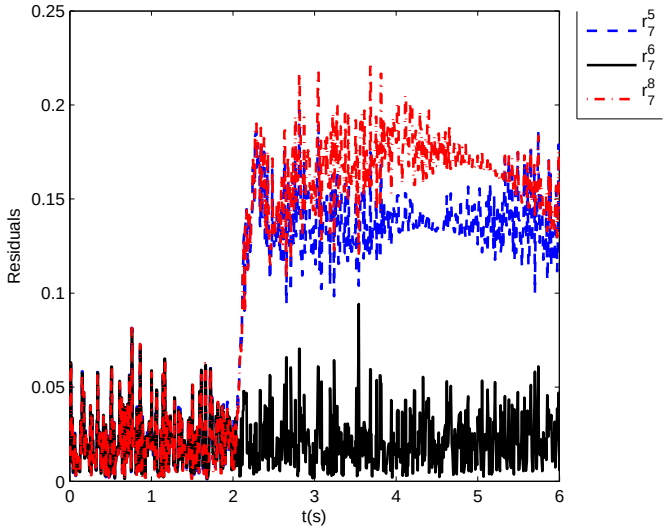


Figure 3. Residuals of buses neighboring bus 7.

VI. Concluding Remarks and Future Directions

In this paper we considered the problem of fault detection and isolation for interconnected nodes with double-integrator dynamics performing consensus. We presented an illustrative example to show the application of the proposed method to fault detection in power systems. Future directions include considering a way to reduce the dimension of the unknown input observers at each node in the current scheme, and explore applicability of other fault detection methods to the problem.

References

[1] A. Cárdenas, S. Amin, and S. Sastry, "Secure control: Towards survivable cyber-physical systems," in *First International Workshop on Cyber-Physical Systems (WCPS2008)*, Beijing, China, June 2008, pp. 495–500.

[2] T. G. Roosta, "Attacks and defenses of ubiquitous sensor networks," Ph.D. dissertation, EECS Department, University of California, Berkeley, May 2008.

[3] "Cyber war: Sabotaging the system," *CBSNews*, November 8 2009.

[4] "Electricity grid in U.S. penetrated by spies," *The Wall Street Journal*, p. A1, April 8 2009.

[5] N. Lynch, *Distributed Algorithms*, 1st ed. Morgan Kaufmann, 1997.

[6] D. Bauso, L. Giarre, and R. Pesenti, "Lazy consensus for networks with unknown but bounded disturbances," in *Proceedings of the IEEE Conf. on Decision and Control*, New Orleans, LA, December 2007, pp. 2283–2288.

[7] S. Amin, A. Cárdenas, and S. Sastry, "Safe and secure networked control systems under denial-of-service attacks," in *Hybrid Systems: Computation and Control*. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, April 2009, pp. 31–45.

[8] S. Sundaram and C. Hadjicostis, "Distributed function calculation via linear iterations in the presence of malicious agents - part II: Overcoming malicious behavior," in *Proceedings of the American Control Conference*, Seatle, WA, June 2008, pp. 1356–1361.

[9] F. Pasqualetti, A. Bicchi, and F. Bullo, "Distributed intrusion detection for secure consensus computations," in *Proceedings of Control and Decision Conference*, 2007, pp. 5594–5599.

[10] M. A. Massoumnia and G. C. Verghese, "Failure detection and identification," *IEEE Transactions on Automatic Control*, vol. 34, pp. 316–321, 1989.

[11] J. Chen and R. J. Patton, *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Kluwer Academic Publishers, 1999.

[12] R. Isermann, "Model-based fault detection and diagnosis: status and applications," in *In Proceedings of the 16th IFAC Symposium on Automatic Control in Aerospace*, St. Petersburg, Russia, June 2004, pp. 71–85.

[13] S. X. Ding, *Model-based Fault Diagnosis Techniques: Design Schemes*. Springer Verlag, 2008.

[14] E. Scholtz and B. Lesieutre, "Graphical observer design suitable for large-scale DAE power systems," in *Proceedings of the IEEE Conf. on Decision and Control*, Cancun, Dec. 2008, pp. 2955–2960.

[15] M. Aldeen and F. Crusca, "Observer-based fault detection and identification scheme for power systems," in *IEE Proceedings - Generation, Transmission and Distribution*, vol. 153, no. 1, Jan. 2006, pp. 71–79.

[16] M. Franceschelli, M. Egerstedt, and A. Giua, "Motion probes for fault detection and recovery in networked control systems," in *Proceedings of American Control Conference*, Seattle, WA, June 2008, pp. 4358–4363.

[17] A. M. H. Teixeira, "Multi-agent systems with fault and security constraints," master's Thesis in Automatic Control KTH-Royal Institute of Technology, 2009.

[18] W. Ren and E. Atkins, "Distributed multi-vehicle coordinated control via local information exchange," *Int. J. Robust Nonlinear Control*, vol. 17, pp. 1002–1033, 2007.

[19] P. Kundur, *Power System Stability and Control*. McGraw-Hill Professional, 1994.

[20] G. Xie and L. Wang, "Consensus control for a class of networks of dynamic agents," *Int. J. Robust Nonlinear Control*, vol. 17, pp. 941–959, 2007.

[21] "Power system test cases," http://psdyn.ece.wisc.edu/IEEE_benchmarks/.

Appendix

$$A = \begin{bmatrix} A_1 & A_2 \\ I_9 & 0_9 \end{bmatrix}$$

$$B = \begin{bmatrix} B_1 \\ 0_9 \end{bmatrix}$$

$$C = \begin{bmatrix} \bar{C} & 0_9 \\ 0_9 & \bar{C} \end{bmatrix}$$

$$\bar{C} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} -25.8844 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -22.7101 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -15.1515 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -15.2672 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.5003 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.5752 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.4452 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.4532 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.6322 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} -697.98 & 0 & 0 & 0 & 697.98 & 0 & 0 & 0 & 0 \\ 0 & -873.11 & 0 & 0 & 0 & 873.11 & 0 & 0 & 0 \\ 0 & 0 & -507.47 & 0 & 0 & 0 & 0 & 507.47 & 0 \\ 0 & 0 & 0 & -488.74 & 0 & 0 & 0 & 0 & 488.74 \\ 12016.97 & 0 & 0 & 0 & -12230.33 & 113.25 & 100.11 & 00 & \\ 0 & 486.52 & 0 & 0 & 148.43 & -763.75 & 128.80 & 0 & 0 \\ 0 & 0 & 0 & 0 & 102.69 & 100.81 & -274.10 & 70.60 & 0 \\ 0 & 0 & 1135.52 & 0 & 0 & 0 & 71.40 & -1276.58 & 69.66 \\ 0 & 0 & 0 & 18635.24 & 0 & 0 & 0 & 75.18 & -18710.42 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 0.1355 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5.4881 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0818 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0685 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.3334 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3.0581 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2.3935 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.4207 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.6126 \end{bmatrix}$$

SCADA-specific Intrusion Detection/Prevention Systems: A Survey and Taxonomy

Bonnie Zhu Shankar Sastry

***Abstract*— Due to standardization and connectivity to the Internet, Supervisory Control and Data Acquisition (SCADA) systems now face the threat of cyber attacks. SCADA systems were designed without cyber security in mind and hence the problem of how to modify conventional Information Technology (IT) intrusion detection techniques to suit the needs of SCADA is a big challenge. We explain the nuance associated with the task of SCADA-specific intrusion detection and frame it in the domain interest of control engineers and researchers to illuminate the problem space. We present a taxonomy and a set of metrics for SCADA-specific intrusion detection techniques by heightening their possible use in SCADA systems. In particular, we enumerate Intrusion Detection Systems (IDS) that have been proposed to undertake this endeavor. We draw upon the discussion to identify the deficits and voids in current research. Finally, we offer recommendations and future research venues based upon our taxonomy and analysis on which SCADA-specific IDS strategies are most likely to succeed, in part through presenting a prototype of our efforts towards this goal.**

I. Introduction

The origin of taxonomy is rooted in bioscience [77], [79]. The idea of taxonomies for attacks and instructions was more borrowed from pathology, where each disease can be treated with a specific method or medication. Thus often those taxonomies require exclusiveness among their components. However, at the current stage of the SCADA-specific intrusion detection field or even intrusion detection in IT field in general, we'd like to argue that such exclusiveness is not suitable for an early attempt in which the eco-space is still under development. Such stringent formality would not provide aid to solve the problem at hand.

When it comes to taxonomy in the field of intrusion detection, John Mchugh has made very keen observation in his critique of evaluation of Intrusion Detection System (IDS) [49]:

> *The point is that the taxonomy must be constructed with two objectives in mind: describing the relevant universe and applying the description to gain insight into the problem at hand.*

which is the philosophy that this paper strives to follow.

A. Introduction to SCADA systems

Defined by IEEE Standard (C37.1-1994) [32] , a Supervisory Control and Data Acquisition (SCADA) system includes all control, indication, and associated telemetering equipment at the master station, and all of the complementary devices at the (Remote Terminal Unit) RTU(s)[1]. A typical SCADA system includes hardware, software and communication protocols that connect together the different layers in the hierarchy. For more detailed expositions on SCADA system compositions, readers please refer to resources such as [80], [39].

Being one of the primary categories of control systems, SCADA systems are generally used for large, geographically dispersed distribution operations, such as electrical power grids, petroleum and gas pipelines, water and wastewater (sewage) systems and other critical infrastructures [80]. They not only provide management with remote access to real-time data from Distributed Control Systems (DCSs) and Programmable Logic Controllers (PLCs) but also enable operational control center to issue automated or operator-driven supervisory commands to remote station control devices and complete high-level exchange among different networks and domains. Consequently, the communication protocols used within the hierarchical system to enable *cyber-physical* interaction[7], [20], [39] have strong implications on the security of SCADA system [20], [86], [3]. The raw data protocols are designed for communication between physical layer and serial/radio links, but can also be tunneled over Internet. They are used for reading raw data from field devices such as voltage, pressure, fluid flow and so on or sending alerts from field devices when leakage detected or overpressure sensed or sending commands remotely from control station to field devices such as flip a switch or turn on or off a break[2].

On the other hand, the high-level data protocols[3], are designed to transmit bulk process data and commands between various applications/databases. They often bridge between the enterprise-network and control-network to provide information for humans.

For example, company A wants a current pipeline pressure reading off the oil pipeline within zones, the area where

This work is supported by the National Science Foundation Award CCF-0424422 for the Team for Research in Ubiquitous Secure Technology (TRUST).

B. Zhu and S. Sastry are with Department of Electrical Engineering and Computer Science, University of California at Berkeley, Berkeley, CA 94720, USA {bonniez,sastry}@eecs.berkeley.edu

[1] RTUs are special purpose data acquisition and control units designed to support SCADA remote stations. These field devices are often equipped with wireless radio interfaces to support remote situations where wire based communications are unavailable.

[2] Here, we name a few most popular ones: Modbus, Profibus, Distributed Network Protocol (DNP3) and Utility Communications Architecture (UCA), Foundation Fieldbus, Common Industrial Protocol (CIP), Controller Area Network(CAN) [39].

[3] Examples in this category are Object Linking and Embedding (OLE) for Process Control (OPC) and Inter-Control Center Communications Protocol (ICCP) [39].

a company has the right to oil exploration, belonging to company B. It sends a request through ICCP to company B. Company B relays this request to one of its Human Machine Interface (HMI) workstations before this request message reaches a set of PLCs and initiates the data transfer processes. Each PLC then provides a response containing the requested information through Modbus [52], [53]. In this situation, the device running the HMI is acting as the client/master and the PLC is acting as the server/slave. Each message contains a function code set by the client/master and indicates to the server/slave what kind of action to perform.

Most industrial plants now employ networked process historian servers storing process data and other possible business and process interfaces, such as using remote Windows sessions to DCSs or direct file transfer from PLCs to spreadsheets. This integration of SCADA networks with other networks has made SCADA vulnerable to various cyber threats. The adoption of Ethernet and TCP/IP for process control networks and wireless technologies such as IEEE 802.x, Zigbee, Bluetooth, WiFi, plus WirelessHART and ISA SP100 [20], [39] has further reduced the isolation of SCADA networks. The connectivity and de-isolation of the SCADA system is manifested in Fig.1.

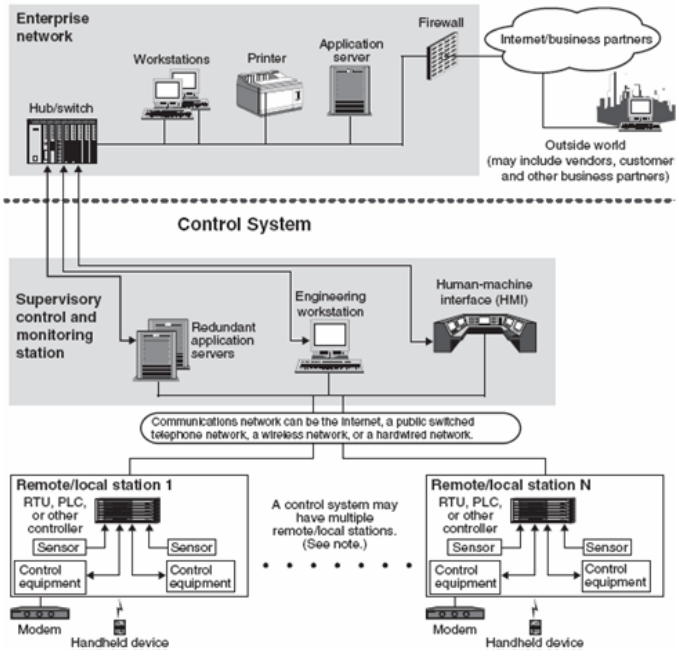


Fig. 1. Typical SCADA Components Source: United States Government Accountability Office Report. GAO-04-354 [23]

Furthermore, the recent trend in standardization of software and hardware used in SCADA systems [39] potentially makes it even easier to mount SCADA-specific attacks These attacks can disrupt and damage critical infrastructural operations, contaminate the ecological environment, cause major economic losses and, even more dangerously, claim human lives [25], [1], [24]. These likely "penalty costs" due to lack of protection and our tendency in *aversion to loss* [33], [83], [75] push us to consider tapping into SCADA systems characteristics and seeking protection measures with reasonable cost-effectiveness [56].

B. Why SCADA-specific Intrusion Detection Systems?

Had we not started with the legacy systems but been freed from difficulties such as interoperability [41], [57] instead, we may apply and implement many known security measures directly, such as rigorous *access control*, end-to-end secure communication protocols with full *authentication* , *encryption* besides *key management systems* and so on [7], [67].

However, there is no such a thing as perfect security or prevention product. An all-encompassing and airtight prevention is not only extremely expensive both in economic and operational sense but also technically and socially infeasible. The arm-race between protections and attacks is a continuous up-hill battle.

Bruce Schneier [75] considers *"Prevention is best combined with detection and response."* The method of an *intrusion alarm* coupled with a *security response* [6], [9], [21], a well-established approach in the traditional security field, has its special immediate appeal for securing SCADA systems [35], [39], [70], [80]. A sound implementation and viable deployment of one Intrusion Detection System (IDS) can manifest itself as an add-on intelligence component to the existing SCADA systems with minimum hardware cost or operational changes, leveraging many entrenched SCADA component infrastructures and technologies.

To this end, the industrial and academic control security community has started to build Intrusion Detection Systems (IDS) specifically for SCADA systems ([17], [54], [55], [57], [71], [74], [81], [82], [90]).

Nevertheless, it is important to realize that when we borrow tools from other fields, there are situations and conditions that our original set of assumptions might not hold. A SCADA system is different from the conventional IT system in the following ways [80], [93]: it is a **hard real-time** system, i.e. having the capability to meet a deadline *deterministically*, with its **timeliness** and **availability** at all times is very critical; its terminal devices have **limited** computing capabilities and memory resources [84]; and more importantly the fact that logic execution occurred within SCADA has a direct impact in the physical world dictates **safety** as the paramount [23], [24].

In particular, we shall point out that the time-criticality of SCADA systems is resulted from their need to meet deadlines deterministically and from their inherited concurrencies as being widely dispersed distributed systems. It includes both the *responsiveness* aspect of the system, e.g. a command from the controller to actuator should be executed in real-time by the latter, and the *timeliness* of any related data being delivered in its designated time period, by which, we also mean the *freshness* of data, i.e., the data is only valid in its assigned time period. Or in a more general sense, this property describes that any queried, reported, issued and disseminated information shall not be stale but corresponding to the real-time and the system is able and sensitive enough to

process requests, which may be of normal or of legitimate human intervention in a timely fashion, such as within a sampling period. In reality, even a command to an actuator is correct or a perfect measurement from a sensor is intact, they become no good if they arrive late to a specified node, Similarly, any replay of data easily breaches this security goal.

Moreover, this characteristic also implicitly implies the order of updates among peered sensors, especially if they are observing the same process or correlated processes. The order of data arrival at *central monitor room* may play an important factor in the representation of process dynamics and affect the correct decision making of either the controlling algorithm or the supervising human operator. In a nutshell, all right data should be processed in *right* time.

In addition to above mentioned operational requirement nuances, comparing to typical IT systems and/or enterprise networks, in the existing SCADA systems, there are no or weak authentication mechanisms at best to differentiate human users or privilege separation or user account management to control access and so on [57]. Such fundamental weakness in access control leaves the door open to attacks. These differences challenge design and implementation of SCADA-specific IDSs.

Meanwhile, among the attempts to date, some authors [17] may consider that SCADA systems usually have a relatively static topology[4], a *presumably* regular network traffic pattern[5] and use simple protocols, hence monitoring them may not be more difficult than doing so in enterprise systems. But to our best knowledge, none of the work ([17], [54], [55], [57], [71], [74], [81], [82], [90]) has been tested on real operational SCADA system network traffic to validate such assumptions. On the other hand, it's a known fact that simulated data may be potentially quite different from real measurements especially in abnormal cases, the focal point of our IDS research. For example, as seen in Fig.2, the department of Energy records a drastic difference in simulated and real power grids measurements and performance during the August 10, 1996 western grid breakup [59] .

Moreover, we believe that the **cyber-physical** security of real-time, continuous systems necessitates a comprehensive view and holistic understanding of network security, control theory and physical systems [80], [93]. The ultimate goal of much needed work is to aid in achieving satisfactory control performance in a continuous 24×7, real-time, realistic environment, where normalized behavior co-exits with benign noises, honest mistakes, natural components and or systems faults plus potential malicious cyber intrusions. However, by convention, certain shared vernacular use in each of these fields may have their own field-specific interpretations[6], particularly regarding several key terminologies used in standard IT IDS research such as *misuse, fault and anomaly*, of which definitions are clarified in section II. Hence we aim to provide a clear definition and precise interpretation besides a set of desired properties, or metrics, for SCADA-specific IDSs.

[4]Under the assumption that there is no wireless sensor network involved.

[5]Due to the scarce accessibility to operational SCADA traces known to the public, we are conservative at taking the leap of faith yet.

[6]One of the barrier facing control security researcher in general is the occupational and cultural including lingo difference between IT and control personnel.

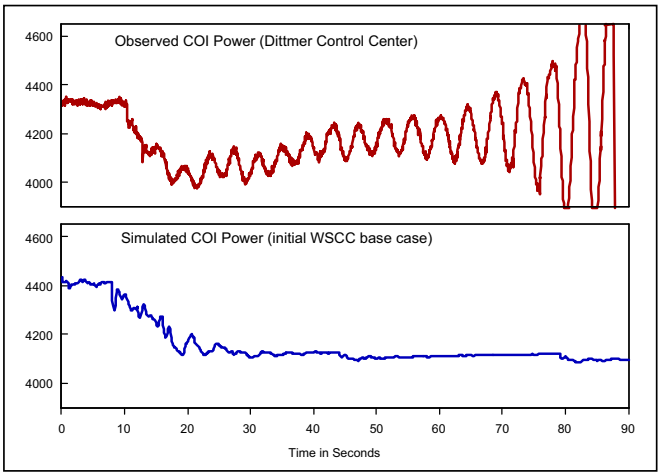


Fig. 2. Comparison of measured and simulated grid performance and measurements during the August 10, 1996 western power grid breakup. The upper panel shows the real grid breakup while the lower panel indicates stability through simulation. Source: Department of Energy

Towards concrete progress beyond generic discussions, it's important for us to survey and evaluate up-to-date research efforts in this area and reflect on the soundness of the overall methodologies. We may want to ask:

- Have these techniques and approaches addressed the specifical needs of SCADA systems? Furthermore,
- Are we simply handicapped by the nuance of current SCADA systems and diving into unrealistically complicated strategies in terms of security engineering efforts? Or
- Are we incorporating and tapping into the entrenched SCADA infrastructure components and technologies?

C. Related Work

Since SCADA-specific IDS research is a rather new arena, we decide to resort to the classics in the standard IT field ([42], [15], [43], [60], [69], [18], [61], [9], [11], [58], [48], [50], [49], [27], [62], [28], [73], [47], [85], [63], [35], [64]), for relevant insights into categorizing *intrusion identifiers* in the context of the *SCADA environment* in which we wish effectively use them and highlighting these that we consider more applicable to our problem space.

1) Landscape: Lough [46] has done a rather thorough job of reviewing the various taxonomies offered by the computer security community, as well as the criteria for evaluating them.

Kent and Mell at National Institute of Science and Technology (NIST) recommend the general guidelines on Intrusion Detection and Prevention (IDP) systems [35].

Killourhy, Maxion and Tan [36] give comprehensive exposition on attack taxonomies.

Alessandri[5] developed a classification of attacks and a description framework for intrusion detection systems. The developed method can be used by IDS designers to predict whether a given design will be able to detect certain classes of attacks. Attacks are classified according to their externally observable characteristics. The identified attack classes are then described in terms of IDS characteristics which are needed to analyze a given class of attacks.

Buhan et al. developed a meta-classification schema of attack taxonomies to provide guidance to the process of choosing the most suitable taxonomy for a security task[13]. They classify *atomic taxonomies* based on the 'grounds of distinction' including: the *who, how* and *what* aspect of the attack. Each atomic taxonomy represents only one dimension of the attack. Then they combine a taxonomy from each of these classes to create a nested taxonomy. Yampolskiy and Govindaraju[14] survey all aspects of computer security including attackers and attacks, software bugs and viruses as well as different intrusion detection systems and ways to evaluate such systems.

As far as the style of a intrusion taxonomy goes, Lippmann et al [58] at Lincoln Lab provided a general and attack manifestation based categorization; Axelsson[9] offered a thorough description; Killourhy et al [36] showed work more align with McHugh's observation [49], similar to ours.

2) Flagship Works: The ongoing work at MIT Lincoln Lab, such as Lippmann et al [58] including attack taxonomy, the use of receiver operating characteristic (ROC) techniques and attack construction, DARPA datasets etc., Haines et al. [27] on extending the DARPA off-line intrusion detection evaluations, and Rossey et al. [73] on the LARIAT (Lincoln Adaptable Real-time Information Assurance Testbed), has been one of early systematic and solid efforts in intrusion detection research. Their experience has long term impact in the field of intrusion detection [48], [50], [49], [47].

Both Stefan Axelsson [9] and John McHugh[50] have thorough work on classification of intrusion detection systems. In particular, Axelsson[9] provided one of most comprehensive and detailed taxonomy not only on the detection principle of the 22 IDS prototypes surveyed but also on certain operational aspects of IDSs in general, both with sufficient qualitative explanations. Whereas McHugh[50] gave a historical review on intrusion detection and some detailed description on a number of contemporary research and commercial intrusion detection systems at the time of his writing. He also noted the difficulties associated with evaluating IDSs.

Backed by substantially operational experience of the Bro (Network Intrusion Detection System) NIDS at the Lawrence Berkeley National Laboratory (LBNL) and numerous other sites, Vern Paxson[61], [62], [28], its primary author, pointed out the disambiguation of "crud" seen in an adversary environment and the analysis of application level semantics[7] among other principles and aspects of viable design and implementation of an IDS for its in situ deployment.

We gain quite insights into how we should conduct sound IDS research through the warnings of difficulties, pitfalls and challenging issues raised by Stefan Axelsson [8], [9], [11], John Mchugh [49], and Vern Paxson [60], [63], [64]. Thus many evaluation and assessment principles on SCADA-specific IDS in this paper and design principles of our follow-on work are derived from their works.

On the other hand, in general, the metrics used for evaluation are the benchmarks that the evaluated subjects should be striving for. Both Axelsson [8], [9], [11] and Mchugh [50], [49] have thorough work on classification and sound metrics of intrusion detection systems[8].

We hope to constructively offer a set of useful metrics to facilitate SCADA-specific IDS research and securing SCADA in general.

To understand the nature of IDS performance, we adapt the unified view framed by Stefan Axelsson [10], where the intrusion detection is considered as a signal detection problem and the normal network traffic is treated as the background data. Indeed, if we view background data & responses as the noise and attack data & responses as the signal, the IDS problem can be characterized as one of detecting a signal in the presence of noise. This school of thought is much in line with the standard control theory [16]. And it's natural to realize that the dataset used has noneligible impact on gauging the design and performance of IDSs.

3) Dataset: As far as the datasets used for constructing attack traffic and/or simulated background traffic, for verification purpose only, are concerned, MIT Lincoln Labs DARPA datasets[58] and KDD Cup dataset [34] derived from them are not only overly used, but also won't be precisely apt and reliable for SCADA-specific IDSs, given that they are not even simulated SCADA network traffic. Beyond McHugh's critique[49], Maxion and Tan [48] further illustrate both the regularity of background traffic and environment conditions affect *false positive rate*. Mahoney and Chan [47] observe that simulation artifacts may render network anomaly detection systems very low false positive rate and claim this evaluation problem can be mitigated by mixing real traffic into the simulation. We will use such observations to verify whether all proposed SCADA-specific IDSs take such precaution when conduct and assess their own work.

D. Contribution

In this paper, we make the following contributions:

- First systematic and thorough effort in investigating and assessing the landscape of up-to-date SCADA-specific intrusion detection techniques and systems;
- Explain the nuance of SCADA-specific IDS and provide clear definitions plus a taxonomy and a set of metrics of SCADA-specific IDS;

[7] As far as viable SCADA-specific IDS solution goes, these are among the top reasons that we prefer Bro over Snort [72], i.e. connection based application level analysis.

[8] Or at least they've posed the questions that IDS research and researchers should address.

- Ease the interoperability between the conventional IT security and control systems research by addressing the intrusion detection problem in the setting of SCADA systems' continuous operation;
- Bring in cross-discipline insights to tailor the special needs entailed by SCADA systems by leveraging entrenched SCADA components and technologies and provide future direction;
- Show a prototype of our efforts in this arena.

E. Organization of the Paper

SectionII presents a set of unified terminologies to facilitate understanding and some reasoning on the difficulties facing IDS due to ambiguities. SectionIII briefly reviews intrusions on SCADA systems and sets the context for sectionIV to give a taxonomy of real-time intrusion detection approaches and to discuss their usage in SCADA systems. Then sectionV describes 9 proposed SCADA-specific IDS with their comparison in sectionVI and evaluation in sectionVII , respectively. Section offers some suggestions on the future direction including some work we are undertaking before sectionIX concludes.

II. Definitions and Difficulties from Ambiguities

To resolve the ambiguity of same terminologies that bear different meanings in control theory (including systems & control and fault detection & isolation) and IT (particularly, operating system and security engineering), we intend to unify the terms to ease the misunderstanding and highlight the end goal of providing engineers and researchers insights into the problems facing securing networked control systems.

Fault: is a non-hostility-induced deviation from the system's specified behavior including honest mistakes caused by honest people and component failures or defects.

Anomaly: refers to maliciously intrusive event and atypical yet non-intrusive behavior including (faulty and overly noisy/messy) actions.

Misuse: includes both malicious and unintentional misuse deviated from system's specified ideal action[9].

Noise: refers to the messiness of benign and innocuous yet non-ideal system and network data due to unintentional interference from natural and technical sources.

Detection: alarm alerts issued in the presence of true anomaly or misuse.

False alarm/positive: alarm alerts issued in the absence of real **anomaly** and/or **misuse** when there is normal traffic/behavior only.

False negative or missed detection: missed detection in the presence of a real intrusion.

Note: Any large network is a very "noisy" environment even at the packet level. Ross Anderson[7] considers that *noise* of the unintentional interference from natural sources such as lightning, electric motors and animals is not within security-centric considerations. He further argues that although these inevitable noise can threaten the integrity of data in a message, communication protocols have been designed with overcoming such concerns in mind, such as TCP ensures reliable transmission to avoid such errors. However, we believe such noises do affect the performance of an IDS by contributing more ambiguities into physical 'analog' sensing[10] and thus potentially 'into digital' network traffic so strongly that might be beyond TCP's original capability and exploitable by attackers to evade detection [28].

Also according to the conversations we had with people from industry, one of the major concerns of theirs is the noise due to physical interference when data are transmitted over communication link.

Moreover, we are referring to the diversity of legitimate network traffic. Bellovin [15] gives accounts on that there are many bad packets on the Internet. Paxson [61] recounts crud seen on a DeMilitarized Zone (DMZ). Many of those pathologies look very similar to genuine attacks.

In general, the ambiguities in network traffic lead to the evasion problem facing Network Intrusion Detection System (NIDS)[69], [28] is a known fact in cyber security and intrusion detection community . Zachary et al. [91] further argue that discerning between normal and malicious traffic is an ill-posed problem, which can be made less ill-posed by restricting the set of admissible solutions through a regularization scheme.

Keep in mind that some of the mostly common used SCADA-specific protocols are byte-coding, such as ModBus, DNP3. When these protocols are tunneled over IP and used in conjunction with TCP, the security implication of the envision problem due to ambiguities would be more potentially damaging, if no proper attention is paid.

III. Intrusions on SCADA Systems

For completeness, we briefly cover cyber intrusions on SCADA systems grouped according to their possible target-based manifestation channels:

1) Control historian, Human Machine Interface (HMI), controller: what's been stored including memory and control functionality;
2) network link between sensors and HMI or controller: what's seen by controller/operator including ID, address, value and time;
3) network link between controller and actuators: what's being sent to actuators including ID, address, action, value and time;
4) modify sensors threshold values and settings through cyber means;
5) modify or sabotage auctors normal settings through cyber means;

Interested readers please refer to [93] for more details.

Before going into the details of the proposed intrusion detection/prevention approaches for SCADA systems , let us first review the categories that an intrusion detection method may fall into.

[9]Unauthorized access should fall into the category of misuse under our definitions.

[10]An extreme case would be channel jamming.

IV. Taxonomy of Intrusion Detection System Approaches

In this section, we adapt a taxonomy of real-time intrusion detection to facilitate the choice for control's researchers.

A. On Real Time Intrusion Detection Types

In the early days of IDS research, two major approaches known as **signature detection**[11] and **anomaly detection** were developed [21], [9], [35]. The *signature detection* matches traffic to a known misuse pattern of the intrusive process and its characteristic traces regardless system normal behavior. Namely, we are watching for known intrusion–the *s*ignal [9]. Supplied with a well-craft intrusion signature and the absence of its variants in real operations, theoretically this approach can achieve high detection rate and low false alarm rate simultaneously. While in *anomaly detection*, we do not watch for known intrusion–the signal–but rather the abnormalities in the observed data in question and alert when something "extremely unusual" is noticed. It's usually based on learning with certain statistical profiling of the usual behavior of the overall system[12] over time without regard to actual intrusion scenarios. Namely, we identify deviation from the learned normal system model and decide whether it's within acceptable range. This approach faces the difficulty to find a snag fitting model for the usual behavior that is comprehensive enough to avoid false alarms yet tight enough to escape false negatives. Ideally, a faithful model can detect *novel* attacks as well.

In between these two approaches, there lie the probabilistic- and specification-based methods for intrusion detection. A **probabilistic approach** is also termed as a *statistical* or a *Bayes* method [38] with probabilistically encoded models of misuse. It has some potential to detect unknown attacks. A **specification-based approach** constructs a model of what is allowed, enforces its predefined policy and raises alerts when the observed behavior is outside this model. It has a high potential for generalization and leverages against new attacks [12]. This technique has been proposed as a promising alternative that combines the strengths of signature-based and anomaly-based detection.

Instead of finding the deviation and unknowns, specification-based method [12], [37] defines what's allowable in terms of network and system traffic behavior/patterns. This method sounds promising. But it might be tedious to enumerate all possibly allowable patterns.

Complementary to the above *direct* knowledge based classification, there are also **behavioral detection** approaches[13]. They capture behavior patterns associated with certain attacks, which are not necessarily illegitimate in the direct semantic sense but wrong in a *contextual* setting thus may require secondary evidence. They may abstract allowable normal interaction as well. Such methods are quite promising, especially used in conjunction with other methods [92].

[11] also refers as misuse detection.

[12] By system, we mean the networked control system or SCADA system, not just the operating system.

[13] A thoroughly stringent and meticulous categorization is not the focus of this paper. Interested readers may refer to [9], [50] for more detailed taxonomies on IDS.

B. Organizational Principles

Pragmatically speaking, what matters most is how this information and technology can assure SCADA and networked control systems in general to provide basic functionality under attacks. Therefore, we don't intend to give the most exhaustive categorization or taxonomy of existing intrusion techniques. Furthermore, its not to say that these characteristics in specific intrusion techniques we want to highlight are mutually exclusive, absent of over-lapping.

Especially given the fact that the modeling, monitoring of the dynamical physical process, fault detection and isolation are traditionally well studied in control engineering field, we want to categorize the intrusion detection techniques to bring out the basics so that control engineers may find easy to relate control field experience upon this new challenge and useful in understanding.

C. Taxonomy Dimensions

- **Approach** refers to the methods we discussed above.
- **Knowledge-based** refers to that methods predominately rely on primary evidence such as semantic definitions, predefined (access) policies, model of legitimate data flow and abstraction of known illegal patterns.
- **Behaviorial-based** refers to that methods also need secondary evidence to make contextual analysis.
- **Basis** refers to the methods' building blocks.
- **Attacks Detected** refers to the detection range.
- **Generalization** refers to whether the detection mechanism can deal *novel* attacks.

D. Taxonomy

Table I gives the overall comparison.

E. Implication and Discussion

Through above comparison in Table I, we can see the strength, limitation and tradeoff of each method. In light of the intrusions we mentioned in III, we believe there's room for direct extension of existing control system dynamical models for intrusion detection at the application layer as a way of using anomaly-based detection methods. To reduce false alarms, reachability theory can be casted in the setting of specification-based detection methods. Similarly, those techniques in stochastic control may also be turn into the use for probabilistic intrusion detection approaches. On the other hand, many fault detection methods may be handy to turn into signature-based intrusion detection rules, provided that we figure out the cyber-physical correlation of these cases. Furthermore, we think that behavioral detection can be done right and effectively for SCADA system when we build up a database for such incidents. We will see more concrete examples in the following section.

Approach	*Knowledge-based* or *Behavioral-based*	*Basis*	*Attacks Detected*	*Generalization*
Signature	Knowledge	Misuse	Known	No
Anomaly	Knowledge	Learned models of normal	Must appear anomalous	Yes
Probabilistic	Knowledge	Model learning	Match patterns of misuse	Some
Specification	Hybrid	Construct normal model	Must violate specs	Yes
Behavioral	Behavioral	Capture behavioral pattern	Match patters of behavior	Yes

TABLE I
TAXONOMY OF INTRUSION DETECTION SYSTEM APPROACHES

V. PROPOSED SCADA-SPECIFIC INTRUSION DETECTION/PREVENTION SYSTEMS

A. Model-Based IDS for SCADA Using Modbus/TCP

As mentioned before, SCADA systems have a relatively static topology and regular traffic and they use simple protocols. Backed-up by this argument, the group at SRI [17] adapted the specification-based approach for intrusion detection to SCADA systems that rely on ModbusTCP, the most widely used application layer protocol for communication between control station to field devices in industrial networks.

This work renders a multi-algorithm IDS appliance containing pattern anomaly recognition, Bayes analysis of TCP headers, and stateful protocol monitoring complemented with customized Snort rules[72]. Alerts are forwarded to the correlation framework. They offer three model-based techniques to characterize the expectedacceptable system behavior according to the Modbus/TCP specification and to detect potential attacks that violate these models. The first technique, the protocol-level technique, is based on building the specifications for individual fields and for groups of dependent fields in the Modbus/TCP requests and responses. The second technique, the communication patterns modeling technique, is based on the analysis of the communication patterns among network components. The detection of violation of the expected communication patterns is done with the help of SNORT rules [72]. The third technique, the service usage patterns modeling technique, is based on learning models that describe the expected trends in the availability of servers and services.

This is the first intrusion detection system built using a formal model of the underlying Modbus/TCP. Its initial experimental results provide evidence that model-based intrusion detection is a promising approach for monitoring process control networks. As stated earlier, model-based techniques may result in false alarms if the models aren't accurate. The authors do not describe the false alarms that their system generated during its evaluation.

B. Anomaly-Based Intrusion Detection

We discuss two anomaly-based intrusion detection systems in this section.

1) AutoAssociative Kernel Regression and Statistical Probability Ratio test SPRT: Yang et al [90] use the AutoAssociative Kernel Regression (AAKR) model coupled with the Statistical Probability Ratio test (SPRT) and apply them to local network consisting of several SUN servers and workstations to simulate a SCADA system.

The authors construct a local network consisting of For the simulated SCADA system. They used a previously developed condition monitoring technique, the Continuous System Telemetry Harness (CSTH), which was originally designed by Sun Microsystems [26], [88] to detect non-hostility induced anomaly but not for intrusion, to monitor the server activity and to build an initial base profile of its normal working status. Then the database is incorporated with a MATLAB-based Process and Equipment Monitoring (PEM) toolbox [29] to establish an initial baseline for the IDS. They consider Simple Network Management Protocol (SNMP) as the most important network traffic statistics.

Their fundamental methodology is pattern matching. Predetermined features representing network traffic and hardware operating statistics, such as link utilization, CPU usage, and login failure, are used by the AAKR model to predict the"correct" behavior. Then new observations are compared with past observations denoted as normal behavior. The comparison residuals are fed into the SPRT to determine to see whether fit within a predetermined confidence interval of the stored profiles. If yes, then an alarm is triggered.

Besides DoS attacks, ping flood, jolt2 attacks, bubonic attacks, simultaneous jolt2 and bubonic attacks, the authors also consider insider attack scenarios.

This work is potentially reproducible and may be used for other intrusion scenarios. However, the threshold value setting in the SPRT to determine false alarm and false negative rates seems arbitrary.

2) Multi-Agent IDS Using Ant Clustering Approach and Unsupervised Feature Extraction: Tsang and Kwong [81] propose an unsupervised anomaly-learning model - the Ant Colony Clustering Model (ACCM) in a multi-agent, decentralized IDS to reduce data dimensionality and increase modeling accuracy. The idea is bio-inspired from nature to construct statistical patterns of network data into near-optimal clusters for classification.

The Multi-Agent System (MAS) is of a tree-hierarchical structure and consists of autonomous agents which can be

assigned to different tasks. Depending on their tasks, these agents are categorized as *monitor agents,decision agents, action agents, coordination agents, user interface agents and registration agents*. They run on distributed subnets with cooperation.

Distributed in different locations, *monitor agents* gather information about the network traffic through packet capture engines. They extract independent features and reduce certain irrelevant and noisy data. The Principle Component Analysis (PCA) applies second-order statistics to extract principle components (PCs) as mutually orthogonal and linear combinations of original features for dimensionality reduction. Then the *decision agents* cluster the preprocessed data into different groups of normal and abnormal patterns. When abnormal patterns of network traffic is detected, they notify the action agents and coordination agents in the attacked subnet. Upon notification, *action agents* issue responses such as logging correlated TCP sessions into database, screening firewalls and redirect attacks to honeypots and so on. Each agent searches the feature space through random walking or jumping by short term memory, picks up and drops data objects according to local density of similarity measure.

When clustering high-dimensional intrusion data, potentially, there are two major problems. One is too many homogeneous clusters are created without convergence. The other is that impure clusters can be formed. The ACCM-based IDS leverages several factors to fine-tune the clustering. Firstly, it combines information entropy and averages the similarity to identify spatial regions of clusters. Secondly, it uses cluster-pheromone to search for compact clusters and object-pheromone to search for objects to be picked-up. This mechanism helps in optimal cluster formation. Thirdly, the short term memory that it employs consists of local regional entropy and average similarity of successfully dropped objects. Fourthly, the model employs a selection scheme to control the ant agent's population diversity.

The MAS offers efficient and decentralized control mechanism for large-scale intrusion detection. Multiple autonomous agents who are capable of different IDS related tasks work on distributed subnets and cooperate as well. The scalability of such multi-agent systems is due to the autonomy and versatility of each agent. The work offers detailed techniques on how to reduce data dimensionality and how to improve the precision of clustering thus improving the accuracy of detection. However, it doesn't give very specific information on how to the handle control networks explicitly and the implementation section is weak as well.

C. Configurable Embedded Middleware-Level Detection

Næss et al [55] present a configurable Embedded Middleware-level Intrusion Detection System (EMISDS) framework that is application specific. EMISDS comes with IDS-aware middleware tools to embed IDS sensors and detectors into an application's middleware layer instead of directly interacting with the low-level system and network interface. The system model is comprised of anomaly and misuse detection. EMIDS uses *interval-based* and *procedural-based* IDS sensors and *misuse-based* IDS detectors. Interval-based sensors are responsible for identifying whether parameter values and method invocation frequencies fall within their predefined ranges or not. They can be automatically injected into the stub and skeleton code by the IDS-aware Interface Definition Language (IDL) compiler. Procedural-based sensors embedded at the entry or exit points of application monitor its execution patterns. Misuse-based detectors reside within the application's source code at those locations where known vulnerabilities exist.

The structure of the application logic of the distributed objects is expressed in its interface definitions. By exploiting this application specific information, EMIDS provides reusable security policies such as predefined ranges for interval-based sensors and stored profiles of acceptable behavior for procedural-based sensors. It computes the execution profiles with a sliding window algorithm[14].

Responses policies such as to log events, delay invocations and determine connections are implemented either in the middleware or in the application layer. They can be configured globally to fit for the specific purpose of the application or particular clients.

The IDL compiler creates configuration files for client or server IDS implementation to specify the interaction among EMIDS' data, policy, profile and response.

The performance evaluation is conducted without implementing intrusions to see the overhead generated by the EMIDS framework and the set of security policies. The end-to-end latencies are checked for all the policies. The interval sensor has little overhead and adds a minor amount to the end-to-end latency.

This approach integrates intrusion detection in the middleware layer which does the resource intensive job of unmarshalling network packets thus saving the IDSs in the embedded components of the SCADA networks from doing it. It has the efficiency and flexibility for IDS reconfigurations including the instrumentation choice on IDS sensors and policies, provided that reconfiguring a middleware layer is cheaper than rewriting the application layer code for embedded systems/devices. However, as pointed out by the authors, there are several inherited fragileness in an embedded IDS system. One is that the response policy may alter the execution path of the application and may result in strange behavior. The other is the possible self-induced denial of service due to certain false positive responses.

D. Intrusion Detection and Event Monitoring in SCADA Networks

Oman and Phillips [57] from the University of Idaho give a very clear exposition on the implementation of a SCADA power-grid testbed for intrusion detection and event monitoring. Their work produce comprehensive intrusion

[14]Examples of two application-based policies for detectors are: defining the maximum number of connections allowed between a client and a server and preventing a client from making excessive connection requests within a certain time frame.

signatures for unauthorized access to SCADA devices besides baseline-setting files for those devices. Details about each SCADA device in the testbed such as its IP address, telnet port, legal commands for the device, are expressed using XML. A Perl program parses the XML profile and creates Snort IDS [72] signatures for legal commands on the RTU to monitor normal operations. For complex events whose signatures can't be automatically generated through above automated mechanism, certain extra steps are taken to produce their customized signatures. For example, failed password attempts, require pattern matching on the RTU's failed response to a bad login attempt. A packet sniffer is used to determine the response, and a customized signature is created to detect login failures before they are graphed. On the other hand, the system maintains a single settings repository which contains one or more baseline setting files for each device to monitor setting changes made either at the local terminal or over the network. The work also provides protection for the baseline data from unauthorized access and modification. Furthermore, their system consists revision control that enables device settings to be compared over time. Lastly, in order to monitor the uptime heath condition of the communication system, the authors use a PerlExpect script that runs every five minutes to log onto the devices and to verify if the issued simple command succeeds.

Evidently, the automated gathering and comparison of device settings over time is very useful to SCADA operators, who typically rely on personal notes and reminders about device settings. Their current prototype automates intrusion detection and settings retrieval for RTUs only . Special attention needs to be paid to the security of their revision control and uptime monitoring/polling, which potentially can be serious vulnerability on its own and a vector for Denial of Service (DOS) attacks[15].

E. Model for Cyber-Physical Interaction

1) Power Plant interfacing Substations through Probabilistic validation of attack-effect bindings (PVAEB): Rrushi and Campbell [74] looked into the attacks on the implementations of IEC 61850 [31], the protocol used for communication between electricity substation and power plant (a nuclear power plant in the paper).

The authors set out to probabilistically build a profile of legitimate data flows along with the main characteristics of the substation information exchanged between (Intelligent Electronic Devices) IEDs and communication services in IEC61850 invoked in an electrical substation interfacing with a power plant.

To abstract the semantic correlation between the dynamics of nuclear reactors in the power plant and those of the generated electricity provision in the substation, they used the `sem` package within the R®software for statistical computing to construct structural equations models[16] estimating the causality relations.

For each logical node of IEC 61850, they apply Bayesian Belief Networks (BBN)[17] via the MSBNx tool to enumerate the probability distributions attributed by its associated legitimate data and potential attack data respectively.

Then they used the Möbius tool to build the Stochastic Activity Network (SAN)[18]models to verify above bindings and to derive detection rules to spot intrusions.

Besides the simulated sensor data and nuclear power plant, the authors also simulated a distributed control system through a host-based network of virtual machines, which was running FreeModbus [87], a free implementation of Modbus protocol on an uClinux operating system [4]. They used the modpoll Modbus master simulator to gather simulated Modbus Protocol Data Units (PDUs) denoting typical status data of various components of a nuclear power plant, which includes the neutron monitoring system.

As noted by the authors, their intrusion detection rules are implementable in electrical substations and all construction of attack-effects are based on *known* failure models. Thus the work's capability to deal with *novel* attacks not clear.

2) Workflow-based non-intrusive approach for enhancing the survivability of critical infrastructures in Cyber Environment: Xiao et al [89] decompose a SCADA system into a physical layer and a cyber layer and propose a separate *workflow* layer above it. They consider that each essential component in the physical layer has a corresponding node in the *workflow*. Mathematically speaking, a *workflow* models both essential functionalities of the underlying physical layer and attack patterns derived domain specific security knowledge. This work leverages the *presumably* existing survivability-related knowledge and protection scheme to incorporate the detections of both *known* attack patterns and *known* unsafe states.

A simplified water treatment system is studied through simulation to illustrate the idea.

As acknowledged by the authors themselves, the system is only able to deal with *known* attacks and faults, which may not be viable for deployment at this stage.

The following two systems worth mentioning albeit lacking enough publicly available description on their technical details.

F. Modeling Flow Information and other Control Systems Behavior to Detect Anomalies

Moran and Belisle at IBM use a commercially available *Network Based Anomaly* solution to passively monitor the **flow** between routers and other network devices. Although

[15]For example, if an attacker gets unauthorized access to those monitoring devices and keeps issuing testing command

[16]The Structural equation modeling (SEM)[66] is a statistical technique for testing and estimating causal relationships using a combination of statistical data and qualitative causal assumptions. It can be used for both theory testing and theory development.

[17]Bayesian Belief Network is probabilistic graphic model that represents a set of random variables and their conational independencies via a directed acyclic graph.

[18]Stochastic Activity Network is a stochastic extension of Petri Nets for unified performancedependability evaluation of discrete distributed systems.

they are using slightly different terminologies than us in their paper [54], they apply quite comprehensive a combination of anomaly-, behavioral- and specification- based techniques to detect deviation from *normal* behavior. Since it's flow-based, this solution focuses more on network layer detection and can't investigated attacks specifically crafted at application layer. No analysis on false alarm or missed detection rate is available.

G. SHARP

Security-Hardened Attack Resistant Platform (SHARP) [71] designed by Pacific Northwest National Laboratory is also a front-end processor and resides between the network connection and all I/O ports within the Intranet inside a Master Terminal Unit (MTU).

It's done through user authentication and privilege escalation protection – unauthorized physical or network access by malicious users or software are detected and blocked. Its threat model also provides self validation, i.e., attacks can be launch from intranet.

VI. Comparison

The overall comparisons of the proposed systems are listed in Table II and Table III. The rationale behind choosing the features we used for comparison is drawn out of operational concerns besides performance issues. Most terms are expected to be self-explanatory. Some of them are derived from works by Axelsson[9] and McHugh[50].

In particular,

- *SCADA-specific* refers to whether SCADA-specific protocols, or the hierarchical structure, or the cyber-physical interaction of SCADA systems are analyzed, and the
- *degree of SCADA-specific-ness* is measured and compared relatively among the systems we studied, more itemized comparison seen Table VI-B.
- *self-security* refers to whether the proposed IDS itself is secure in the sense it will fail-safe.
- *fallacy analysis* refers to whether the proposed system contain discussions on the false alarm and false negative (miss detection) rate[9].
- *Unit of Analysis* refers to the base unit upon which the proposed system makes intrusion detection decision.

Furthermore,

- *Data Processing* refers to the location of monitored data being processed for intrusion detection and analysis purpose, namely, central or distributed location. Similarly,
- *Data Collection* refers to the location where the intrusion detection sensors are placed.
- *Granularity* refers to whether these data are processed continuously or in batch.
- *Type of Response* refers to the IDS passively watches traffic or actively contributes to the decision of relaying the traffic.
- *Interoperability* refers to whether the proposed IDS has the capability of interacting without likely SCADA components.

A. Intrusion Detection

For more qualitative aspects , we'd like to look into the intrusion detection methods used in each system, seen in Table IV, where

- *Detection Type* refers to the intrusion types that we listed above in IV-A.
- *Intrusion only* refers to whether the proposed IDS can detect only intrusion or both intrusion & non-malicious fault or is extensible in achieving the both.
- *Detection Method/Algorithm* refers to the detailed algorithm for computation purpose that the proposed IDS employs.

B. SCADA-Specific-ness

We explicitly compare how SCADA's special needs are addressed in each proposed system with results shown in Table V, where the terms are mostly self-explanatory or were mentioned earlier. Note that we refined the desired *security properties* of the proposed IDS to its *timeliness* and *availability*. *Timeliness* is particularly stressed in light the fact that SCADA systems are *hard real-time* systems while the desired property of *availability* further breaks down to the *self-security* and *type of response* of the IDS, two items stipulated by the 24×7 operational requirement of SCADA systems.

VII. Evaluation

A. Design Pitfalls and Evaluation Criteria

Looking at IT standard IDSs, McHugh [49] critiqued many aspects of the DARPA/MIT Lincoln Lab evaluation. In terms of modeling, by which we mean not only the conventional mathematical system modeling employed in the standard control theory but also what's implied in the general sense of *abstraction* of features as its classic usage in machine learning. More specifically, both signature and probabilistic IDSs model misuse, the *illegal* behavior of an intrusion while anomaly-based IDSs empirically and statistically model normal system usage and behavior. And specification-based IDSs define what is allowable under protocol and policy specification. All these model-based approaches bear certain common drawbacks:

- Inaccurate models can lead to false alarms and/or missed detections.
- Modeling can be expensive and difficult if the system and/or user activity is complex.

When it comes to the application of abstraction and classification, Anderson states [7] "In general, if you build an intrusion detection system based on data-mining techniques, you are at serious risk of discriminating."

Paxson has a similar argument, even more from a technical point of view [62], [78] that one of the pitfalls of machining learning based IDS techniques is the lack of illumination for the rationale behind many approaches on how they decide to take such approach; and why they succeed in doing so or why they fail in achieving.

Name of System	Publ. year	Degree of SCADA Specific	Specific Domain	Detection Prevention Principle	Malicious Intrusions only?	Threat model	Time of Detection	self-Security	Fallacy Analysis	Unit of analysis
PVAEB [74]	2008	high	electrical power	proba.	fault & intrusion	no	N/A	low	no	packet
IBM NADS [54]	2008	medium	N/A	anomaly, spec, behavioral	extensible	outsider not explicit	Non-real	low	no	flow--based
SRI Modbus [17]	2007	high	N/A	spec. proba.	extensible	outsider	real	medium	no	packet
WFBNI [89]	2007	high	water treatment system	signature	unintent. faults unsafe states	not explicit	on-line prediction	low	no	N/A
SHARP [71]	2008	medium	N/A	spec. encryp.	extensible	insider or outsider	on-line	high	no	N/A
IDEM [57]	2007	high	electrical power	signature	yes	unauth. access	real	low	no	packet
AAKR--SPRT [90]	2006	high	N/A	anomaly	yes	insider & outsider	real	low	no	packet
EMISDS [55]	2005	low	N/A	anomaly, spec., signature	yes	no	real	low	no	procedural interval
MAAC--UFE [81]	2004	medium	N/A	anomaly	yes	both	real	N/A	yes	N/A

TABLE II
COMPARISON OF INTRUSION DETECTION SYSTEM APPROACHES

Name of System	Data Proc.	Data Coll.	Scalab--ility	Granul--arity	Audit Source	Type of Response	Inter-oper.	Imple-ment.	Deploy. ment	Real traces
PVAEB [74]	centr.	centr.	medium	batch	host	passive	N/A	yes	no	testbed
IBM NADS [54]	centr.	dist.	high	cont.	network	passive	yes	yes	no	N/A
SRI Modbus [17]	dist.	dist	high	cont.	both	active	yes	yes	no	testbed
WFBNI [89]	centr.	dist.	high	cont.	network	passive	N/A	yes	no	simulation
SHARP [71]	centr.	centr.	low	cont.	network	active	yes	no	no	N/A
IDEM [57]	centr.	centr.	low	cont.	network	passive	yes	yes	no	testbed
AAKRSPRT[90]	centr.	centr.	low	cont.	host	passive	yes	yes	no	testbed
EMISDS [55]	dist.	dist.	high	batch.	both	N/A	N/A	no	no	simulation w/o intrusion
MAACUFE [81]	dist.	dist.	high	N/A	both	active	N/A	yes	no	KDD-cup

TABLE III
COMPARISON OF INTRUSION DETECTION SYSTEM APPROACHES: CONTD.

Name of System	Detection Type	Intrusion only	Detection Method / Algorithm
PVAEB [74]	anomaly	fault intrusion	Structural Equation Modeling, Bayesian Belief Networks, Stochastic Activity Networks
IBM NADS [54]	anomaly, behavioral specification	N/A	net flow matching
SRI Modbus [17]	spec., prob.	extensible	descriptive statistics, simple rule based
WFBNI [89]	signature	fault intrusion	matching fault model
SHARP [71]	spec.	extensible	N/A
IDEM [57]	signature	yes	N/A
AAKRSPRT[90]	anomaly	yes	AAKR, SPRT, pattern matching
EMISDS [55]	anomaly, spec. signature	yes	simple rule based, sliding window
MAACUFE [81]	anomaly	yes	ACCM, PCA

TABLE IV
COMPARISON OF INTRUSION DETECTION METHOD IN EACH PROPOSED SYSTEM

Name of System	Security Properties			Inter. oppp	Use of SCADA Components					Interaction between Cyber – Physical
	Time--liness	Availability			Domain/ Industry	HW	SW	communication		
		Self Security	Type Response					hardware	protocol	
PVAEB [74]		low	passive	N/A	electrical power			simulated IED	IEC 61850 DNP3	yes
IBM		low	passive	yes					Modbus	
NADS [54]										
SRI Modbus [17]		medium	active	yes	N/A				Modbus	
WFBNI [89]		low	passive	N/A	water					yes
SHARP [71]		high	active	yes	N/A					
IDEM [57]		low	passive	yes	electrical power	yes				
AAKRSPRT [90]		low	passive	yes	N/A				SNMP	
EMISDS [55]	yes	low	passive	N/A	N/A					
MAACUFE [81]		N/A	active	N/A	N/A		yes			

TABLE V
COMPARISON OF SCADA'S SPECIAL NEEDS BEING ADDRESSED IN EACH PROPOSED SYSTEM

According to Axelsson [9], McHugh [50] and Paxson [62], we shall look for

- soundness
- completeness
- timeliness
- choice of metrics, statistical models, profiles
- system design
- feedback: or how to decide actionable events
- social implications

The SCADA-specific angles we look at are: What are their contributions, limitations or room for improvement, extensibleness in terms of

- How do they frame the work including assumptions, logics and conclusions?
- What kind of security properties do they want to achieve? Do they achieve and how?
- What are their trust model, threat model and attack scenarios? How plausible?
- What are the illuminations they bring into the problem

space;
- What's the selling point of their approach?
- What kind of detection algorithms they've used that suit SCADA systems particularly well
    1) either through leveraging the entrenched components and/or technologies used in the specific SCADA physical systems under their study;
    2) or restrict their attention to a more focused and potentially narrowed workspace that are more relevant to specific SCADA physical system under their study when applying generic methods.
- What are the subtle points they bring out that might have been simply left out by a non-SCADA-security expert?
- What's unique in the cyber-physical interactions?
- How is the detection performance measured in terms effectiveness and efficiency? Effectiveness is reflected through high detection rate and low false alarm rate; and efficiency overheads.

B. Evaluation Results

Intrusion detection research for SCADA systems to date has been quite limited, with the three most prominent and critical deficiencies being

- the lack of a well-considered threat model;
- the absence of addressing false alarm and false negative (mis-detection) rates; and
- the need to empirically ground the development of IDS mechanisms in the realities of how such systems operate in practice, including the diversity of traffic they manifest and the need to tailor IDS operation to different SCADA environments.

From the above evaluation of existing IDSs for SCADA systems, we can see that the current bottleneck problems faced by research and design henceforth implementation and deployment of IDS for SCADA are the scarce access to operational SCADA system (network and system traffic) traces and the lack of prudent yet novel threat models, or attack scenarios.

Barely any of these systems has a performance evaluation on the false alarms that it generates. However, given the availability demand of SCADA systems, we believe this is an issue that must be addressed well before IDS can be implemented and deployed in SCADA systems at large scale.

In contrast to what we explained in II regarding the potential seriousness of ambiguity-induced envision problem faced by the network IDS and more so by the SCADA-specific IDS, none of the work we surveyed has touched upon this issue yet.

VIII. Future Directions

Ultimately, any viable technical solutions and research directions in securing SCADA systems must lie in the conjunction of computer security, communication network and control engineering. However, the very large installed base of such systems means that in many instances we must for a long time to come rely on retrofitted security mechanisms, rather than having the option to design them in from scratch. This leads to a pressing need for deployable, robust, SCADA-specific intrusion detection systems (IDS).

We shall aim to capture the characteristics of a specific SCADA system under study with full situational awareness, including the dynamics of the physical plant being monitored, its communication patterns, system architecture, network traffic behavior, and specific application-level protocols used including the envision problem.

A. Our Work-in-Progress

We propose a JIE[19], a *viable* intrusion detection and self-hardening system for SCADA systems [94].

In terms of the functionalities of intrusion detection and prevention, our proposed JIE would be able to

- efficiently detect and block cyber intrusions into SCADA systems in real operational environments, and in real-time,
- without interrupting the control performance of the protected system,
- without creating extra operational burdens or operational reservations due to false alarms,
- in the presence of both malicious and messily benign network traffic. The system must operate in a real-time, robust fashion, with performance adequate to meet the demands of the dynamic cyber-physical interactions inherent to SCADA systems.

In particular, an earlier detection and resilient estimation scheme for SCADA systems in an uncertain network environment is currently explored more technically. Without any prior knowledge of the occurrence time and distribution of the outliers or anomalies, this online recursive algorithm robustly identifies and detects them among the measurements by using a robustified window-limited sequential Generalized Likelihood Ratio Test. The choice of this fixed yet approximately optimal window size provides guaranteed delay to detection time under the constraint of false alarm rate conditions when identifying outliers. Further, this resilient and flexible estimation scheme robustly rectifies and cleans data upon both isolated and patchy outliers while maintain the optimality of the nominal condition.

In response to the ambiguities in network traffic, our earlier detection algorithm utilizes robust statistical tools to resolve the issue of identifying two signals in a *least favorable* setting. We are also paying extra attention at the network detection level to reduce the impact of the potential envision problem.

[19]This is the 40th hexagram of *I Ching*, or, *Yi Jing*, *The Book of Changes*, comprising of 64 hexagrams plus their commentaries and transformations as strategic interpretation of chance event. It literally means *Problem Solving* or *Deliverance*. The essence of this strategy is: Don't trouble troubles until trouble troubles you; If it does, then act quick.

IX. Conclusion

As argued by Rakaczky [70], the ease of deployment requires the intrusion detection/prevention strategy to minimize the associated personnel overhead.

The model-based system for SCADA system using Modubs/TCP addresses Modbus protocol encapsulated within TCP/IP. The idea can be generalized to other control system protocols as well.

Since SCADA networks are built of resource-constrained embedded systems, the IDS using the middleware-level detection has the advantage of directly accessing message signatures and parameter values without decoding the raw network packets. But there is a tradeoff in the risk involved in handling embedded responses to attacks.

Both model-based intrusion detection and middleware-level intrusion detection build models to specify the normal behavior of the network traffic and compare the SCADA traffic against these models to detect potential anomalous behavior. Model-based detection is an important complement to signature-based approaches.

The specification-based IDS has an inviting advantage to SCADA systems and networked control systems in general.

X. Acknowledgement

The authors gratefully acknowledge Professor Vern Paxson for sharing his insight and expertise with us on intrusion detection and network security besides his thorough review of the first draft. Moreover, his guidance, teaching and research work have greatly shaped this paper and its follow-on work-in-process. Special thanks also go to several anonymous reviewers and numerous domain experts with whom we've been lucky enough to have had many stimulating discussions.

References

[1] *Cybersecurity of PCS/SCADA Networks: Half-baked Homeland Security*, June, 2006 http://www.bechteltelecoms.com/docs/bttj_v4n2/Article04.pdf

[2] EPRI *Anomaly-Based Intrusion Detection in SCADA (Supervisory Command and Data Acquisition* http://www.epriweb.com/public/RS_1002598.pdf

[3] AGA Report No.12, *Crptographic Projection of SCADA Communications Part1: Background, Policies and Test Plan*, American Gas Association, March 2006

[4] K. Albanowski, and D.J. Dionne, *Embedded Linux Microcontroller Project*, http://www.uclinux.org

[5] Dominique Alessandri *Attack-Class-Based Analysis of Intrusion Detection Systems*. Ph.D Thesis, 2004. University of Newcastle upon Tyne, School of Computing Science. Newcastle upon Tyne, UK.

[6] Julia Allen, Alan Christie, William Fithen, John McHugh, Jed Pickel, Ed Stoner *State of the Practice of Intrusion Detection Technologies*,TECHNICAL REPORT CMU/SEI-99-TR-028, January 2000 http://www.sei.cmu.edu/pub/documents/99.reports/pdf/99tr028.pdf

[7] Ross Anderson, *Security Engineering – A Guide to Building Dependable Distributed Systems*, 2001, Wiley. ISBN 0-471-38922-6.

[8] Stefan Axelsson *Research in Intrusion Detection Systems: A Survey*, Technical Report. Department of Computer Engineering, Chalmers University of Technology, Göteborg, Sweden, 1999.

[9] Stefan Axelsson *Intrusion Detection Systems: A Survey and Taxonomy*, Technical Report, Department of Computer Engineering, Chalmers University of Technology, Göteborg, Sweden, 2000

[10] Stefan Axelsson *A preliminary attempt to apply detection and estimation theory to intrusion detection* Technical Report 00-4, Department of Computer Engineering, Chalmers University of Technology, Göteborg, Sweden, March 2000.

[11] Stefan Axelsson *The Base-Rate Fallacy and the Difficulty of Intrusion Detection*. In ACM Transaction on Infromation and System Security (TISSEC), 3(3), pp. 186-205, ACM Press, ISSN: 1094-9224, 2000

[12] Ivan Balepin, Sergei Maltsev, Jeff Rowe, and Karl Levitt *Using Specification-Based Intrusion Detection for Automated Response*, in the Proceeding of the 6th International Symposium, RAID 2003, Recent Advances in Intrusion Detection, Pittsburgh, PA, September 8-10, 2003.

[13] Buhan, I. and P. Hartel, *The state of the art in abuse of biometrics*. Technical Report TR-CTIT- 05-41 Centre for Telematics and Information Technology. University of Twente, Enschede, 2005.

[14] Roman V. Yampolskiy and Venu Govindaraju *Computer Security: a Survey of Methods and Systems*, Journal of Computer Science 3 (7): 478-486, 2007.

[15] Steven Bellovin *Packets found on an internet*. SIGCOMM Computer Communiation Review. 23, 3 (July 1992), 26-31.

[16] Frank Callier, Charles Desoer, *Linear System Theory*, Springer-Verlag, New York, 1991

[17] Steven Cheung, Bruno Dutertre, Martin Fong, Ulf Lindqvist, Keith Skinner, Alfonso Valdes, *Using Model-based Intrusion Detection for SCADA Networks*, SCADA Security Scientific Symposium, 2007

[18] H. Debar, M. Dacier, and A. Wepsi. *A Revised Taxonomy for Intrusion-Detection Systems*. IBM Research Report. 1999.

[19] Sarang Dharmapurikar and Vern Paxson, *Robust TCP Stream Reassembly in the Presence of Adversaries*, USENIX Security 2005

[20] Dacfey Dzung, Martin Naedele, Thomas Von Hoff and Mario Crevatin *Security for Industrial Communication Systems*, Procedings of the IEEE, VOL. 93, NO. 6, Page 1152 - 1177, JUNE 2005

[21] Carl Endorf, Jim Mellander, *Intrusion Detection & Prevention*, McGraw-Hill Professional, 2004, ISBN 0072229543

[22] Wei Fan, Matthew Miller, Salvatore J. Stolfo, Wenke Lee, Philip K. Chan: *Using artificial anomalies to detect unknown and known network intrusions*. Knowl. Inf. Syst. 6(5): 507-527 (2004)

[23] GAO: United States Government Accountability Office, *Critical Infrastructure Protection Challenges and Efforts to Secure Control Systems*, Report to Congressional Requesters, March 2004,http://www.gao.gov/new.items/d04354.pdf.

[24] GAO: United States Government Accountability Office, *Critical Infrastructure Protection Multiple Efforts to Secure Control Systems Are Under Way, but Challenges Remain*, Report to Congressional Requesters, GAO-07-1036, September, 2007 http://www.gao.gov/new.items/d071036.pdf

[25] Mark Grimes, *SCADA Exposed* http://www.toorcon.org/2005/slides/mgrimes/mgrimes-scadaexposed.pdf

[26] Kenny Gross, Keith Whisnant, Aleksey Urmanov, Kalyan Valdyanathan, Sajjit Thampy, *Continuous System Telemetry Harness*, Tech. Rep., [Online] Available: http://research.sun.com/sunlabsday/docs.2004/talks/1.03_Gross.pdf, 2005.

[27] Joshua W. Haines, Lee M. Rossey, Richard P. Lippmann, and Robert K.Cunningham, *Extending the DARPA off-line intrusion detection evaluations*, Proceedings of DARPA Information Survivability Conference & Exposition II, 2001. DISCEX '01. Volume 1, 12-14 June 2001 Page(s):35 - 45 vol.1

[28] Mark Handley, Christian Kreibich and Vern Paxson, *Network Intrusion Detection: Evasion, Traffic Normalization, and End-to-End Protocol Semantics*, Proc. USENIX Security Symposium 2001.

[29] J.Wesley Hines and Dustin Garvey, *The Development of a Process and Equipment Monitoring (PEM) Toolbox and its Application to Sensor Calibration Monitoring*, The Fourth International Conference on Quality and Reliability, 9 - 11 August, 2005, Beijing, P.R. China.

[30] International Electrotechnical Commission, *IEC TS 62351: Power systems management and associated information exchange Data and communications security*, 2007.

[31] International Electrotechnical Commission, *IEC 61850: Communication Networks and Systems in Substations*, part 1 through 9, 2004.

[32] IEEE Std C37.1-1994, *IEEE Standard Definition, Specification, and Analysis of Systems Used for Supervisory Control, Data Acquisition, and Automatic Control*, The Institute of Electrical and Electronics Engineers, Inc. Published 1994, New York, NY.

[33] David Kahneman, Amos Tversky *Prospect Theory: An Analysis of Decision under Risk*. Econometrica 47, 263-291.(1979)

[34] KDD Cup Datasets, http://kdd.ics.uci.edu/

[35] Karen Kent, Peter Mell, *Guide to Intrusion Detection and Prevention (IDP) Systems (DRAFT)*, Recommendations of the National Institute of Standards and Technology, Special Publication 800-94, August 2006.

[36] Kevin Killourhy, Roy Maxion, and Kymie Tan, *A Defense-Centric Taxonomy Based on Attack Manifestations*. In International Conference on Dependable Systems & Networks (DSN-04), pp. 102-111, Florence, Italy, 28 June - 01 July 2004. IEEE Computer Society Press, Los Alamitos, California, 2004.

[37] Calvin Ko, *Execution Monitoring of Security-critical Programs in a Distributed System: a Specification-based Approach*, Dissertation, Department of Computer Science, University of California at Davis, 1996.

[38] Christopher Kruegel, Darren Mutz, William Robertson and Fredrik Valeur, *Bayesian Event Classification for Intrusion Detection*, in Proceedings of the 19th Annual Computer Security Applications Conference (ACSAC 2003)

[39] Ronald Krutz, *Securing SCADA systems*, Wiley, 2006.

[40] Landon Lewis, Dale Peterson, *SCADA Honeynet Results from the PCSF Annual Meeting*, available https://www.pcsforum.org/library/files/1174588590-PCSF_SCADA_Honeynet.pdf

[41] Ted Lewis,*Critical Infrastructure Protection in Homeland Security – Defending a Networked Nation*, John Wiley & Sons, Inc., Hoboken, New Jersey, 2006

[42] D. Denning. *An Intrusion-Detection Model*. IEEE Transactions on Software Engineering, 13(2), Feb. 1987.

[43] T. Lunt. *Detecting Intruders in Computer Systems*. In Proceedings of the 1993 Conference on Auditing and Computer Technology. 1993.

[44] Zhuowei Li, Amitabha Das, Jianying Zhou, *USAID: Unifying Signature-Based and Anomaly-Based Intrusion Detection*,In PAKDD, pages 702-712,2005.

[45] Richard P. Lippmann, David J. Fried, Isaac Graf, Joshua W. Haines, Kristopher R. Kendall, David McClung, Dan Weber, Seth E. Webster, Dan Wyschogrod, Robert K. Cunningham, and Marc A. Zissman, *Evaluating Intrusion Detection Systems: the 1998 DARPA Off-Line Intrusion Detection Evaluation*, in Proceedings of the 2000 DARPA Information Survivability Conference and Exposition (DISCEX), Vol. 2, January 2000, IEEE Press.

[46] D. L. Lough. *A Taxonomy of Computer Attacks with Applications to Wireless Networks*. PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, April 2001

[47] Matthew V. Mahoney, Philip K. Chan *An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection*, RAID 2003: 220-237

[48] Roy A. Maxion, Kymie M.C. Tan *Benchmarking anomaly-based detection systems*, Proceedings of International Conference on Dependable Systems and Networks, DSN 2000. Pages 623-630

[49] John McHugh, *Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory*, Proc. ACM TISSEC 3(4) 262-294, 2000.

[50] John McHugh, *Intrusion and Intrusion Detection*, Published online: 27 July 2001, Springer-Verlag

[51] A. David McKinnon. *Supporting Fine-grained Configurability with Multiple Quality of Service Properties in Middleware for Embedded Systems*. Doctoral Dissertation, School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA, December 2003.

[52] Modbus IDA. *Modbus messaging on TCP/IP implementation guide* v1.0a, June 4, 2004.

[53] Modbus IDA. *Modbus application protocol specification* v1.1a, June 4, 2004.

[54] Brian Moran, Rick Belisle, *Modeling Flow Information and Other Control System Behavior to Detect Anomalies*, in Proceeding of S4: SCADA Security Scientific Symposium, Miami, FL, Janunary 2008

[55] Eivind Nss, Deborah A. Frincke, A. David McKinnon, David E. Bakken, *Configurable Middleware-Level Intrusion Detection for Embedded Systems*, The 25th ICDCSW, 2005

[56] Tevfik Nas, *Cost-Benefit Analysis: Theory and Application*, SAGE Publications, February 1996

[57] Paul Oman, Matthew Phillips, *Intrusion Detection and Event Monitoring in SCADA Networks*, book chapter of *Critical Infrastructure Protection*, Pages 161-173, Springer Boston, 2007

[58] Richard P. Lippmann, David J. Fried, Isaac Graf, Joshua W. Haines, Kristopher R. Kendall, David McClung, Dan Weber, Seth E. Webster, Dan Wyschogrod, Robert K. Cunningham, and Marc A. Zissman, *Evaluating Intrusion Detection Systems: the 1998 DARPA Off-Line Intrusion Detection Evaluation*, in Proceedings of the 2000 DARPA Information Survivability Conference and Exposition (DISCEX), Vol. 2, January 2000, IEEE Press.

[59] Pacific Northwest National Laboratory, U.S. Department of Energy *The Role of Synchronized Wide Area Measurements for Electric Power Grid Operations* Position Paper for the National Workshop Beyond SCADA: Networked Embedded Control for Cyber Physical Systems (HCSS-NEC4CPS),November 8-9, 2006 http://www.truststc.org/scada/papers/paper23.pdf

[60] Vern Paxson, Sally Floyd, *Why We Don't Know How To Simulate The Internet*, Proceedings of the 1997 Winter Simulation Conference, December 1997. pages 1037–1044

[61] Vern Paxson, *Bro: A System for Detecting Network Intruders in Real-time*. Computer Network Journal 23-24 (December 1999), 2435-2463.

[62] Vern Paxson, *Topics in Network Intrusion Detection*. Tutorial, 8th ACM Conference on Computer and Communications Security (CCS-8), November, 2001.

[63] Vern Paxson, *Strategies for Sound Internet Measurement* Proceedings of ACM Internet Measurement Conference, October 2004.

[64] Vern Paxson, *Considerations and Pitfalls for Conducting Intrusion Detection Research* Keynote, Fourth GI International Conference on Detection of Intrusions & Malware, and Vulnerability Assessment (DIMVA), July 2007. http://www.icir.org/vern/talks/vp-IDS-Pitfalls-DIMVA07.pdf

[65] Thumanoon Paukatong, *SCADA Security: A New Concerning Issue of an In-house EGAT-SCADA*, 2005 IEEE/PES Transmission and Distribution Conference & Exhibition: Asia and Pacific Dalian, China

[66] J. Pearl, *Causality: Models, Reasoning, and Inference*, Cambridge University Press, ISBN 0521773628, second edition, 2001.

[67] Charles Pfleeger, Shari Pfleeger, *Security in Computing*, 3rd ed., Prentice Hall, Upper Saddle River, NJ, 2003, ISBN 0-13-035548-8

[68] Niels Provos , Thorsten Holz, *Virtual Homeypots From Botnet Tracking to Intrusion Detection*, Addison-Wesley, Boston, MA 2008

[69] Thomas H. Ptacek and Timothy N. Newsham,*Insertion, Evasion, and Denial Of Service: Eluding Network Intrusion Detection*, Secure Networks techncial report, 1998

[70] Ernest Rakaczky, *Intrusion Insights Adapting Intrusion Prevention Functionality for Process Control/SCADA Systems*, position paper in *Beyond SCADA: Networked Embedded Control for Cyber Physical Systems*, Pittsburgh, Pennsylvania, Noverember, 2006 http://www.truststc.org/scada/papers/paper24.pdf,

[71] Eric Robinson, Brad Woodworth, Ron Pawlowski, *Security-Hardened Attack-Resistant Platform (SHARP)*, Pacific Northwest National Laboratory I3P Security Tools Team, available https://www.thei3p.org/projects/pcs_publications.html

[72] Martin Roesch, *Snort - Lightweight Intrusion Detection for Networks*, Proceedings of LISA '99: 13th Systems Administration Conference, USENIX

[73] Lee M. Rossey, Robert K. Cunningham, David J. Fried, Jesse C. Rabek, Richard P. Lippmann, Joshua W. Haines, and Marc A. Zissman, *LARIAT: Lincoln Adaptable Real-time Information Assurance Testbed*, Proceedings of IEEE Aerospace Conference, Volume 6, Page(s):6-2671-2676, 6-2678 - 6-2682 vol.6 March 9-16, 2002

[74] Julian Rrushi and Roy Campbell, *Detecting Attacks in Power Plant Interfacing Substations through Probabilistic Validation of Attack-Effect Bindings*, in Proceeding of S4: SCADA Security Scientific Symposium, Miami, FL, January 2008

[75] Bruce Schneier, *Beyond Fear Thinking Sensibly about Security in an Uncertain World*, Copernicus Books, Springer-Verlag, September 2003

[76] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, and S. Zhou, *Specification-based anomaly detection: a new approach for detecting network intrusions*, in Proceedings of the 9th ACM Conference on Computer and Communications Security, pages 265 274. ACM Press, 2002.

[77] G. G. Simpson. Principles of Animal Taxonomy. Columbia University Press, New York, 1961, Fourth printing 1969.

[78] Robin Sommer, Vern Paxson, *Outside the Closed World: On Using Machine Learning For Network Intrusion Detection*, Proc. IEEE Symposium on Security and Privacy (to appear), 2010

[79] P. H. A. Sneath and R. A. Sokal. Numerical Taxonomy. W. H. Freeman and Company, San Francisco, 1973.

[80] Keith Stouffer, Joe Falco, Karen Kent, *Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems*

Security – Recommendations of the National Institute of Standards and Technology, Special Publication 800-82, Initial Public Draft, September 2006

[81] Chi-Ho Tsang, Sam Kwong, *Multi-Agent Intrusion Detection System in Industrial Network using Ant Colony Clustering Approach and Unsupervised Feature Extraction*, In Proceeding of IEEE International Conference on Industrial Technology Page 51- 56, ICIT 2005.

[82] Patrick Tsang, Sean Smith *YASIR: A Low-Latency, High-Integrity Security Retrofit for Legacy SCADA Systems*, Dartmouth Computer Science Technical Report TR2007-603, Spetember 24, 2007.

[83] Amos Tversky, David Kahneman *Loss Aversion in Riskless Choice: A Reference Dependent Model*. Quarterly Journal of Economics 106, 1039-1061. 1991

[84] United States. Congress. House. Committee on Homeland Security. Subcommittee on Economic Security, Infrastructure Protection, and Cybersecurity. *SCADA systems and the terrorist threat: protecting the nation's critical control systems: joint hearing before the Subcommittee on Economic Security, Infrastructure Protection, and Cybersecurity with the Subcommittee on Emergency Preparedness, Science, and Technology of the Committee on Homeland Security, House of Representatives, One Hundred Ninth Congress, first session, October 18, 2005* Serial No. 109-45, Washington: U.S. G.P.O.: For sale by the Supt. of Docs., U.S. G.P.O., 2007. http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=109_house_hearings&docid=f:32242.pdf

[85] Jacob W. Ulvila, John E. Gaffney, Jr., *Evaluation of Intrusion Detection Systems*, Journal of Research of the National Institute of Standards and Technology, Volume 108, Number 6, November-December 2003, Pages 453-473.

[86] Lisa Vaas, *Hole Found in Protocol Handling Vital National Infrastructure* eWeek, http://www.eweek.com/article2/0,1759,2107265,00.asp, March 23, 2007

[87] C. Walter, *FreeMODBUS library*, http://www.freemodbus.org/

[88] Keith Whisnant, Kenny Gross, Natasha Lingurovska, *Proactive Fault Monitoring in Enterprise Servers*, in Proceedings of the 2005 International Conference on Computer Design, pp. 3-10, June 2005.

[89] Kun Xiao, Nianen Chen, Shangping Ren, Limin Shen, Xianhe Sun, Kevin Kwiat, Michael Macalik, *A Workflow-based Non-intrusive Approach for Enhancing the Survivability of Critical Infrastructures in Cyber Environment*, in Proceedings of Third International Workshop on Software Engineering for Secure Systems (SESS'07)

[90] Dayu Yang, Alexander Usynin, and J. Wesley Hines, *Anomaly-Based Intrusion Detection for SCADA Systems*, International Atomic Energy Agency (IAEA), Technical Meeting on Cyber Security, Idaho, 2006

[91] John Zachary, John McEachen and Dan Ettlich *Conversation Exchange Dynamics for Real-Time Network Monitoring and Anomaly Detection*, Proceedings of the Second IEEE International Information Assurance Workshop (IWIA04), 2004

[92] Stefano Zanero, *Behavioral Intrusion Detection*, in Proceedings of 19th International Symposium on Computer and Information Sciences - ISCIS, pp. 657-666, October 2004.

[93] Bonnie Zhu, Anthony Joseph and Shankar Sastry, *Taxonomy of Cyber Attacks on SCADA Systems*, 2008.

[94] Bonnie Zhu and Shankar Sastry, *Jie: A Viable Intrusion Detection System for SCADA Systems*, working paper

[95] Bonnie Zhu and Shankar Sastry, *BEAVER (Berkeley Efficient And Viable Electric Ranger) for Critical Infrastructures*, 2008

NOTES

NOTES (cont.)

NOTES (cont.)

NOTES (cont.)