9-2007

Collaborative, Trust-Based Security Mechanisms for a National Utility Intranet

Gregory M. Coates

COLLABORATIVE, TRUST-BASED SECURITY MECHANISMS FOR A NATIONAL UTILITY INTRANET

THESIS

Gregory M. Coates, Major, USAF

AFIT/GIA/ENG/07-05

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

AFIT/GIA/ENG/07-05

COLLABORATIVE, TRUST-BASED SECURITY MECHANISMS FOR A NATIONAL UTILITY INTRANET

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Cyber Operations

Gregory M. Coates, BS

Major, USAF

September 2007

AFIT/GIA/ENG/07-05

COLLABORATIVE, TRUST-BASED SECURITY MECHANISMS FOR A NATIONAL UTILITY INTRANET

Gregory M. Coates, BSEE

Major, USAF

Approved:

Dr. Kenneth M. Hopkinson (Chairman)	Date
Maj Scott R. Graham, PhD (Member)	Date
Lt Col Stuart H. Kurkowski, PhD (Member)	Date

Abstract

Plans by utility standards organizations and privately-owned companies to transition control and monitoring of the US power grid and other utility infrastructures from simple, proprietary protocols to open, IP-based architectures and standards will reduce operating costs and expand customer support options but will also face several serious obstacles to implementation. First, TCP/IP and the Internet were never designed for the hard real-time packet delivery required by SCADA systems. Second, the alarming rise each year in reported corporate downtime, financial loss, and espionage from insiders and Internet attackers, often using widely available exploits, foreshadows an increasing vulnerability of utility data and control systems. With the swift move to embrace IP-based control systems, there is surprisingly little available research regarding means to ensure continuous, safe, and secure operation of these critical infrastructures in the face of determined cyber threats.

This thesis investigates network security policies and mechanisms for control system networks using a mix of TCP and UDP transport protocols over IP. It recommends flexible, scalable, modular, and cost-effective security solutions that can be added in strategic locations to protect existing legacy architectures and accommodate transition to IP standards. User-definable rules and responses enact the unique policies of organizations that must operate with zero failures in environments with varying levels of uncertainty and trust.

This thesis proposes and evaluates a comprehensive and collaborative security concept, defined as a **trust system**, that is based on a best-of-breed application of standard IT network security mechanisms and IP protocols. The **trust system** provides seamless, automated command and control for suppression of network attacks and other

suspicious events. It also supplies access control, format validation, event analysis, alerting, blocking, and event logging at any network-level and can do so on behalf of any system that does not have the resources to perform these functions itself.

This thesis simulates layering mechanisms for encryption, authentication, traffic filtering, content checks, and event correlation over real-time data acquisition, control, and protection signaling in order to mitigate malicious activities from both internal and external sources. Latency calculations are used to estimate limits of applicability within a company and between geographically separated company and area control centers, scalable to hierarchical regional and national implementations.

A successful solution at any level requires balancing the protection of private communities of interest while fostering a combination of centralized and distributed emergency prediction, mitigation, detection, and response. To achieve this, while meeting strict time constraints, secure and dynamic peer-to-peer mechanisms are assisted by bandwidth guarantee algorithms in automatically sharing critical status information within and between organizations to enhance real-time situational awareness and prevent catastrophic power outages that would otherwise cascade across large control and reliability boundaries.

To Dad and Mom for Your Prayers and Support

Acknowledgments

I would like to express my sincere appreciation to my faculty advisor, Dr. Ken Hopkinson, for his patience, guidance, recommendations, and assistance throughout the course of this thesis effort. I would also like to thank my committee member and Deputy Department Head, Major Scott Graham, Section Leader Major Duane Harmon, and instructor and friend, Major Paul Williams, for their encouragement and advice during tough times.

I am indebted to Cedarville University student interns Ben Wiley and Gabe Greve for donating their expertise in writing and troubleshooting simulation code for the experiments, data analysis, and concept implementation. This work would not be complete without their time and talent.

Special thanks also go to Dr. Rick Raines, Mr. Tim Lacey, and Mrs. Stacey Johnston for the excellent instruction and administrative support provided throughout my coursework by the AFIT Center for Cyberspace Research.

I am also thankful for all of the outstanding AFIT instructors and staff who continue to operate, maintain, and mold this institution for the benefit of the Department of Defense, the Dayton community, and future commissioned officer, non-commissioned officer, and civilian students.

Gregory M. Coates

Table of Contents

Page

List of Figures

Figure Page

List of Tables

COLLABORATIVE, TRUST-BASED SECURITY MECHANISMS FOR A NATIONAL UTILITY INTRANET

I. Introduction

1.1 Background

The U.S. utility industry operates and maintains a significant portion of national critical infrastructure, supplying electrical generation and transmission, nuclear power production, water and waste management, oil and gas, and other critical services to consumers; seaports, airports, and other transportation systems; and numerous manufacturing plants, government offices, and businesses throughout the country.

Systems used to manage these complex networks, often with thousands of monitored nodes, have to be capable of reliable and accurate hard real-time or near real-time responses to fluctuations and emergency situations. Traditionally, each company purchased and installed its own proprietary systems and protocols from various vendors with no overall guiding interoperability standards adhered to by the community as a whole.

In system design, interoperability and security were often of a lower priority than efficiency and functionality. Many companies took comfort in the uniqueness and complexity of their systems as a means of security from would-be attackers. The need for interoperability was not critical for larger companies that could control the cradle-to-grave supply of services, from generation to transmission and distribution, to meet customer demands for an entire metropolitan area.

In the electrical power industry, deregulation has resulted in fragmenting many of

the previously held monopolies so that each privately-owned company specializes in only one function of the power grid (i.e. generation, transmission, distribution, etc.) with less wide-area visibility. It has also served to increase competition among these companies resulting in a greater need for management efficiencies and protection of company-sensitive data from unauthorized disclosure to competitors. These new trends point to a need for greater collaboration and situational awareness while providing strict network security in an environment prone to variable trust relationships.

1.2 Problem Statement

In recent years, the utility community has drifted away from the proprietary systems and protocols that once dominated the industry toward adoption of more open, networked communication standards for control and data acquisition, patterned after the efficiencies and lower cost of technologies seen in the Internet. The increased competition has made the lower cost and interoperability of IP-based, plug-and-play, Commercial-of-the Shelf (COTS) technologies attractive. These signs point to the eventual development of a Utility-specific Intranet, patterned after, yet unconnected to, the global Internet.

The Transmission Control Protocol (TCP), riding upon the Internet Protocol (IP) is the most common Internet standard for reliable information transfer with delivery confirmation. In November 1999, the TCP/IP framework was mandated by the International Electro-Technical Committee (IEC), a standards organization for the community, so that every modern computer and operating system integrated into the SCADA network will have a TCP/IP network stack.

Whether the legacy proprietary protocols were any less vulnerable to attack because of their obscurity is unlikely, however with the shift to IP-standards and common control system operating systems (e.g. Windows®, Linux®, Solaris®, UNIX®) it is certain that they are becoming more vulnerable to a wider audience of skilled and amateur attackers, familiar with the numerous IP-based exploits, techniques, and attack tools freely downloadable from the Internet [1].

Power engineers wanting to maintain strict processes and speed of operation claim that the vast majority of common IT security mechanisms will upset the delicate balance and cannot be applied to SCADA networks. IT personnel familiar with the security mechanisms used to defend more delay-tolerant office networks see these as the most secure measures for protecting computer systems against the potential threats from malicious code and online exploits for which they are all too familiar. Both parties are at odds as to the role, priority, and best implementation of security countermeasures.

1.3 Research Objectives, Questions, and Hypotheses

The purpose of this thesis research is to investigate the claims from both sides regarding employment of common, delay-inducing network security mechanisms to real-time SCADA and near real-time wide-area measurement systems (WAMS).

It is the hypothesis of this author that an acceptable, low-cost form of standard IT security measures may be applied to a Utility Intranet to secure communications from potential attackers, provide automated responses to identified attacks and suspicious activity, and increase situational awareness throughout the network within the real-time reaction timelines for SCADA operations.

1.4 Research Focus

The focus of this research has been on security for electrical power grid devices within a company. The concepts and results, however, are applicable to all levels of the Utility Intranet from company-level substation automation and control center operations to area-wide, regional, and even National Interconnection organizations (or any non-utility communications network for that matter).

1.5 Investigative Questions

Research was designed to answer the following questions:

1. What delay will be induced by each security component?
2. What accidental and malicious actions can the security mechanisms identify and mitigate?
3. Which mechanisms are the most appropriate for each possible operational configuration and each envisioned attack scenario?

1.6 Methodology

To begin with, it was assumed that future Utility Intranet SCADA networks will resemble IT network architecture. A collaborative **trust system** capability has been derived as a hybrid solution comprised of the most secure IT security mechanisms and standard IP protocols while focusing on the distinct requirements of the SCADA community.

To test the hypotheses specified in Section 1.3, a C++ implementation of a simplistic **trust system** was created that could evaluate and respond to incoming messages read in from a scenario file. The delays for processing at the **trust system** were

measured and summed with the delays for sender-to-receiver encryption, transmission, and propagation, to render the total per-packet and per-scenario latency values.

1.7 Assumptions and Limitations

The delays for router queuing and processing as well as encryption and decryption were estimated based on measurements presented in the literature. While these occurrences are responsible for the greatest amount of end-to-end delay, they do not detract from the **trust system** functionality and delay, which is in addition to transit delay that already exists in a SCADA network.

Detailed IEC 61850 message structure was not available for this thesis research. Message types for scenarios were selected only to illustrate the types of messages that might be present in a Utility Intranet but do not necessarily duplicate the IEC standards format. The messages chosen, however, are likely to be larger than SCADA messages for the same purpose because of full-character representation of some data vice integer representations and abbreviations likely with real-world optimizations to keep packets as small as possible. The messages defined for use in this thesis also contain the additional overhead of TCP, IP, larger IPV6 address, and encryption. The **trust system** results accurately represent the delay for **trust system** evaluation of real-world messages of the same general size.

1.8 Implications

This thesis research shows that, even with TCP/IP and UDP/IP communications, Internet Protocol Security (IPsec) encryption, *firewall rules*, *format check*, and *access control* functions, the recommended security schema can perform within near real-time

and at the high end of real-time response time constraints. It is therefore deduced that with further optimizations, the same schema can be improved to perform satisfactorily in many real-time scenarios.

1.9 Preview

Chapter 2 describes requirements of real-time SCADA network communications and the challenges facing those who attempt to secure them. It also presents the results of investigating on-going research in the field related to SCADA security. Finally it suggests the ways in which the **trust system** concept can solve existing security problems.

Chapter 3 describes the recommended **trust system** implementation in detail.

Chapter 4 demonstrates functionality of the **trust system** simulation and presents several realistic scenarios for attacks against a SCADA network. It also presents the calculated delay estimates for each scenario.

Chapter 5 concludes this thesis with recommendations for future research in **trust system** code optimization, refinement of IEC 61850 message structure, and bandwidth guarantees.

II. Literature Review

2.1. Chapter Overview

The purpose of this chapter is to present relevant background material and existing research as the foundation for investigative questions, assumptions, and direction guiding this thesis work.

2.2 Supervisory Control and Data Acquisition Overview

In North America, the term Supervisory Control and Data Acquisition (SCADA) is only applied to either a central system that monitors and controls a complete site or a system spread out over a long distance (i.e. on the order of kilometers or miles) for large-scale distributed measurement and control [2]. It is interesting to note that that throughout the rest of the world, even a single system that performs supervisory control and data acquisition functions, regardless of its size or geographical distribution, is referred to as a SCADA system, including those that only monitor without performing control functions [2].

There is a distinction between supervisory control and real-time (or process) control. Whereas, the real-time control system within a utility provides automated control of a process that is external to the SCADA system, the supervisory control function is implemented by a SCADA system that is overlaid onto the automated real-time control system. SCADA servers provide a human operator with alarms, status, performance data, and statistics of the real-time process. The SCADA system is typically not critical to controlling the industrial process in real-time, because the separate (or integrated) real-time automated control system is designed to respond quickly enough to

compensate for process changes within the time-constraints of the process. The SCADA system, however, allows the operator to poll for information or issue commands in the event of a failure in the automated process and must still meet stringent time constraints.

SCADA systems are found throughout the public utility industry and are integral to operation of our national critical infrastructure. SCADA systems are used to monitor and control geographically separated utility sites such as oil and gas pipelines and refineries, electrical power generation facilities and transmission grids, air traffic control towers, railways, maritime ports, water and waste management facilities, chemical plants, manufacturing facilities, and telephone and cell phone networks, including 911 emergency services [3, 4]. Due the mission critical nature of a large number of SCADA computer systems, attacks could result, directly or indirectly, in massive financial and sensitive data losses, destruction of facilities, or loss of life.

Scenarios such as massive power blackouts, oil refinery explosions, or waste mixed with drinking water due to SCADA system compromise, failure, or degradation have the potential to inflict significant damage to human life and critical infrastructure at local, regional, or national levels. If synchronized with a physical attack or the aftermath of a natural disaster, cyber attacks on SCADA systems could greatly escalate fatalities in a region already rendered unable to coordinate a timely response or ill-prepared to offer necessary shelter, clean water, and contamination control, perfect methods for inciting terror once again in America.

One can imagine the disastrous, synergistic effect of an explosion in a nuclear facility releasing nuclear contamination in the vicinity of a large population area immediately following a winter storm or summer hurricane that limits traversal of major

roadways and at the same time that the city's water system has been contaminated with sewage or bacteria and its electricity blacked out for well over a week. The combination of prolonged extreme (either sub-freezing or above 100 degree) temperatures, disease, and radioactivity would account for numerous deaths. The effects of Hurricane Katrina alone, in 2005, resulted in well over 1400 confirmed deaths, this amidst early warning and active emergency response efforts [5].

Meticulously planned and well-executed cyber attacks, whether conducted solely by remote network access or in conjunction with a malicious insider, is not an impossible scenario. What if similar actions were coordinated by terrorist agents to attack multiple cities within a region simultaneously?

2.3 The Threat to Utility Operations

2.3.1 Threat Sources.

Potential sources for cyber attacks and operational disruptions (whether accidental or intentional) on SCADA and other utility resources are listed in Table 1.

Table 1. Sources and Motivations for Utility Disruptions and Attack [6]

Source	Reason
Industrial sabotage or theft	Financial advantage in insider trading or competing vendor partnerships
Concentrated physical and cyber attack	Destruction, terror, or activism
Vendor compromise	Easier to target the supplier than the defended infrastructure itself [7]
Technical design error or environmental influence	Hardware or code; network design, installation and configuration; or interferences from other technologies in the environment
Natural disasters	Earthquakes, tornadoes, volcanoes, fire, thunderstorms, and snow storms
Operator error	Misjudgment, misconfiguration, or failure to remember operational details, resulting in dangerous and costly results

2.3.2 Specific Threats.

Theoretical scenarios abound; however, many businesses and engineers are incredulous or simply lack the resources or technical expertise to plan and maintain security upgrades that might eat into company profits or potentially affect performance. There is also an “if it ain’t broke, don’t fix it” mentality that can still be found regarding modifying or rethinking control system operations and cyber security implementations. Table 2 summarizes the potential threats to utilities from the sources listed in Table 1.

Table 2. Summary of Threats from Potential Sources of Attack or Disruption [1]

Source						Threat
Industrial Sabotage	Physical and Cyber Attack	Vendor Compromise	Design Error/ Environmental Influence	Natural Disaster	Operator Error	
X	X	X			X	Improper application of software patches
X	X	X	X	X	X	Plant shutdown for maintenance and start-up after maintenance (many harmful events occur as a result of plant maintenance shutdown and start-up)
X	X	X			X	Access lock-out (locked accounts, admin usernames and passwords changed)
X	X	X			X	Removal or misconfiguration of connectivity paths
X	X	X		X	X	Physical destruction of systems, resources, or infrastructure
X	X	X			X	Downloading malicious code (i.e. autonomous worms randomly searching for propagation paths, viruses, Trojan horses, etc.)
X	X	X			X	Denial of Service (DoS) and Distributed-denial-of-service (DDoS) attacks, such as those that overwhelm network bandwidth
X	X	X				Control message spoofing
X	X	X				Data acquisition message spoofing so everything looks normal to prevent response or bad to prompt dangerous responses
X	X	X				Password or message sniffing
X	X	X				Installation of backdoors to the network
X	X	X				Unauthorized data or code access, use, theft, modification, re-routing, and/or deletion
X	X	X				Unauthorized access to or modification of audit logs, firewall logs, and IDSs signatures/alerts
X	X	X				GPS timeserver corruption
X	X	X	X			Electromagnetic interference (EMI) and radio frequency interference (RFI)
			X			Noise on power lines
			X			Interdependence with other networks and support elements

Table 3 details potential avenues of attack or disruption in today's utility networks that require either elimination or defenses. It also lists the specific **trust system** functions that can be applied as a defense-in-depth strategy along these pathways.

Table 3. Potential Attack Routes Requiring Elimination or Defenses [1]

Attack Routes	**Trust System Mitigating Functions**
Internet connections	*Firewall rules*
Business or enterprise network connections	*Firewall rules*, network *Access Control Matrix (ACM)*
IT/Vendor connections to SCADA framework[6]	*Firewall rules*, network *ACM*
Connections to other networks	*Firewall rules*, network ACM
Compromised VPNs	Network logon enforcement, nodal and network-level *ACMs*, *Suspicious Event Handler*
Back-door connections through dial-up modems	Nodal and network-level *ACM*s, source tracking
Unsecured wireless connections discovered by war-driving laptop users	*ACM*, source tracking, encryption and authentication enforcement, network logon enforcement
Malformed IP packets, in which packet header information conflicts with actual packet data	Packet *format analysis*, *Suspicious Event Handler*
IP fragmentation attacks, where a small fragment is transmitted that forces some of the TCP header field into a second fragment	Packet *format analysis*, *Suspicious Event Handler*
Vulnerabilities in SNMP, which is used to gather network information and provide notification of network events	Packet *format analysis*, *Suspicious Event Handler*
Open computer ports, such as UDP and TCP ports that are unprotected or left open unnecessarily	*Firewall rules*
Weak authentication protocols in SCADA elements	Encryption and authentication enforcement
Maintenance hooks or trap doors, which are means to circumvent security controls during SCADA system development, testing, and maintenance	Nodal and network-level *ACMs*
E-mail transactions on control network	Traffic prioritization, antivirus scans, DoS detection and blocking, firewall rules
Buffer overflow attacks on SCADA control servers, which are accessed by PLCs and SCADA HMIs	Packet *format analysis*, *Suspicious Event Handler*
Leased, private telephone lines	Nodal *ACM*
GPS conditioned timeserver	*Firewall rules*, packet *format analysis***, trust systems'** collaboration synchronized off of network-level **trust system** internal clock as back-up time-stamping source

2.3.3 Open Source Intelligence.

Even for legacy control systems with proprietary hardware and software, the knowledge needed to cause a widespread power blackout is readily available on the Internet, where SCADA vendor websites post manuals, downloadable software, and source code for major applications [8]. Vendor sites often list well-known customers with detailed case studies of how these customers have implemented their systems and which products they have. In fact, it has been found that "over 90% of major SCADA and automation vendors have all of their technical manuals and specifications available on-line to the general public" [8].

Many corporate websites list their training materials and operating manuals, presentations about vulnerabilities and what they think hackers could do, firewall policies, network diagrams, spreadsheets listing accounts and DNS or IP addresses, backup and sample configuration files for the control systems, protocol documentation, as well as documentation of simple penetration testing techniques, examples, and hacker scripts [8].

2.3.4 Real-world Incidents.

There have been several well-known, real-world incidents affecting SCADA systems, and very likely many others never publicized, that clearly illustrate the vulnerability of our critical infrastructure [7].

1. During the Cold War, the US provided Trojan firmware to the Soviet Union, causing a pipeline to explode in one of the world's largest non-nuclear explosions [7]. SCADA software, hardware, or firmware can be maliciously produced and sold to US

companies by foreign or domestic entities with the intent to destroy the power supply to a region.

2. In 1992, a former Chevron employee disabled its emergency alert system in 22 states, which wasn't discovered until an emergency happened that needed alerting [7].

3. In 1997, a teenager broke into the NYNEX telephone network and cut off Massachusetts' Worcester Airport for six hours, affecting air and ground communications [7].

4. In 2000, former employee Vitek Boden, exploited a wireless link to the SCADA system for the Queensland, Australia, Maroochy Shire sewage control system, releasing a million liters of sewage into the coastal waterways over a period of four months [7].

5. Also in 2000, the Russian government announced that hackers, acting together with a company insider, succeeded in bypassing Gazprom security measures and gained control of the system regulating gas flows for the world's largest natural gas pipeline network [7].

6. Some computers and manuals seized from Al Qaeda terrorist safe houses in Afghanistan contained SCADA information regarding dams and related structures, but no implementation plan [7]. Terrorists have been searching for critical infrastructure targets-of-opportunity for many years.

7. In 2001, hackers broke into CAL-ISO, California's primary electric power grid operator, and weren't discovered for 17 days [3:75].

8. In 2003, the Ohio Davis-Besse nuclear power plant safety monitoring system was offline for five hours due to the Slammer Worm [7].

9. In 2005, Hurricane Katrina disrupted a few refineries in the southern coast of the US, affecting gasoline prices world-wide [7]. Shutdown by cyber attack has the potential to affect supplies of gasoline, electricity, or water and corresponding global stock prices.

2.4 Changes in the SCADA Environment

SCADA systems evolved from proprietary hardware and software platforms used in the 1960s to acquire data from and control real-time systems. The networks and protocols used in SCADA systems were also proprietary and customized to meet the specific needs of the industrial world [1].

There was no Internet or World Wide Web (WWW) at the time, and the SCADA systems were self-contained, so they were generally considered safe against malicious intrusions from the outside, but have always been vulnerable to threats from the inside. Even when the Internet emerged and SCADA systems began to incorporate standard hardware and software platforms that had known vulnerabilities, the mentality of most SCADA operators and managers remained the same. The SCADA community believed that external hackers were not interested in their applications and probably did not know much about the existence and configuration of SCADA systems. Even in the 1980s and early 1990s, most documented SCADA security incidents were either initiated by disgruntled employees or were the result of accidents. SCADA systems were not even considered IT systems, and were assumed to be relatively less vulnerable to IT-type cyber attacks. Even to this day, many SCADA systems are perceived as either nearly invulnerable to cyber attacks or uninteresting to potential hackers [1].

Within the last few years, several changes have begun to impact SCADA system operation, design, communications, and security—increasing the risks, vulnerabilities, and the complexity of defining network security measures for this unique environment.

The restructuring of the utility industry has increased competition while driving the need for more efficient operations and better coordination among utility companies. Two major elements are involved. The first element of restructuring is regulatory. Using the electric power industry as an example, power grids, historically, were centrally controlled and operated. Changes in the regulatory structure now encourage independent ownership of generators and favor the emergence of competitive mechanisms by which organizations can enter into bilateral or multilateral power generation contracts. The second element in restructuring is a consequence of the first, involving large-scale operation of the grid. In the past, this was a centralized task. In the restructured climate, a number of competing power producers must coordinate their actions through a set of independent service operators (ISO). The process of restructuring has occurred incrementally. In its earliest stages, large monopoly-style utilities that might have owned beginning-to-end power production and delivery processes were broken into smaller companies with typically specialized roles in only generation, transmission, or distribution. At the same time, there has been a slow but steady growth in the numbers of long-distance contracts.

Stress on the electric power grid continues to rise in the current deregulated environment as the demand for power grows with increasing population and infusion of technology into businesses and homes. With increasing demands world-wide for electric power, the grid is being operated closer and closer to its limits. Despite this reality, the

generation and transmission capacity of the grid has not been widely upgraded to accommodate greater output and flows. Deregulation has served to exacerbate this situation.

The deregulated utilities have been forced to split into separate companies, each devoted to different aspects of the power grid, in place of the vertically integrated structure that existed in the past. Generation, distribution, and transmission systems all have separate owners under this new arrangement. The transmission system, in particular, is typically owned and controlled now by the ISO in each region of the grid. This operating arrangement is problematic in the sense that none of these entities has an incentive to upgrade the transmission infrastructure. Ostensibly, this is the responsibility of the ISO, but they lack an economic incentive for adding new transmission lines in the same way that a generation company has a clear motive to add new power plants to the grid.

The new structure of the power grid has led to increased competition between utilities that might have cooperated with one another in the past. This complicates the proper detection and response to faults that occur in the electric power grid since information that might have been shared in the past is seen as proprietary for economic reasons [9].

There is also an emerging trend in many organizations comprising SCADA and conventional IT systems toward consolidating overlapping activities. For example, control engineering might be absorbed or closely integrated with the corporate IT department. In addition, integrating SCADA data collection and monitoring with

corporate financial and customer data provides management with an increased ability to run the organization more efficiently and effectively [1].

This drive for efficiency and cost savings has led SCADA system and architecture designers to begin patterning utility communications after the rapid changes occurring in the larger Information Technology (IT) and networking industry by becoming more open and at the same time more interconnected. For economic and efficiency reasons, the primitive legacy systems are being upgraded using Commercial-Off-The-Shelf (COTS) hardware and software, and are being migrated from isolated in-plant networks using proprietary hardware and software to standard data formats and network protocols, particularly Transport Control Protocol (TCP) for end-to-end control. This trend is motivated by cost savings achieved by consolidating disparate platforms, networks, software, and maintenance tools [10]. The downside of this transition has been to expose SCADA operating systems to the same vulnerabilities and threats that plague Windows and Linux-based PCs and their associated networks connected to the Internet.

2.5 A Future Utility Intranet

Most researchers anticipate that an Internet-like Utility Intranet (also referred to as a Utilities Network or Superstructure), dedicated to the power grid and mostly isolated from the public Internet, will emerge in the coming decade, with TCP likely to be the primary transport protocol [10]. Another reason SCADA is likely to migrate to a Utility Intranet is due to the higher polling rates that would be possible with the increased bandwidth available in the new communications infrastructure [9]. Given the stricter response thresholds of SCADA systems, this presents an extreme challenge in providing

for their security in an environment where connections to the Internet (whether known or not) are almost certain to exist, providing a tempting avenue for attempted cyber attacks.

The move to a universal protocol among all utilities is slow at best but will probably be dominated by the use of Ethernet as a common carrier for data because of the ease of use and low cost of Ethernet LAN systems. Many newly developed SCADA applications and many future variants will use various protocols but ride over IP [11].

The power industry is turning towards next-generation communications systems in order to meet the increased demands that are being placed on the electric power grid. These standards point toward the future adoption of a private Utility Intranet based on Internet technology to improve the efficiency and reliability of the power grid. The Utility Intranet is likely to begin as an effort to improve upon the monitoring, protection, and control of individual utilities and, with communication standards, will lead to the interconnection of the utilities' data networks in the same way that the electric power grid has become integrated over time. The introduction of a Utility Intranet has many potential benefits such as increased information sharing, greater protection and control of the grid, and the enhanced ability to share power in complex situations such as bilateral load following. However, great care must be taken to ensure that network capacities, communication protocols, security, and quality of service (QoS) requirements are appropriately managed to ensure that the Utility Intranet will be able to meet the demands that are placed on it by increasing consumption rates [9].

Traditionally, SCADA systems and corporate IT systems have focused on very different information assurance priorities. Whereas IT system priorities are confidentiality, authentication, integrity, availability, and non-repudiation, SCADA

systems emphasize reliability, real-time response, tolerance of emergency situations, personnel safety, product quality, and plant safety, usually to the exclusion of any security mechanism that might hinder these.

Now, with the compatibility and overlap of the two networks, both SCADA and corporate IT will have to develop complementary security models. Current issues such as dial-in modems connected to one system compromising the other, the possibility of unprotected, rogue corporate Internet connections rexposing the SCADA network, the real-time deterministic requirements of SCADA systems, and 24/7 operations require deconfliction of the disparate cultures of SCADA and IT [1]. A good example of this sort of problem is the routinely scheduled downtime for IT organizations to upgrade, patch vulnerabilities, perform backups, and so on [1]. Such downtime cannot be tolerated for most SCADA systems [1].

Throughout this transition to a Utility Intranet, SCADA system networks must be well defended yet maintain the same level of service required by their customers [3]. Blindly layering standard IT security mechanisms on top of SCADA networks will not work without accounting for their unique requirements and time constraints; therefore, it is important to first understand current and future SCADA architectures and operational philosophies.

2.6 Substation Integration and Automation

The electrical power substation integration and automation system is the combination of equipment and communications infrastructure by which raw data measurements and system health status updates are processed and transmitted from

remote substation equipment to SCADA systems and historical databases for human interaction. It is also the means by which commands or polls for information are communicated in the reverse direction.

Substation integration involves integrating protection, control, and data acquisition functions into a minimal number of platforms to reduce capital and operating costs, reduce panel and control room space, and eliminate redundant equipment and databases [7].

Substation automation (SA) involves the deployment of substation and feeder operating functions and applications ranging from SCADA and alarm processing to integrated volt/Var control in order to optimize the management of capital assets and enhance operation and maintenance efficiencies with minimal human intervention [7].

Substation integration and automation can be broken down into five levels. The lowest level is the power system equipment, such as transformers and circuit breakers. The middle three levels are Intelligent Electronic Device (IED) implementation, IED integration, and substation automation applications. The focus today is on the integration of the IEDs. Once this is done, the focus will shift to what automation applications should run at the substation level [7].

The highest level of substation integration and automation is the utility enterprise. There are three primary functional data paths from the substation to the utility enterprise:

1. Operational data to the SCADA system
2. Non-operational data to the data warehouse
3. Remote access to the IED

2.7 Operational Data to the SCADA System

2.7.1 SCADA System Components.

Historically substation field devices had no standardized way to present information to an operator. They were distributed across a plant, making it difficult to gather data from all of them manually, therefore, the purpose of the SCADA system was to gather information from the field devices and other controllers, then present it to the human operator in easy to understand graphics.

The most common substation automation data path is conveying this operational data from the substation to the utility's SCADA system every 2 to 4 s. Operational data (also called SCADA data) includes instantaneous values of power system analog and status points such as volts, amps, MW, MVAR, circuit breaker status, and switch position. This data is time critical for the utility's dispatchers to monitor and control the power system (e.g., opening circuit breakers, changing tap settings, equipment failure indication, etc.). The operational data path to the SCADA system uses the communication protocol presently supported by the SCADA system [7]. The SCADA system itself has the following four components:

1. Multiple field devices (i.e. power equipment, IEDs, RTUs, and PLCs)
2. Substation data concentrator
3. SCADA master station, HMI, and databases
4. Communications infrastructure

The first two components are within the substations themselves. The third component interfaces to the company control center, engineering center, and corporate

offices. The communications infrastructure is the interconnecting transport mechanism that ties the SCADA system together

2.7.2 Traditional Field Devices.

The bulk of supervisory control and data acquisition is performed automatically at the substation level [2]. For years, Programmable Logic Controllers (PLCs) and Remote Terminal Units (RTUs) carried the load.

The first PLCs used simple software to duplicate the functionality of a rack of interconnected relays [12]. In the last few years higher end models have been supplemented with analog inputs and outputs (I/O). The low end PLCs are not even addressable (i.e. they cannot be used as a slave to another device or as a component in a control system) [12].

PLCs scan their I/O by electrically reading each I/O point. In a system with lots of I/O points it can take some time to completely scan all the points. PLCs can be used as stand-alone devices but they are difficult to configure, requiring ladder logic programming [12]. When a substation contains lots of I/O that must be monitored or controlled, PLCs are not the best choice, because they are not usable as the master controller in a control system, neither are they appropriate for use as protocol converters or for controlling other IEDs [12].

RTUs are more sophisticated than PLCs and have the intelligence needed to control a process (or multiple processes) without intervention from a more intelligent controller or master [12]. RTUs offer interrupt driven digital inputs, time stamped sequence of events, data logging, intelligent communications, multitasking sequential

control, process identification control, alarm logging, modular construction, and easier programming than PLCs [12]. Additionally, an RTU can serve both as the master controller or a slave controller--in fact, it can be used as both a slave and master simultaneously in a "vertically deployed control system" [12]. An RTU can be used in conjunction with IEDs as a protocol converter or controller for the IEDs [12].

Because of today's advancements in microprocessor technology, a single IED is capable of performing numerous protection, control, auto-reclose, self-monitoring, and communication functions that used to require separate RTU and PLC devices [13].

2.7.3 Intelligent Electronic Device (IED) Implementation and Integration.

IEDs are a key component of substation integration and automation technology. An IED is any device that incorporates one or more processors with the capability to receive or send data/control from or to an external source (e.g., electronic multifunction meters, digital relays, controllers, and regulators). Their primary function is to process the incoming analog signals, convert the values directly to a digital form, and forward the information via their communications link to a substation automation (SA) controller (also known as a data concentrator). IED technologies help utilities improve reliability, gain operational efficiencies, and enable asset management programs including predictive maintenance, life extensions and improved planning. IEDs can also issue control commands, such as tripping circuit breakers to maintain a steady state if they sense anomalies or dangerous changes in voltage, current, or frequency. Many IEDs are now capable of peer-to-peer communications for high-speed protection functions in which any node can initiate sessions and is able to poll or answer polls from other devices [7:7-6].

Nearly all electric utilities are implementing IEDs in their substations. New substations will typically have many IEDs for different functions, and the vast majority of operational data for the SCADA system will come from these IEDs, with a smaller amount of direct (i.e. hardwired) input acquired by PLCs.

Typically, there are no conventional RTUs in new substations. Instead, the RTU functionality is addressed with a mix of IEDs and PLCs using digital communications. Older substations, that still have a conventional RTU installed, can integrate the RTU with IEDs, integrate the RTU as just another IED, or retire the RTU altogether and use a combination of IEDs and PLCs as with new substations [7].

IEDs being implemented in substations today contain valuable information, both operational and non-operational, needed by many user groups within the utility. Each device has some internal memory to store data such as analog values, status changes, sequence of events, and power quality, usually in a first-in, first-out (FIFO) queue, and is integrated with digital two-way communications [7].

2.7.4 Substation Data Concentrator.

The data concentrator polls each IED or PLC for updates according to the utility's SCADA data collection rates (e.g. status points every 2 sec, tie line and generator analogs every 2 sec, and remaining analog values every 2 to 10 sec). Current systems must perform protocol translation, converting all of the IED protocols from the various IED suppliers. Some experts believe that, "even with the protocol standardization efforts going on in the industry, there will always be legacy protocols that will require protocol translation" [7:7-5].

The substation controller collates the data received from the IEDs, performs logic calculations, time synchronization, filtering, and pre-processing or reformatting of the substation data to meet presentation requirements of the master control station, operator workstation clients, or other intended data receivers [14]. The substation controller will usually have a PC-based substation host processor, or substation HMI, that supports an archival relational database, GUI, and Windows® Office-like applications. It stores all analog and status information available for the substation that is required for both operational and non-operational purposes (e.g. fault-event logs, oscillography, etc.).

The substation host processor and substation controller are optional--either, none, or both may be present [14]. A substation controller may be PC-based (in which case the substation controller itself would be the host processor). It could also be a PLC, data concentrator, or hybrid combination of any of these options [14].

In a truly flat architecture, where substation-level data collation and re-formatting functions are not required, the IEDs may communicate directly with the remote SCADA operator clients. The remote clients can then conduct the same data selection tasks by polling, requesting, or browsing only the specific data required from a particular IED [14].

Small, secondary substations may have only a data concentrator with no host processor for user interface or historical data collection. In this case, IED data is sent to a larger primary substation, which has a complete substation integration and automation system, to combine the information and interface with the SCADA system.

It is expected that future technological improvements in substation devices will continue to increase the decentralized gathering/processing of data and alarm

handling/filtering at the field device (rather than the master control station), direct IED communications with multiple master stations and databases (reducing the need for data concentrators), and peer-to-peer status sensing/reaction by neighboring field devices.

2.7.5 SCADA Master Control Station and Human Machine Interface.

The data concentrator forwards all data required for operational purposes to the SCADA system. The operational data is then compiled and formatted in such a way that a control room operator can make appropriate supervisory decisions that may be required to adjust or over-ride normal PLC or IED controls.

A Human-Machine Interface (HMI) computer presents the process data to a human operator and is the standardized means through which the human operator monitors, controls, and interacts with the industrial process and its multiple remote substation field devices. A typical SCADA operator screen shot is depicted in Figure 1.

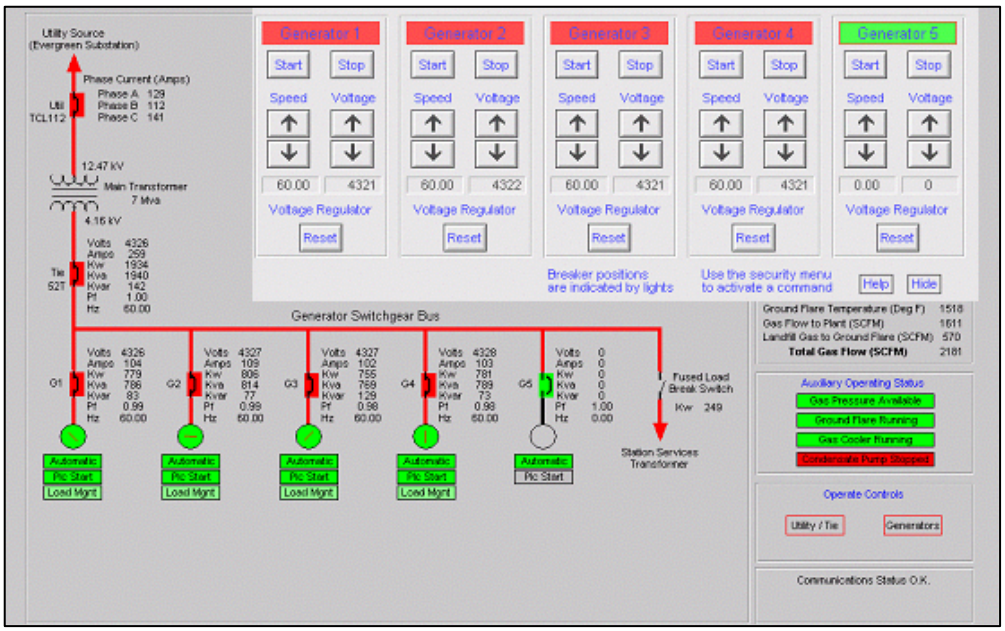


Figure 1. Example SCADA HMI Control Screen [15]

A master control station (or simply master station) is compromised of the supervisory servers and software responsible for communicating with the field devices in substations and then to the HMI software running on client workstations in the control center. In smaller SCADA systems, the master control station can be composed of a single PC. In larger SCADA systems, the master control station may include multiple servers, distributed software applications, and geographically separated disaster recovery sites. Today, most major operating systems (e.g. Windows®, Linux®, Solaris®, UNIX®, etc.) are used for both master control station servers and HMI workstations [2].

SCADA host control functions are almost always restricted to basic site over-ride or supervisory-level capability. For example, an IED may govern the generation rate of a generator in a power plant, but the SCADA system may allow an operator to change the control set point for the current and effective load on the generator, and will allow any alarm conditions such as extreme frequency or voltage fluctuations to be recorded and displayed. While the feedback control loop is closed through the IED, the SCADA system monitors the overall performance of that loop.

Use of newer IEDs and intelligent PLCs, capable of autonomously executing simple logic processes, is increasing [2]. Instead of relying on operator intervention, or master control station automation, IEDs may now be required to operate almost entirely on their own to react to emergencies and perform safety-related tasks [2].

2.7.6 SCADA Databases.

SCADA systems typically implement a distributed operational database, commonly referred to as a tag database, which contains data elements called tags (or

points) [2]. A point represents a single input or output value monitored or controlled by the SCADA system [2]. Point values are normally stored as value-timestamp combinations (i.e. the value and the timestamp when the value was recorded or calculated) [2]. A series of value-timestamp combinations is the history of that point [2]. It's also common to store additional metadata with tags such as path-to-field-device and register, design time comments, and alarm information [2]. Data may also be correlated by a Historian, often built on a COTS database management system, to allow historical trending and other analytical work [2].

2.7.7 Communications Infrastructure and Transmission Media.

A system to meet hard real-time or near real-time detection, decision, and reaction times is strongly dependent on a robust, reliable communications architecture. The internal substation integration and automation infrastructure and the connections between utility organizations will become increasingly critical data highways for situational awareness and response, requiring attention to security, reliability, and, most of all, low latency. Specific intra-company design criteria include high bandwidth, low bit error rate, multi-point access, and some degree of redundancy [16].

Electrical utilities have employed a wide range of transmission means to meet short and long-range communication needs, driven more by cost-efficiency than security. SCADA systems traditionally relied upon radio or direct serial and modem connections for communications with substations. Now there is a growing trend in the use of spread-spectrum satellite and inherently non-secure wireless technologies such as Wi-Fi/Wi-MAX, General Packet Radio Service, Enhanced Data rates for Global Evolution, CDMA

Data Service, and home-grown 900MHz radio solutions. Power line carrier, microwave, and fiber optics systems are the most popular technologies for wide area protection [16].

Optical fiber is an ideal solution for Utility Intranet communications. Thousands of miles of optical fiber have already been installed as part of the power line facilities [16]. Since optical fiber is immune to electromagnetic and radio frequency interference and crosstalk present in power plants, substations, and powerline transmission paths, fiber-based LANs reduce error rates from a few errors per minute (with copper) to only a few errors per month, even at data rates above one gigabit per second (Gbps) [17]. Optical fiber's low attenuation and high bandwidth also provide the ability to transmit signals over long distances.

Wavelength Division Multiplexing (WDM) systems present a new alternative for optical fiber network connectivity with much greater advantages in cost, flexibility, and scalability. Since light waves of different lengths do not interfere with one another, multiple wavelength signals can be transmitted through the same optical fiber without error [17]. By allowing multiple high-speed communications applications to share the same fiber simultaneously, WDM opens the door to optical fiber's tremendous bandwidth capability allowing transmission and propagation speeds of more than one Terabit per second [17]. WDM systems create completely independent, fully transparent paths over each fiber[17]. This allows the combination of multiple application protocols over the same fiber without any issues of latency, speed, proprietorship, or software setup [17]. A multi-channel WDM link behaves as multiple virtual fiber pairs, letting utilities mix and reconfigure protocols as needed [17].

2.8 Non-Operational Data to the Corporate Data Warehouse

The most challenging data path is conveying the non-operational data to the utility's data warehouse. The non-operational data path to the data warehouse conveys the IED non-operational data from the substation automation system to the data warehouse, either being pulled by a data warehouse application from the SA system or being pushed from the SA system to the data warehouse based on an event trigger or time. Non-operational data consists of files and waveforms such as event summaries, oscillographic event reports, or sequential events records, in addition to SCADA-like points (e.g., status and analog points) that have a logical state or a numerical value. This non-operational data is not needed by the SCADA dispatchers to monitor and control the power system [7].

The trend in IP-capable utility operations is for the data concentrator to send both operational and non-operational data through a firewall, separating the operational and corporate LANS, to the corporate Intranet, to be maintained in a corporate data warehouse for common, client-server or mainframe access by various company user groups such as operations, planning, engineering, SCADA, protection, distribution automation, metering, substation maintenance, and IT personnel. This setup provides multi-user simultaneous access, throughout the organization, for up-to-date information.

2.9 Remote IED Access

The remote access path to the substation traditionally uses either a dial-in telephone connection or a network connection. There are interfaces to substation IEDs to acquire data, determine the operating status of each IED, support all communication

protocols used by the IEDs, and support standard protocols being developed. There may be an interface to the Energy Management System (EMS) that allows system operators to monitor and control each substation and the EMS to receive data from the substation integration and automation system at different periodicities. There may be an interface to the Distribution Management System (DMS) with the same capabilities as the EMS interface [7].

2.10 Time Constraints

Timeliness of message delivery is critical to the electrical grid. Traditional short circuit protection systems measure local signals and respond in 4-40ms to disturbances in the local area. For the purposes of this paper, 4ms is considered as a benchmark for worst-case response time requirements in local protection.

Wide Area Protection and Control (WAPaC) systems gather information from multiple locations on the system and issue wide area controls as necessary to respond to disturbances in a somewhat longer time frame. Depending upon the distance from the origin of the disturbance and type of disturbance, there may be a time lag on the order of seconds before the disturbance reaches systems that are hundreds of miles away. If high-speed communication channels are available for signaling, it would be possible to get an early warning of an impending disturbance in time to set some supervisory control strategies in place. Today's wide area communication topologies, are capable of delivering messages from one area of a power system to multiple nodes on the system in as little as 6 ms. Assuming a decision calculation time of 50 ms, a disturbance on a system could be detected and a corrective response delivered in less than 200 ms [16].

Even assuming as much as 200 milliseconds delay in transmission and processing, enough early warning would be available in most cases so that supervisory control of critical functions could be implemented. If, in addition, the nature of the disturbance was known, each key control and protection system could be switched to a defensive posture appropriate for the particular problem [16]. Table 4 summarizes typical time constraint thresholds that must be met for SCADA and utility protection responses.

Table 4. Time Constraints for Electric Utility Operations

Systems	Situation	Response Time
Substation IEDs; Primary short circuit protection and control	Routine power equipment signal measurement	Every 2-4ms
	Local-area disturbance [6]	< 4ms from event detection to sending notification [14]
		4 - 40 ms automatic response time
Backup protection and control; Wide-area protection and control (WAPaC)	Transient voltage instability	Often ≤ 180 ms to convey 14+ trip signals to disconnect generators at the top generating station [16]
	Frequency instability, must respond faster than generator governors to trip generators instantaneously	Could require < 300ms response time (by load shedding) for high rates of frequency decay; requires detection within 100ms to allow operator response in 150 to 300ms [16]
	Dynamic instability	A few seconds
	Poorly damped or un-damped oscillations	Several seconds
	Voltage instability	Up to a few minutes
	Thermal overload	Several minutes for severe overloads, rarely less than a few seconds for minor occurrences [16]
SCADA	Emergency event notification	< 6 ms
	Routine transactions	< 540 ms [3]
	Routine HMI status polling from substation field devices	Every 2 secs

2.11 SCADA Protocols and Standards

2.11.1 Legacy Proprietary Protocols.

SCADA protocols have always been designed to be very compact and efficient, however, RTUs and other automatic controller devices were being developed before the advent of industry-wide standards for interoperability. As a result, manufacturers invented a multitude of SCADA and control system protocols. Especially among the larger vendors, there was the incentive to create their own proprietary protocol to "lock in" their customer base. It wasn't until the late 1990's that manufacturers began to shift toward more open communications like Modicon MODBUS over RS-485. By 2000 most vendors offered completely open interfacing such as Modicon MODBUS over TCP/IP.

2.11.2 Transition to Open Protocols.

The development of Distributed Network Protocol (DNP) 3 was a comprehensive effort to achieve open, standards-based interoperability between substation computers, RTUs, IEDs, and master stations (except inter-master-station communications) for the electric utility industry. It is still used within US utilities such as water companies and electricity suppliers for the exchange of data and control instructions between master control stations and substation controllers [1].

In the early 1990s, the Electric Power Research Institute (EPRI) decided that an effort was needed to define a more robust standard than DNP3 to serve the SCADA needs of the electric utilities. The result was the Utility Communications Architecture (UCA).

In 1999, UCA 2.0 migrated to International Electrotechnical Commission (IEC) Standard IEC 61850 for Substation Automation. Both are networkable and object-oriented, which makes it possible for a device to describe its attributes when asked [18]. This capability allows self-discovery and pick-list configuration of SCADA systems [18]. IEC 61850 is part of the Common Information Model (CIM) developed by IEC Technical Committee (TC) 57 that also includes the utility communications standards listed in Table 5 and visually depicted in Figure 2 [1].

Table 5. Sample of Standards Comprising the Common Information Model

IEC Standard	Title
IEC 61970	Power Systems and Programming Interfaces for Integrating Utility Applications
IEC 61968	Distribution Equipment and Processes
IEC 61334	Distribution Automation Using Distribution Line Carrier Systems
IEC 60870-5	Distribution
IEC 60870-5-103	Telecontrol Equipment and Systems: Transmission Protocols - Companion Standard for the Informative Interface of Protection Equipment
IEC 60870-6	Transmission
IEC 60870-6-101/104	Telecontrol Protocols Compatible with ISO and ITU-T Recommendations
IEC 60870-6-TASE.2	Inter-Control Center Communications Protocol (ICCP)
IEC 61850	Communication Networks and Systems in Substations
IEC 60834	Performance and Testing of Teleprotection Equipment of Power Systems

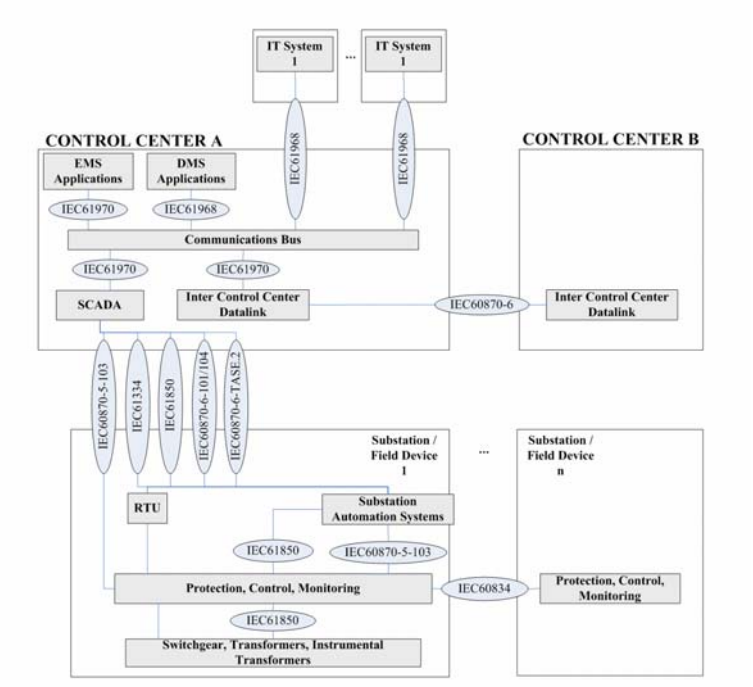


Figure 2. TC57 Standards Used in Substation and Control Center Communications [19]

2.11.3 IEC 61850, Communication Networks and Systems in Substations.

The IEC 61850 standard defines common data formats and communication methodologies to allow devices to communicate across IP-based networks [9]. IEC 61850 is a layered architecture that separates the functionality required for electric utility applications from the lower-level networking tasks [1].

IEC 61850 defines a total of 13 different *Logical Groupings* of data that could originate in the substation (see Figure 3) [14].

Logical Node Groups	Group Designator
System Logical Nodes	L
Protection functions	P
Protection related functions	R
Supervisory control	C
Generic References	G
Interfacing and Archiving	I
Automatic Control	A
Metering and Measurement	M
Switchgear	X
Instrument Transformer	T
Power Transformer	Y
Further power system equipment	Z
Sensors	S

Figure 3. IEC 61850 Logical Node Groups and Group Designators [14]

Each of the *Logical Groups* are further subdivided into *Logical Nodes* (86 total), each composed of data that represent some application-specific meaning and intended to provide separate sub-categories of data [14]. Figure 4 provides an example of *Logical Node Groups*.

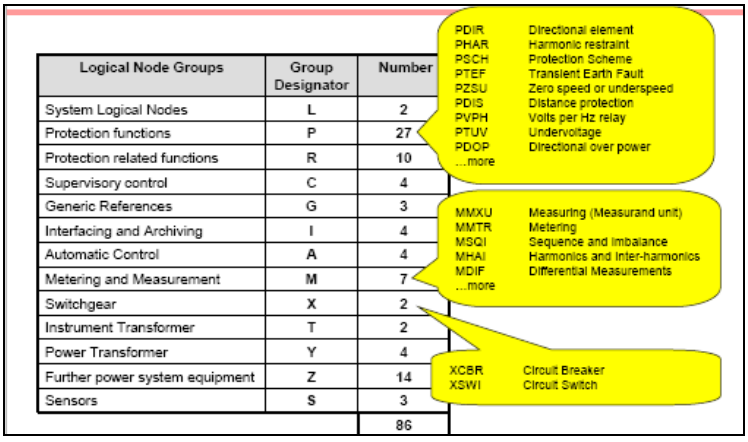


Figure 4. IEC 61850 Logical Nodes [14]

Logical Nodes are comprised of *Data Classes* (355 total), which are divided among seven categories as detailed in Figure 5 [14].

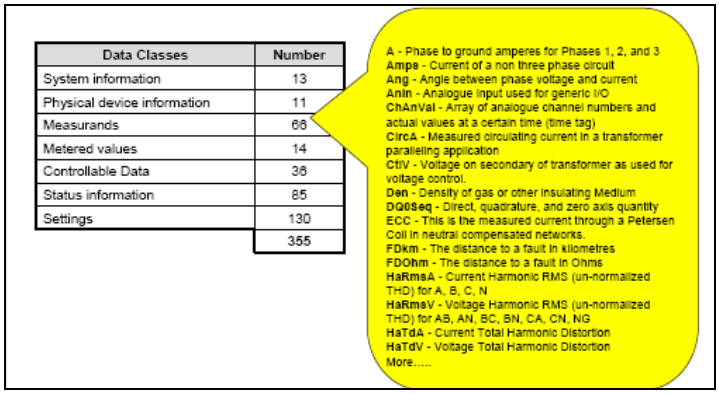


Figure 5. IEC 61850 Data Class Categories [14]

The container is the *Physical Device* (network address), and contains one or more *Logical Devices*. Each *Logical Device* contains one or more *Logical Nodes*. Each *Logical Node* then contains a pre-defined set *of Data Classes*, each of which contains data [14]. Figure 6, depicts the multiple functions supported by IED-1.

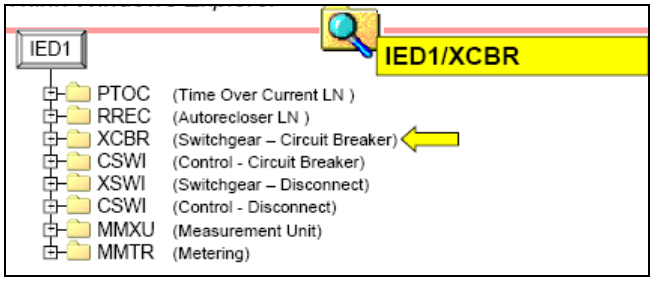


Figure 6. Example of Browsing IED-1's Functions [14]

Because IEC 61850 supports self-description, an operator can see what data a device has by communicating with it and browsing its contents. Control center personnel, via the HMI, browse the devices directly and subscribe to the data they require – there is no need for an intermediate cross-reference of data. Figure 7 depicts the ability to drill down through folders on the IED for data values.

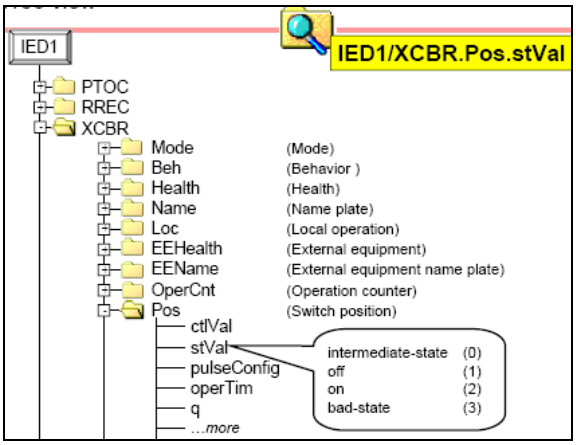


Figure 7. Example of Browsing IED-1 for Data [14]

2.11.4 GOOSE and GSSE.

Generic Object Orientated System-wide (Substation in some literature) Events (GOOSE) and Generic Sub-Station Event (GSSE) define a high-speed, Ethernet-based, object-model protocol to be used for high-speed multi-device communications between protection devices. The GOOSE and GSSE services are used for fast multicast communication between a publisher and one or more subscribers. The abstract services are used for such operations such as protection event notification. Upon detecting an event, the IED(s) use a multi-cast transmission to notify those devices that have

registered to receive the data [6]. Collisions are quite possible in an Ethernet network in this scenario, so the GOOSE messages are re-transmitted multiple times by each IED [14]. "IEC 61850 supports both client-server and peer-to-peer communications. "It is the peer-to-peer communications ability that is used to exchange GOOSE messages between IEDs" [14]. GOOSE requires peer-to-peer communications between relays, quite possibly from different vendors. Configuring the requisite publisher/subscriber model could be a very daunting task, especially when each vendor will have their own proprietary configuration program [14]. Because of this, IED vendors are required to provide a descriptor file for their IEDs in Extensible Markup Language (XML) format. The eventual goal is for the devices to transmit their configuration in XML upon request. The use of XML and the substation configuration language defined by IEC 61850 will provide visibility into the data available from any vendor [14].

There is still great room for improvement. IED suppliers acknowledge that their expertise is in the IED itself – not in two-way communications capability, the communications protocol, or added IED functionality from a remote user. Though the industry has made some effort to add communications capability to the IEDs, each IED supplier has been concerned that any increased functionality would compromise performance and drive the IED cost so high that no utility would buy it. Therefore, the industry has vowed make competitive cost and high performance as priorities over network security enhancements as standardization is incorporated into the IED [18].

Figure 8 illustrates GOOSE, GSSE and other substation-level communications that will ride over Ethernet and Internet Protocol.

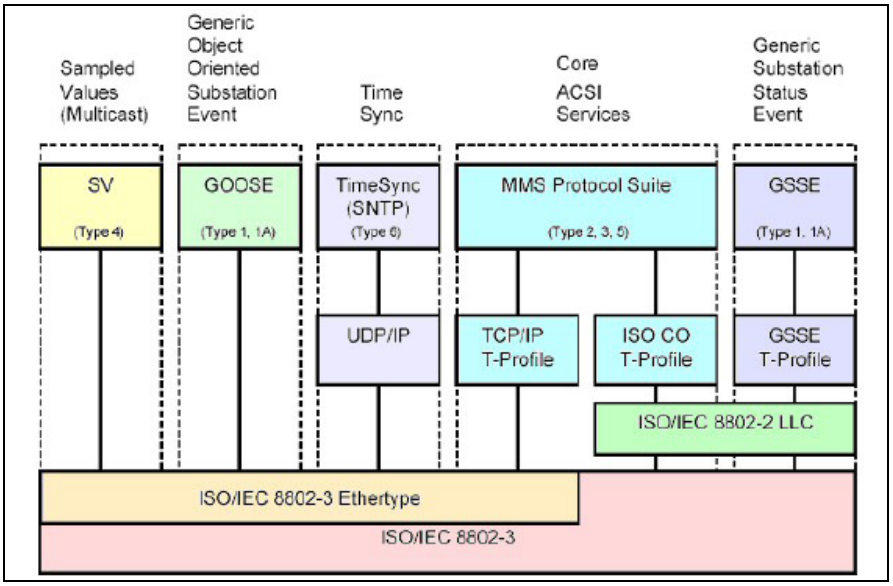


Figure 8. Ethernet as the Foundation for All Future Substation Communications [19]

2.11.5 Problems with TCP/IP for Time-constrained Traffic.

TCP as a transport protocol has several undesirable properties that make its deployment problematic in situations and applications that have time dependencies.

TCP's tightly integrated congestion control mechanism, designed to work well when transmitting large quantities of data, can interfere with time-critical transmissions. TCP slow-start and congestion control will induce instability during periods of peak message traffic, such as emergency situations, precisely when guaranteed delivery of urgent information is required [10]. Unless a nonstandard TCP implementation is selected or bandwidth guarantees are provided, standard TCP functionality will be intolerable for real-time traffic [10].

TCP is a primarily point-to-point protocol that is inefficient in many types of monitoring applications where the same message needs to be shared with multiple other nodes [10].

The large overhead associated with TCP headers and the three packet handshake, required to establish a connection, creates significant delay. The congestion in the network will increase by several magnitudes as the number of simultaneously communicating sensor nodes increases over time along with the resulting number of systems monitoring them. If the network grows large enough, this could become a significant cost [10].

TCP lacks any provision for priorities. Messages are delivered in a strict first-in-first-out (FIFO) order without exception. A Utility Intranet will support many applications and message types, some having lower priority, and many shipping very large files. Because TCP lacks any notion of priority, low priority file transfers compete for the same resources as do high-priority, urgent notifications. If several TCP connections are all transmitting relatively unimportant non-operational information across a section of the network and a new TCP connection is initiated with extremely important emergency information, the most important connection will only receive its "fair share" of the connection rather than the high priority that it deserves.

TCP's behavior results in a network with very high utilization rates that are shared in what can loosely be described as a fair manner between TCP connections that are making use of it. This high utilization makes it difficult to initiate a new TCP connection or to ramp up an existing connection if new time-critical information becomes available when network utilization is high. The lengthy connection re-establishment and re-send times could result in time-critical data finally arriving stale to its intended destination.

The greedy bandwidth consumption approach underlying TCP ensures that when this happens, routers will become overloaded, a common occurrence in the modern Internet, resulting in further incoming packets being dropped until space in the router's incoming queue is cleared [10]. The back-off and slow-start that this priority connection will undergo attempting to establish a connection under congested conditions will also add significant delay [9].

2.11.6 UDP/IP Research Approaches.

Some messages forwarded within a Utility Intranet are not strictly real-time. Monitoring and assessing the impact of an evolving power shortage or some other slower contingency involves tracking data that escalates over periods measured in minutes. Still other forms of data such as power generation statistics and consumer usage data can change over hours or days [10].

In the case of non-real-time but still time-dependent communications, in the range of minutes, one solution is to investigate new or real-time protocols, middleware mechanisms, or a better use of existing transport protocols to seek to overcome these problems. Hopkinson, et al., have proposed the use of what are termed epidemic communication schemes, built upon UDP, for coordinated, wide-area SCADA protection using primary and backup wide-area agents [20]. Their assumption was that delays due to TCP/IP delivery guarantees and packet overhead would be intolerable. With less overhead than the same message employing TCP headers, no connection establishment or teardown, and no slow start and congestion avoidance, UDP messaging alleviates much of the overall traffic congestion on the same network for non-real-time (i.e. one minute or

greater) updates [10]. The point-to-multipoint efficiency of UDP also lends itself to decentralized peer-to-peer communications [20].

In the new protection system they propose, software agents would be embedded in each of the conventional protection components (i.e. an IED) to construct component information into informational messages or commands to trip breakers. Each agent would proactively search for relevant information about known primary and remote faults, then relay misoperations (e.g. breaker failures) and fault responses by communicating peer-to-peer with any other available agents at the same substation or at remote substations or control centers. In all test cases, the agent sharing and group awareness approach allowed the same information to be learned much faster and more reliably than standalone alternatives. Agent interactions could compensate for problems with better performance, even in the face of system malfunctions, increased traffic loading, and decreasing bandwidth, than in traditional TCP schemes or point-to-point legacy protocols [20].

In their simulations, three types of agents were envisioned and implemented: primary agents, backup agents, and load agents. Primary agents were responsible for the first zone protection, 100% of the transmission line, and backup agents for the third zone protection (i.e. the first zone plus all the transmission lines connected to the remote end of the first zone). Load agents were only responsible for sending their current state, usually their current phasors, to the backup agents. An agent, at initialization, could either receive a list of the agents in its own protection zone with which it could communicate or, otherwise, learn this information through a network topology discovery algorithm [20].

An IED, for example, could be loaded with software agents that perform control and/or protection functions. Agents embedded within an IED perceive their environment through local sensors and act upon it through the IED's actuators. Sensor inputs might include local measurements of the current, voltage, and breaker status. Actuator outputs might include breaker trip signals, adjusting transformer tap settings, and switching signals in capacitor banks. Agents might even interface with systems such as SCADA master stations.

Primary and backup agents followed a differential philosophy to detect a fault. At every time-step, they read their local current phasors and sent this information to their agent counterparts. Once an agent received the phasors from its protection zone's remote end, or ends, it calculated the differential current and decided whether a fault occurred or not. After detecting a fault, the agents took action based on preset rules [20].

One drawback to the software agent scheme proposed is that, while newer, processor-based IEDs might have sufficient embedded memory, disk, and computational capacity to be loaded with and effectively use these agents, most older systems have such limited resources that they could not.

An interim solution to be used with slower legacy systems might be a separate low-cost, computer or other PC-based box attached at key points in the infrastructure to gather these inputs and perform calculations on behalf of the protection components themselves. This box could then issue messages directly to other equivalent boxes that would translate them into simplistic, understandable instructions to protection components or directly to the protection components themselves that supported this

scheme. The latency for computational analysis, message formulation, and transmission must then be figured into estimated response times.

A similar software agent concept is central to the **trust system** security functionality proposed and evaluated in Chapters III and IV of this thesis.

2.11.7 TCP/IP Research Approaches.

The greatest difficulty with applying common network protocols for SCADA communications is meeting the strict time constraints. In SCADA systems, "the shortest deadlines are seen in relay control algorithms for equipment protection systems, which must react to events within fractions of a second. For near real-time response (i.e. less than one second) delivery guarantees are attractive. Since UDP does not provide this guarantee, TCP/IP alternatives can be investigated.

A Virginia Tech research team has proposed a scheme that they have called PS-TCP/IP because it is a fully TCP/IP-compatible power system communication network [21]. The PS-TCP/IP concept envisioned a utility TCP-IP network (separated either physically from the Internet, or possibly behind a NAT proxy and firewall for security) with IP addresses assigned to each power system device and an undefined but assumed method for management of traffic flows to lessen congestion. The team made two fundamental assumptions. First, they assumed that "only utility applications will be running on the PS-TCP/IP, so network traffic planning and congestion control can be well managed and the response time can be guaranteed" [21]. Second, they assumed that, "since it is a private network, the security issue can be well managed" [21]. The paper's caveat is that "utility companies can build PS-TCP/IP together with their original Intranet; however, a "firewall" must be installed to ensure the security of utility

communications" [21]. In reality, the security situation may be more complex than that and must be evaluated organization by organization. Many companies have begun to mix e-mail and office automation traffic on the same network, making it more difficult to identify malicious packets in a mix of thousands of web interactions and e-mails.

For the purpose of this thesis, it was necessary to make similar assumptions and recommendations for the most ideal security posture necessary for basic analysis before progressing to a more complex state, namely bandwidth guarantees and a Utility Intranet primarily separate from the Internet. In a similar manner, IP addresses were assigned to each system in the simulation network but did not follow the team's recommended address assignment schema and were chosen as larger IPV6, versus IPV4, addresses.

2.12 Current State of SCADA System Protection

The old paradigm was to install a system, let it run unattended, and replace it in about five years or more. For newer PC-based systems, utility companies have to wrestle to cope with more dynamic operating procedures and financial planning (i.e. install a system, patch it at least every week, perform backups and virus scans, upgrade or replace incremental capabilities each year, and train personnel on the changes) without impacting 24/7 operations and quarterly profits [7].

On the positive side, the SCADA constituency is becoming increasingly aware of their systems' vulnerabilities and is taking action through increased emphasis on information systems security peculiar to the needs of SCADA users. In addition, standards organizations concerned with data acquisition and control are developing guidelines and standards for the security of SCADA systems. National laboratories have

established SCADA test beds to evaluate the most effective security measures. Organizations such as the National Institute of Standards and Technology (NIST) have initiated programs focusing on SCADA security [1]. The negative side is that these standards, guidelines, and security measures have not been universally applied to critical infrastructure applications because of lack of funds, management apathy, other issues perceived as higher priority, and lack of guidance in some sectors [1].

Conventional IT cyber security approaches generally focus on standalone products (i.e. firewalls, IDSs, router ACLs, etc.) that are associated with individual devices on a network. This point-oriented security approach is vulnerable to attacks that circumvent the one particular security control. In addition, other parts of the network might be unaware that an attack is occurring. Security researchers have noted that what is needed is a coordinated security paradigm that takes advantage of the capabilities of devices such as routers and switches that are cognizant of network activities on a larger scale. What is necessary is to develop an adaptive network and application-aware solutions that address security as a collaboration of defense mechanisms operating as a defense system to identify threats and respond accordingly [1].

The future power grid will begin to support higher levels of integration and federated systems services [22]. The **trust system** concept was intended to support the goals for current and future SCADA systems, as listed in Tables 6 and 7.

Table 6. Requirements for Current SCADA Systems

Requirement	Description
Quality of Service (QoS)	SCADA systems are deterministic. QoS, precise interrupt timing, reliability, and low latency are more critical than throughput [6].
High Availability	Real-time SCADA systems cannot afford delays that may be caused by information security software and that interfere with critical control decisions affecting personnel safety, product quality, and operating costs.
Security	Security in the utility community has a very unique meaning which is quite different to that used in IT networking. NERC Form 715 defines [1] security as "a system's capability to withstand system disturbances arising from faults and unscheduled removal of bulk power supply elements without further loss of facilities or cascading outages." If the NERC definitions of adequacy and security were modified to apply to SCADA systems in general, they might read as follows: Security: A system's capability to withstand system disturbances arising from faults or unauthorized internal or external actions without further loss of facilities, compromise of human safety, and loss of production [1].
Legacy device interface	Most plant components in existence today have minimal computing resources. They do not usually have excess memory capacity that can accommodate relatively large programs associated with security monitoring activities [1].
Self-describing	Available data is discoverable
Automated	Advancements in systems are requiring fewer operators and more automated SCADA control. As the master station software is more and more capable of analyzing data, it has to present less to the operator [6].

Table 7. Goals for Future SCADA Systems [22]

Goal	Description
Self-healing/adaptive	Correct problems before they become emergencies
Dynamic	Interactive with consumers and markets
Optimized	Make the best use of resources and equipment
Predictive rather than reactive	Prevent emergencies ahead of time rather than solve them after they occur
Distributed assets/information	Share resources across geographical and organizational boundaries
Integrated	Merge all critical information
More secure	Protected from threats from all hazards

2.13 Specific Challenges to SCADA Security and Recommended Solutions

2.13.1 Per-User Authentication and Access Control.

2.13.1.1 SCADA Security Issues.

In the SCADA environment, a control operator might need to enter a password to gain access to a device in an emergency. If the operator types in the password incorrectly a few times, a conventional IT security paradigm, which presumes an intruder trying to guess the password, will lock out the operator. Locking out the operator is not a good thing in real-time control environments [7].

Many systems require no authentication at all. When accounts do exist, username and password information is almost always sent in the clear in both human-to-machine and machine-to-machine applications [7]. In practice, SCADA systems or consoles tend to be configured with the same username and password or with standard defaults like *console*, *administrator*, or *anonymous*.

RTU test sets, used to issue commands to an RTU, are commonly available on the market. The systems don't authenticate and have little to no data validity checking.

2.13.1.2 Recommendations from Literature

For operators on local control devices, passwords might be eliminated or made extremely simple [1]. In situations where the passwords might be subject to interception when transmitted over networks, encryption should be considered to protect the password from compromise.

Access controls should be implemented for all SCADA systems. Role-based access controls might be used at the supervisory level of SCADA operations [1].

In addition, access might also be restricted based on two-factor authentication and digital certificates or challenge-response tokens [1]. Options include biometrics, smart card identification, and other authentication technologies.

Procedures should be implemented to monitor access controls for authorized access, un-authorized access, and unsuccessful un-authorized access attempts.

2.13.1.3 Objections and Questions from Utilities.

Currently, biometrics are not completely reliable. Depending on the characteristic being examined, there might be a high number of false rejections or false acceptances. There are also issues possible with throughput, human factors, or system compromises.

Given the real-time nature of SCADA operations, how would password policies be applied to prevent lockout in emergency situations? In addition, how would rights be managed for each person that may need to perform multiple, changing roles?

It is costly to keep access control lists (ACLs) of who should connect to whom up-to-date as the network evolves over time. It may not be practical to reconfigure all monitoring systems rapidly when a problem arises unless there are automated communications to push updates to each affected node in the network [10].

2.13.1.4 Trust System Solutions.

The **trust system** interacts with an existing authentication mechanism such as a logon server to enforce multi-level, role-based access based on the success or failure of credentials provided by the one that is logging in. For this thesis, the most restrictive policy was assumed and is suggested, requiring initial logon of every new user as well as every system that is coming back online. By tracking the time, conditions, and status of all logons and monitoring, correlating, and even blocking suspicious logon activity

(tracked by username, IP address, credentials, and distance), the **trust system** provides comprehensive logon state and security situational awareness.

The **trust system** also relaxes standards in situations where it has a greater level of trust that the user (or system) is who they say they are, based on the quantity and reliability of the credentials provided to logon and the source of the logon. It differs from most IT security schemas by providing more chances to a user who, after one or two tries, is highly close to being correct, but appears to have simply forgotten or mistyped a few characters of their password. It also simplifies access in emergency situations by assuming that any logon is a priority, to speed this process, and by implementing a one-time network logon which is good for any system or data in the local network enclave to which the individual is entitled, based on their assigned role, instead of separate logons for different systems or higher-level roles when the user is still at the same computer. The pre-defined user role and the access level, calculated from the credentials provided, are used to allow and disallow access to systems, folders, files, and data elements for each user. In the event of lost, misplaced or forgotten credentials, the **trust system** can allow an elevation request from the user to another user with the same logged-on access level desired by the requestor. In this way, assuming proper (preferably visual) verification occurs, they can be approved temporary access at the higher access level required to perform their job. This might be the case if, for instance, they accidentally left their smart card at home or experience biometric read errors and cannot otherwise gain root (or other level) access with only a username and password. Use of this feature, of course, should be the exception and not the norm.

The **trust system** can perform data and validity checking on incoming commands and messages on behalf of field equipment (i.e. from an RTU, PLC, or IED test set or admin laptop); however, access control at the SCADA field equipment, first, and then authentication at the network logon server (i.e. a network-level logon) is preferred before any further communication is allowed with the SCADA node. This can be facilitated by the **trust system**. Authentication by any device connected to the IED requires an IP port on the SCADA field device for connection and an IP-enabled test set or laptop (preferably using encryption) capable of supplying authentication credentials.

Distribution of trust agents throughout the network allows a much more decentralized and efficient implementation of this authentication scheme and all other **trust system** functions.

2.13.2 Prevention of Data Interception or Alteration.

2.13.2.1 SCADA Security Issues.

Traditional RTUs, PLCs, and IEDs are designed for efficiency to prioritize task execution using microprocessors with limited memory and computational capacity, stringent real-time constraints, low bandwidth links, and minimal attention to security policies [18]. They typically send information without transmission security and many use wireless connections susceptible to interception [1].

Packet-based SCADA protocols usually provide message integrity checking at the data link layer to find errors caused by electrical noise and other transmission errors [18]. Since these checks do not include encryption technology, to protect against malicious interference with data flow, and their algorithms are well-documented and publicly

available, they only provide protection against inadvertent packet corruption caused by hardware or data channel failures [18].

2.13.2.2 Recommendations from Literature.

Digital certificates and cryptographic keys should be used and managed for encryption and digital signatures relating to SCADA system elements [1].

Transmission errors are best detected and handled close to the source or physical medium (i.e. at the data link layer) while protection from network content alteration is best achieved as close to the application layer as possible (i.e. the network layer or above) [18].

When packets are routed through a corporate LAN or Utility Intranet, message IP addresses must be visible for each router and switch along the way to read and select the appropriate path to route it to its destination. Traditional security solutions implemented at the network layer or above are usually proprietary VPN schemes or standards-based (e.g. IPsec) protection schemes" [18]. For these public-key cryptosystems, key management, including certification that the public key actually belongs to the person named, is an important issue that has to be handled by the organization. More importantly, they can require relatively long processing times that may be incompatible with the real-time requirements of SCADA control systems [1].

As a result, symmetric-key cryptosystems, which can perform much faster, may be more suitable for use in the SCADA environment, however, key management becomes much more difficult. Although, symmetric-key cryptography has not yet been widely applied to SCADA systems, it is applicable to data transmitted over a long-distance SCADA network and could be added to protect its most critical portions [1].

2.13.2.3 Objections and Questions from Utilities.

Older systems can't support the computational burden of block encryption [18].

Encryption, configuration control, and other strong security measures usually reduce the ease of management of SCADA systems. Complexity is the bane of efficient SCADA operations.

IP already adds nearly 30% more overhead to SCADA communications, encryption will add too much latency.

The TCP security model, SSL, permits a client of a server to authenticate a server and then encrypt sensitive data such as a credit card number, but that capability does not account for the varying levels of trust and other issues that arise between mutually suspicious operators [10].

2.13.2.4 Trust System Solutions.

Research for this thesis, indicates that IPsec public key encryption can be used in some cases for non-real-time communications and has the potential, with faster processing, to reduce latency to the point where it could be applied to real-time communications.

For legacy systems and applications that do not, or cannot, provide encryption at the IP-level or above, the **trust system** in gateway-configuration, with IPsec tunnel mode, can act as an encryption gateway. This can occur by encrypting the unencrypted incoming packets, adding an IP header with destination address of the next **trust system** along the way to the destination, and forwarding it. When the packet is received by the **trust system** closest to the destination, it strips the address, decrypts the packet revealing the destination address, and forwards it, unencrypted, to the destination.

For systems that can be loaded with software **trust system** agents, the agent middleware can interact to package the data with IPsec encryption at the host before it is passed on to the physical/data link layer for transmission.

IPsec delay is highly processor-dependent. Until technological improvements are made in the SCADA hardware installed in utility networks to allow fast enough processing and less queuing delay, stand-alone symmetric key hardware can be added to the network to encrypt packets after they leave the source, switch, and possibly the first router, at the physical layer, and decrypt the packet before passing it to the destination router, switch, and recipient. In that case, the basic IP-to-IP *firewall rules* checks of the **trust system** could still be performed on a packet in transit and fixed-length message-types could be deduced. However, unless the **trust system** itself were implementing the symmetric key encryption, the **trust system's** *format module* and some *access control matrix* checks would be negated because it could not see the encrypted data inside the packets, including the message type. Once the data was decrypted, though, full **trust system** checks could be performed at the host level, catching at delivery instead of stopping malicious activity closer to the source.

2.13.3 System Hardening.

2.13.3.1 SCADA Security Issues.

Once SCADA systems are installed in an operational production network, they are rarely, if ever, patched. SCADA system device banners are rarely disabled, giving out device and software names, versions, and manufacturers (important sources for

manuals of technical and operational information that could be used to attack and compromise them).

2.13.3.2 Recommendations from Literature.

Unused physical ports, banners, and network services should be disabled and patches should be kept up to date [23]. Operating system and application patches should be applied as they are made available, always testing for negative impacts on system functionality first [23].

2.13.3.3 Objections and Questions from Utilities.

The Microsoft Service Pack 2 fix for the Blaster worm turned off anonymous logons by default for the DCOM service, requiring authentication. The OPC standard for data transfer runs without authentication. Blindly implementing SP2 would have broken SCADA systems running OPC that was not designed for logons [7]. This illustrates the complexity of transitioning to COTS products where one-size-fits-all vendor patches may not always work for unique, partially legacy-based, and time-critical control configurations.

2.13.3.4 Trust System Solutions.

While it is assumed that unused ports are disabled by default by SCADA administrators, to supplement interface-level defenses, the **trust system** software agent on a system, acting as middleware between the transport and physical/data-link layers, can perform interface-level access control via its *ACM* for useable ports that are configured ON (or OFF) yet for which connection and access should be restricted only to specific IP addresses and authorized user/role combinations.

A developmental testbed must be established (either within each company or at area or regional level for economy) to duplicate utility systems down to the company substation level for the purposes of testing COTS patches, software, and upgrades prior to deploying them to the production network. This could also be a role for the NSTB in conjunction with a regional or national utility control center.

Most utilities employ redundant servers for reliability. After testing and approval have occurred on developmental duplicates of operational configurations, patches should be loaded onto an offline production system within the company that will be employing the patch and functionality verified prior to rotating the offline system back into operation. This procedure can also be used to regularly exercise the company's backup systems and restoral procedures or to run antivirus scans.

Oversight and accountability for testing, approval, assigning suspense dates, and tracking compliance for patches must be established at regional and national levels to ensure continuity of security posture across the entire Utility Intranet.

Throughout changes from primary to backup, the **trust system** must have all systems configured in its ACM. **Trust system** *ACMs* require each network node to log on and off of the network as they connect, shutdown, or are disconnected. The **trust systems** then update one another as they learn that one system has gone offline and another is online, to maintain situational awareness and accurately deconflict suspicious events.

2.13.4 Secure Software Engineering.

2.13.4.1 SCADA Security Issues.

Security is often an afterthought or not even considered in SCADA operating system design and implementation, hence secure coding practices are not required. They usually have no input validation, non-secure programming syntax and commands, and are vulnerable to buffer overflow, memory dump, etc.

Manufacturers haven't been forced to improve SCADA security and there is little incentive for vendors or developers to do so on their own [18]. Telecommunications equipment and services sold to utilities is "big business, averaging 3.5 million dollars annually and rising, according to UTC Research" [22]. It's hard for manufacturers to financially justify investing extra manpower and dollars to develop, implement, and maintain additional security features and practices that don't make them any more competitive or increase profits over their peers who don't.

2.13.4.2 Recommendations from Literature.

Development of an open, yet secure real-time operating system is encouraged, along with a review of existing SCADA protocols and IEC standards for security.

Federal and state governments should provide sufficient incentives to encourage private sector investment and development in SCADA security.

2.13.4.3 Potential Solutions.

Requirements documents for new systems, protocols, standards, and software should explicitly state security capabilities required and secure coding practices expected to prevent such avoidable security mistakes prior to new software development and marketing.

The **trust system** can only question or stop potentially malicious commands and input in packets created by deployed code or intended to exploit known vulnerabilities in deployed code. Any source code purchased or downloaded for utility systems should be scanned for examples of non-secure code and associated vulnerabilities in order to determine specific signatures for the **trust system** or other intrusion detection systems to look for. This is not easy and there are few, if any, automated tools for this purpose, however, it is reasonable to assume that such a tool could be developed to search for instances of potentially bad practices to narrow the examination in source code. Economy would be achieved by performing this function once per application at a national or regional level, even after deployment, and going back to vendors to re-code specific sections more securely.

An advocate on behalf of utility companies could be established at the national interconnect level to perform vulnerability scans, penetration tests, and code reviews in conjunction with fly-off tests of vendor solutions. The same entity should also consolidate community requirements and hold vendors accountable for developing solutions that meet not only time, safety, and reliability, but security specifications as well, in their designs.

2.13.5 Non-secure, Backdoor Connections.

2.13.5.1 SCADA Security Issues.

SCADA administrators and industrial automation analysts are often deceived into thinking that because their industrial networks are on separate systems from the corporate network, which is often connected to the Internet, they are safe from outside attacks.

Security is most easily compromised at the SCADA host (master station and HMI) or control room level. If the SCADA computers are logging data out to some back-office data repository like SQL server, Oracle, or PI Historian, then the SCADA computers must be on the same network as they are or have a path to access them. This means there is a path to the SCADA systems and eventually to the remote substation field equipment through the corporate network. Often these connections are left open 24x7 to allow full-time logging, which provides an always available path through the network for someone to attack [8].

A data concentrator or substation host processor at a substation mediates all communications to and from IEDs by forwarding the message to the appropriate IED and routing the response back to the original caller.

Modems are commonly imbedded in substation end device equipment such as IEDs, PLCs, and RTUs to allow vendors to poll them over dial-in phone lines to support the product or as an easy way to retrieve non-real-time data from them. These modems will often have default usernames and passwords that aren't changed or backdoor usernames and passwords that can't be easily changed by the customer. Some will accept calls from any source that knows their number.

Though there is no access control, monitoring, or authentication employed on these connections by the utilities themselves, company employees often have a false sense of security because they assume these end devices are protected by the non-corporate vendor network connections [1].

Unsecured remote desktop applications like X-Terminal, PC Anywhere, and Exceed are frequently used for remote visibility and administration within utilities and over the Internet from home or vendor offices [18].

Many IEDs are IP-enabled with much of the data to and from them traversing non-secure wireless networks [18].

2.13.5.2 Recommendations from Literature.

It is best to not to allow any communications to the substation from outside the secure utility network [18].

For existing dial-up lines either require strong user authentication, encrypt communications, or eliminate them altogether [18]. Dial-back modems should, at a minimum, implement separate lines for incoming and outgoing call back. This helps protect the integrity of the phone switch [18].

Eliminate all connections to the Internet from the SCADA network. Do not enable Web-mail for remote e-mail access. Instead, maintain accounts and servers for outside communications on a physically separate office LAN that does not connect to the SCADA network. Also, implement a secure VPN solution for any remote desktop access from either network.

For corporate connections to the SCADA network**, c**onsider web-based thin client solutions that enable plant, management, production, and maintenance personnel to view read-only, real-time process graphics from a remote location [11]. A user can use a standard web browser and Utility Intranet connection to see animated displays of manufacturing activity, thus allowing a more informed decision-making process [11]. At the same time, thin clients can protect data by not allowing users to change values.

The US military has gone to great efforts to ensure that its classified networks are in no way connected to the Internet or to its unclassified systems. Classified digital information is encrypted to the highest level in-transit and hard drives are locked in safes with two-person integrity checks on the lock when not in use or attended. Buildings where classified information is processed implement strict physical access control and require positive identification and need to know for entrance. They also use metal sheets or mesh in the walls and are designed without windows to prevent unencrypted electromagnetic emanations from computer workstations and screens from being detectable or visible from outside the building. Operational security is strictly enforced and trained at least annually so employees recognize sensitive information and even sets of data that, by themselves are unclassified, but when linked together, can give indications of classified operations or intentions.

Though it is not necessary to employ the exact same measures as for national security secrets, critical infrastructures must be seriously evaluated and approached with the same well-planned, deliberate, security-conscious mindset.

2.13.5.3 Trust System Solutions.

One main function of the **trust system** is to implement a firewall. Whether loaded onto its own server (as a hardware firewall) or as a software agent running on a SCADA node (as a software firewall) the **trust system** filters out unauthorized packets, adding security to any incoming connections to which it is attached based on a whitelist of known to-be-authorized traffic by source and destination addresses, port, protocol, and message type. This is in contrast to more typical and more error-prone blacklist rules, which attempt to account for every type of traffic that would not be authorized.

This is easier to do with a controlled network where only a finite number of message types, protocols, and source-destination IP address pairs is possible. Accurately defining authorized traffic for web searches, e-mail, and other unpredictable common office exchanges that occur with numerous Internet servers, clients, and applications is nearly impossible as most Internet-connected business networks can attest from seeing their share of viruses, compromises, and zero-day exploits.

Keeping e-mails and coordination between utility organizations on a separate network may be more of a hassle (potentially two separate client computers—one connected to the SCADA Utility Intranet and the other to the Internet for WWW searches, coordination with vendors, etc.) but is the most secure configuration for the operational network. On a Utility Intranet, separate from the Internet, a global address list and DNS servers can be maintained within areas and regions to feed legitimate e-mail addresses and IP lookup information for utility-specific clients and e-mail servers. Unless this separate Utility Intranet is compromised by an insider (maliciously or through infected disks or thumb drives) the only outside avenue of attack would be through a rogue connection to the Internet, a wireless access point, or dial-in through the telephone network.

2.13.6 Systems In Need of Maintenance.

2.13.6.1 SCADA Security Challenges.

Many critical infrastructure systems have a history of deferred maintenance that has to be addressed before implementing a security system.

2.13.6.2 Trust System Solutions.

Implementing **trust systems** in gateway configuration such that they only interact with each other, or with nodes that can support **trust system** agents, adds security to a network that is seamless to any nodes that cannot yet be loaded with a **trust system** agent themselves.

2.13.7 Timely Detection and Elimination of Malicious Code.

2.13.7.1 SCADA Security Challenges.

SCADA systems do not use antivirus software.

2.13.7.2 Recommendations from Literature.

Antivirus software should be implemented wherever possible [24], [25].

2.13.7.3 Objections and Questions from Utilities.

The computational overhead associated with running antivirus software, updating virus signature databases, and quarantining or deleting malicious code require computing cycles that might seriously affect the real-time performance of SCADA system components. Automatically updating virus databases from Internet antivirus sites exposes SCADA systems to more viruses and attacks.

2.13.7.4 Trust System Solutions.

As a rule of thumb in SCADA systems, scans (antivirus or otherwise) should be conducted on systems rotated temporarily offline and only returned to service when discovered discrepancies are remediated, for minimal impact to operations. Regularly rotating a system offline (replaced by its backup) for scans is also a way to exercise

backup contingency plans ensuring hot spares will always be functional in an emergency situation.

The **trust system** can be loaded with or call antivirus software to run virus scans on e-mails and their attachments traversing the company's piece of the Utility Intranet.

Antivirus updates can be downloaded to media manually from the Internet on a separate network. After scanning the media, it can be hand carried for loading onto the Utility Intranet and distribution. Another solution might be to work with antivirus vendors to mail or ship disks regularly with the latest updates. It is also important to fully test any update on developmental (or test) SCADA systems before loading onto the utility production network, to ensure patches and antivirus detection/cleanup actions will not accidentally break SCADA applications.

2.13.8 Resource Exhaustion Attacks.

2.13.8.1 SCADA Security Issues.

A denial-of-service (DoS) attack is an attempt to either temporarily or indefinitely disable a network system or resource or simply it make it unavailable to legitimate users. Methods of attack can include flooding a link to prevent legitimate network traffic, preventing a particular individual from accessing a service, or disrupting service to a specific system or person. A DoS attack is the greatest problem during times of peak loading like an emergency[18].

A DoS can disrupt a server by sending more requests than it can handle, thereby preventing access to a service; consume computational resources, such as bandwidth, disk space, or CPU time; disrupt configuration information, such as routing information; or disrupt physical network components.,

. A DoS attack may include execution of malware intended to max out the CPU's usage, trigger errors in the microcode of the machine, trigger errors in the sequencing of instructions, to force the computer into an unstable state or lock-up, exploit errors in the operating system to cause resource starvation and/or thrashing, or crash the operating system itself.

2.13.8.2 Recommendations from Literature.

First, there are fewer avenues of attack from the outside on networks that are physically isolated from the Internet [18]. Perimeter defenses with appropriately configured alternate routes can provide some defense (i.e. relief) in the face of DoS attacks, presuming that the alternate links do not become saturated [3].

"Defense on telephone system requires managing QoS by giving preferential dial tone to critical users while denying peak-load service to ordinary users" [18].

Filtering is often ineffective, as the route to the filter will normally be swamped so only a trickle of traffic will survive. However, by using an extremely resilient stateful packet filter that will inexpensively drop any unwanted packets, surviving a DDoS attack becomes much easier. When such a high performance packet-filtering server is attached to an ultra high bandwidth connection, communication with the outside world will be unimpaired so long as not all of the available bandwidth is saturated, and performance behind the packet filter will remain normal as long as the packet filter drops all DDoS packets.

Having a separate, emergency block of IP addresses for critical servers, with a separate route can be invaluable. A separate route can also be cost effective because it can be used for load balancing or sharing under normal circumstances and switched to

emergency mode in the event of an attack. WAN-link failover will work as long as both links have DoS/DDoS prevention mechanisms.

SYN cookies modify the TCP protocol handling of the server by delaying allocation of resources until the client address has been verified. This seems to be the most powerful defense against SYN attacks. SYN floods can also be prevented using delayed binding or TCP splicing [26].

Content-based DoS can be prevented using deep packet inspection. Attacks originating from dark addresses or going to dark addresses can be prevented using bogon (bogus IP) filtering.

Automatic rate filtering can work as long as rate-thresholds are set correctly and granularly. Routers have some manually-set rate-limiting and ACL capability. Most routers can be easily overwhelmed under DoS attack. If rules are added to take flow statistics out of the router during the DoS attacks, they further slow down and complicate the matter.

Application front end hardware is intelligent hardware placed on the network before traffic reaches the servers [26]. It can be used on networks in conjunction with routers and switches. Application front end hardware analyzes data packets as they enter the system, and then identifies them as priority, regular, or dangerous [26].

Intrusion-prevention systems are effective if the attacks have signatures associated with them. However, the trend among the attacks is to have legitimate content but bad intent. IPSs which work on content recognition cannot block behavior based DoS attacks. An Application-Specific Integrated Circuit (ASIC) based IPS can detect and block denial of service attacks because they have the processing power and the

granularity to analyze the attacks and act like a circuit breaker in an automated way. A rate-based IPS (RBIPS) must analyze traffic granularly and continuously monitor the traffic pattern to determine if there is a traffic anomaly [26].

2.13.8.3 Trust System Solutions.

A **trust system** at the network perimeter can enforce encryption and authentication policies for packets entering the network, requiring a malicious DoS packet to have the proper key in order to stand any chance of entering the protected enclave. The packet must also meet the **trust system** *firewall rules* for source IP address, destination IP address, destination port, and authorized message type combinations. In the case of an encrypted DoS (less likely from an outside source than a misconfigured or malfunctioning system) the packet would enter the network but repeated identical or similar packets in a very short period of time would be detected as a potential DoS suspicious event and blocked very shortly after beginning. To prevent resource exhaustion of the **trust system** itself, the **trust system** is capable of communicating further down the line to query other **trust systems** to identify the path the packet has traveled and notify them to discover and block similar activity closer to the source.

2.13.9 Cyber Intrusion Detection.

2.13.9.1 SCADA Security Challenges.

Lack of network security countermeasures in many utility networks makes it nearly impossible to detect cyber intrusions. Current substations generally do not have firewalls or intrusion detection systems (IDS) installed, so it is not possible for those

companies to know if, when, and by whom they are being targeted [18]. Those that do have some IDS capabilities often do not update or monitor them regularly. They also rarely have the expertise to use them effectively unless they hire security specialists.

Because of the varying ages and sophistication of some SCADA system components, many do not even have logging capabilities. Available audit trails are usually not turned on because of the drain on processor performance and limited memory. In general, substation automation systems that do have logging enabled don't log who is attempting to obtain access to them [18]. With no logs or audit trails, activities of malicious insiders are effectively untraceable and there is no easy way to define security policies and traffic filtering for what is usual or unusual activity in the SCADA network [18]. In many cases if an incident occurred there would be no way to tell if it were malicious or accidental [7].

There are few SCADA-aware firewalls and, though the National SCADA Testbed at Idaho National Lab is working to develop intrusion detection capabilities for existing control systems, it is not the SCADA system developers' priority [18].

2.13.9.2 Recommendations from Literature.

Utility organizations should implement network rings (or layers) of defense, also known as defense-in-depth [8].

To start with, there should be perimeter monitoring on remote, unattended SCADA system elements [1].

Firewalls can be used to screen message traffic between a corporate IT network and a SCADA network on the Utility Intranet. This configuration can protect SCADA

systems from penetrations that have occurred on the corporate side. Some issues that have to be considered when applying firewalls to SCADA systems are the delays introduced into data transmissions, the skill and overhead required to set up and manage firewalls, and the lack of firewalls designed to interface with some popular SCADA protocols [1]. While most firewalls do not support SCADA protocols, this situation is being researched by a number of organizations and some SCADA-aware firewalls are under development [1].

Perimeter defenses should employ two layers of firewalls that will conduct stateful data inspection. One firewall would be installed between the Utility Intranet wide-area-network (WAN) side and your corporate LAN and a second strong firewall would wall off the organization's SCADA networking systems from both the internal corporate network, with its preponderance of non-real-time e-mail, web, and office automation traffic, and the mixed content traversing the external Intranet between utility organizations. This would provide at least two layers of firewalls between the SCADA networking systems and the external Utility Intranet [8]. Only trusted connections will be allowed to link into the SCADA system behind the outer firewall in the outer trusted zone, but will also have to pass the scrutiny of the inner firewall policy sets as an added layer of protection from compromise [3]. Firewalls must be SCADA-aware to recognize and protect critical traffic to and from SCADA supervisory control elements [1].

The perimeter router can compliment these defensive systems by implementing strict access control lists to deny all access and only allow access by exception rules (i.e. a whitelist of authorized traffic).

The SCADA network must also employ tightly managed subnetting to ensure an exclusively private network, which will effectively hide the SCADA system from outside entities and the utility's public network in general [3]. If IPsec is used, care must be taken to deconflict incompatibilities with Network Address Translation.

The network connections and any DMZ should be equipped with several types of intrusion detection systems. Network IDS devices should monitor the traffic on the network links and in the DMZ. Host-based IDSs should ensure that key files on critical systems and DMZ servers are not manipulated [3].

After-the-fact analysis of audit trails is a useful means to detect past events. To aid response measures, it is best to record as much of the communications traffic, as possible, however disk storage is very expensive and often cost prohibitive. Monitoring, on the other hand, implies real-time capture of data as a system is operating. Both techniques are successfully employed in IT systems and will yield similar benefits in SCADA networks [1]. For the logging of data on every packet, or even just the suspicious ones, to be practical, low-cost, high capacity storage and an IDS that can distinguish legitimate SCADA messages from unauthorized and malicious counterfeits [18].

2.13.9.3 Objections and Questions from Utilities.

IDSs, firewalls, and antivirus software might slow down certain SCADA operations. Their benefits to SCADA need to be proven to outweigh the potential negative affects on efficiency, safety, and ROI of operations [1].

Oftentimes, SCADA systems go down due to other internal software tools or employees that accidentally gain access into the SCADA network. Any time a system

goes down, even for maintenance, there's no certainty it will come back online smoothly. Adding more complex software to interact with these finicky systems could prove more disastrous.

At this time, IDSs are not available for some SCADA protocols [1].

2.13.9.4 Trust System Solutions.

The **trust system** performs firewall and IDS functions at any level of the network, even on individual systems (i.e. host-level).

Simulations have shown that **trust system** *firewall rules* and *format module* delays are sufficiently small enough to be a relative non-factor even for near real-time (less than one second delivery time) communications.

The **trust system** logs suspicious event details. Because it can unpackage and inspect each packet that crosses its path, the **trust system** can easily log all packets (suspicious or not) on behalf of any system that cannot implement logging or audit trails itself (assuming sufficient storage is available), significantly improving historical reconstruction of network events, including low-and-slow attacks that escalate over days, weeks, or months.

2.13.10 Insider Threat.

2.13.10.1 Objections and Questions from Utilities.

Privacy rights issues inhibit screening and profiling of some individuals.

2.13.10.2 Trust System Solutions.

Regardless of company hiring policies, as long as employees are required to acknowledge and authorize "consent to monitor", the **trust system** can track and log all

actions, suspicious or not, that cross its path and attribute them to an authenticated username and source system. If an individual's action are authorized by the **trust system** but later determined to be malicious, historical records will allow the piecing together of the individual's time-stamped actions.

Consent to monitor should be outlined and signed off by each employee in contracts at the time of hiring and can also be setup as a reminder (and to cover any non-company vendors, etc. that be on the network) by displaying a logon banner with the legal phrasing to which the user must click an agreement button in order to connect. The same banner would also allow the documentation of actions for legal prosecution should an attacker attempt to conduct malicious actions.

The **trust system** logs are much more complete when a **trust system** agent is loaded onto each SCADA node vice two **trust systems** in tunnel mode, which might only see the traffic between them and not node-to-node traffic at the edges (e.g. between two nodes on the same switch).

2.13.11 Limited physical security.

2.13.11.1 SCADA Security Challenges.

Whether due to budget restraints or to low priority, many substations and other remote sites are left with inadequate or lackadaisical physical security procedures, assuming that no one would really be that interested in SCADA equipment [1]. Un-locked and un-guarded facilities can allow an attacker to simply scale a fence to enter an equipment room and plug in to access the SCADA network.

2.13.11.2 Recommendations from Literature.

Fences, locks, motion-detectors, and security cameras can provide greater physical security for facilities and remote substation yards. Tamper-resistant or tamper-proof enclosures for SCADA system components are a good second line of defense to prevent unwanted meddling, compromise, or damage [1]. The use of authenticated entry and metal detectors is highly recommended for control centers and substations. In addition, emergency action plans should be updated with procedures for dealing with armed entry attempts and those procedures regularly exercised.

2.13.11.3 Objections and Questions from Utilities.

No one would really be interested in SCADA equipment and most of our substation yards have fences around them. Even if someone could get in, they wouldn't even know what to do with the equipment or have passwords to logon.

2.13.11.4 Trust System Solutions.

The **trust system** *ACM module*'s logon credentials check can require smart card, voice recognition, biometric, or other physical credentials for logon authentication before granting network access, supplementing enforcement of physical security for network actions.

The **trust system** can also notice and alert on the loss of an expected message from a specific node or connectivity loss that might have resulted from disconnection or damage to network components, aiding rapid recognition and recovery.

It is difficult to defend even the most conscientiously monitored IP networks with highly trained analysts, especially when the ability to dictate and monitor every crucial update and configuration of installations throughout the US is not available and

determined attackers are constantly crafting and testing new ways to steal information or disrupt operations. Even with its best efforts, a large, highly targeted company can expect multiple system compromises each year, and those are just the ones they catch.

The US military has gone to great efforts to ensure that its classified networks are in no way connected to the Internet or to its unclassified systems. Classified digital information is encrypted to the highest level in-transit and hard drives are locked in safes with two-person integrity checks on the lock when not in use or attended. Buildings where classified information is processed implement strict physical access control and require positive identification and need to know for entrance. They also use metal sheets or mesh in the walls and are designed without windows to prevent unencrypted electromagnetic emanations from computer workstations and screens from being detectable or visible from outside the building. Operational security guidelines are strictly enforced and refreshed at least annually so employees can recognize sensitive information and sets of data, that by themselves are unclassified, but when linked together, can give indications of classified operations or intentions.

Though it is not necessary to employ the exact same measures as for national security secrets, critical infrastructure operations and sensitive information must be seriously evaluated and approached with the same well-planned, deliberate, security-conscious mindset.

2.13.12 Proactive Vulnerability Assessment.

2.13.12.1 SCADA Security Issues.

Few, if any, proactive deception measures, vulnerability discovery, or fingerprinting of attackers, attack techniques, and zero-day exploits is conducted by companies.

2.13.12.2 Recommendations from Literature.

Regular vulnerability scans and analysis should be conducted and best practices from the SCADA Honeynet Project should be implemented on the Utility Intranet [1].

2.13.12.3 Trust System Solutions.

As a rule of thumb in SCADA systems, scans (vulnerability or otherwise) should be conducted on systems rotated temporarily offline. Scanned system may be returned to service when discovered discrepancies are remediated.

Vulnerability scanning would be a separate function from the **trust system**, but the schedule for legitimate scans that traverse the network must be updated in trust system rules so they are not assumed malicious and blocked.

The results of vulnerability scans should be used to improve the security posture of the scanned systems and to identify temporary security holes in the network that may require new **trust system** rules or signatures until a more permanent patch or upgrade to remove the vulnerability can be implemented.

The **trust system** could also be loaded with and run vulnerability scanner software, in limited instances, to gather additional information for its analysis of a suspicious event. Examples might be a port scan to a single port to determine its open or

closed status or to send a test message that should be blocked and result in the return of a RST/ACK, in order to verify proper functionality of a **trust agent's** defenses.

2.13.13 Lack of Centralized System Administration.

2.13.13.1 SCADA Security Issues.

Most utility companies have no single entity responsible for network administration. Users are usually their own system administrators (with root-level access) often with no reason to have those privileges [18].

2.13.13.2 Recommendations from Literature.

The principle of least privilege "should be applied in granting system access permissions to users and applications and in allowing access to files" [23].

Assign specific, certified individuals with roles and responsibilities as operations network administrators to monitor, modify, and maintain overall SCADA system and network health. They should work hand-in-hand with administrators of corporate LAN systems, perimeter IT devices, and security administrators if they are not the same individuals.

Restrict root-level access only to administrators and engineers that need higher level privileges (e.g. root) to perform their jobs. Normal IT practice is for administrators to logon as a regular user when they aren't performing immediate administrative functions. When administrative actions requiring root-level access are required, they should then either elevate their privilege with another password or logon with a different username and password to the specific system they need to access. Use of the root-level privilege should be reserved only for specific functions that require that privilege and

only for as long as that higher privilege is necessary, then the individual should logoff as root or return to a lower-level privilege.

2.13.13.3 Objections and Questions from Utilities.

Which of the 300 corporate personnel can monitor, control, and be certified on the more than 10,000 devices in the network, especially if an operator is the only one working on an evening shift and an emergency occurs—it's just easier for everyone to have the same rights and to be their own administrators.

2.13.13.4 Trust System Solutions.

The **trust system** does not provide administration but automatically enforces a well-planned access control policy. It tracks access attempts, generating detailed records of actions that occurred on the network which support network management and reduce the burden on human administrators, allowing companies to do more with less.

An **alert correlator** would bring synergy and speed as well as comprehensive situational awareness, management, and control to network security, administration, and operations personnel in response to all types of alerts, through its filterable, combined displays.

Unique specialties in SCADA, IT, and security administration within and organization, working together, creates a resident body of expertise, confidence, and trust that can proactively and continuously assess and improve the overall security posture of systems and network design. This capability is crucial to defining efficient security policies, rules, and signatures as well as effectively detecting and responding to network security incidents.

2.13.14 Integration of Security into Network Design and Planning.

2.13.14.1 SCADA Security Issues.

SCADA networks were designed for efficiency and simplicity without initial consideration for security. Security, if given any real concern at all, was often a low-priority afterthought.

2.13.14.2 Recommendations from Literature.

Employ a demilitarized zone (DMZ). Access to SCADA data summations from substations and sensors, if made publicly available to the Utility Intranet, should be redundantly ported to special web-enabled database servers, which live exclusively in the DMZ. Additionally, the remote sensors and substations should remain isolated. DMZ servers merely reflect the collected data concatenated and stored in core database servers and should be alternately available via application servers in the heart of the next generation security enclave to decision makers in the central SCADA control center, so no critical system resources will be lost [3].

SCADA networks should be segmented off into their own IP segment and use proper subnet masking techniques to protect the Industrial Automation environment from other network traffic like file and print commands. [8].

A company's SCADA and internal intranet IP addressing schemes should be separated from the company's public (to the Utility Intranet) network and from each other if possible.

If trusted connections link into the SCADA system behind the outer firewall in the outer trusted zone, they must still have to pass the scrutiny of the inner firewall policy sets as an added layer of protection from compromise. [3]

Additionally, the SCADA network must employ tightly managed sub-netting to ensure an exclusively private network, which will effectively hide the SCADA system from outside entities in the Utility Intranet and the utility's public network in general [3].

Use smart switches instead of hubs.

"Data mining out at the edges leaves administrators with the power to configure a security policy appropriate to their installation; deciding what data to share with others and what forms of authorization will be required before access is permitted. For example, a policy might dictate that normally, Node A limits itself to reporting voltage data and the phase- angle of the power phasor, measured locally, but when the ISO announces a "contingency," Node A may be willing to report far more detailed data. Node A would require a configuration certificate authorizing contingency-mode reporting, and could log this information for subsequent audit" [10].

Do not allow wireless connections if at all possible. Those that remain should require authentication and strong encryption added (not inherent WEP, which is easily cracked).

In addition to technical and administrative security controls, various physical security measures can be applied to protect SCADA systems. Backup, duplicate, geographically separated control centers can provide redundancy and, therefore, protection against human attacks and natural disasters. On a smaller scale, a hot backup standby SCADA system at the supervisory control center provides a means to continue operating if the primary system is disabled. As an additional security layer, the SCADA control center could be located in a remote area in an unmarked, inconspicuous building [1].

2.13.14.3 Trust System Solutions.

The trust system enforces access restrictions between IP addresses that should not be allowed to communicate with one another via specific message types and interfaces. Because the trust system analyzes and reassembles packets, it can, where necessary, replace IP addresses and provide network address translation for the purposes of hiding or making routable, IP addresses of nodes behind it.

2.13.15 Security Policies and Procedures.

2.13.15.1 SCADA Security Challenges.

Policies and procedures constitute the foundation of security policy infrastructures. Implementing effective policies and procedures can reduce liabilities and ensure subsequent prosecution of violations. Unfortunately, developing, documenting, and enforcing effective security policies are some of the most difficult measures to manage. Only a conscious, ongoing, proactive network security program can have realistic success over the long term [8]. Most utility companies lack effective, enforceable security policies and procedures.

2.13.15.2 Recommendations from Literature.

Utility organizations at all levels of the Utility Intranet, from the smallest SCADA office to regional and area control centers, should implement comprehensive, flexible, and testable security policies for each environment and for their interactions with other entities (i.e. between SCADA and corporate, inter-company, company to area control center, etc.). These policies should not be drafted in a vacuum, but instead with input

from all stakeholders (e.g. operators, engineers, IT, and management). Finally, plan and implement security policy management and assign responsibility and oversight.

It is important for SCADA operators, engineers, and administrators to work with IT departments to develop well thought out system operation and contingency plans in the event of problems, including the gamut of potential network security incidents [16]. Over the years, information system security professionals have developed a number of generally accepted best practices to protect networks and computing infrastructures from malicious attacks. They are an excellent starting point, however, these practices cannot be applied directly to SCADA systems without accounting for the different requirements of SCADA as compared to IT systems.

2.13.15.3 Trust System Solutions.

The **trust system** enforces the security policy with which it is configured. It also learns and proactively implements blocks or suggests new firewall rules, to security analysts, for suspicious activity not originally anticipated.

Security logs generated by the **trust system** document suspicious events and **trust system** response details for after action review, analysis of security policy for updates, and **trust system** configuration changes.

2.13.16 Cybersecurity Priorities.

2.13.16.1 SCADA Security Issues.

Cybersecurity is a low priority to most utility owners because of long-held misconceptions of invulnerability. First, there is industry denial about how much they are actually connected to the Internet. There is an increasing trend in connections from

the corporate network to the SCADA network for activity and performance reports. The same corporate offices are connected to the Internet for e-mail and web access. Remote login over the Internet and telephone lines for monitoring and administration of SCADA systems has also been growing in popularity for years. In the very beginning, control systems were less visible than IT systems and many were not even connected to external networks. Their components required detailed technological knowledge to implement and operate, so the myth of security-through-obscurity had some basis in fact, but that is not so anymore.

Fear of economic impact has resulted in isolationism (i.e. reluctance to ask for help or report network security incidents). A press release out of Washington, dated April 7, 2002, stated that, according to an FBI survey, most large corporations and government agencies have been attacked by computer hackers, but more often and more frequently they do not inform authorities of the breaches. The survey found about 90% of respondents detected computer security breaches within the previous year but only 34% reported those attacks to authorities. Many respondents cited fear of bad publicity about computer security. There is much more illegal and unauthorized activity going on in cyberspace than corporations admit to their clients, stockholders, and business partners or report to law enforcement [8].

2.13.16.2 Recommendations from Literature.

Education of decision makers in the industry is key to dispelling the myths. Vulnerability assessments have already demonstrated unauthorized access to SCADA and Distributed Control Systems. Examples from contracted penetration testing, using no zero day exploits, indicate the level of naivety among SCADA users. A common

misperception among SCADA operators and managers is that “the threat is low because outsiders know nothing about our systems”. “They were appalled to then learn that teams were able to, in a matter of minutes, gain access to the SCADA control network through unsecured Wi-Fi access from the neighborhood, unknown and unprotected dialup lines, and the Internet. Although organizations were adamant about the fact that their operations network was not connected to the Internet, the teams more often than not identified an interconnection between the production and office network, with no airgap, and the office network then connected to the Internet. The teams discovered network diagrams that in many cases didn’t match reality and laptops, not tracked or accounted for, allowed to connect to the production network from the outside (spreading viruses and worms) [7].

A combination of scheduled vulnerability assessments to include remote and internal scans (even lab results can suffice), human engineering analysis (i.e. looking for written-down passwords, accessible network equipment, phishing techniques, etc.), and operational security assessments (i.e. searching for and piecing together sensitive information from public websites and records), can prove just how vulnerable a particular network or system is and can demonstrate the negative operational impact that could be created by an attacker.

2.13.16.3 Objections and Questions from Utilities.

A well known CIO stated in the 2002 issue of CIO magazine that “most public utilities rely on a highly customized SCADA system. No two are the same, so hacking them requires specific knowledge,” referring to the company’s unique design and access

to that customized software [1]. He also stated that "cyber terrorism may not be nearly as worrisome as some would make it. That's because it's utterly defensible" [1].

2.13.16.4 Trust System Solutions.

Even with technical training, regular application of the latest patches, security software and hardware, and dedicated specialists for round-the-clock monitoring, even the most heavily defended IT networks see their share of system compromises throughout the year from Internet connections.

The **trust system** records suspicious event details useful for IT and security personnel to prove to management the types and quantities of attacks attempted against the network when suggesting investment in security purchases.

Unnecessary ports, obviously, should be closed. As another line of defense, though, the **trust system** protects the unprotected (i.e. systems that for one reason or another have open ports which for which there should be no communication). In this case, the **trust system** blocks incoming packets destined for that port and IP address combination.

Institution of a national utility certification program that ranks companies and areas on their production, training, efficiency, environmental impact, rates, customer satisfaction, and security performance would increase healthy competition for customers now able to pick and choose their energy sources.

A certification program, coupled with external vulnerability assessments and mandatory incident reporting, would reward companies with good management, policy, and security measures, encouraging network security investment.

It would soon become apparent that the number of attacks on a company is irrelevant as compared to the ability to quickly and consistently detect and prevent breaches, which are the hallmarks of a security conscious organization.

The **trust system** makes it easy to gather and analyze attack data for reporting and proving successful mitigation by a company, allowing it better protect its operations while gaining a higher security certification than its peers, and potentially higher profits due to consumer confidence.

2.13.17 Economics and Return on Investment.

2.13.17.1 SCADA Security Issues.

Deregulation has resulted in a greater focus on efficiency and return on investment (ROI) rather than on security. Industry consolidation, increased competition, and low profit margins in some sectors have reduced investment in technology and production upgrades. Utilities are now operating closer and closer to their limits as they attempt to keep up with growing energy demands. The minimal reserve capacity, such as in the electric utility industry, has resulted in systems that are less resilient to accidents and attacks.

Obviously, corporate management officers are truly concerned about the safety of their country and the nation's critical infrastructure, but, when they have to make budgetary and commitment decisions for their own organizations, these security concerns can be easily superseded by multiple economic, cultural, and financial issues. Budget meetings revolve around maximizing profits while juggling other pressing investments

required to out-perform lower cost competition, fund deferred maintenance, and achieve harmony between conflicting institutional cultures and priorities [1].

Most senior managers of utility companies view security costs as a competitive economic issue. They do not see a market incentive for spending large amounts of capital on information security technology. Just as some companies assert that regulations requiring expenditures for pollution controls negatively impact their bottom line, many claim that the costs of SCADA security will put them at a competitive disadvantage with companies that do not implement similar measures. In addition, many managers do not see investments in their individual organizations having much effect on the overall public welfare.

Few senior managers think of securing SCADA systems as more than just purchasing and installing hardware and software. More importantly, an organization has to invest in hiring qualified personnel, instituting an on-going training program so individuals remain current in a highly dynamic field, developing and managing flexible security policies, daily monitoring network traffic, and continuously assessing and improving security measures. Without those who can properly operate and maintain security systems, and provide the human operational understanding and on-the-fly decision-making for which no machine can adequately substitute, security hardware and software, by itself does them little good.

2.13.17.2 Recommendations from Literature.

To level the playing field, the government must develop and enforce standards for securing SCADA systems that apply to all organizations in an industry so that all the participants bear the costs equally [1].

2.13.17.3 Objections and Questions from Utilities.

A utility consists of dozens to hundreds of substations, each with many IEDS, enterprise-wide upgrades, re-programming, or replacement for IEDs and legacy systems is too costly [18].

2.13.17.4 Trust System Solutions.

The **trust system** concept recognizes the needs and financial resources of various organizations can be quite different. It focuses on minimizing cost and maximizing flexibility in implementation.

As a caveat, the **trust system** does require a particular amount of hard disk storage for its applications and performs better, especially when conducting more security functions, with faster processors and larger/faster memory, so there are some limits to the modular add-on capability without also upgrading memory, hard disk, and processing capacity.

Simulations for this thesis were conducted on a personal computer (PC) to evaluate performance by the most simple, cost-effect COTS hardware solution, however, the **trust system** would be an open software solution that can be added to any user hardware (better hardware just performs better) and interact with any existing operating system or protocol. All **trust system** modules are software programs that work either together or alone, so a company that does not need all of the **trust system** functions can simply pick-and-choose and purchase only what they need. The idea is that is the company already has a good firewall or IDS that they could interact with the **trust system** (or vice versa) to keep up to date on their discoveries and actions. In this way, the **trust system** is more of a security manager.

The same is true for a company that needs to invest in one or two modules now and add additional functionality later. **Trust system** software would be easily upgradeable by simply installing additional, add-on software modules or upgrades to existing modules.

While the **trust system** provides a security framework, companies themselves implement their own security policies and can enable, add, or tweak security thresholds and rules unique to their organization.

Until a company is able to transition fully to processor-based, IP-enabled master stations and field equipment that can implement distributed **trust system** agents on all critical nodes, the **trust system**, loaded onto separate security boxes in the network (i.e. systems, such as a **trust PC, server, or router**) provides security functionality on behalf of the limited-capacity legacy systems that send traffic across its path.

Server memory and SCSI drives are recommended for extended operation, even in workstations and HMIs. When defining hardware purchase requirements, organizations should plan for excess memory and disk space upfront or ensure servers and workstations have plenty of expansion capacity to accommodate future performance enhancing memory, disk, and processor upgrades as technology improves and costs decrease.

2.13.18 Information Security Expertise and Responsibility.

2.13.18.1 SCADA Security Issues.

SCADA maintenance and administration are fractured with no single cyber security overseer [18]. SCADA and distributed control systems have traditionally been

the exclusive domain of electrical engineers. With the transition to standard hardware and software platforms, Internet protocols, and connections to corporate enterprise networks, IT personnel are becoming more involved with SCADA systems. Thus, there are conflicting cultures and priorities and differing stances on implementing IDSs, firewalls, authentication, and encryption [1].

Operators, power engineers, and management often try to guard their operational systems from IT personnel, who are the smartest on network security, because of assumptions they can't or won't understand operational impacts or that they will disrupt working operational capabilities for unnecessary security restrictions [18].

Unfortunately, SCADA security discussions typically devolve to "SCADA personnel largely working in a vacuum and telling the IT security community that they don't understand SCADA protocols" [18].

Great research is being done on both sides (i.e. IT security engineers and SCADA engineers), but the SCADA security torch continues to be carried by a handful of people focused only on the control system environment.

2.13.18.2 Recommendations from Literature.

IT personnel must understand operations and the impact of security mechanisms to the degree that they can make them transparent to the operators and power engineers. It is important that the IT security community is involved. It doesn't take much work for them to extend their existing body of knowledge in order to take some of the increasing burden off of power engineers and operators.

It is important for both sides to understand that SCADA network security discussions are the same as any other security discussion (i.e. operating systems, services,

web-based, XML, SNMP, TCP/IP, UDP/IP, etc.) but with different message formats and very strict time constraints. While the overall concepts are the same, it is important to understand that the applications and priorities are going to be slightly different in SCADA security versus IT security.

2.13.18.3 Objections and Questions from Utilities.

Power engineering is already a relatively small subset of electrical engineering and power engineers who are interested in, let alone like, understand, or are enthralled by information technology are even harder to come buy [18].

IT personnel don't understand SCADA systems, protocols, and operational requirements and may degrade or cause downtime in the 24/7 operations with their restrictive policies and security measures.

2.13.18.4 Potential Solutions.

It is recommended that each organization assign an Information Systems Security Officer (ISSO) to maintain and monitor all security policies for control, office, and engineering networks.

Assign a small, qualified security team (reporting to the ISSO) with administration privileges over security systems to objectively evaluate security posture and effectiveness; update security systems, signatures, and rules; and analyze suspicious network events, logs, and security alerts.

Power engineers must know enough about information technology and information security to be effective in adhering to security policies and assisting in defining workable system security requirements, solutions, and operating procedures.

Instead of arguing over who is more qualified to offer architecture recommendations and protocol design changes, all parties (i.e. SCADA, IT, and network security) must recognize that they each bring important skillsets and insight to the table. They must form a team that gains familiarity with each other's requirements and works together to define, implement, and maintain a workable security policy.

2.13.19 Security Training.

2.13.19.1 SCADA Security Challenges.

Companies have little or no investment in security training.

2.13.19.2 Recommendations from Literature.

Management should ensure design of a specific, documented and testable security training plan for each user role and require initial training and qualification, quarterly updates, and annual refreshers. Integrate security scenarios and responses into regular exercises. A certification program should be established for security analysts and all who will perform administrative functions affecting network performance, security, and safety.

2.13.19.3 Objections and Questions from Utilities.

Financial constraints leave little money for expensive information security courses.

2.13.19.4 Trust System Solutions.

Organizations should develop, by experience, in-house experts (or area/regional support teams) that will continuously document lessons learned, best practices, and provide tech support and training for the benefit of all employees.

Detailed traffic captures and suspicious event parameters, logged by the **trust system,** are no-cost, and can be used to train employees to recognize regular traffic patterns, signatures and history of the most common and most dangerous suspicious events, and proper **trust system** responses. A spare **trust system** can be used offline connected to a single laptop (attacker) as a training simulator or test platform by which the laptop can launch attacks or requests to see how the **trust system** will respond, as configured by the company's security policy. This simulator configuration provides an interactive learning and testing environment.

2.14 Chapter Summary

Overall, the majority of the SCADA community has been quick to embrace the transition to IP-based standards. In some cases it has already begun to adopt the IT business practices of non-SCADA corporations, such as connecting the corporate management LAN to the operational (i.e. production) network for updates and improved communications. We have also seen the continuation of remote connections for administrators and operators, now over Internet and telephone connections, for business efficiency. In all of these endeavors, it appears that security has taken more or less a back seat to functionality in SCADA design, instead of being considered in parallel at the outset. As a result, it has been temporarily ignored or passively entertained but mainly left to vendors and research labs to figure out and recommend as an after-the-fact configuration, if necessary, while IP-standards documentation, testing, and deployment is already underway without clearly defined security standards, policies, strategies, and technical support at all levels of the community.

It is difficult to defend even the most conscientiously monitored IP networks with highly trained analysts, especially when the ability to dictate and monitor every crucial update and configuration of installations throughout the US is not available and determined attackers are constantly crafting and testing new ways to steal information or disrupt operations. Even with its best efforts, a large, highly targeted company can expect multiple system compromises each year, and those are just the ones they catch. As a move toward an interconnected Utility Intranet is realized, a US-wide operational utility network, with some potential overseas offices, will only be as secure as the least secure organization within it. Since these companies are currently privately owned with little hierarchical oversight or technical support, the chances for non-secure practices and backdoors from Internet and telephone connections by even a single company to allow viruses, worms, DoS, Trojans, sniffers, and rootkits to filter into the network and wait for the next opportunity to infect and disrupt SCADA operations is only a matter of time and nearly impossible to prevent.

Although non-SCADA Internet-connected corporations and home computers are low hanging fruit for amateur script kiddies who have a plethora of online tactics, techniques, and procedures for manipulating cyberspace, others desire more of a challenge or are simply greedy enough to look for alternative targets of opportunity in order to improve there situation in life. They are undaunted to make the extra effort to understand utility SCADA and emergency management technologies and dream of ways to defeat them. Then there are those terrorist-sponsored individuals, groups, and organizations who simply live to control through fear, death, propaganda, and anarchy or nation-state military or paramilitary units that work to maintain a technological advantage

and have been gathering and testing every bit of communications network intelligence they can find, steal, or buy in preparation for the chance to take down or disrupt an American utility when it is advantageous to their cause.

As computer processors have increased in speed, we have begun to quickly exceed the ability to humanly react fast enough to escalating, well-planned network attacks and must rely more and more on automated security technologies to detect and respond accurately in order to prevent damage, disruption, or loss and buy time for the decisions and actions that only humans can make to maintain continuity of operations. This will require ever-increasing technological capabilities and refreshes, regular system updates and security checks, active security monitoring of network traffic, and regular training to detect and respond to network reconnaissance and attacks before or at least as they happen, and recover in the event of successful attacks.

Operators and administrators will have to know their own systems and their vulnerabilities as well as the impact of network transport and security systems in the same network. IT administrators will have to understand the specific security measures that apply to SCADA networks versus more delay-tolerant office LANs, and security analysts will have to be employed and trained to provide low-level expertise and integration of security mechanisms that complement and don't hinder their company's operations. Communication and between all of these roles is critical. Above all of these cultural changes, will be a pervasive and necessary lack of trust, because any IP packet that attempts to enter the network or is received by one of its systems could be from a compromised system elsewhere in the network.

Fortunately, the cost for increased commercial-off-the-shelf (COTS) technologies remains the same or decreases over time making it possible for companies to continuously improve their network functionality, efficiency, and security. Yet, technology means nothing if the humans, especially at the lowest level, are not well trained to select, install, configure, maintain, analyze, and improve it, since they are the one's who best know their own operations and are responsible each and every day to supply the basic needs of millions of American citizens.

III. Methodology

3.1 Chapter Overview

The purpose of this chapter is twofold. First, it will explain the concept and real-world applications of a **trust system,** as an integrated suite of flexible and configurable trust-based security mechanisms with pick-and-choose, add-on security capabilities for existing and future IP-based SCADA, real-time control, and Emergency Management networks. Second, this chapter will explain the models and scenarios developed to simulate communications that would be present in a collaborative control network relying upon non-real-time transport protocols such as UDP and TCP. The purpose of the simulations was to implement the proposed functionality of these delay-inducing security mechanisms and to estimate the impact of the induced delay on utility control communications. The goal of the experiments was to evaluate the hypothesis that stringent security mechanisms can be implemented in SCADA environments, using a mix of non-real-time protocols for communication and wide-area information sharing, while meeting strict real-time thresholds for emergency response.

3.2 The Trust System Concept

3.2.1 What the Trust System Is.

The concept of a **trust system** is to provide a non-proprietary system, system of systems, or software agents that plug into an existing network, somewhat transparently, to perform the functions of correlating data and identifying risk levels for corresponding events and status updates that point to negative impacts on utility services. The **trust**

system, at its core, is a software agent performing active security analysis and response. In a network where nodes have sufficient unused hard drive capacity, memory, and processing power, the agent would be loaded directly onto the node and provide an active interface between incoming messages and the node's code, data, and applications, similar to other software firewalls. It could also be set to monitor outgoing messages

3.2.2 What the Trust System Does.

The **trust system** intercepts status messages or commands from network nodes destined for the master control station or other nodes in the network. For companies with some legacy nodes, this would require protocol gateway plug-ins for the **trust system** to interpret and analyze packets delivered in different protocols and formats.

The **trust system** validates input and identifies security risks or bad data, initiating appropriate alerts and response actions. It then assigns data types to each of the good data elements in each message. Next, it determines if the recipient is authorized to read all of the data types in the message, particularly when a recipient is external to the company (i.e. not a company employee or source IP address outside the company network). If not, it sanitizes the parts of the message that are not allowed to be passed to the recipient before forwarding it or simply deletes the message altogether. Finally, good data elements (i.e. those that appear legitimate because they pass all checks for corruption and valid data ranges/values) are transferred to database systems for company Intranet display and to archiving systems for historical and trend analysis. The archived data is then viewable and accessible only to those with the appropriate credentials, need to know, and rights to access those data elements.

Trust systems monitor communications within the company's SCADA network and between the company's SCADA network and other organizational enclaves in the Utility Intranet. The same concept can be applied to monitoring the company's office LAN, DMZ, and Internet VPN connections, which should not be connected to the SCADA network, if at all possible.

System status updates are communicated between SCADA and emergency management systems in a series of messages with potentially tens, hundreds, or thousands of data elements per message. Not all of the data will be needed by every system or by every user that views the correlated status summaries. Some data may be strictly for historical analysis or accountability reasons in the event of a resource, security, or safety incident. Other data may relate to operational or financial performance and be considered company-sensitive and limited in release.

Because of the wide range of users and systems involved in interconnected utility operations that need to share data in an effort to increase situational awareness and prevent emergency situations, there is also a need to restrict what data is readable, depending on the need-to-know of a user and the sender's trust that the recipient is who they say they are and is not going to share the data with someone who does not have the need to know. Hence the reason for assigning *data types* (e.g. operational, financial, network, etc.) and releasability *caveats* (i.e. company-sensitive, company-restricted, no vendors, no competitors, etc.) to all *data elements* (e.g. values, variables, entries, files, folders, etc.) in the network. The *data type* and *caveat* must match the *role* and *access operations* (i.e. rights, such as read, write, copy, etc.) assigned to a specific user, in order

for that user to perform that specific *access operation* on that specific *data element*. This is defined and enforced in the **trust system** *Access Control Matrix (ACM).*

3.2.3 Flexibility in Implementation of the Trust System

In today's heterogeneous utility networks, where most legacy nodes are unable to provide the resources needed by a loaded agent, the trust system is a flexible solution that can be implemented in multiple different ways, depending on the company's current architecture and needs, without jeopardizing existing control functions.

For legacy networks, the **trust system** can be implemented as a **trust box** (i.e. a server in front of a group of unprotected nodes that screens incoming packets and generates security alerts to a security server and security analyst workstations). The **trust box** would also act as an encryption gateway, maintaining secure tunnels with the **trust box** in front of the master control station server and other servers with which the nodes it protects must communicate. This thesis investigates the functionality of a standard PC (desktop) hosting the **trust system** software. Consider, instead, if one or two distributed, high-speed cluster servers with processing speeds close to 200 gigaflops per second were assigned to the task. At the time of this writing, priced between $20,000 and $50,000 [27], a cluster server is not extravagantly expensive compared to the threat of lost revenue and respect that might result from a security incident in the operational control network.

For the sake of flexibility and cost savings to utility companies, these various **trust system** functions would be implemented as separate software plug-in capabilities that could each be purchased separately, to perform as standalone capabilities, or installed

with other **trust system** modules for more robust capabilities. The loose coupling of pick-and-choose options for the plug-in **trust system** modules (i.e. simple software installs) makes it financially palpable, scalable, and easily upgradeable over time

The **trust system** can also be implemented as a system of systems. Every function of the proposed **trust system** may not be needed by every organization. If a company already has a good firewall and intrusion detection systems, it would not necessarily need to purchase these plug-in capabilities for their **trust system**. In this case, the **trust system** can theoretically interact with the data provided by these existing network devices and complement them by providing its own unique capabilities in a synchronized conglomerate of distributed systems. Key to the effectiveness of such a scheme would be an **alert correlator** to deconflict duplications of both data and alert traffic and to interpret and consolidate the protocols and information before presenting an integrated picture to a security analyst, network administrator, or engineer's screen.

Since every utility and utility company's network will be different, each individual company must perform its own individual network needs assessment and simulation to determine security and financial feasibility and identify weak points and points of failure in its own network design. It is then up to that company to implement the best network design with the level of redundancy and defense-in-depth that is economically feasible and corresponds to due diligence in protecting national infrastructure and utility services. The **trust system**'s cost-effective, modular acquisition and employment options are well-suited for meeting a wide range of implementation requirements. A logo for the functions supported by the **trust system** is depicted in Figure 9.

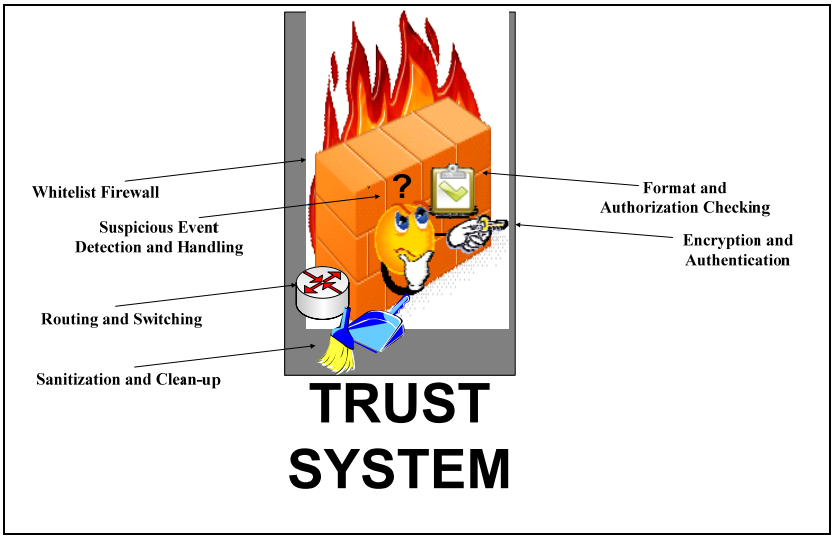


Figure 9. Trust System Logo with Capabilities Summary

3.2.4 Passive vs. Active Mode Implementations.

Trust systems may be implemented in an *active (or router) mode*, where the **trust system** is implemented on a hardware device inline with all communications between the SCADA master control station and the nodes it controls and between the company's SCADA network and its outgoing connection to the rest of the Utility Intranet as depicted in Figure 10. This device may be a specialized **trust box** or a **trust-enabled router** which is also responsible for network routing of all packets on the link. In this implementation, it may itself stop or correct malformed or malicious packets that it inspects. The advantage is the ability to block malicious traffic immediately as it's detected. A block is constituted by a DENY entry being added to the *firewall rules* (for a specific IP address, interface, protocol, port, and/or message type combination) or a lowered *trust level* and/or *access level* (for a specific user or system). The disadvantage is

that the hardware device is a potential single-point of failure on that link. If the entire hardware device fails, the link is down; however, alternate or redundant routes can alleviate this problem, as in any IP network. The simulations and experiments for this thesis assume a **trust system** in *active mode* to demonstrate its blocking functionality.

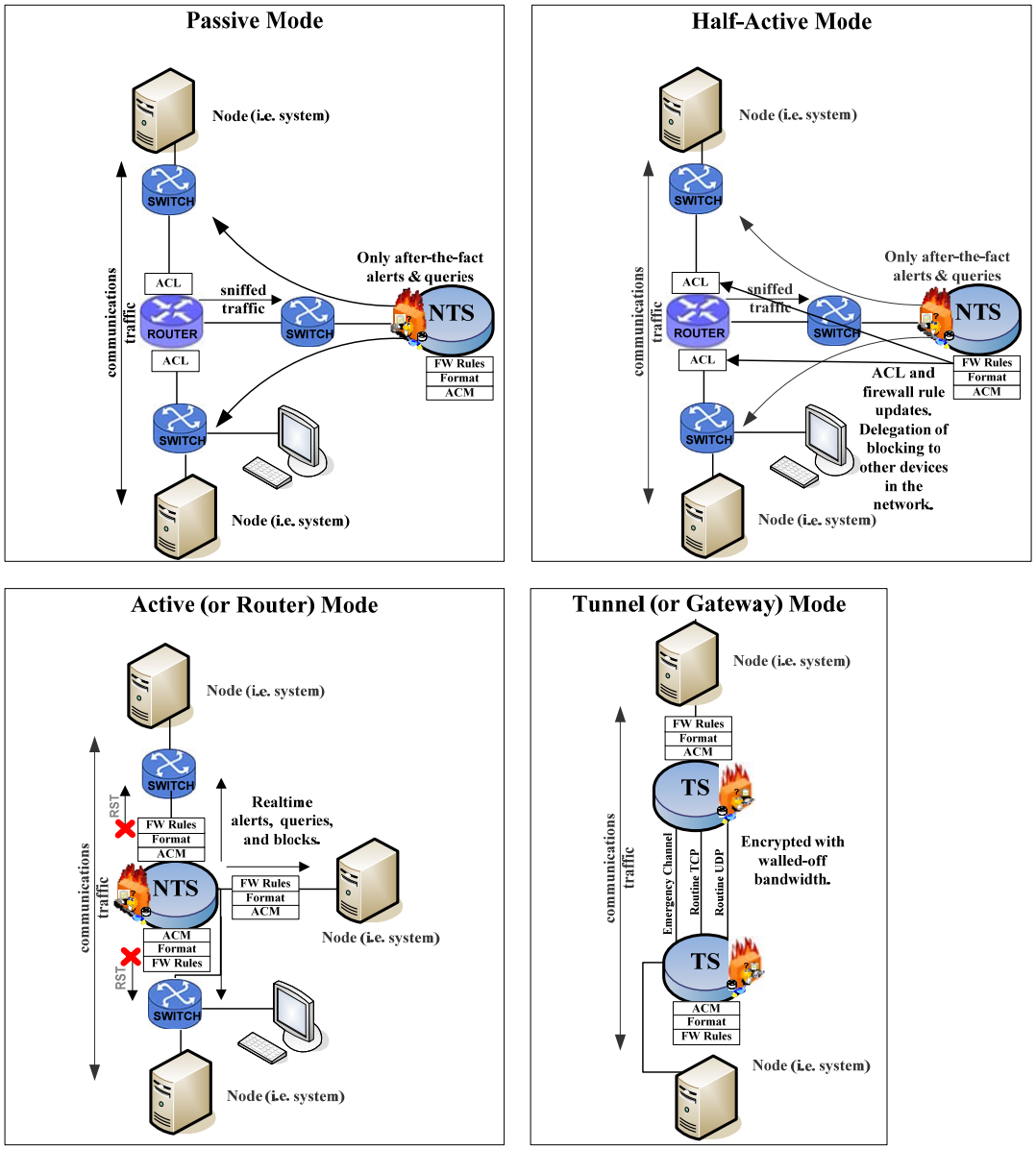


Figure 10. Trust System Modes and Configuration Options.

In *passive mode*, a **trust device** is connected to a hub or switch on the link between the SCADA master control station and the nodes it controls and between the company's SCADA network and its outgoing connection to the rest of the Utility Intranet as depicted in Figure 10. In this case, the **trust system** simply sniffs packets as they pass by, saves a copy to analyze, and alerts if a security or trust rule has been broken or has the potential of being broken. The advantage to this mode is that the **trust device** is not in-line with the communications, so a failed **trust system** does not block the communications link. The disadvantage is that the **trust device** cannot stop, only report, malicious packets it sees and it will do so after the packet has passed the **trust system** and is likely to already be delivered to the intended recipient.

A way to implement blocking with a *passive mode* **trust system** is for the **trust system** to interact with a separate firewall or router ACL to block further packets by source_IP, interface, transport protocol, and message type combinations but there will still be some delay and a chance that one malicious packet will be delivered to its destination before other similar packets are blocked. This is also known as *half-active mode*.

In the case where some nodes cannot be loaded with **nodal trust agents** or afford the clock cycles required for encryption, the trust system may be implemented in either passive or active *gateway mode* as depicted in Figure 10. In this implementation, **trust system** boxes or routers provide firewall and other security features for the nodes behind them. They also create an encryption gateway between themselves to protect communications between **trust systems**. This mode can also be referred to as *tunnel mode*, since IPsec would be implemented in IPsec tunnel mode.

Figure 11 depicts peer-to-peer and master-slave configurations of **trust systems**.

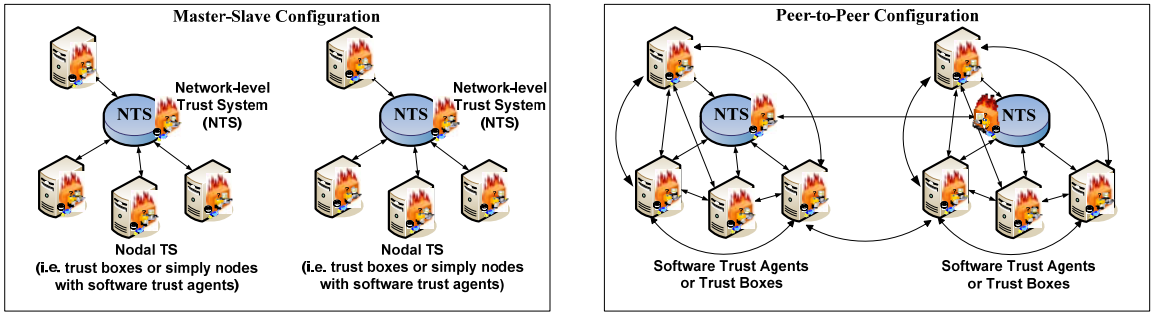


Figure 11. Trust System Configurations

3.3 Real-world Applications for the Trust System

3.3.1 Inter-Company and Inter-Area Protection.

While not the norm in present day SCADA architectures, the concept of a Utility Intranet can make possible unprecedented situational awareness between utility companies, control and engineering centers, and neighboring utility control areas. Sharing of automatic status updates will enable near-real-time situational awareness for trusted ISOs, control authorities, or reliability coordinators who can, in turn, direct actions to prevent catastrophic overloads or underloads and ensure equity of resources within their areas of responsibility and oversight.

The **trust system**, when placed at strategic locations such as connections between adjacent utility companies, outgoing connections from utility companies to area control and engineering centers, tie lines between control and transmission areas, specifically between control centers and between engineering centers, and between reliability coordinators in different ISOs provides low-cost network security to traditional SCADA

networks with their mix of legacy, proprietary systems and protocols and newer standards-based solutions. Appendix A illustrates the proposed hierarchical structure of information sharing, support, and command and control. Obviously, an understanding of appropriate and inappropriate information flows (e.g. who, what, when, where, and how) is critical to network security planning and design in general but more so in the design and configuration.

Just as status updates in electric power utilities are sent from field equipment via IEDs, RTUs, or PLCs and on to SCADA master control stations every few seconds, or even milliseconds, either the same updates, a subset of those updates (i.e. only significant changes from the previous update), or a summary report can be easily forwarded on to connected control area authorities and adjacent electric utility companies on the Utility Intranet. When substation automation applications do not support this forwarding, the **trust system** can be configured to initiate this on their behalf whenever it sees a qualifying message cross its path.

Situational updates shared between adjacent utility companies will facilitate automatic recognition of changing conditions that might affect their levels of generation or transmission. Neighboring companies that receive reliable status updates will have earliest warning of creeping load changes versus current power generation levels. Early warning and impact realization will prompt timely decisions on the right combination of load shedding and adjusted generation rates necessary to absorb or make up for the rapid changes in power flows from adjacent companies. This will aid private companies in preserving service to their own customers while preventing potential blackouts and

alleviating the associated financial costs and loss of public trust that can result from outages.

Monitoring systems in neighboring Utility Company Operations Centers can then automatically update their operational picture with a wider perspective of power capabilities and emergencies in the immediate area while area controllers, ISOs, and reliability coordinators would have a complete picture of currently segmented utility operations owned by private companies. Control Areas can forward area-wide status updates and emergency notifications to a Regional Utility Operations Center and to their adjacent Area Operations Centers for improved regional situational awareness. Appendix B depicts the cross-flow of information within and between various Utility Intranet enclaves.

The **trust system** can facilitate this message forwarding right now between utility company networks and control areas for which numerous existing SCADA applications do not cooperate in this manner. When the **trust system** inspects and then reassembles a packet, it can check its own *ACM* for the list of recipients external to the company network who are authorized to receive that message type, translate the message into a new packet with the proper format understandable by those receivers, and then forward the original message internally, as normal, and the new message to those external destination IP addresses.

In the event a neighbor noticed a spike or increasingly dangerous situation, in what amounts to a macroscopic version of the local *neighbor_trip*, *backup_trip*, and *intertrip* messages that are proposed to occur through embedded software agents within a single company, a similar trip message might be generated from an adjacent neighbor

company to ensure the owning company is aware of the impending emergency and can approve or disapprove the requested action, even if its own systems are malfunctioning.

The Northeast Blackout of August 14, 2003, the largest in North American history, illustrated this very scenario. Due primarily to malfunction, accidental shutdown, and internal miscommunication, systems failed to report problems to the control center within one company, which later denied any need for concern when it received phone calls from a neighbor company warning they had indications of abnormal readings along their shared borders of the transmission grid [28].

The controllers continued to operate, blind to the actual situation, for hours before the cumulative affect (there were also power lines that had sagged in the heat to where they contacted overgrown trees) created a system-wide point-of-no-return. A series of cascading transmission line outages traveled through Ohio, around the Great Lakes in Michigan, through Canada, and into New York State in only ten seconds [9]. Once it began, the blackout that cascaded from Cleveland to the Northeastern United States took just seven minutes total [9]. Nearly 10 million people in the province of Ontario (one-third of the Canadian population) were without power and 40 million people in eight U.S. states (one-seventh of U.S. population U.S.). The financial losses due to the outage were estimated at $6 billion [29]. In a highly reliable and secure environment, trip messages from one company to another, especially from a trusted partner that has the interests of both companies at heart, might be trusted to automatically trip breakers in another company. This would require a complete culture change from the way electric utilities are currently operated. Today such company to company initiated actions would likely be rejected for fear of false trips due to technological or human errors, outside

hackers/crackers/attackers, or corporate sabotage/espionage. This is where the **trust system** will assist in validating traffic and providing assurance to utility managers.

Those companies hesitant to allow automatic actions to their systems by neighbor companies would be more amenable to the option to approve or deny the trip requests first or to allow neutral ISOs and reliability coordinators the ability to send commands to company SCADA systems or breakers in reaction to a growing power outage seen within their control area. It is also conceivable that the control area authorities that recognize such a situation could contact the company to direct actions and, if granted proper permissions, initiate breaker trips remotely when the required reaction time does not allow for coordination. Either way, shared electronic status readings are more credible than just word of mouth, and a master control station receiving conflicting reports from its own substations and its neighbor's control center could alarm to warn the operator and would have prevented the 2003 blackout.

In the future, such security mechanisms as those investigated in the **trust system** simulations, when layered over ever-increasing bandwidth and connectivity between utility organizations, would enable the creation and operation of Regional Utility Operations (or Control and Security) Centers to ensure integrity and fair use of the power grid and a utility-specific capability for network security response, technical assistance, and law enforcement liaison for companies within its regional span of control.

3.3.2 Internal Traffic Protection.

Internal to a utility company, the **trust system** provides firewall functionality between SCADA nodes and between the SCADA network and any connected office

environments, restricting traffic only to authorized protocols and message types, while compensating for bandwidth congestion and enforcing prioritization of packets. It can ensure the fastest reliable delivery of important real-time and emergency traffic, unhindered by delay-tolerant background traffic such as routine e-mails and Intranet web browsing that might be present simultaneously.

The **trust system** envisioned will not only enable the sharing of automatic power flow status and corrections but would guard security enclaves and commercial communities of interest, protecting company-sensitive data from access by or accidental transmission to competitors, vendors, and other entities accessing the Utility Intranet that do not necessarily have the need-to-know, based on their duty position, or role.

3.3.3 Preventing Single Points of Failure.

The goal of the **trust system** is to be completely transparent to the controlled utility process and robust in the face of adversity. The **trust system** is meant to be layered over the existing process and communications scheme through adding, in a sense, optional, independent security-layers to the network stack. Even if the entire **trust system** was disabled, it should be completely decoupled from the industrial process such that nothing in industrial operations would break or slow down.

If a single **trust system agent** at a node (i.e. a **nodal trust system**) which inspects messages attempting to access enter through a device interface (at the physical and network layers) and monitors access operations attempts (e.g. read, write, copy, etc.) at the application level, should it fail, should not prevent functionality of the node in sending and receiving communications. The industrial operations would perform as

always without skipping a beat, except the layered on security measures would no longer be in effect. Review of the **trust system** generated security logs would indicate a gap in the regular entries expected (i.e. at a point in time where it is known that regular traffic was flowing across the link and **trust system** analysis detail should be present). This absence of log entries for a significant period would be sufficient to quickly verify that the **trust system** is not functioning.

The best implementation of the **trust system** within an organization is in a distributed manner with a **network-level trust system (NTS)** as an overseer. Each distributed **trust system** would be independent, but keep the **NTS** up to date so that it can maintain the big picture for the sake of correlating related events in multiple parts of the network. In the face of lost communications with the **NTS,** a **trust system** agent loaded on a node, referred to as a **nodal trust system**, could operate on its own to protect its node and keep its neighbor **nodal trust systems** up to date, collaborating to ensure security in their interactive node-to-node communications. The **NTS** might either have another trust system in the network pre-defined as an alternate, should it fail, or in the case of a leaderless situation, **nodal trust systems** might hold an election to designate a new **NTS** with the greatest resources available (above a minimum requirement) for that function.

3.4 Trust System Concepts and Terminology

3.4.1 Roles and Categories.

There are many different types of users requiring access to various SCADA and IT system data within the interconnected Utility Intranet. Example user *roles*, for the

purposes of this paper, are listed in Table 8. Several different *roles* may belong to the same *category* of users (e.g. for those within the same organization) that requires distinction as a group for the purposes of releasability.

Table 8. Example Roles for Various Utility Intranet Users

Category	Trust Level	Role
IED_vendor	-2	vendor_sales_rep
	-1	vendor_programmer
	-1	vendor_engineer
my_company	0	SCADA_operator
	0	SCADA_administrator
	0	power_systems_engineer
	0	SCADA_maintenance
	0	IT_administrator
	0	network_security_analyst
	0	company_ISSO
	0	management
	0	planner
	-1	dispatcher_in_training
	0	my_master_station
trusted_power_grid_organizations	0	area_controller
	0	area_engineer
	0	area_ISSO
	0	independent_system_operator
	0	reliability_coordinator
	0	regional_transmission_operator
	0	interconnection_controller
	0	adjacent1_master_station
	0	adjacent1_trust_system
	0	adjacent1_email_server
	0	adjacent1_SCADA_operator
adjacent_competitor_company	-1	adjacent2_master_station
	0	adjacent2_email_server
	-2	adjacent2_SCADA_operator

A *role* could be arbitrarily defined to describe any group of individuals. For this thesis, the *role* has been specifically defined as a job position. This *role*-based access may vary over time for a particular individual, depending on the individual's assigned tasks, the data and tools they need to know and use, and the level of trust the company has in their experience, performance, and current level of training.

3.4.2 Data Elements and Rights.

Each user *role* is associated with a set of rights (i.e. permissions) for *access operations* (e.g. view, read, write, copy, delete, move, execute, etc.) on specific elements of data and code available on the various systems in the network. Potential *data types* that can be found on a Utility Intranet might include those listed in Table 9.

Table 9. Example Data Types

Abbreviation	Data Type
OC	Operations-specific (SCADA) Code
DC	Development Code
DD	Developmental and Test Data
OD	Operations Data
SD	SCADA-specific Data
ND	Network Data (IT)
NC	Network Code and Configurations (IT)
ED	Emergency Management Data
OA	Office Automation and Common Drives
LG	Logs
IW	Internal (Intranet) Web Pages
IC	Internal (Intranet) Web Code
XW	External (Internet) Web Pages
XC	External (Internet) Web Code
SE	Security Data and Security Code
CT	Coding Tools

Potential data access operations by Utility Intranet users might include those listed in Table 10.

Table 10. Example Access Operations

Abbreviation	Access Operation
r	read/view/open
c	copy
w	write
a	append
p	paste
m	move/cut
d	delete
s	save
x	execute

3.4.3 Access Levels.

An *access level* determines what data a user or device should be allowed to receive, see, and interact with. More specifically, an individual's *access level* is dependent upon two factors: an individual's *role* and the *Access Credentials Control Number (ACCN),* an integer (0-4), calculated from the number and reliability of successful logon credentials presented for logon to the network.

While very minute internal failures, outages, or limitations that have no effect on other companies or services provided to customers do not need to be known by competitors, when an event (or factors) are detected that could contribute to a widespread (outside of the company) emergency, some of these data elements, previously kept internal to the company, may need to be communicated. In this event, either the *access level* of the user or device to be informed must temporarily increase or the *access level* of the data element must temporarily be decreased to make more "company sensitive" information available or releasable. This also means there must be a mechanism to track *access level* state changes and a method to revert to the original level once the emergency situation is resolved. The easiest way to deal with this is at the **trust system** when assigning *access caveats* to *data elements*. Normally, some *data elements* might have company-sensitive or company-restricted *caveats* assigned. Data given a *restricted caveat* can never be sent to an external agency that is not authorized to see this *caveat* (financial reports might be an example). Sensitive data (*caveat* = sensitive) may not be released, in general, to external organizations, except in certain circumstances. If, for instance, all of the following conditions are met: emergency = true & external_impact = possible & caveat = sensitive, then the data is releasable to a particular list of authorized IP

addresses. There is no need for raising or lowering *access levels* of users or *access caveats* for data because once data is released it is then known and present outside the company network and cannot be taken back. Another tag for released = true could also be set with a release date traceable to the release list so it is always known to whom and when sensitive information elements were released. If not released, then released = false and it is understood that this information has never (deliberately and electronically) been made available to anyone outside the organization without the need to know.

Access levels and *access caveats* are not the same as security classifications or security clearances assigned to government data and personnel, respectively. Data of varying security classifications (e.g. Confidential, Secret, Top Secret) traditionally has not been maintained on the same physical network or connected networks and must maintain physical separation. Although this has been the procedure to date, in the future, technology may provide strict logical separation control, for data of different classifications on the same storage media, that prevents any chance of remanence, bleed-over, tapping, theft or inadvertent access by anyone not holding the appropriate security clearance for the data they attempt to view or access. Research in this area is not the purpose of this thesis.

Data, folders, and files could have an associated *data type* as well as a release restriction, or *access caveat*, such as "company sensitive" applied to them. In this case, authorized access to both parameters and the proper *read* and *execute* rights would be required to view and use the folder, file, or data.

3.4.4 Trust Levels.

In addition to *access levels*, there are *trust levels* for both users and systems. *Trust levels* designated for the purposes of this thesis are depicted in Table 11.

Table 11. Example Trust Levels

Trust Level	Degree	Example
-0	Full trust (i.e. High)	Control Area (CA) employee, Reliability Coordinator (RC), or Independent Systems Operator (ISO)
-1	Cautious trust (i.e. Med)	Employee of a partner company
-2	Suspicious (i.e. Low)	Employee of a partially-trusted competitor company
-3	Untrusted (i.e. None)	Employee of an untrusted competitor company or other untrusted source

The *trust level* is an integer to be subtracted from (or a negative integer that is added to) the *ACCN* (a positive integer from 0 to 4) of a user or system. A *trust level* of 0 is good and a trust level of -1 to -3 means something has occurred to cause the *trust system* to begin regarding further traffic from a particular source with greater suspicion. A lowered *trust level* decreases the *ACCN*, and, therefore, the *access level* of the user or system.

If the **trust system** detects false or corrupted data from a node (e.g. a malformed or spoofed packet, DoS attack, or corrupted data), it must decide if it should initiate a maintenance trouble ticket or security alert, lower the *trust level* for that system, initiate a switchover to a redundant backup system (if available), or change its priorities for primary and backup sources of information for particular data elements that were originally supplied by that system?

3.4.5 Multi-level Access.

The assignment of *access levels* and rights over data elements can prevent unauthorized disclosure of sensitive data or even the existence of such data for multiple users at different *access levels*.

Each individual's account is tied to specific rights (i.e. permissions) over specific types of data by its assigned *role* (and category, if applicable). One right would be for reading operational status message data elements. Another might be for executing a diagnostic program. This applies not only to a utility company's employees and systems but to partners and competitors, which would normally have no authority to initiate actions on that company's systems. User and system *roles* prevent a user from viewing data, files, folders, or systems in the network for which they are not authorized. Those they can read, are prevented from modification if the user or system does not have authorization according to the *ACM*.

Permissions for writing and executing code or initiating actions on or by the utility company's systems (tripping a breaker, increasing/decreasing generation or load, shutting down, switching over from primary to backup, etc.) require specific *access levels*. Rights (i.e. permissions) not only apply to accessing *data elements* in messages and in folders but also to accessing systems and sections of code (used by the system), etc.

3.5 Trust System Modules Overview

Appendix C illustrates the primary functions of the **trust system** in a flowchart of operations. White blocks indicate functions simulated to illustrate the **trust system**

capabilities. Gray objects indicate servers that are important to the SCADA network and comprehensive security monitoring that are assumed, but not simulated, in experiments for this thesis. Gray diagonally shaded blocks indicate **trust system** functions not necessary to be simulated in the research for this thesis yet important to the overall **trust system** capability.

3.6 Firewall Rules Module

3.6.1 Firewall Rules Check.

The **trust system** is configured with signatures for authorized communications traffic, similar to a firewall whitelist. This is the opposite approach to blacklist firewall rules, which specify unauthorized traffic. The **trust system** *firewall rules* filter incoming packets on the combination of source and destination IP pairs, message type allowed, protocol, source and destination ports, and **trust system** interface receiving the packet. In the *firewall rules* depicted in Table 12, only port 500 (IPsec) is allowed, to ensure all communications are encrypted.

Table 12. Firewall Rules and Outbound Routing Table Excerpt

Rule	SourceIP	SourceName	DestinationIP	DestinationName	Port	Prot	Message Type	In	Out	Denied	Distance
1	2001:DC98:5634:2110:BD1C:BA89:7325:0239	mg_IED_239	2001:DC98:5634:2110:BD1C:BA89:7325:3200	mg_SCADA_master_station	500	UDP	control	1	2	FALSE	1050
2	2001:DC98:5634:2110:BD1C:BA89:7325:0239	mg_IED_239	2001:DC98:5634:2110:BD1C:BA89:7325:3200	mg_SCADA_master_station	500	TCP	control	1	2	FALSE	1050
3	2001:DC98:5634:2110:BD1C:BA89:7325:0239	mg_IED_239	2001:DC98:5634:2110:BD1C:BA89:7325:3200	mg_SCADA_master_station	500	UDP	status	1	2	FALSE	1050
4	2001:DC98:5634:2110:BD1C:BA89:7325:0239	mg_IED_239	2001:DC98:5634:2110:BD1C:BA89:7325:3200	mg_SCADA_master_station	500	TCP	status	1	2	FALSE	1050
5	2001:DC98:5634:2110:BD1C:BA89:7325:0239	mg_IED_239	2001:DC98:5634:2110:BD1C:BA89:7325:3200	mg_SCADA_master_station	500	UDP	intertrip	1	2	FALSE	1050
6	2001:DC98:5634:2110:BD1C:BA89:7325:0239	mg_IED_239	2001:DC98:5634:2110:BD1C:BA89:7325:3200	mg_SCADA_master_station	500	UDP	neighbor_trip	1	2	FALSE	1050
7	2001:DC98:5634:2110:BD1C:BA89:7325:0239	mg_IED_239	2001:DC98:5634:2110:BD1C:BA89:7325:3200	mg_SCADA_master_station	500	UDP	backup_trip	1	2	FALSE	1050
8	2001:DC98:5634:2110:BD1C:BA89:7325:0239	mg_IED_239	2001:6B03:105E:A993:28CA:E7BB:A4B3:3200	mg_CA_SCADA_master_station	500	UDP	status	1	3	FALSE	301100
9	2001:DC98:5634:2110:BD1C:BA89:7325:0239	mg_IED_239	2001:6B03:105E:A993:28CA:E7BB:A4B3:3200	mg_CA_SCADA_master_station	500	TCP	status	1	3	FALSE	301100
10	2001:DC98:5634:2110:BD1C:BA89:7325:0239	mg_IED_239	2001:DC99:31AC:9AAD:FC29:6C1A:80EA:3200	adjacent_company1_SCADA_master	500	UDP	status	1	3	FALSE	101100
11	2001:DC98:5634:2110:BD1C:BA89:7325:0239	mg_IED_239	2001:DC99:31AC:9AAD:FC29:6C1A:80EA:3200	adjacent_company1_SCADA_master	500	TCP	status	1	3	FALSE	101100
12	2001:DC98:5634:2110:BD1C:BA89:7325:0239	mg_IED_239	2001:40B1:1B4D:A52D:2CF3:1002:D228:3200	adjacent_company2_SCADA_master	500	UDP	status	1	3	FALSE	101100
13	2001:DC98:5634:2110:BD1C:BA89:7325:0239	mg_IED_239	2001:40B1:1B4D:A52D:2CF3:1002:D228:3200	adjacent_company2_SCADA_master	500	TCP	status	1	3	FALSE	101100
14	2001:A344:4DD1:F76F:D2CB:3B09:5629:1000	mg_logon_server	2001:DC98:5634:2110:BD1C:BA89:7325:4050	mg_network_trust_system	500	TCP	control	2	0	FALSE	100
15	2001:A344:4DD1:F76F:D2CB:3B09:5629:1000	mg_logon_server	2001:DC98:5634:2110:BD1C:BA89:7325:4050	mg_network_trust_system	500	TCP	logon_evaluated	2	0	FALSE	100
16	2001:DC98:5634:2110:BD1C:BA89:7325:3901	mg_SCADA_admin_workstation_1	2001:DC98:5634:2110:BD1C:BA89:7325:4050	mg_network_trust_system	500	TCP	control	2	0	FALSE	100
17	2001:DC99:31AC:9AAD:FC29:6C1A:80EA:3200	adjacent_company1_SCADA_master	2001:DC98:5634:2110:BD1C:BA89:7325:3200	mg_SCADA_master_station	500	UDP	get_status	3	2	FALSE	100
18	2001:DC99:31AC:9AAD:FC29:6C1A:80EA:3200	adjacent_company1_SCADA_master	2001:DC98:5634:2110:BD1C:BA89:7325:3200	mg_SCADA_master_station	500	UDP	query_packet	3	2	FALSE	100
19	2001:DC98:5634:2110:BD1C:BA89:7325:0239	mg_IED_239	2001:A344:4DD1:F76F:D2CB:3B09:5629:1000	mg_logon_server	500	TCP	control	1	2	FALSE	1050
20	2001:DC98:5634:2110:BD1C:BA89:7325:0239	mg_IED_239	2001:A344:4DD1:F76F:D2CB:3B09:5629:1000	mg_logon_server	500	TCP	logon_request	1	2	FALSE	2
21	2001:A344:4DD1:F76F:D2CB:3B09:5629:1000	mg_logon_server	2001:DC98:5634:2110:BD1C:BA89:7325:0239	mg_IED_239	500	TCP	control	2	1	FALSE	1050
22	2001:A344:4DD1:F76F:D2CB:3B09:5629:1000	mg_logon_server	2001:DC98:5634:2110:BD1C:BA89:7325:3901	mg_SCADA_admin_workstation_1	500	TCP	logon_denied	2	2	FALSE	100

3.6.2 Encryption Check.

All messages sent and received between systems on the SCADA network should use encryption, such as network-layer (i.e. layer 3 of the OSI model) IPsec, if it does not prevent delivery within time-constrained thresholds. The SCADA network nodes simulated this thesis were assumed to communicate only over a single encrypted source and destination port (port 500 for IPsec) for inbound and outbound messages.

Incoming messages are decrypted by the **trust system** with its private key and the sender's public key. If a message was both sent and received on port 500 and

successfully decrypted, the message passed the *firewall rules* encryption check. If, however, the message was received unencrypted, the *firewall rules* encryption check failed immediately. If the message was encrypted, but the **trust system** did not have the proper key to decrypt the message, the *firewall rules* encryption check also failed, because either the sender or the **trust system** had the wrong key.

3.6.3 Firewall Rules Scorekeeper.

If a message does not pass the *firewall rules*, the passed and failed parameters, known as *labels*, are updated in the *firewall rules scorekeeper* (FWR-SK) with the *label* name, value, and score (0=passed or 1=failed). At this point, if one of the *firewall rules labels* failed, the packet has failed the *firewall rules* check and the packet may be discarded and ignored; however, for the purposes of search into maximum delay, the **trust system** is allowed to fully analyzed every packet, forwarding it through all trust system checks (i.e. *firewall rules, format,* and *ACM* checks) before documenting all passed and failed parameters and discarding bad packets. Therefore, the updated *firewall rules scorekeeper* is forwarded to the next **trust system** module, the *format module*.

3.7 Format Module

3.7.1 Input Validation and Format Checks.

If a message passes the *firewall rules* check within the *firewall rules module*, the *firewall rules scorekeeper* is forwarded to the *format module* component of the **trust system** for format and input validation. The **trust system** differs from a standard firewall (which usually looks only at lower-level IP addresses, ports, and protocols) in that it also

inspects a message's packet and header sizes and contents as well as its application data. Data that does not meet expected types, values, or ranges is recognized by the **trust system** as a *suspicious event* and rejected.

By checking packets against expected size, field content, or data ranges, the **trust system** identifies corrupted or malicious packets. It may then auto-correct (if the proper value is known), discard, or discard and poll the sender for a resend. Such efforts can help to prevent database contamination and improper or erroneous actions by the intended recipient. The **trust system** uses the following rules to analyze packets in the scenarios designed to support this thesis:

1. Compare message payload length to the expected length for that message type
2. Compare content and values to expected values or range for that message type
3. Compare message source_IP to logged_on_IP of that system_name
4. Compare message source_IP to logged_on_IP of that username (if message was user initiated and not strictly system-to-system)

If there is an expected value for overall packet length for the message type, this is checked first. If no overall length is set for that particular message type, or if the overall length is correct, the **trust system** separates the packet into its individual components by reading and assigning each header and data value to *label* variables, specific to that message type. The variables are then compared to the expected values for that specific message type. If values are within expected ranges, or exactly match the expected value or list of values, the *label* passes the format check.

3.7.2 Format Scorekeeper.

Similar to the *firewall rules* check, a *format scorekeeper (FOR_SK)* keeps track of which labels passed or failed and is forwarded with the *FWR-SK* to the next **trust system** module, the *Access Control Matrix (ACM).*

By forwarding along all *scorekeepers* from the previous checks (IP addresses, ports, protocols, etc. in the case of the *firewall rules* check and header and payload values in the case of the *format* check) to the next module (in this case the *ACM*), that next module has documentation of the previously evaluated parameter names and values that it might need for its own checks and also, when it comes time to create a log entry of the results or to re-assemble the original packet for forwarding onto the destination, all of the data and header information is maintained

3.7.3 Data Tagging.

Before the *FOR-SK* is forwarded to the *ACM*, each *label* is tagged with a particular *data element* type and *caveat*. This tag is used by the *ACM* for access control and can also be used for data archiving in a historical database or on a server, so that later access by users and systems can be checked against a **trust system** *ACM* (either at the network level or on the database/server itself) for authorization. The *data element* type tag (i.e. OD, ND, OC, etc.) and other metadata parameters (such as creation date, original name, author, and data types/caveats; copies made by date and username, changed name and/or type/caveat; etc.) can also be carried along with the data (or file) when it is copied, pasted, modified, and attached to e-mails. This metadata can allow the **trust system** to

evaluate access authorization for attachments in e-mails or access from the LAN, even as an original document is renamed or modified over time.

3.8 Access Control Matrix (ACM) – Logon Security.

3.8.1 Initial Network Logon Control.

The **trust system** *Access Control Matrix (ACM)* maintains the current *name* (username or systemName), *role*, and *access level* entries for all authorized network users and systems that it, or the nodes it protects, may need to interact with on the network. While the values for these entries are pre-configured and usually do not change very often, the *logon ACCN, effective ACCN,* and *logon IP* are initialized at zero until a user (or system) logs on to the network. After an approved logon, the IP address from which the logon occurred is entered (i.e. the *logon IP*) and the calculated values for *logon ACCN* and *effective ACCN* are updated in the *ACM*.

When the user (or system) logs off, the values are reset to zero again. In this way, the **trust system** always knows the users (and systems) that are logged on and from which location (the logon IP), at any given time. The *trust level* is normally zero (i.e. -0) at initial logon, and is only changed by the **trust system** if it detects behavior that lowers its trust in the user (or system).

It is recommended that communications between the logon server and network trust system be via a dedicated (i.e. directly connected) and encrypted connection. The purpose of a dedicated connection is to prevent spoofed network *logon_evaluated* messages being sent to the **network trust system**. In this way, for the **trust system** to receive spoofed credential analysis, either the logon server must receive a spoofed

logon_request message, authenticate the false credentials and forward the results to the **trust system** or the logon server must be compromised in order to send incorrect messages to the **trust system**.

No group accounts should be allowed, instead all users should be required to logon and authenticate individually with username and logon credentials to gain access to network resources.

When a user (or system) attempts to logon to the network, a *logon_request* message, containing the user-supplied logon credentials (e.g. password, smart card, PIN, biometrics, etc.), is sent to the network logon server. The logon server evaluates the credentials and informs the **network trust system** of which credentials passed or failed, in a *logon_evaluated* message.

For a logon server capable of hosting a **nodal trust system** agent, the **trust system** functionality could be performed on the logon server itself and the **network trust system** informed, after-the-fact, of the results.

The **trust system** uses the analysis of successful and failed credentials, provided by the logon server, in the *logon_evaluated* message, to calculate a *logon ACCN*, using the criteria outlined in Table 13**.** The greater the number of credentials provided and the greater the reliability of those credentials, the greater the *logon ACCN (LACCN)*. For full administrator (i.e. root-level) access, at least two credentials with a total *effective ACCN (EACCN)* of at least 4 must be provided. This is to lower the possibility of simple password cracking attempts on accounts gaining high-level privileges.

Table 13. Example Logon ACCNs Assigned Based on Supplied Credentials

Credentials	Logon ACCN	Summary of Access Granted
Authorized username, incorrect password	0	No Access
Authorized username, correct password	1	Basic access, unless elevated by another logged-on user (same role) with a higher access level (effective ACCN of 2, 3, or 4)
Authorized smart card, incorrect PIN	2	Basic access, unless elevated by another logged-on user (same role) with a higher access level (effective ACCN of 3 or 4)
Authorized smart card, correct PIN or Authenticated biometrics	3	Intermediate access, unless elevated by another logged-on user (same role) with a higher access level (effective ACCN of 4)
Any combination of the above successful credentials for which the sum of the individual *logon ACCNs* is ≥ 4	4	Full (root) access

After calculating the *logon ACCN*, the **trust system** then adds the current *trust level* for the user (or system) to the *logon ACCN* to give the *effective ACCN*. The *trust level* is a negative integer indicating the level of trust that has been lost, normally -0. If the *effective ACCN* is zero, the logon is denied. If the *effective ACCN* is not zero, the **trust system** checks its *Access Control Matrix (ACM)* to determine the *role* assigned for that username. This is essentially *role*-based access. The **trust system** also determines the authorized combination of *access operations* on *data types*, based on the *effective ACCN* for that *role*.

As an example, if user John R. Smith attempts to logon with the correct username, jrsmith, and with the following credentials: the correct password, smart card with PIN, and one biometric credential (i.e. fingerprint, eye scan, voice recognition, etc.) his assigned *logon ACCN* would be calculated from Table 13 using 1+3+4=7. According to Table 13, he needs an *effective ACCN* of 4 to achieve root level access if the *ACM* affords that to his *role*. The **trust system** checks the *ACM* to find that the username jrsmith is

assigned the *role* of SCADA_engineer and a *trust level* of 0. It then calculates the *effective ACCN* (*effective ACCN = logon ACCN + trust level* = 4+0 = 4).

The minimum amount of successful credentials supplied (i.e. only username and password) provides only the lowest level of access, equivalent to a basic_office_user for the purposes of this paper. It is assumed that this might be a secretary or other office worker that has access to office automation tools and inter-office communications but no need-to-know regarding operational data or code.

3.8.2 Work Schedule Restricted Access.

The **trust system** can also check each logon attempt for certain positions against a work schedule of authorized users. This way it could detect unusual activity such as an employee coming in after hours or when they are not scheduled to work, in order to attempt something malicious. If no malicious actions were performed (e.g. someone came in on a weekend to catch up on some office work) the log entry could be verified and ignored/annotated/deleted. This type of check would also alert for malicious logon attempts by an outside attacker that has compromised a username and password and after hours, on weekends, or during the shift the person with that username is not scheduled to work attempts to use the stolen/cracked username and password to gain access to the network. This would be easily detected by matching the logon attempt against the facility's physical entry records.

3.8.3 Simultaneous Logon Control.

If a user, already logged on at one IP address, attempted to logon from a second IP address, the **trust system** would check its *simultaneous_logon_limit* parameter to ensure that the maximum number of simultaneous logons for a single user would not be violated before issuing a *logon_approved* message. It would also verify that it was reasonable for the user to be logging on from the source IP by comparing the *time_from_last_activity = current_time – time_of_last_activity* for the IP address of the original logon to the time required to travel between the physical locations of the two logon IP addresses, to ensure it is reasonable for the user to have traveled to the new location to logon. A query message would also be sent to the screen of the computer at which the user first logged on, displaying a message requesting that they approve or deny the simultaneous logon. In this manner, if the *simultaneous logon* was spoofed, and the original user was at their workstation, they could click to DENY the logon. If the response was APPROVE ,or if no response was received within 15 seconds and the **trust system** had no other reason to believe the *simultaneous logon* was not legitimate or reasonable, the logon would be approved and a second entry with the same *username*, but different *logon IP* and *ACCN* values, would be entered into the *ACM*. If at any time the DENY *query_response* was received, or activity was observed from the keyboard or mouse of the original IP address, it would indicate a *suspicious event*. The **trust system** also maintains a record, while a user is logged on, of the credentials used to logon at each location. If, for example, a user logged on with a smart card at the first location, and attempted to logon with the same smart card at the second location, the **trust system** would query the system where the first logon occurred to ensure the smart card had been

removed. If not, this would obviously be a *suspicious event* prompting a *logon_denied* for the second logon attempt.

Logging off from either the first or second location, deletes one of the simultaneous entries from the *ACM*. Then, logging off from the other location, doesn't delete the last entry, but returns *ACCN* and *logon IP* values to zero.

3.9 Access Control Matrix (ACM) - Access Operations Security

3.9.1 Distributed Access Control Matrices.

The systems themselves (also referred to as nodes within this thesis) are authorized to only send certain message types to and receive only certain message types from specific other systems, and only on specific interfaces that match their routing tables. All of these restraints are enforced by the *ACM*. The primary *network-level ACM* is hosted on the **network-level trust system (NTS)**. Table 14 depicts an example portion of an *ACM*.

Table 14. Network Trust System ACM Excerpt

ID (username or systemMAC)	Role	Access Level	Logon ACCN	Trust Level	Effective ACCN	Logon IP
rhadams	SCADA_ operator	Standard	3	-0	3	2001:DC98:5634:2110: BD1C:BA89:7325:3931
jsboone	management	Standard	0	-0	0	Not Logged on.
mdjefferso	office_ worker	Standard	2	-0	2	2001:A344:4DD1:F76F: D2CB:3B09:5629:2005
hrlincoln	vendor_ engineer	3	3	-0	3	2001:DC98:5634:2110: BD1C:BA89:7325:3921
jrsmith	SCADA_ engineer	Standard	4	-1	3	2001:DC98:5634:2110: BD1C:BA89:7325:3923
dktruman	SCADA_ engineer	3	0	2	0	Not Logged on.
smwashingt	SCADA_ administrator	Standard	4	-0	4	2001:DC98:5634:2110: BD1C:BA89:7325:3901
master_station _MAC	my_SCADA _ master	Standard	4	-0	4	2001:DC98:5634:2110: BD1C:BA89:7325:3200
IED-239_ MAC	IED	Standard	4	-0	4	2001:DC98:5634:2110: BD1C:BA89:7325:0239

Each node (IED, RTU, PLC, data concentrator, SCADA master control station, etc.) that has the necessary hard drive storage and processing capacity available could maintain a local software *ACM* hosted on the node itself (in the form of a **nodal trust system**), or in the case of legacy systems, have a network device installed in front of the node to host the **trust system** software and protect one or more nodes behind it as depicted in Figure 11. An example *nodal ACM* is depicted in Table 15.

Table 15. Example Nodal Access Control Matrix

ID	Name	Role	Access Level	Logon ACCN	Trust Level	Effective ACCN
jhadams	John H. Adams	SCADA_operator	Standard	3	-0	3
hrlincoln	Harry R. Lincoln	vendor_engineer	3	4	-0	3
dktruman	Daniel K. Truman	SCADA_engineer	3	3	-1	2
smwashingt	Sally M. Washington	SCADA_engineer	Standard		-0	
MPL_SCADA_ master _station _MAC	MPL SCADA master control station (primary)	SCADA_master_ station	Standard	4	-0	4

In this case, the node does not allow access to everyone that is able to logon to the network and instead maintains entries for specific individuals (usernames) and systems (IP addresses) authorized to logon through the node's terminal interfaces or to access the node's data or code via the SCADA network.

For the purposes of this thesis, it is assumed that *local* (i.e. *nodal*) *ACMs* at each node send an update to the *network-level ACM* on the SCADA **network trust system** each time the node approves an update to its *local ACM*. A node would only need to approve an update to its own *ACM* if connectivity to the **network-level trust system** and logon server were lost and the **nodal trust system** needed to act independently. In this case, if a user attempted to logon directly to the node (for instance at a laptop or terminal connected to a remote substation IED interface or substation controller), the node would have to use its current *ACM* version to verify the username and password. A successful logon results in adding a *logon* and *effective ACCN* to the *local ACM* to maintain the state of logged on users. The node will also send a *logon_request* message to the **logon server** and *ACM_update* to the **network-level trust system** as soon as connectivity is restored, in order to update the *network-level ACM*.

In normal circumstances, the *network-level ACM* always sends an update message to each of the appropriate local *nodal ACMs* whenever it approves a change to its own *network-level ACM* based on a network-level logon and verifies the *nodal ACMs* match the *network-level ACM*.

3.9.2 Standard Access Levels.

The SCADA *network-level ACM* has entries for all individuals authorized access to the SCADA network. A *nodal ACM* maintains entries for all individuals authorized to access the node and all systems authorized to communicate with it. Most *access levels* here are categorized as Standard, in which case the **trust system** will refer to its own *Standard Access Levels Table (SALT)*. Table 16 shows an example of a few *SALT* entries. Using the *Standard Access Levels Table*, the **trust system** performs a lookup of the user's (or system's) authorized *access operations* based on their *role* and *ACCN*.

Table 16. Example Standard Access Levels Table

Role	Effective ACCN	Access Operation	Data Type													
			OC	OD	DC	ND	NC	ED	EC	OA	LG	SE	SC	IW	IC	XW	XC
any	1 or 2	r	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0
		w	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
		x	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
		a	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
		c	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
		d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		s	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
SCADA_operator	4	r	1	1	1	1	0	1	1	1	1	1	0	1	0	1	0
		w	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0
		x	1	0	1	0	0	0	1	1	0	0	0	0	0	0	0
		a	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0
		c	0	1	1	0	0	1	0	1	1	1	0	0	0	0	0
		d	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
		s	0	1	0	0	0	1	0	1	1	0	0	0	0	0	0
	3	r	1	1	1	1	0	1	1	1	1	1	0	1	0	1	0
		w	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
		x	0	1	1	0	0	0	1	1	0	0	0	0	0	0	0
		a	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
		c	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
		d	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
		s	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
SCADA_administrator	4	r	1	1	1	1	0	1	1	1	1	1	0	1	0	1	0
		w	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0
		x	1	0	1	0	0	0	1	1	0	0	0	0	0	0	0
		a	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0
		c	0	1	1	0	0	1	0	1	1	1	0	0	0	0	0
		d	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
		s	0	1	0	0	0	1	0	1	1	0	0	0	0	0	0
	3	r	1	1	1	1	0	1	1	1	1	1	0	1	0	1	0
		w	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
		x	0	1	1	0	0	0	1	1	0	0	0	0	0	0	0
		a	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
		c	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
		d	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
		s	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0

Note from Table 16, which has been extracted from an overall *Standard Access Level Table*, that an operator with the highest *access level* (*ACCN*=4) is granted *read* (i.e. *r*) and *execute* (i.e. *x*) privileges (i.e. user *access operations*) for operational SCADA code and logs, but not the ability to change (i.e. *write*) them.

The operator can also read operational data (e.g. SCADA status values) and emergency data (e.g. operational alerts) and execute SCADA code (e.g. sending OPEN and CLOSE commands to breakers). The operator can only read, not modify, network data (e.g. congestion statistics, server health, links up or down, etc.). Table 17 depicts data types that might be available on various systems in the network.

Table 17. Example Data Types Found on Utility Intranet Systems

SystemName	Data Type (accessible on each system)													
	OC	OD	DC	SD	ND	NC	ED/EC	OA	LG	IW	IC	XW	XC	SE
SCADA master station	X	X				X	X							
trust system						X	X							X
RTU/ IED/ PLC	X	X					X							X
router/switch					X	X								X
network IDS														X
firewall					X	X								X
ops database		X			X	X								X
historical DB		X			X	X	X							X
web server										X	X	X	X	X
power simulator	X	X			X	x								X
log server									X					X
common drives					X	X		X						X
office workstation					X	X		X						X
operator workstation	X	X			X	X								X

Only *data elements* that are readable to the operator would be visible when the operator explorers the network and system directories, folders, and files. For example, suppose that the complete SCADA network file structure for the company is depicted in Appendix E.

Because there is a data element assigned to each saved directory, folder, and file, when logged onto the company's segment of the Utility Intranet, the file structure the operator sees might look like that depicted in Appendix F.

In this fashion, the SCADA operator is only allowed to see the files for which the SCADA_operator *role* is granted access to read (i.e. has the need to know) and for which the proper logon credentials were provided.

When logged onto the separate office network from an office computer, the operator (or any other user) is granted access to the office LAN and shared drives. Read-only access is granted to network data, the company intranet, and the external website.

Note also that an IT network administrator has a very different set of authorized *data elements* and *access operations* as shown in Table 18.

Table 18. Example IT Network Administrator Standard Access Levels

Role	Effective ACCN	Access Operation	Data Type														
			OC	OD	DC	ND	NC	ED	EC	OA	LG	SE	SC	IW	IC	XW	XC
IT_network_administrator	4	r	1	0	0	1	1	0	1	1	1	1	1	1	1	1	1
		w	0	0	0	1	1	0	0	1	0	0	0	1	1	1	1
		x	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0
		a	0	0	0	1	1	0	0	1	1	0	0	0	0	0	0
		c	0	0	0	1	1	0	0	1	1	1	0	1	1	1	1
		d	0	0	0	1	1	0	0	1	1	0	0	0	0	0	0
		s	0	0	0	1	1	0	0	1	1	0	0	0	1	1	1
	3	r	0	0	0	1	1	0	0	1	1	1	0	1	0	1	0
		w	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
		x	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0
		a	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0
		c	0	0	0	1	1	0	0	1	1	0	0	1	0	1	0
		d	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
		s	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0

With an *effective ACCN* of 4, the network administrator can read and append logs but only has read access to security data (i.e. security alerts, packets blocked by the **trust system** or firewall, etc.). Network administrators also have full read, write and execute privileges for network data and code, allowing them to modify configurations for IT components (routers, switches, servers, etc.) and write scripts as necessary to improve network performance; however, they do not have access to operational data or code, as a SCADA operator or SCADA engineer does. Only a security analyst can modify security code (such as firewall and **trust system** rules). Also in the example, the IT network administrator was given read, write, and execute privileges for the internal intranet and external website code to backup the webmaster and assist in responding to cyber attacks to the company's web-accessible resources.

While this is only a theoretical example, data categorization and user access levels could be tailored to provide more or less restriction to the various user *access roles* depending on company and utility needs.

3.9.3 Manually-Entered Access Levels.

Manually-entered *access level* entries in the *ACM* assign a specific maximum *effective ACCN* to an individual. The manual entry allows the security administrator (or **trust system**) to assign a specific *ACCN*, different from the *Standard Access Level*, to an individual user. For instance, suppose new operators undergo a one month on-the-job training regimen and evaluation before being allowed to work unsupervised. During this time, the company may assign a permanent maximum *access level* of 3 to the username,

allowing the new operator, with no further *trust level* restrictions, to only read but not modify operational and emergency data until fully qualified in their position.

Suppose a particular vendor supports its products by providing in-house trouble-shooting assistance to the company. For this reason the vendor's engineers are permitted to logon (on-site or remotely) to the utility company's SCADA nodes that were purchased from the vendor in order to analyze performance and install system updates.

The last two times a particularly grumpy engineer named Harry Lincoln logged-on remotely for routine updates, he did not request that the system be taken offline or inform the control center that his troubleshooting could kick off test alerts that might be mistaken for real events. In addition, when he finished, he updated (appended) the logs with the problem and fix action in a less than professional manner. After explaining the company's expectations and operational impact the first time, and seeing the same behavior again, the utility company security administrator manually entered an *access level* of 3 (Table 19) so that the next time the vendor engineer logged on remotely he would only have read access to operational data and code. In this way, modifications would only be allowed when he was supervised on-site with access temporarily elevated to a full *Standard Access Level* (i.e. maximum *effective ACCN* of 4). Later, when it is deemed that the individual has been performing satisfactorily, the *Standard Access Levels* could be permanently restored for this engineer.

Table 19. Example Nodal Access Control Matrix Entries.

ID	Name	Role	Access Level	Logon ACCN	Trust Level	Effective ACCN
hrlincoln	Harry R. Lincoln	vendor_engineer	3			
dktruman	Daniel K. Truman	SCADA_engineer	3	4	-0	3
smwashingt	Sally M. Washington	SCADA_administrator	Standard	4	-0	4

Suppose a particular disgruntled employee, Daniel Truman, has given notice that he is quitting and Friday will be his last day. His *role* is SCADA_engineer, so he normally has access to documentation of network configurations, SCADA code and production statistics, as well as other sensitive operational information.

Based on an open display of anger yesterday, the utility company decides to manually enter an *access level* of 3 which allows him to read SCADA data and code as he passes continuity on to another engineer over the next few days, but prevents him from copying or modifying any sensitive data or code that might be sold to give another utility company a competitive advantage, used destructively against the company's SCADA network, or taken out of the company to start his own enterprise.

In this way, the utility company regains control of its own network while maintaining a degree of flexibility in a dynamic environment.

In addition, specific access can be configured in the **trust system** *ACMs* if an employee's actions are suspicious. For example, an employee or vendor with a SCADA_engineer *role*, normally able to see and use operational data and operational code, can be restricted from access to operational code based on two times when they have not followed the company policy of developing and testing code on a development system before porting it to the operational system.

3.9.4 Access Level Elevation.

If John Smith had forgotten his smart card or was somehow unable to provide biometric credentials, he would only be granted a *logon ACCN* of 1 with a correct password and username combination, which would authorize him basic_user_access. If John Smith as SCADA_engineer needed to perform root-level functions, he would have to provide the proper credentials or he might be elevated to a higher access level as a result of being vouched for by another member of the company with at least the level of access John Smith wants to attain. Note also that all users authorized root level access must be authenticated by at least two credentials (for *ACCN* = 3 or higher) to gain this access.

One purpose of the elevation function is to enforce security control, based on visual identification, while offering multiple logon options in emergency situations. This prevents a single forgotten or mistyped credential from resulting in an account lockout at a critical moment. It also provides a secure alternative to leaving accounts un-protected (without passwords) to prevent lockouts.

The goal of the network logon method is to provide a one-time logon, with access to all the systems and data a user needs and only at the appropriate, or necessary, level. It does not, however, conform in a straightforward manner to the IT security principle of Least Common Privilege but could be modified to enforce it more strictly. Making the logon process faster, to prevent logon delays with the associated credential checks and *ACCN* calculation can be critical to responding to real-time emergencies, but optimizing calculations and search routines is left to follow-on research.

Note that an *effective ACCN* of either 1or 2 equates to basic_user_access for all *rol*es unless vouched for and escalated by a user with the same *role* that has been authenticated at the higher privilege requested. It is recommended that a user whose access has been elevated by such an *elevation_request* not be allowed to then elevate another user.

This concept embodies two-credential integrity, where one credential can be a person that can visually vouch for the one requesting elevation. Note the recommendation of visual confirmation vice phone calls. Company security policy would have to deal with or disallow the potential phone call dilemma where an attacker uses social engineering techniques to pretend to be or copy/record the true voice of a legitimate user and replay it over the phone stating "I forgot my <credentials>, can you vouch for me?" to try to gain higher-level access after compromising a password and logging on at a lower access level.

When a password is incorrectly entered, the **trust system** would also execute an algorithm to determine how close the entered password is to the X previous passwords, represented by a percentage. Off-by-one might render a 95% depending on the overall password length. A "close" password might indicate a legitimate user that mistyped the password. Other characteristics such as trying the password again or changing the case of one or two letters in the next attempt might indicate a legitimate user that has forgotten the exact password. In the same way, monitoring the entries could also indicate a dictionary attack when each attempt is completely different and does not come close to matching any previous passwords, especially if a single character changes in alphabetical or numerical order. In this manner a potential legitimate user might be afforded five

logon attempts before locking out the password, whereas a dictionary attack might be detected in as few as two attempts.

Combining known password attack signatures from a network IDS with the **trust system** algorithm would provide an even more intelligent analysis. Analyzing host- and network-based IDS alerts with **trust system** security alerts would provide a significantly improved picture of attack attempts from outside the network, successful bypassing of the firewall, continuation of the attack once inside the company's outer security defenses, and data returned to the attacker.

3.9.5 Message Sanitization.

When a particular recipient (system or user) is authorized, per the *ACM*, to receive a particular message type, but only allowed to receive a subset of the *data elements* contained in the message, the **trust system** can sanitize the message before forwarding it to the intended recipient. In this way the code of the system sending the message does not have to be changed to send different messages to different destination users or systems. This is especially usefully in cases where legacy systems and systems of different protocols are in use in the same company's SCADA network or in the destination network. The **trust system** in each network provides sanitization and can bridge communications between dissimilar networks with a protocol gateway capability as described in Section 3.12.1. The **trust system** can maintain a list of multi-cast addresses for particular message types and situations to implement multicast on behalf of sending nodes that do not have this capability, and sanitized multicast to recipients that

can benefit from the message information (e.g. a status update) but who are restricted from receiving some sensitive parts of the data in the message.

As an example of the sanitization process, suppose a status update from a node to the SCADA master control station contains the following data elements: OD, ED, ND.

Also suppose that status messages are also to be relayed to neighboring company control centers so they have a clearer picture of adjacent voltage drops or rises that may affect their contracts or require an adjustment on their part to maintain balance in the power grid. Now suppose that under normal conditions, there is no need for a neighbor company that is a competitor to know if a single bus is undergoing maintenance. The **trust system** can be configured to filter out certain data elements (in this case network data, ND) from messages to another company. All that the node has to know is to send duplicates of the regular status reports it has been designed to send to its SCADA master control station and other recipients. The **trust system** checks the *access level* (i.e. *role* and *effective-ACCN*) of each recipient IP address (and username logged on at that IP), each *data element* type of each *label* in the message, and the *caveat* of the *data element* types, against its *ACM*. By doing so, it ensures unnecessary or company-sensitive *data elements* are removed from the message before sending it to an IP address or user that is not authorized the need-to-know, or in which there is less than full trust.

Sanitization prevents unauthorized information leakage of company-sensitive information and is ideal for an environment with legacy systems or continually evolving requirements. Simple changes to the **trust system** *sanitization* rules and *ACM* can accommodate routing and *sanitization* changes quickly.

In the same manner, even e-mail attachments could be checked for files not authorized for the intended recipient.

For the purposes of demonstration in the simulations for this thesis, *sanitization* was implemented by replacing each *to-be-sanitized* character with a space character, to effectively blank-out the original information, in the *sanitized* output message. A real-world implementation would not allow recipients to have any indication that information was even missing.

3.9.6 Access Violation Attempts.

If a user or system attempts an *access operation* (i.e. an *operation_request* message is received), the *data type* of the data for which access is requested and the *access operation* on that *data type* is checked against the *ACM* for the individual (or system's) current *role* and *effective ACCN*. If the requestor is not authorized to access that particular *data type*, or is authorized access to the *data type* but not authorized to perform on that *data type* the operation requested, the attempt will be denied by the **trust system** and initiate a *suspicious event*. An *operation_denied* warning message will be sent to the screen of the source IP address. Figure 12 depicts a sample denial warning.

Access Operation Denied: Username *jrsmith* not authorized to delete SCADA code.
If you believe this restriction is in error, contact net-security-team@mpl.com.

Figure 12. Warning to Requestor's Screen for Denied Operation Message

3.9.7 ACM Scorekeeper.

Similar to the *FWR-SK* and *FOR-SK*, the *ACM scorekeeper (ACM-SK)* keeps track of failed logon, simultaneous logon, and elevation attempts. It also updates failed *access operation* attempts. When the *ACM* has completed all *ACM* checks, if any check has failed in the *scorekeepers*, all three *scorekeepers* are forwarded to the *Suspicious Event Handler (SEH)* module.

3.9.8 Supplemental Access Control Policies and Procedures.

Another threat to critical infrastructure might come from a state-sponsored or terrorist source. If online attempts to gain access are sufficiently thwarted, the only network access method may be kidnapping or armed assault. If a company employee were held at gunpoint and forced to logon remotely to the network at escalated privilege, company security policies might require a beeper or cell phone message to a specific beeper number and beeper ID or cell phone number and phone ID to which the requestor must confirm the access attempt, deny, or confirm but send distress with GPS location. The problem here would be the possibility of stealing beeper and cell phone numbers for administrators from the company's phone bills at the phone company or through a compromised online account.

Despite all these efforts, the greatest threat is often from the inside. Policy (security and termination) plays an important role in ensuring that network access is discontinued as soon as an employee is no longer to be employed by the company. In the event of a disgruntled employee or corporate espionage, before the situation is realized and a decision is made to separate the individual, all actions by that individual are logged

by username and timestamp. Later, these records can assist the company in holding the individual accountable for any damage or malicious intent.

3.9.9 Maintaining a Secure State.

Before a change to the *ACM* is authorized and implemented, the **trust system** should auto-check the proposed *ACM* policy change to ensure both a proper domination relationship and a secure state are maintained using such methods as the *-property and Simple Security Principles for mandatory and discretionary access control [30]. These methods have not been simulated for this thesis.

3.10 Suspicious Event Handler (SEH) Module.

3.10.1 Alert Counter.

After the **trust system** evaluates a message according to its *firewall*, *format*, and *ACM rules*, if any parameter failed, the *firewall rules*, *format*, and *ACM scorekeepers* are forwarded to the **trust system** *Suspicious Event Handler* (*SEH)* component. The *SEH* uses the failed parameters to determine when to generate a security alert and of what type. Some types of *suspicious events (SE)* will create an immediate *security alert*. Others will start an *alert counter*.

The *alert counter* is set in order to continue to monitor *suspicious events* that the *SEH* cannot yet determine to be a security issue (e.g. a failed logon that might be a legitimate user that has forgotten or mistyped a password). The *SEH* increments the alert counter for each occurrence until the configured threshold for that type of alert is reached. Once the *counter threshold* has been reached, the *SEH* generates a

security_alert message. It may also lower the *trust level* of a particular message type, protocol, interface, username, system, or any combination of these parameters as a result. A lowered *trust level* may lower the *effective_ACCN* in the *ACM* and may also require a blocking (i.e. deny) action in the **trust system** *firewall rules*.

As an example, suppose the *alert counter* threshold is set to 3 with a duration of 60 seconds for bad data packets detected, as shown in Appendix C. The *alert counter* for the suspicious event is initially 0. When the first bad data packet is received, the alert counter is incremented to 1. After the second bad data packet, the *alert counter* equals 2. If three packets are received from the same source in a 60 second period, with data values that do not conform to the expected range, a *security alert* is generated and further messages on that interface from that source are blocked by updating the *firewall rules*.

3.10.2 Tracking Suspicious Events by Suspicious Event ID.

When a *suspicious event* notification is received by the *SEH* (i.e. when the *SEH* is forwarded *scorekeepers* containing failed parameters) it initiates a new *suspicious event ID (SEID),* characterized by its *SEID number,* which is the date-timestamp that the event was first detected (i.e. when the first packet was received by the **trust system),** and two or three parameters known as *trackers* taken from the *scorekeepers.* The *SEID* is an object containing all of the *scorekeepers,* the *SEID number,* and the *trackers*. Table 20 summarizes the *tracker* values assigned for different *suspicious event* types, indicated in the *scorekeepers*.

Table 20. Trackers for Possible Trust System Suspicious Events

Suspicious Event (SE) Type	Trackers
Logon SE	Tracker1: username
	Tracker2: source_IP
Access Control SE	Tracker1: username
	Tracker2: source_IP
	Tracker3: <object_of_operation>
Firewall Rules SE	Tracker1: source_IP
	Tracker2: destination_IP
	Tracker3: <failed_label>
Message Format SE	Tracker1: source_IP
	Tracker2: destination_IP
	Tracker3: <message_type>

The purpose of the *trackers* is to be a reference point for correlating similar packets that may be part of a larger event. Each time the *SEH* receives a *suspicious event*, before creating a new *SEID*, it compares the *trackers* for the incoming *scorekeepers* to the *trackers* of currently open *SEIDs*. If there are no matches, it checks recently closed *SEIDs* as well. If any of the *trackers* match, the *SEH* will determine if the new activity is part of a previous *SEID* and, if so, update a currently open *SEID* or re-open a closed, related *SEID*.

3.10.3 Blocking.

When a blocking action is required, the *firewall rules* allow the **trust system** to deny packets based on any combination of message type, protocol, interface, username, or system IP address. If the traffic was previously allowed by a whitelist rule in the *firewall rules*, the *Deny* column is simply changed from *false* to *true*. If the necessary granularity for the blocking rule does not already exist, a new rule is added for the activity experienced and the *Deny* column is set to *true*.

3.10.4 Trust Assignment and Authorization.

By recognizing bad or malicious packets from a particular source, especially if it occurs more than once, the **trust system** can begin to lower its trust in further packets from that source and even switch to another more trusted source as its primary, trusted input for particular *data elements*, alerts, or status updates.

Lowering the *trust level* for users or systems lowers their *effective ACCN*, restricting some of their access to critical data and restricting their privileges (i.e. operations on the data to which they still have access).

3.11 Outgoing Message Handling

3.11.1 Re-encryption.

If a message passes all **trust system** checks and is to be forwarded on to the original destination, after any required *sanitization* takes place, the **trust system** reassembles the payload in the original order and must re-encrypt it before forwarding it on to its intended destination.

Messages created by the **trust system** (i.e. queries, control messages, alerts, warnings, logon_approved/denied, operation_denied, etc.). An exception is made in the case where original message was un-encrypted and it is believed that the source may not be encrypting properly. In this case, UDP messages will simply be blocked and ignored and TCP messages will result in a RST/ACK to close the connection. A *query_encryption* message may also follow, to the closest **trust system** to the source IP, requesting an investigation and confirmation of the encryption problem and actions taken to prevent further unencrypted traffic from the source.

3.11.2 Addressing and Routing.

For a **trust system** to be able to perform *format* and *ACM checks*, it must be able to decrypt packet payloads to inspect the data inside.

This is simpler if the encryption is accomplished solely by **trust systems**. If the encryption is performed by a **nodal trust agent** on the source or a **gateway-mode trust router** along the path, that **trust system** will know the next **trust system** down the line, closest to the destination (i.e. a **trust router** or **trust agent** on the destination system), and be able to encrypt the packet with its own private key and the public key of the down-range **trust system,** then apply an IP header with its IP address as a source and the down range **trust system's** IP address as the destination, before forwarding it on. The down-range **trust system** will strip the IP header, decrypt and inspect the contents. If it is a **trust router,** and the packet passes all checks**,** it will apply another IP header with its IP address as the source and forward the unencrypted packet to the destination system. If the down-range **trust system** is a **nodal trust agent** on the destination system, the agent

will decrypt the packet, complete its checks, and if the packet passes, deliver the unencrypted original packet to the operating system.

If the source operating system applies IPsec encryption to its packets, this requires the source to encrypt the packet payload with the public key of the next **trust system** along the path and with its own private key, in order for the **trust system** to be capable of decrypting and inspecting its contents. Then that intermediate **trust system,** after inspecting the packet, will repackage the original packet (with its original IP header), encrypt the packet with its private key and the public key of the next **trust system** closest to the destination, and add a second IP header to route it to that **trust system**.

The other option is for the sender operating system to encrypt the packet with IPsec and when the packet is received at the destination, the **trust system** there performs decryption on behalf of the operating system, checks it, and passes it up to the next layer in the OSI stack if it passes all checks. Although this requires fewer **trust system** checks and reduces end-to-end delay, the time to stop a bad packet is the greatest, only occurring at the destination.

3.12 Other Required or Augmenting Capabilities Not Simulated

3.12.1 Protocol Gateway.

Legacy RTUs, PLCs, and IEDs were developed with proprietary protocols and prior standards such as MODBUS, DNP3, Fieldbus, etc. This may require specific protocol gateway plug-ins to translate input delivered in various protocols to a common format.

3.12.2 Summary and Full Reporting Modes.

To eliminate network traffic over bandwidth-constrained communication lines, a message from the **trust system** could toggle between full reporting (when all seems normal or there is good bandwidth and un-necessarily delayed traffic flow) or summary reporting (in the event of high congestion or when the node is involved in an emergency situation). If necessary, based on bandwidth congestion or lines down, the **trust system** could send a squelch message to less important nodes to send minimal update info and not overwhelm the line. Integrating security alerts with network management alerts would provide a more intelligent view of the impact of an attack in progress to bandwidth usage, although the **trust system** has an inherent bandwidth calculation algorithm used to determine TCP and UDP connection capacity.

3.12.3 Key Management.

There is the potential for packets to be sniffed and the key cracked, enabling an attacker to spoof messages to and from nodes internal to the utility company or from outside utility entities. Changing keys often can help to prevent this. Especially when a compromise is detected or a suspicious event that might have resulted in a key compromise is suspected, the key should be automatically changed. For this reason, it is recommended to have a key change and distribution process, initiated at random times at least once per week, potentially once per day, and if congestion is not a problem, per message via a reliable means (i.e. TCP).

If an emergency is initiated in the middle of a key update, some nodes will have changed over to the new key and some may be in transition or not have received the

update yet. Both the old and new key would need to remain valid until a response is received from all recipient nodes that the new key has been updated.

3.12.4 Node Discovery.

All nodes on the network are required to logon on to the network logon server in the same way that a user must, in order to participate on the network. A system provides its own unique credentials, such as IP address, MAC address, a unique node ID or node name, and IPsec authentication. When the logon server evaluates the node's *logon_request* message, it forwards a *logon_evaluated* message to the **trust system,** which then identifies if there is any security reason to mistrust or deny the logon and reports back to the logon server with a *logon_approved* or *logon_denied* message. The **trust system** also calculates an *ACCN* (equal to 4 if there is no reason to mistrust the system) and updates its *ACM* to show the *node_name* and IP address as logged on to the network. Whenever information is received indicating the node has gone down or is disconnected for an extended period of time (e.g. expected messages are not received and a subsequent ping or status check with the **nodal trust system** receives no response), the logon entry is deleted, and the logon server is sent a *logon_denied* message, requiring the node to logon once more to join the network when it comes back online. This realization would also prompt a *maintenance_alert* message.

3.12.5 Alert Correlation.

Very often network security, network management, and the operators impacted the most by configuration changes are physically separated, hindering timely communications between these parties. Ideally, all would be co-located in the same

control center room. Whether this is the case or not, correlating network management system (NMS) and security alerts can facilitate a network management and security synthesis that, together, provide instant awareness of the impact of security configurations and cyber attacks on network performance and operations and the impact of network outages on operator capabilities and network security posture.

By gathering all alerts from SCADA, EMS, network management, and network security platforms, an **alert correlator** can convert them to similar formats to display to a company, CA, or RC control room. The alerts could be easily filtered to show only a subset of the total alerts (i.e. just the operational alerts, only emergencies, only network security alerts, or any combination or subset). In this way, a control room could be properly staffed with operators, engineers, network security analysts, and network management experts that can operate in real-time off the same sheet of music and understand the complete impact of outages and emergencies on availability, performance, security, and safety of the entire network.

The **trust system** should include or work in conjunction with a network security correlation tool that would evaluate network security alerts from other security mechanisms (i.e. network and host-based intrusion detection systems and firewalls) in the network and initiate (or recommend to a human analyst) corrective or mitigating actions based on a simulation or estimation of network and utility service impact of such actions (whether automated or human-in-the-loop). In fact, if malformed packets, bad or corrupted data, or DoS indicators were detected, the cause could be a system (i.e. software or hardware) malfunction or malicious attack, so evaluation of alerts from both security and engineering/maintenance perspectives is essential, further justifying the

integration of alerts from a network security, capable of informing and interacting with the **trust system**, with an overall **alert correlator** which is fed network security, network management, and operational alarms.

3.13 Assumptions for Development of Experiments

3.13.1 Protocols and Standards.

For the purposes of this paper and its experiments, an IPV6, TCP- and UDP-compliant structure was used for messages. UDP was the protocol-of-choice for non-real-time updates and **trust system** queries, to alleviate network congestion. TCP was used for emergency traffic and real-time or near real-time traffic that either required reliability or would be implemented as TCP by its manufacturer. For example, it can be assumed that network logon operations would be designed as standard TCP/IP traffic by an IT vendor. Furthermore, it was assumed that in an emergency situation a logon should be a high priority event (warranting reliability and confirmation) to ensure engineers and operators gain fastest access to the network to implement response actions. For simplicity, even in non-emergency situations, logons are deemed high priority.

Previous work by Birman, et al., demonstrated the feasibility of UDP messages for sending breaker trip messages between peer nodes on a SCADA network, within just a few seconds, when there is no network delay. Delivery times were only a few seconds longer in the face of network congestion or communication links. It is, however, necessary for some emergency situations to be resolved in fractions of a second, often in 100ms or less. Hard real-time notifications might even need to be made in 4ms or less.

For such messages to be received, processed, and reacted to, these UDP techniques do not provide the necessary reliability and transit time guarantees.

In an attempt to resolve the more real-time requirements, the SCADA network was simulated as capable of UDP messaging for non-emergency traffic, and dedicated TCP bandwidth for emergency traffic. For example, normal status updates are sent as UDP datagram packets. Walled-off TCP bandwidth is reserved for emergency commands, including neighbor trip attempts and emergency status updates. Criteria for emergency handling would be defined in the **trust system** specification and would typically be indicated by protocol (TCP) and message type. A dangerous security event might also warrant the **trust system** sending an emergency alert notification. The **trust system** implements a prioritization of each packet in its incoming and outgoing queues to ensure that only the highest priority packets are checked first and sent first. Less important data that is moved to the back of the line, so to speak, would be checked against a staleness factor for the message type and queue. It would only continue through the **trust system** process to be checked and forwarded if the time delay from waiting in the queue for higher priority packets to be processed did not make the data stale or obsolete, in which case it would be discarded because a more current update has already arrived.

3.13.2 Encryption Delay.

For this thesis, IPsec encryption is assumed for all messages between nodes on the SCADA network in order to assess its impact. The SCADA network nodes modeled for this thesis only communicate over a single encrypted port (port 500 for IPsec) for

inbound and outbound messages. Nested application-level encryption would add additional overhead and require the **trust system** to update and cache application specific keys, but would greatly increase the security of transmissions. Only IPsec was simulated for the purposes of this research effort but application-layer encryption would be useful to investigate in follow-on research.

3.13.3 Network Message Formats.

Various packets can be expected to traverse the Utility Intranet. Commands from HMIs to SCADA master control stations, commands and polls from SCADA master control stations to substations (i.e. data concentrator, RTUs, PLCs, and IEDs), file transfers for IED configuration and PLC programming, and status updates and alerts from substation power systems, network management systems, and network security systems (e.g. firewalls and intrusion detection systems). Even low-priority corporate e-mail and file sharing has been allowed to traverse some utility networks.

Packet sizes for messages vary depending on purpose, payload, and protocol. For the purposes of the simulations and experiments for this thesis, example message types selected for the model network are defined in Table 21. The format for each message type is illustrated in Figure 13.

Table 21. Message Types Defined for Simulations

Type	Message	Description
1	*get_status*	Request for power *status* from a node.
2	*status*	Contains power *status*. Response to a *get_status* or sent by a node to inform others of its *status*.
3	*set*	Command to set breaker *status* as open or closed.
4-6	*intertrip, neighbor_trip, backup_trip*	Command to trip (i.e. open) a breaker.
7	*logon_request*	Generated by a workstation, terminal, or node when a user attempts to logon to the network. Sent to the logon server along with credentials supplied by the user. Also sent by a node reconnecting to the network.
8	*logon_evaluated*	Generated by the logon server after receiving a *logon_request*. Specifies logon server's analysis of the user's credentials (which were authenticated and which failed).
9	*logon_denied*	Response to *logon_ request* and *logon_evaluated* messages. Relays the verdict that the logon is disapproved. Includes the *ACCN* (0 because access is denied) and any *ACM* or *trust level* changes regarding that username. May also inform the node's *ACM* to locally deny any further attempts if the **network trust system**'s *SEH* detected a dictionary attack and believed the logon attempts to be malicious.
10	*logon_approved*	Response to *logon_ request* and *logon_evaluated* messages with the verdict that the logon is approved. Includes the user's *ACCN* and any *ACM* or *trust level* changes regarding that username.
11	*security_alert*	Warning of suspicious event that violated the security policy. Includes actions taken by the **trust system** and a link to further detail.
12	*ACM_update*	Identifies most current *ACM* settings. Sent from **network trust system** to **nodal trust systems** (and vice versa) to promulgate *ACM* changes.
13	*suspicious_event _log*	Historical log entry record (or update) of a *suspicious event.* Contains per-packet detail, **trust system** evaluation, and **trust system** response.
14	*query_packet*	Query by the **trust system** to find out if the source IP actually sent a packet believed to be spoofed.
15	*query_ACM*	Query by the **trust system** to find out if a system has the latest ACM.
16	*query_ simultaneous_ logon*	To prevent malicious logon. Sent by the **trust system** whenever a currently logged-on username attempts to logon-from a second IP address. Creates an alert to the screen at the IP address where the initial logon occurred, prompting for an APPROVE/DENY response from the user.
17	*query_response*	Response to a query from the **trust system**.
18	*elevation_request*	Occurs infrequently when a user, who is authorized to perform duties at a higher level, but on this occasion does not have enough of the credentials present to authenticate at the higher level to perform a duty they are required to perform. The request is sent by the user, after a successful logon at a lower than desired access level, to another user currently logged on with the same role but higher effective ACCN, requesting they vouch for their authorization and approve the trust system granting a higher effective ACCN than they provided credentials for. Typically only an emergency measure.
19	*elevation_approved*	Sent by the **network trust system** to the node that originated an *elevation_request* with the verdict that the elevation is approved. Includes the user's new *effective ACCN*.
20	*elevation_denied*	Sent by the **network trust system** to the node that originated an *elevation_request* with the verdict that the elevation is denied.

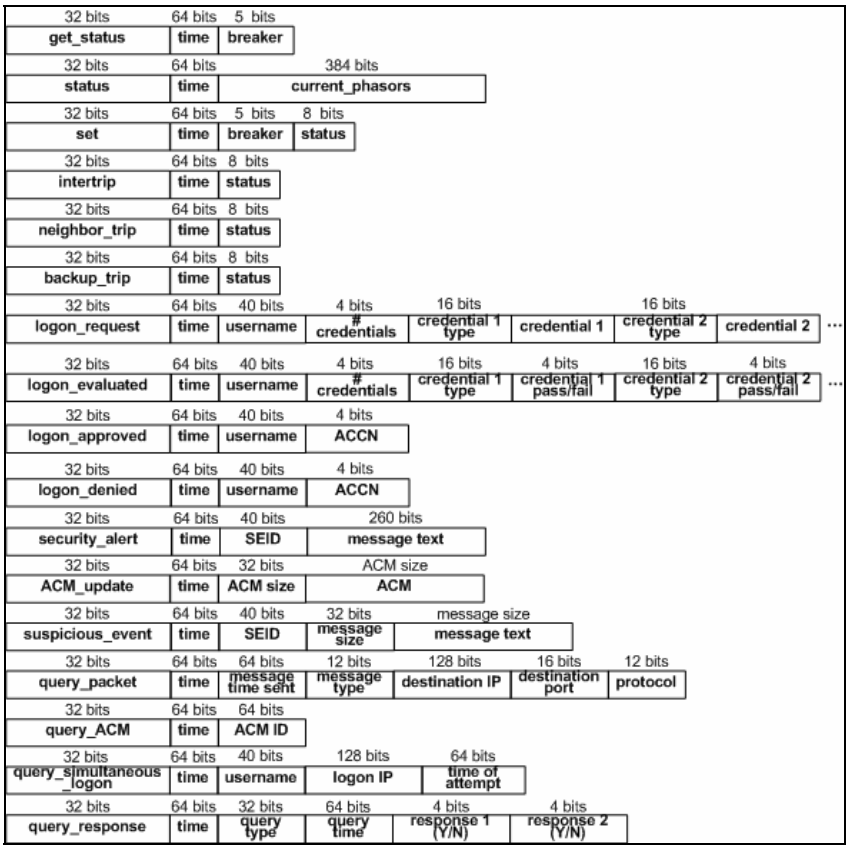


Figure 13. Format for Scenario Message Types

3.13.4 Background Traffic.

Besides operational traffic, other company network traffic, including office automation (e.g. e-mail, web, etc.) and network management traffic (i.e. SNMP, etc.), might be present simultaneously on the same external communications links between organizations (even between internal offices). Here again, exact network loading and bandwidth consumption will be company-specific. The capability to inject large volumes of random background traffic into the scenarios was a limitation of the simulator and would be a good follow-on test of the robustness of the **trust system**.

IV. Analysis and Results

4.1 Chapter Overview

The purpose of this chapter is to present the calculations and simulation results for **trust system** interception, evaluation, and response to real-time and non-real-time traffic expected in a Utility Intranet that includes substation automation, wide-area notifications, and malicious actions by a determined and intelligent foe. The primary results are delay estimates of the time required for trust system checks, encryption mechanisms, end-to-end delivery per packet, and scenario resolution, to include attack mitigation. The second goal of this chapter is to estimate the potential applications for these security technologies and honestly evaluate limitations in defensive capabilities and real-time response.

4.2 Investigative Questions Answered

This chapter indicates that IPsec encryption can be used carefully in a SCADA environment to provide security and that a **trust system,** properly configured and maintained, will either prevent, quickly detect and mitigate, or provide sufficient evidence after-the fact to determine where and how malicious activity occurred in the network. It supports the hypothesis that TCP and UDP can be used with bandwidth guarantees to meet real-time delivery requirements. This chapter also shows that the automated actions of the **trust system** can provide comprehensive, all-in-one, layered security, reducing the need for a large team of security analysts while giving those few security analysts required the exact tools they need to answer difficult questions regarding intrusion footprints.

The following sections of this chapter explain the measurements, foundational calculations, simulation scenarios, and resulting delays determined in bringing to life and supporting the **trust system** concept.

4.3 Scenario Files

4.3.1 Input Files.

A text file comprised of the *firewall rules* was read in by the **trust system** main.cpp program prior to processing a scenario packet. The number of rules in this file was kept to a minimum, including only the rules that applied for the scenarios. This required the least *firewall rules* check delay, allowing the **trust system** simulation to avoid the low level of accuracy (i.e. 10ms increments) of the Microsoft Windows® system clock and reasonably substitute the average **trust system** check delay values measured for each message type and transport protocol.

An input scenario text file was created to specify IP packet details for each scenario to be read into the **trust system**. Specifically, each component (i.e. *label*) of each packet's headers and data were specified as variables and assigned the appropriate value for that scenario. An actual IP packet would be received as a sequence of digital bits (i.e. ones and zeros) for which the **trust system** would need to strip off the appropriate number of bits for each component in turn and assign it to the appropriate variable to be evaluated. However, for simplicity of generating and reading the scenarios, the components (i.e. *labels*) in each packet in the input file were represented from the start as a mix of integers, floating point variables, or doubles for numerical values and as hexadecimal or ASCII values for string or character equivalents.

Some of the more complex scenarios required multiple interactions between network servers, SCADA nodes, and the **trust system.** In some cases, the **trust system** was required to send and receive multiple packets to and from other systems in the network. This was required either to query for additional information needed to improve the accuracy of its decisions, block unauthorized activity, respond to logon server evaluations of logon credentials, or send updates such as changes to ACMs, access control lists, firewall rules, and assigned *ACCN* values, each of which defined improvements to the overall network security posture.

To account for the end-to-end delay that would be experienced by these packets as they traversed the network, their message type (indicating message size in bits) and the source and destination IP addresses (indicating total distance to travel) were read into the simulator and used to calculate their impact on received time of the next message and the overall scenario's completion time. It was assumed that as soon as the trust system completed its processing of one packet, it was immediately ready to read in and begin processing the next packet, calling this the received plus queue time at the **trust system**. Packet sent time was then determined by subtracting the calculated transmit and propagation delay on the link from source to **trust system** and estimated queuing delay **in** the **trust system** input queue from this received plus queue time**.** Any delays for sending packets due to human response time (i.e. in typing a password or reading an elevation request before responding) could easily be tacked onto the total transit time to account for delays at the source before the packet was sent.

4.3.2 Output File.

The output file entry generated for each scenario demonstrated an alert for suspicious activity, a log of the results of each of the **trust system** the checks (i.e. the passed or failed parameters), a log of actions taken by the **trust system** in response, documentation of the time to complete each check, and the total time to complete all **trust system** functions for a packet.

It was assumed that the **trust system** was able to provide small network security alerts (i.e. with only minimal, summary information), either directly to the screen of a network security analyst or to an **alert correlation** system, on the network, where combined security and network management alerts could be evaluated for further action required or dismissal of false positives (i.e. verifiably legitimate events the **trust system** algorithms categorized as suspicious). This was simulated by the **trust system** code writing the text of these entries to the output file under headings for each scenario and each packet in the scenario. A more detailed log of the parameters that passed or failed the **trust system** checks and the values of those failed parameters were posted to the same output file to simulate logs sent to an archive for historical purposes. The same detailed data would, then, be available for analysts to request if they needed further detail in their evaluation of a security alert, without automatically overwhelming their screens with potentially unnecessary excess data. And easy way to implement this is a link in the summary alert allowing the analyst to then open and drill down into the related, more detailed historical record available on a separate data store. Event and packet statistics (such as estimated bandwidth available on the link, response times, etc.) could also be calculated and posted to log entries.

4.4 Delay Measurements and Calculations Approach

To simulate the operational feasibility of a network, two factors are of paramount importance: delay and congestion. Both are contingent upon the bandwidth available throughout the network, propagation characteristics, store-and-forward operations (i.e. queuing delay) by individual devices within the network, the presence or absence of redundant paths and systems, system or connectivity failures, and the time required for **trust system** checks.

4.4.1 Trust System Delay.

The **trust system** is able to measure statistics on delay for each received packet and each **trust system** check, to include the time to complete a *firewall* rules check, format check, logon check, *access control check*, and *sanitization*. Summing these values gives the total time to complete all **trust system** checks necessary before forwarding the packet on to its destination (i.e. if it is a good packet) or throwing away a bad packet.

It was discovered, however, that the Microsoft Windows XP® system clock updates in 10ms increments, which did not provide the microsecond granularity necessary for the small execution times of these individual functions. To estimate overall **trust system** check times, time trials for the various message types were conducted 650, 000 times, for both TCP and UDP, to determine the minimum, maximum, and average delay for each message type. For each trial, two different *firewall rules* files were used, the first with the matching rule at the top of the list, so that it would be found immediately, and the second with the matching rule at the bottom of a list of 2000

firewall rules, giving the slowest times for rule matches. These results are depicted in Appendix D.

It was hypothesized that dividing the *firewall rules* times by the number of rules in the firewall (i.e. 2000) would provide the cost per rule, which could then be extrapolated to estimate delays for smaller and larger rules lists. It was also thought that dividing the *format check* times by the number of packet elements checked would provide an accurate cost per value, however the **trust system** *format checks* implementation was sufficiently different for each *label*, that the results varied greatly and were not easily extrapolated to estimate the required time for messages with greater or fewer data values. It was, therefore, necessary to run time trials for each message type and average the results.

The measurements were conducted using a PC with Intel® Pentium® 3.00GHz CPU and 3.50GB RAM, running the Microsoft Windows XP Professional® operating system with Service Pack 2. Each message type ran through complete **trust system** checks 50,000 times and the results were average for each trial. Each trial was repeated 15 times for a total of 650,000 samples taken per message type.

Minimum, average, and maximum values were recorded. These results, using both UDP and TCP versions per message type, are depicted in Appendix D.

4.4.2 Network Transit Delay.

Processing delay is the measure of the time required to examine a packet's header and determine where to route the packet. Processing delay would also include the time

required to check for bit errors in the packet that occurred in transmitting the packet's bits from the source node to the router, trust system, and destination node [31].

The total processing delay within the **trust system** from the time the first check begins on a packet to the time it is ready to be forwarded on to its destination is designated as $d_{proc(TrustSystem)}$. This value is derived from actual measurements of execution time of the code's checks.

Queuing delay, $d_{queue,}$ is the time while a packet sits in the output queue to be transmitted onto the link by the source node and each router or trust system along the way [31].

$$d_{queue(node)} = 3(size_{queue})size_{packet})/(rate_{incoming\_link}) \quad (1)$$

where:

$size_{queue}$ = *queue size of router, node, trust system (B)*

$size_{packet}$ = *packet length including headers (bits)*

$rate_{incoming\_link}$ = *incoming link rate (bps)*

It would also include the time waiting in the input queue to be processed, which depends on the priority it is assigned and the quantity and size of higher priority packets that are processed ahead of it. In the case of the **trust system**, d_{queue} has been divided into two parts: an incoming queuing delay, which is the time a packet waits in the incoming queue before processing of the packet begins by the **trust system,** and an outgoing queue delay which is the time the same packet waits to be transmitted onto the link by the **trust system**.

Transmission delay, $d_{trans,}$ is the amount of time required to push (i.e. transmit) all of the packet's bits onto the communications link at the source and each router or **trust system** along the way [31].

$$d_{trans} = length_{packet} / rate_{outgoing\_link} \quad (2)$$

where:

$$\text{size}_{packet} = \textit{packet length including headers} \text{ (bits)}$$

$$rate_{outgoing\_link} = \textit{rate of the link} \text{ (bits/sec)}$$

The $rate_{outgoing\_link}$ can vary due to link congestion and dynamic bandwidth assignment algorithms.

Propagation delay, d_{prop}, is the total time required for the packet to propagate from the outgoing interface of one node in the link to the incoming interface of the next node, for each node along the path traveled by the packet. If $distance_{link(i\ ->\ i+1)}$ is the link distance (in meters) between a network device or system, $node_{(i)}$, that is about to transmit a packet onto the link, and the next device or system, $node_{(i+1)}$, poised to receive the packet and, if $speed_{prop(i\ ->\ i+1)}$ is the *propagation speed* of the signal across that link (in m/s), then the propagation delay (in milliseconds) across a series of *n* links, $d_{prop(end\text{-}to\text{-}end)}$, is given by Equation 3 [31].

$$d_{prop(end\text{-}to\text{-}end)} = \sum_{i=1}^{n} \left(\frac{distance_{link(i\ ->\ i+1)}}{speed_{prop(i->i+1)}} \right) (1000) \quad (3)$$

For simulation experiments, the fiber cabling between each node within a company's network was assumed to be of the same capacity, therefore, $distance_{(source->destination)}$, the

distance between the source and the destination nodes, could be used to approximate the total end-to-end link sum, $\sum_{i=1}^{n} distance_{link(i \rightarrow i+1)}$, so that Equation (3) reduces to Equation (4) for LAN communications within a single company.

$$d_{prop(end\text{-}to\text{-}end)} = \left(\frac{distance_{link(source \rightarrow destination)}}{speed_{prop(constant)}} \right)(1000) \qquad (4)$$

Higher speed links were used for most inter-organization (i.e. company to company or company to CA) communications, requiring use of Equation (3) for their propagation delay.

The *propagation speed* is dependant upon the physical medium of the link. For these experiments, all internal company links were assumed to provide a total 100Mbps. The distances between fixed nodes were maintained in the **trust system's** *firewall rules* with their traffic rules and were used to calculate available throughput and the legitimacy of receive times for incoming packets, especially when certain packets were expected only at regular intervals. Values were adjusted within reasonable boundaries, for calculated current congestion values.

Device processing delays are specific to each uniquely manufactured device in the network and can be expected to continue to decrease over the next several years as better and faster network technologies are developed. In this simulation, reasonable delay estimates were used for network components such as routers, switches, and fiber optic cabling as depicted in Table 22. For the speed of light in fiber, a value of 2.0×10^8 meters

per second was assumed. Processing delay for routers and switches was assumed to be approximately the same. A minimum value of 0.09 milliseconds and a maximum value of 2 milliseconds were used. Constant queue size for all nodes was estimated to be a medium range of 300B. The greater the queue size, the greater the overall processing delay per packet.

Table 22. Network Device Delay Figures for End-to-End Calculations

Device	Delay Type	Delay Estimate (ms)
source	d_{trans}	$length_{packet} / rate_{outgoing\_link}$
router	$d_{proc(router)}$	min=.09ms, max=2ms
	$d_{queue(router)}$	$3(queue\_size)(message\_size)/(rate_{incoming\_link})$
	d_{trans}	$length_{packet} / rate_{outgoing\_link}$
switch	$d_{proc(switch)}$	min=.09ms, max=2ms
	$d_{queue(switch)}$	$3(queue\_size)(message\_size)/(rate_{incoming\_link})$
	d_{trans}	$length_{packet} / rate_{outgoing\_link}$
trust system	$d_{decryption}$	$3(queue\_size)(message\_size)/(rate_{incoming\_link})$
	$d_{proc(TrustSystem)}$	message type-specific, use Appendix D
	$d_{queue(TrustSystemIncoming)}$	$3(queue\_size)(message\_size)/(rate_{incoming\_link})$
	$d_{queue(TrustSystemOutgoing)}$	$3(queue\_size)(message\_size)/(rate_{incoming\_link})$
	$d_{packet-reassembly}$	$d_{(TrustSystem-FWR)}$
	$d_{encryption}$	message type- and processor-specific, use Appendix E
	d_{trans}	$length_{packet} / rate_{outgoing\_link}$
fiber optic cable	$d_{prop(link)}$	$distance_{link}/(2.0*10^8$ m/s)*1000
destination	$d_{queue(destination)}$	
	$d_{proc(destination)}$	min=.09ms, max=2ms
total:	$d_{end-to-end}$	sum of the above values

End-to-end delay, $d_{end-to-end,}$ is the one-way latency of a packet from source to destination and was calculated using Equation (5) for the sum of all of the values in Table 22.

$$d_{end\text{-}to\text{-}end} = d_{trans(source)} + (d_{proc(switch)} + 2*d_{queue(switch)} + d_{trans(switch)})(quantity_{switches}) \quad (5)$$

$$+ (d_{proc(router)} + 2*d_{queue(router)} + d_{trans(router)})(quantity_{routers}) + (d_{proc(TrustSystem)}$$

$$+ d_{queue(TrustSystemIncoming)} + d_{queue(TrustSystemOutgoing)} + d_{trans(TrustSystem)})(quantity_{TrustSystems})$$

$$+ d_{trans(source)} + d_{prop(link)} + d_{queue(destination)} + d_{proc(destination)}$$

4.4.3 Encryption Delay.

All packets in the SCADA network simulated were assumed to be encrypted and authenticated for greater data security. The **trust system** simulation code does not actually perform any encryption or decryption, so, to estimate IPsec encryption delay, the research of Niedermayer, Klenk, and Carle [32] was used as the basis for extrapolating values for each message type. Their work indicated much better performance of IPsec as compared to SSSL. Of the multiple IPsec Authentication Header (AH) and Encapsulating Security Payload (ESP) variations that they measured, the best, worst, and mid-range performers were selected for use in this thesis. The results of their measurements demonstrated minimal difference between the performance of AH-only, ESP-only, and AH plus ESP; therefore, the obvious solution, for maximum security was to use both AH and ESP.

128-bit Encryption Standard (AES-128) with SHA-1 authentication was the fastest performer with average security. Unfortunately, SHA-1 has been cracked and a 128-bit key is not nearly as secure as a 192- or 256-bit key. Triple (3DES) with the stronger SHA-256 authentication was the worst performer, according to their measurements, creating a huge encryption time delay that also proved problematic with meeting real-time requirements for the scenarios. 192-bit Blowfish with SHA-256

performed in the middle range overall and appeared to be the best fit for both better security and lower delay. Plotting the rise and run for their results, within the range of message sizes used for this thesis' experiments, yielded the slope and general equations for extrapolating IPsec encryption delay with both AH and ESP (for maximum security) listed in Table 23.

Table 23. IPsec Encryption and Authentication Delay Equations

Encryption/ Authentication Scheme	**IPsec Mode**	**Encryption Delay (ms)**	**#**
AES-128/ SHA-1	trans	$d_{AES\text{-}128/SHA\text{-}1(transport)} = 1000((10.253)(size_{payload})+19337.5)/speed_{CPU}$ $size_{payload}$ = size of message payload to be encrypted (bits) $speed_{CPU}$ = processor speed in the encrypting node (Hz)	(6)
AES-128/ SHA-1	tunnel	$d_{AES\text{-}128/SHA\text{-}1(tunnel)} = 1000((10.52)(size_{payload})+19698)/speed_{CPU}$	(7)
Blowfish-192/ SHA-2(256)	trans	$d_{Blowfish\text{-}192/SHA\text{-}2(transport)} = (1.17)(1.146)\, d_{AES\text{-}128/SHA\text{-}1(transport)}$ $= (1.3408)\, d_{AES\text{-}128/SHA\text{-}1(transport)}$	(8)
Blowfish-192/ SHA-2(256)	tunnel	$d_{Blowfish\text{-}192/SHA\text{-}2(tunnel)} = (1.3408)\, d_{AES\text{-}128/SHA\text{-}1(tunnel)}$	(9)
3DES/ SHA-2(256)	trans	$d_{3DES/SHA\text{-}2(transport)} = (2.75)(1.17)\, d_{AES\text{-}128/SHA\text{-}1(transport)}$ $= (3.2175) d_{AES\text{-}128/SHA\text{-}1(transport)}$	(10)
Blowfish-192/ SHA-2(256)	tunnel	$d_{3DES/SHA\text{-}2(tunnel)} = (3.2175) d_{AES\text{-}128/SHA\text{-}1(tunnel)}$	(11)

Appendix E lists the results of these calculations for each message type traversing the simulated network.

4.4.4 Concurrency.

Although the **trust system** C++ code for this simulation did not implement concurrent processes, a realistic implementation would use pipelining to increase the speed of execution. Separate measurements of completion time for each check (i.e. *firewall rules*, *format*, and *ACM* with *sanitization*) on each packet traversing the path of

the **trust system** allow calculation of best case performance with concurrent processing of more than one packet at a time.

4.5 Scenarios Approach and Simulation Network

The escalating scenarios in Chapter 4 simulate IP packet traffic of various sizes and various message types between SCADA nodes and network servers that is intercepted and analyzed by the **trust system.** The reaction of the **trust system** to each message, by accurately allowing legitimate traffic, blocking malformed packets and unauthorized traffic due to user errors or malicious attempts, or sanitizing information in messages that the receiver is not authorized read, demonstrated the successful execution of the **trust system** concept and supporting computer code. Delay measurements were calculated based on maximum response times measured for the **trust system** and average and high-end ranges for each network component (i.e. routers, switches, and cabling) along the way. The total time for each scenario was also calculated. These delay figures indicate the impact of **trust system** operations on control system time constraints.

Figure 14, illustrates a simple, two-company slice of an interconnected Utility Intranet, illustrating the various SCADA and IT systems simulated in the experiments for this thesis.

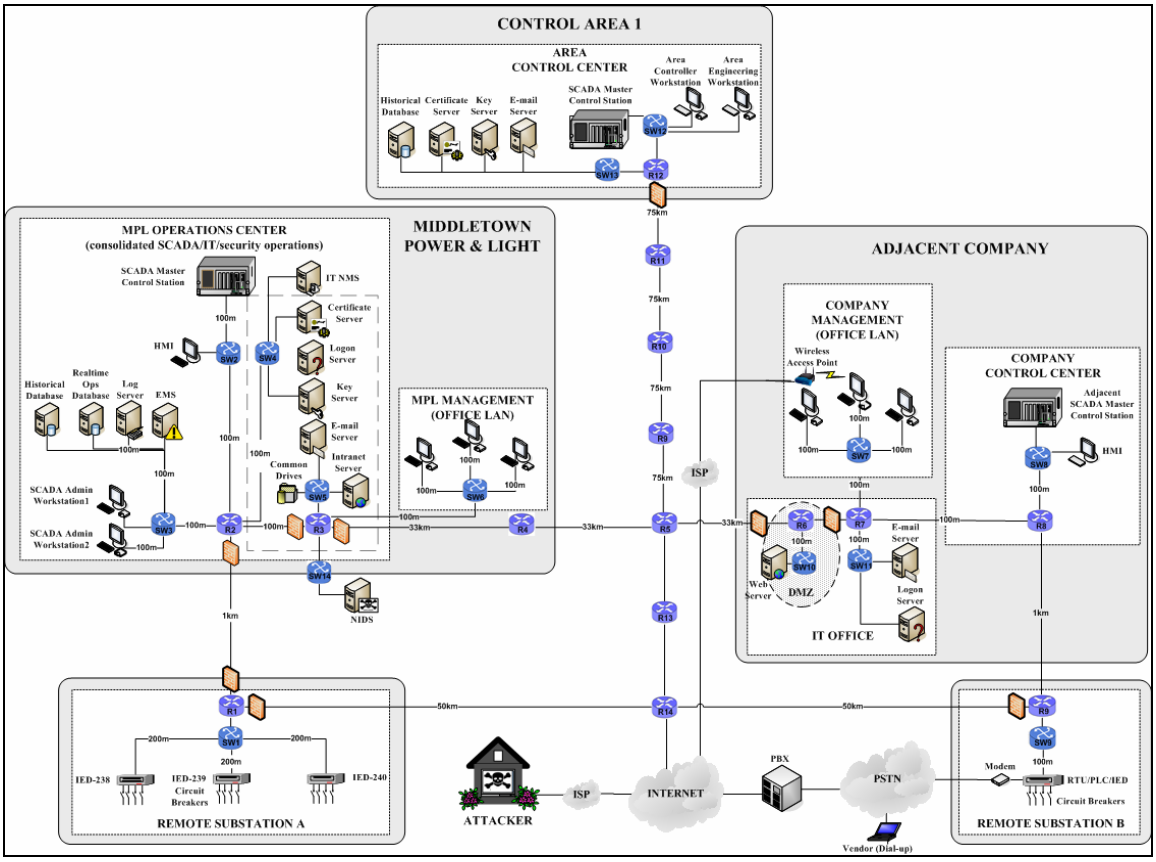


Figure 14. Typical Network Diagram

The simple, yet realistic scenarios to demonstrate the concepts proposed in this thesis are based upon a fictitious electric utility company, Middletown Power and Light (MPL), and its personnel, a nearby utility company with some poor security habits, and their area control (or operations) center. The scenarios are not intended to represent any particular real-life company or employee. Figure 15 depicts the same network as Figure 14, yet replacing the standard firewalls with more comprehensive, strategically placed **trust systems** in the network for a minimal **trust system** implementation. The diagram shows the components and the distances used in calculations. To illustrate applicability to highly remote communications, the two company control (or operations) centers are

100km apart from each other and 30km away from the nearest substations that they control. The CA Operations Center is approximately 300km away from each company. Of course, the ideal **trust system** implementation would implement all router/switch combinations as **trust routers** and include **trust agents** on nearly all nodes in the network.

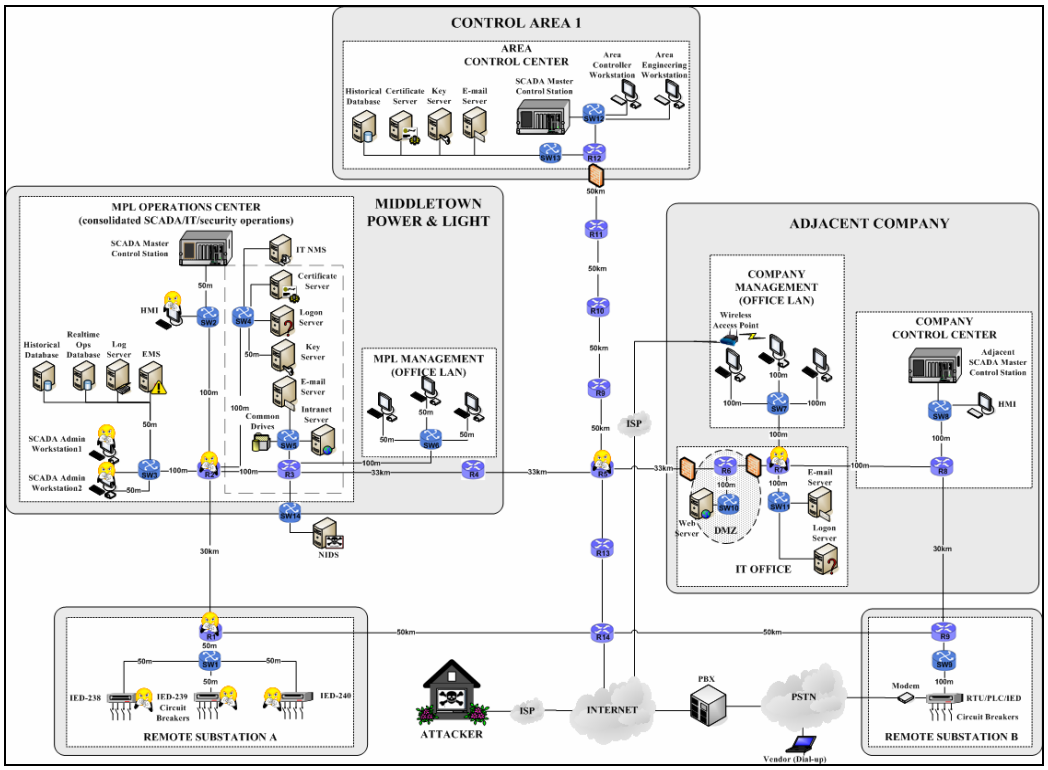


Figure 15. Scenarios Network Diagram (Minimal Trust System Implementation)

4.6 Baseline Simulation Scenarios

4.6.1 Overview.

The purpose of the baseline scenarios was to demonstrate **trust system** functionality in a benign environment with properly formatted traffic sent by legitimate

users or systems. They indicate the delay induced by the **trust system** in a network without background or malicious traffic and estimate the associated impact on the types of day-to-day traffic expected on a Utility Intranet.

4.6.2 Scenario 1 - Legitimate Status Update.

A legitimate UDP *status* update packet, Packet 1-1, was transmitted within MPL's SCADA network from IED-239 (in Substation A) to MPL's SCADA master control station. The input parameters defined for Packet 1-1 (as received at the MPL **network trust system**) are depicted in Figure 16. Note that IPv6 tunnel mode and **trust system** gateway (i.e. router) mode have been employed. Specifically, the IED-239 **nodal trust system** encrypts the message from IED-239 using its private key and the **network trust system's** public key then adds an IP routing header to send it to the **trust system** gateway closest to the destination, trust_router2, which happens to be hosting the **network trust system**. Ellipses throughout the rest of the examples indicate omissions, for brevity.

```
MESSAGE UDP
//begin IPv6 outer header for tunnel mode
        IP2_traffic_class             3290
        IP2_flow_label                4E28C
        IP2_source_address            2001:DC98:5634:2110:BD1C:BA89:7325:4239
                                      //nodal_TS@MPL_IED-239
        IP2_destination_address       2001:DC98:5634:2110:BD1C:BA89:7325:4050
                                      //network_TS@MPL_trust_router2
//end IPv6 outer header for tunnel mode
//begin AH header
        …
//end AH header
//begin ESP header
        …
//end ESP header
//begin IPv6 inner header for tunnel mode
        IP1_traffic_class             C450
        IP1_flow_label                13B87
        IP1_payload_length            101
        IP1_source_address            2001:DC98:5634:2110:BD1C:BA89:7325:0239
                                      //MPL_IED-239
        IP1_destination_address       2001:DC98:5634:2110:BD1C:BA89:7325:3200
                                      //MPL_SCADA_master_station
//end IPv6 inner header for tunnel mode
//begin UDP header
        …
        UDP_source_port               500
        UDP_destination_port          500
        UDP_length                    26
        …
//end UDP header
//begin message data
        message_type                  status
        time_message_created          12:00:00.0000-20Jun07
        busNumber                     1006
        busName                       HUNTLEY_
        CAname                        CA1
        companyName                   _MPL
        nominalVoltageKV              +0220.000
        busVoltPu                     +0000.984
        VoltKV                        +0137.581
        busAngleDeg                   +0013.790
        loadMW                        +0017.610
        loadMvar                      +0320.740
        gen_MW                        -0236.740
        genMvar                       +0234.020
        switchedShuntsMvar            +0200.000
        actGshuntMW                   +0009.110
        actBshuntMvar                 -0006.760
        month_AMR_collect_start       0:00:00.0001-01Jun07
        customers                     20
        month_AMR__total_usage        479,015.996
        daily_ave_AMR_usage           24,563.731
        AMR_usage_kWh_today           13,404.326
//end message data
//begin ESP trailer
        …
//end ESP trailer
//begin ESP ICV
        …
//end ESP ICV
```

Figure 16. Packet 1-1 (UDP Status, IED-239 to MPL Master Station)

The **trust system** simulator code, at the time of this writing, did not implement multicast or a carbon-copy list, however this capability was simulated by sending the exact same message, with the same originating timestamp, to each of the other (external) destination IP addresses allowed to receive the message. In this case, the same *status* message was forwarded to both the CA1 control center and a neighboring competitor company (adjacent company 1) control center. In a network without a **trust system agent** loaded on IED-239, the **network-level trust system (NTS)** could create and send duplicates of the message to the CA and neighbor destinations based on its list of carbon-copy recipients and on behalf of the IED, which could not multicast the message.

According to the **trust system** *firewall rules*, both external destinations (i.e. outside the MPL SCADA network) were authorized to receive a *status* message, but the adjacent competitor company was not allowed to receive all of the same MPL status data that would be given to a fully-trusted organization, like the CA control center. Instead, the adjacent company was only granted access to the minimal amount of performance parameters required for it to recognize or respond to emergency situations occurring within MPL's span of control. Although *firewall rules* and *format checks* all passed, the *ACM* identified *data elements* in the message, specifically financial rate and customer usage data (i.e. data type FN) which the competitor was not authorized to read. As a result, the **trust system** sanitized the *status* message, as depicted in the packet detail of Packet 1-2, Figure 17. The **trust system** demonstrated sanitization of the message that would be forwarded to the adjacent company by replacing each character of the financial data elements with an X.

```
MESSAGE UDP
//begin IPv6 outer header for tunnel mode
        IP2_traffic_class             F065
        IP2_flow_label                C13AA
        IP2_source_address            2392001:DC98:5634:2110:BD1C:BA89:7325:4050
                                      //network_TS@MPL_trust_router2
        IP2_destination_address       2001:DC99:31AC:9AAD:FC29:6C1A:80EA:4057
                                      //network_TS@adjacent1_trust_router7
//end IPv6 outer header for tunnel mode
//begin AH header
        …
//end AH header
//begin ESP header
        …
//end ESP header
//begin IPv6 inner header for tunnel mode
        IP1_traffic_class             530E
        IP1_flow_label                8B7A0
        IP1_payload_length            101
        IP1_source_address            2001:DC98:5634:2110:BD1C:BA89:7325:0239
                                      //MPL_IED-239
        IP2_destination_address       2001:DC99:31AC:9AAD:FC29:6C1A:80EA:3200
                                      //adjacent_company1_master_station
//end IPv6 inner header for tunnel mode
//begin UDP header
        …
        UDP_source_port               500
        UDP_destination_port          500
        UDP_length                    26
        …
//end UDP header
//begin message data
        message_type                  status
        time_message_created          12:00:00.0000-20Jun07
        busNumber                     1006
        busName                       HUNTLEY_
        CAname                        CA1
        companyName                   _MPL
        nominalVoltageKV              +0220.000
        busVoltPu                     +0000.984
        VoltKV                        +0137.581
        busAngleDeg                   +0013.790
        loadMW                        +0017.610
        loadMvar                      +0320.740
        gen_MW                        -0236.740
        genMvar                       +0234.020
        switchedShuntsMvar            +0200.000
        actGshuntMW                   +0009.110
        actBshuntMvar                 -0006.760
        month_AMR_collect_start       XXXXXXXXXXXXXXXXXXX
        customers                     XX
        month_AMR__total_usage        XXXXXXXXXX
        daily_ave_AMR_usage           XXXXXXXXX
        AMR_usage_kWh_today           XXXXXXXXX
//end message data
//begin ESP trailer
…
//end ESP trailer
//begin ESP ICV
…
//end ESP ICV
```

Figure 17. Packet 1-2 (Sanitized Status, IED-239 to Adjacent Master Station)

In comparison, Packet 1-3 (Figure 18) was sent to the company's CA control center, which was authorized to receive financial data because of its responsibility for regulating electrical power costs, usage, and generation, was unsanitized and had identical data as Packet 1-1, sent to the MPL operations center.

```
MESSAGE UDP
//begin IPv6 tunnel mode outer header
        IP2_traffic_class             DA2F
        IP2_flow_label                1CC43
        IP2_source_address            2001:DC98:5634:2110:BD1C:BA89:7325:4239
                                      //nodal_TS@MPL_IED-239
        IP2_destination_address       2001:DC98:5634:2110:BD1C:BA89:7325:4050
                                      //network_TS@MPL_trust_router2
//end IPv6 tunnel mode outer header
//begin AH header
        …
//end AH header
//begin ESP header
        …
//end ESP header

//begin IPv6 tunnel mode inner header
        IP1_traffic_class             95C0
        IP1_flow_label                B9602
        IP1_payload_length            101
        IP1_source_address            2001:DC98:5634:2110:BD1C:BA89:7325:0239
                                      //MPL_IED-239
        IP1_destination_address       2001:DC99:31AC:9AAD:FC29:6C1A:80EA:3200
                                      //CA1_master_station
//begin UDP header
        …
        UDP_source_port               500
        UDP_destination_port          500
        …
        UDP_length                    26
//end UDP header
//begin message data
        messageType                   status
        time_message_created          12:00:00.0000-20Jun07
        busNumber                     1006
        busName                       HUNTLEY_
        CAname                        CA1
        companyName                   _MPL
        nominalVoltageKV              +0220.000
        busVoltPu                     +0000.984
        VoltKV                        +0137.581
        busAngleDeg                   +0013.790
        loadMW                        +0017.610
        loadMvar                      +0320.740
        gen_MW                        -0236.740
        genMvar                       +0234.020
        switchedShuntsMvar            +0200.000
        actGshuntMW                   +0009.110
        actBshuntMvar                 -0006.760
        month_AMR_collect_start       24:00:00.0001-01Jun07
        customers                     20
        month_AMR__total_usage        479,015.996
        daily_ave_AMR_usage           24,563.731
        AMR_usage_kWh_today           13,404.326
//end message data
//begin ESP trailer
```

```
…
//end ESP trailer
//begin ESP ICV
…
//end ESP ICV
```

Figure 18. Packet 1-3 (Unsanitized Status Update, IED-239 to CA1 Control Center)

No suspicious event or security alert was warranted; therefore, the sanitized message was forwarded on to the adjacent company and the original message was forwarded to the MPL master control station and CA1 control center. The packet details were logged to the historical database.

Table 24 summarizes the end-to-end delay totals for each of the three packets to reach their destinations, comparing IPsec mode options using Blowfish-192/SHA-2(256), maximum measured **trust system** values for a *status* message, and **trust system** processor speeds ranging from 3GHz to 12GHz. Internal to the MPL network, the IED was able to deliver a status update within 1.62ms, well within the normal 2sec time constraint and sufficient for an emergency notification. External communication was also possible in less than 4.3ms over distances as great as 300km. The greatest delay dependency resides in the routers along the path. Routers with large queue size and high processing delay were simulated in conjunction with tunnel mode IPsec to provide an idea of worst case delivery with non-real-time routers. Results indicated fractions of a second transit time, though not hard real-time. Routers (or **trust routers**) that will handle real-time traffic must be optimized for minimal processing delays.

Table 24. Scenario 1 Delay Summary

Source	Destination	IPSec Mode	Queue Size (B)	d_{proc} (ms)	Trust System Delay (ms)	Link Delays (ms)	Router/ Switch Delays (ms)	Per Packet Delay (ms) UDP Only 3GHz	4GHz	8GHz	12GHz
IED-239	my_master_station	none	300	0.09	0.5281	0.3999	0.4922	**1.4342**	**1.3022**	**1.1042**	**1.0381**
		transport	300	0.09	0.5922	0.4226	0.5485	**1.6095**	**1.4614**	**1.2393**	**1.1653**
		tunnel	300	0.09	0.5934	0.4281	0.5485	**1.6167**	**1.4683**	**1.2458**	**1.1716**
		tunnel	1500	2.00	0.5934	4.3937	8.6221	**13.6559**	**13.5075**	**13.2850**	**13.2108**
IED-239	adjacent_master	none	300	0.09	0.5281	1.3124	1.1930	**3.0475**	**2.9155**	**2.7175**	**2.6515**
		transport	300	0.09	0.5922	1.3868	1.3176	**3.3428**	**3.1948**	**2.9727**	**2.8986**
		tunnel	300	0.09	0.5934	1.3868	1.3176	**3.3445**	**3.1962**	**2.9737**	**2.8995**
IED-239	CA_master_station	none	300	0.09	0.5281	2.2461	1.1680	**3.9563**	**3.8243**	**3.6262**	**3.5602**
		transport	300	0.09	0.5922	2.3267	1.2820	**4.2470**	**4.0990**	**3.8769**	**3.8029**
		tunnel	300	0.09	0.5934	2.3267	1.2820	**4.2488**	**4.1004**	**3.8779**	**3.8037**

4.6.3 Scenario 2 - Legitimate Area Summary and Emergency Trip.

A legitimate UDP *area_summary* message, Packet 2-1, was received from MPL's CA1 control center. These messages summarize power status for the hundreds, thousands, or tens of thousands of buses within the control area that would be of interest to a particular company. The summary indicated rising load requirements in nearby towns managed by other electrical utility companies in the same control area. Typical packet size is around 2.4MB and was simulated by sending 9600 *status* packets, similar to the example above, each approximately 250B in size; however in reality, maximum packet fragment sizes might be as large as 1500B.

The calculated transit time, from send to receive, for a single *status* packet would have been a minimum of 4.25ms in IPsec tunnel mode with a 3GHz **trust system** processor (as determined from Scenario 1). For 2.4MB, the estimated receive time would be approximately 9600 times that delay, equivalent to 40.8sec for MPL to receive and process the complete update from its area operations center. At the high end of the delay spectrum, with large router/switch processing delays (d_{proc} = 2ms) and larger queues

(1500B), delivery of one UDP *status* packet would have taken an estimated 37.88ms, requiring 363.652sec = 6min 3.7sec to receive and process the entire 2.4MB equivalent.

However, only a few seconds after receiving the first bit of the 2.4MB *area_status* message, a legitimate TCP emergency *trip* message, Packet 2-4, was also received by MPL from the CA1 control center. The parameters defined for Packet 2-4, as received at the **network trust system**, are depicted in Figure 19.

```
MESSAGE TCP
//begin IPv6 tunnel mode outer header
        …
        IP2_source_address            2001:6B03:105E:A993:28CA:E7BB:A4B3:4050
                                      //network_TS@MPL_trust_router3
        IP2_destination_address       2001:DC98:5634:2110:BD1C:BA89:7325:4050
                                      //network_TS@MPL_trust_router2
        …
//begin IPv6 tunnel mode outer header
//begin AH header
        …
//end AH header
//begin ESP header
        …
//end ESP header
//begin IPv6 tunnel mode inner header
        …
        IP1_source_address            2001:6B03:105E:A993:28CA:E7BB:A4B3:3200
                                      //CA1_master_station
        IP1_destination_address       2001:DC98:5634:2110:BD1C:BA89:7325:0239
                                      //MPL_IED-239
        …
//end IPv6 tunnel mode inner header
//begin TCP header
        TCP_source_port               500
        TCP_destination_port          500
        …
        TCP_control_flags             111000  //URG, ACK, PSH, RST, SYN, FIN
        …
//end TCP header
//begin message data
        message_type                  breaker_trip
        time_message_created          13:00:00.0000-20Jun07
        status                        OPEN
//end message data
//begin ESP trailer
        …
//end ESP trailer
//begin ESP ICV
        …
//end ESP ICV
```

Figure 19. Packet 2-4 (TCP Emergency Trip Message from CA1 to IED-239)

Because the packet was an emergency packet, indicated by the message type (*trip*), TCP protocol (emergency TCP bandwidth is reserved for extremely time-critical communications), and URG control flag being set, it was moved to the front of the **trust system** input queue, and allowed to interrupt the evaluation of the non-emergency UDP *area_status* summary packet. The **trust system** processed the emergency *trip* message before completing all of the *status* messages, simulating the capability of the **trust system** to break evaluation of a single UDP 2.4MB area summary packet to devote all of its efforts to handling the emergency event. Concurrent processes with sufficient memory and processing speed could allow simultaneous processing by the **trust system** with little impact to real-time response to the emergency. The emergency packet passed all **trust system** checks, warranting no suspicious event or security alert.

The source (i.e. CA1) was a trusted, neutral third party that MPL had given permission to initiate emergency actions on its systems, when warranted, so the packet was forwarded directly to the intended destination, IED-239. A copy of the same packet was also sent to the SCADA master station for awareness in the MPL operations center. The MPL SCADA master station would, in turn, issue its own *trip* command in response and the node would respond to whichever message it received first and discard the second.

After tripping its breaker, IED-239 replied in response with a multicast TCP emergency *status* packet to the MPL master control station and the CA control center indicating the now open breaker, as depicted in Figure 20.

```
MESSAGE TCP
//begin IPv6 tunnel mode outer header
        …
        IP2_source_address            2001:DC98:5634:2110:BD1C:BA89:7325:4239
                                      //nodal_TS@MPL_IED-239
        IP2_destination_address       2001:DC98:5634:2110:BD1C:BA89:7325:4050
                                      //network_TS@MPL_trust_router2
        …
//end IPv6 tunnel mode outer header
//begin IPv6 tunnel mode inner header
        …
        IP1_source_address            2001:DC98:5634:2110:BD1C:BA89:7325:0239
                                      //MPL_IED-239
        IP1_destination_address       2001:6B03:105E:A993:28CA:E7BB:A4B3:3200
                                      //CA1_master_station
//end IPv6 tunnel mode inner header
//begin TCP header
        TCP_source_port               500
        TCP_destination_port          500
        …
        TCP_control_flags             111000  //URG, ACK, PSH, RST, SYN, FIN
        …
//end TCP header
//begin message data
        message_type                  status
        time_msg_created              13:00:00.0218-20Jun07
        breaker                       1003
        status                        open
//end message data
//begin ESP trailer
        …
//end ESP trailer
//begin ESP ICV
        …
//end ESP ICV
```

Figure 20. Packet 2-2 (TCP Trip Response from IED-239 to MPL Master and CA)

The delay calculation results are summarized in Appendix F. The emergency *trip* alone took an estimated 22-205 ms (regular TCP, tunnel mode IPsec) to execute using regular TCP control protocol, avoiding the blackout events occurring in nearby cities from spreading or affecting customers supplied by MPL. Using an abbreviated TCP protocol (by eliminating an ACK from three-way handshakes and graceful closes and by only ACKing with data, whenever possible) would reduce the response time nearly 40%. The packet detail was also logged to the MPL log server and historical database.

After the **trust system** handled and forwarded IED-239's trip response *status* packet, it immediately returned to completing its checks on the rest of the UDP *area_summary* update packet (i.e. the rest of the 9600 packets simulating a single 2.4MB packet). The total time calculated for the **trust system** to evaluate this message was between 40.82sec ($d_{proc(router/switch)}$= 0.9, 300B queue sizes) and 6min, 3.9sec ($d_{proc(router/switch)}$ = 2.0, 1500B queue sizes), from start to finish, including the delay in evaluating the emergency *trip* and response messages. Of course, the trip action may have stabilized the overall area power status, rendering this message's data stale and not worth continuing to process.

The robustness of the **trust system** code created for these simulations was demonstrated in its handling of over 9600 packets while re-prioritizing its actions to handle an emergency.

4.6.4 Scenario 3 - Successful Root Logon by a Legitimate User.

A *logon_request*, Packet 3-4, was sent from SCADA_admin_workstation1 by user Sally Washington, a SCADA administrator with username *smwashingt*. Returning after a 2-week vacation, she had forgotten and mistyped her password as depicted in Figure 21.

```
MESSAGE TCP
//begin IPv6 tunnel mode outer header
        …
        IP2_source_address              2001:DC98:5634:2110:BD1C:BA89:7325:4051
                                        //nodal_TS@MPL_SCADA_workstation1
        IP2_destination_address         2001:DC98:5634:2110:BD1C:BA89:7325:4050
                                        //network_TS@MPL_trust_router2
//end IPv6 tunnel mode outer header
//begin AH header
        …
//end AH header
//begin ESP header
        …
//end ESP header
//begin IPv6 tunnel mode inner header
        …
        IP1_source_address              2001:DC98:5634:2110:BD1C:BA89:7325:3901
                                        //MPL_SCADA_admin_workstation1
        IP1_destination_address         2001:A344:4DD1:F76F:D2CB:3B09:5629:1000
                                        //MPL_logon_server
//end IPv6 tunnel mode inner header
        …
//begin message data
        message_type                    logon_request
        time_message_created            08:00:00.000-21Jun07
        username                        smwashingt
        number_of_credentials           4
        credential_1_type               PASS
        credential_1                    !#V8k12g4x
        credential_2_type               CARD
        credential_2                    1D43EF3409193A389BB067867D3A80C3249B8
        credential_3_type               PIN
        credential_3                    10465891
        credential_4_type               FING
        credential_4                    ÿØÿà JFIF  dd1i=&ÿÛ„OXÖS+x°rGŽe-Ô&9 |6C‡ùw?]
                                        ± ÊÍ/‘ L Ëíë´F±O K³XüÞGÇr qBÒåj Õ©•ã É5²ÇY
                                        u “™Òß`é°³ÆUªTj¤Ö¼ö ˜Úµ®Ê‰p®Êaµºm$<'×Lhø Ìbx
                                        ÿØÿà  JFIF     d d  1i  =&ÿÛ „ OXÖS+x°rGŽe-Ô
                                        &9 |6C‡ùw?]± ÊÍ/‘ L Ëíë´F±O K³XüÞGÇr qBÒåj Õ
                                        ©•ã É5²ÇY au “™Òß`é°³ÆUªTj¤Ö¼ö ˜Úµ®Ê‰p®Êaµºm
                                        $<'×L hø Ìb_x$ ½\-|h¡• ò™žä ø Æÿ œ7ÖÆJ†^©gÇ²
                                        Ç\·åiIû]z 8Û×X{rÈåŠ¶ÜsJÄnÙQ0•æ>³é¬½5ä6ÆÊT3×·
                                        "³™‰)™ §ût§ÄWµ]`cø·i éë3·M 7™í3ÙÓéÿ9Õ'6¸ X¢“
                                        9“Ü§ºÈú G¤h #ˆÀˆþ  ´Gö□@¾¿u_ë /Œü¤ÿ,6)Ž'í,þ ý
                                        º¯ ^É3-"f å1 ¾ÑþÏM9rèlÕ d‰3îo<¿5-¥t×#ô vÚ “+
                                        D”¬)Í¡ŸÓÔÞœŽ²§üEÃ)þ9úÆ…ÌÉ¡™ VaÜ  4Õ§W¨w¦ˆOÇ
                                        !ô†–ÓøjÅºÇeÕd5¬æìþ›ˆµ-],eEC-/„IFÑËSnÚ¼f )kŠÄ
                                        ê9J êìÜù4ZûÕŒÒõä[Dï:Û §di[/ŽÜ‡ áû×°ëšµR§‘m§1
                                        $_ÌX³□^šNB†n ¬ˆZ<bMuû |Œ»eËÆ³Õ½«¯’ ƒí9 ÷ ¢"w
                                         $›y ß“Q¬Ùæñ<¿vk -dŸqµý»ô…è|p mN(|³Å~+‡ Ê ¬œ
                                        ×  gq ‘@ûÞ®¾E÷rËã ®@ñÇñr”YF¢òøÄrž`ÆD¡ë’ŸÌ  ¬
                                        •ü: Ö2pßñ  Sm›Üî²¦Rœ+y /aI □{Y7  F'Èkåà\]©Œ□
                                        ¬ èm<ù!FÞ…-¤3§-®šø/'®ã    wöÉo Ü·)çîúm¯"I, ´
                                        \]š¿^KnÍ–í3¼A–ÿ Û¬y/žÖ±€²`oÚaGP(˜ôôÚtó°+7U´á
                                        ŽqÐ8qãìŒOM"ÉÕ¯Êa›òPIO¾vë#øi ±   â#  -òôˆéÿÿ
                                        Ÿ×G¶ÿ ÏG¤íõ rartG-S Ë‰{wßùaõÔ   Žá$¿Ì{~
//end message data
//begin ESP trailer
        …
//end ESP trailer
//begin ESP ICV
        …
//end ESP ICV
```

Figure 21. Packet 3-4 (First Failed Logon Attempt, Wrong Password).

The password entered, *!#V8k12g4X*, did not match the stored password of *!#V8k12g$X*; therefore, the logon attempt was denied. The **trust system** recognized the similarity to the correct password and increased the threshold from three tries to five, to provide her more opportunities to log on.

Sally tried a second time. The second logon attempt (Figure 22) was denied because the last character was a lowercase letter *x* instead of the expected uppercase *X*.

```
MESSAGE TCP
…
//begin message data
        message_type                    logon_request
        time_message_created            08:00:15.000-21Jun07
        username                        smwashingt
        number_of_credentials           4
        credential_1_type               PASS
        credential_1                    !#V8k12g$x
        credential_2_type               CARD
        credential_2                    1D43EF3409193A389BB067867D3A80C3249B8
        credential_3_type               PIN
        credential_3                    10465891
        credential_4_type               FING
        …
//end message data
…
```

Figure 22. Packet 3-15 (Second Failed Logon Attempt, Wrong Case)

The third logon attempt was denied because of a typo, an @ sign instead of a *2*, as illustrated in Figure 23.

```
MESSAGE TCP
…
//begin message data
        message_type                logon_request
        time_message_created        08:00:28.000-21Jun07
        username                    smwashingt
        number_of_credentials       4
        credential_1_type           PASS
        credential_1                !#V8k1@g$x
        credential_2_type           CARD
        credential_2                1D43EF3409193A389BB067867D3A80C3249B8
        credential_3_type           PIN
        credential_3                1046589
        credential_4_type           FING
        …
//end message data
…
```

Figure 23. Packet 3-23 (Third Failed Logon Attempt, Typo)

The fourth logon attempt (Packet 3-31) supplied the proper password. Since all of the credentials supplied were correct, the logon server, sent the evaluated credentials to the in a *logon_evaluated* packet, depicted in Figure 24.

```
MESSAGE TCP
//begin IPv6 header
        …
        IP1_source_address              2001:A344:4DD1:F76F:D2CB:3B09:5629:1000
                                        //MPL_logon_server
        IP1_destination_address         2001:DC98:5634:2110:BD1C:BA89:7325:3901
                                        //MPL_SCADA_admin_workstation1
//end IPv6 header
//begin AH header
        …
//end AH header
//begin ESP header
        …
//end ESP header
//begin TCP header
…
//end TCP header
//begin message data
        message_type                    logon_evaluated
        time_message_created            …
        username                        smwashingt
        number_of_credentials           4
        credential_1_type               PASS
        credential_1_pass               YES
        credential_2_type               CARD
        credential_2_pass               YES
        credential_3_type               CPIN
        credential_3_pass               YES
        credential_4_type               FING
        credential_4_pass               YES
//end message data
//begin ESP trailer
        …
//end ESP trailer
//begin ESP ICV
        …
//end ESP ICV
```

Figure 24. Packet 3-33 (Logon Credentials Evaluated by the Logon Server)

The logon was successful, so no security alert was generated. The **trust system** assigned an *effective ACCN* of 4 to username smwashingt in its *ACM*, granting Sally root-level privileges in a *logon_approved* message, as depicted in Figure 25.

```
MESSAGE TCP
…
//begin IPv6 header
        IP1_source_address              2001:DC98:5634:2110:BD1C:BA89:7325:4050
                                        //network_TS@MPL_trust_router2
        IP1_destination_address         2001:DC98:5634:2110:BD1C:BA89:7325:4239
                                        //nodal_TS@MPL_IED-239
        …
//end IPv6 header
        …
//begin message data
        message_type                    logon_approved
        time_created                    …
        username                        smwashingt
        effective_ACCN                  4
//end message data
…
```

Figure 25. Packet 3-37 (Successful Logon by SCADA Administrator)

The **trust system** then generated a historical log entry for the historical archive. Appendix G lists the calculated end-to-end delay measurements for Scenario 3.

4.7 Malicious Activity Scenarios

4.7.1 Scenario 4 – Unencrypted Remote Logon Attempts.

Before the **trust system** and other security mechanisms were installed on the MPL SCADA network, an attacker, intent upon disrupting MPL's operations, first accessed an adjacent utility company's network through an unsecured rogue office connection to the Internet. Unknown to Sally Washington, the attacker then sent a spoofed e-mail to Sally's co-worker at MPL with a Valentine's Day card attachment using a compromised e-mail account and source IP address from the adjacent utility company. Because the source IP address and SMTP were allowed and e-mail attachments were not being scanned by the firewall or by antivirus software on the e-mail server or workstations at that time, the e-mail easily traversed the MPL firewall and was loaded to the MPL e-mail server. When the co-worker was logged onto shared

SCADA_admin_workstation_1 and opened the e-mail a malicious Trojan horse program was loaded onto the computer's hard drive. The malicious code could sniff and record keystrokes from the attached keyboard and from the Ethernet switch connecting the company control center workstations to the SCADA network. The installed malicious code, using a non-disabled FTP service on the workstation, forwarded the results each evening to the compromised computer in the adjacent company to which the attacker had remote administrator access. The sniffer captured keystrokes, including the username and local hashed password caches as workers logged on throughout the day, and reported them back to the compromised system for the attacker to extract.

Soon after this undetected incident, MPL management, concerned about improving the security of its operations after increasing reports of network intrusion attempts, had a trial **trust system** installed at a strategic location within its network and revised its security policies.

While scanning the Internet, the attacker came across MPL's external website which listed the names of company managers and technical support. Sally Washington was listed as the point of contact for SCADA technical matters with her e-mail address, smwashingt@middletownpl.com. The attacker then guessed, correctly, that Sally's network username might be the same, or at least similar, to the beginning of her e-mail address.

The next evening, the attacker attempted to logon remotely to the compromised MPL SCADA_admin_workstation1, with Sally's username, in an attempt to gain SCADA administrator privileges.

The first attempt was a remote logon from the compromised computer in MPL's neighboring company office using a common password, *password12*. The three packets crafted by the attacker were SYN (Packet 4-1) and ACK (Packet 4-3) control messages used in initiating and completing a three-way TCP handshake and the actual *logon_request* message (Packet 4-4), depicted in Figure 26.

```
MESSAGE TCP
//begin IPv6 outer header for tunnel mode
            …
            IP2_source_address            2001:DC99:31AC:9AAD:FC29:6C1A:80EA:4057
                                          //network_TS@adjacent_trust_router7
            IP2_destination_address       2001:DC98:5634:2110:BD1C:BA89:7325:4050
                                          //network_TS@MPL_trust_router2
            …
//end IPv6 outer header
…
//begin IPv6 inner header
            …
            IP1_source_address            2001:DC99:31AC:9AAD:FC29:6C1A:80EA:5231
                                          //adjacent1_office_workstation
            IP1_destination_address       2001:A344:4DD1:F76F:D2CB:3B09:5629:1000
                                          //MPL_logon_server
            …
//end IPv6 inner header
…
//begin message data
            message_type                  logon_request
            time_message_created          19:00:00.0000-30Jun07
            username                      smwashingt
            number_of_credentials         1
            credential_1_type             PASS
            credential_1                  password12
//end message data
…
```

Figure 26. Packet 4-4 (Remote Logon Attempt, Wrong Password and Unencrypted)

The attempt was detected by the MPL **trust system** *firewall rules encryption check* and blocked closest to the MPL WAN boundary because packets are required to be encrypted with the proper key. In this first attempt, the attacker was not even able to establish a connection with the logon server (the initial SYN packet was rejected) because the traffic was not encrypted.

In the second attempt, only packet 4-4 was sent, without establishing a connection, and was again rejected because it was not encrypted.

Each time a non-encrypted packet was received and rejected, the MPL **trust system** queried the adjacent company's **trust system** regarding whether the source IP address was properly encrypting its traffic (possibly needing to turn on encryption or update to the current key). The parameters defined for the *query_encryption* message are illustrated in Figure 27. The adjacent company's **trust system** would then query it's own nodes to determine the answer. If the result was that the source had the current key and was encrypting its traffic (which was the case), the adjacent company **trust system** (on its own, or prompted by the MPL **trust system**) would then query to determine if the source had actually sent the packet the MPL **trust system** claimed to have received.

```
MESSAGE UDP
//begin IPv6 header
            …
            IP1_source_address            2001:DC98:5634:2110:BD1C:BA89:7325:4050
                                          //network_TS@MPL_trust_router2
            IP1_destination_address       2001:DC99:31AC:9AAD:FC29:6C1A:80EA:4057
                                          //network_TS@adjacent1_trust_router7
            …
//end IPv6 header
…
 //begin message data
            message_type                  query_encryption
            time_message_created          …
            key_ID                        22:19:43.215-29Jun07
    //end message data
 …
```

Figure 27. UDP Encryption Check for Unencrypted Packet Source IP

The *query_response*, depicted in Figure 28, indicated that encryption was in effect at the node and the key was current.

```
MESSAGE UDP
//begin IPv6 header
        …
        IP1_source_address              2001:DC99:31AC:9AAD:FC29:6C1A:80EA:3201
                                        //nodal_TS@adjacent1_SCADA_master_station
        IP1_destination_address         2001:DC99:31AC:9AAD:FC29:6C1A:80EA:4057
                                        //network_TS@adjacent1_trust_router7
        …
//end IPv6 tunnel mode inner header
        …
//begin message data
        message_type                    query_response
        time_message_created            …
        encryption_on?                  yes
        key_current?                    yes
//end message data
```

Figure 28. UDP Response to Encryption Query

Next the adjacent company **trust system** queried the node to see if it had actually sent the packet. The parameters defined for a *query_packet* message, are depicted in Figure 29.

```
MESSAGE UDP
//begin IPv6 header
        …
 IP1_source_address                     2001:DC99:31AC:9AAD:FC29:6C1A:80EA:4057
                                        //network_TS@adjacent1_trust_router7
        IP1_destination_address         2001:DC99:31AC:9AAD:FC29:6C1A:80EA:3201
                                        //nodal_TS@adjacent1_SCADA_master_station
        …
//end IPv6 header
…
//begin UDP header
        …
  destination_port                      500
        protocol                        UDP
        …
//end UDP header
//begin message data
        message_type                    query_packet
        time_message_created            …
        rcvd_queryPacket_type           control
        queryPacket_dest_IP             2001:A344:4DD1:F76F:D2CB:3B09:5629:1000
        queryPacket_dest_port           500
        queryPacket_protocol            UDP
//end message data
…
```

Figure 29. Query to Verify the Source IP Actually Sent the Status Request

In a third attempt the source IP address was spoofed to look like an MPL address. The *query_response* from the system in the adjacent company's network that the attacker had pretended to be, indicated that it had not sent the packet, as depicted in Figure 30.

```
MESSAGE UDP
…
//begin message data
        message_type                    query_response
        time_message_created            …
        query_type                      query_packet
        query_time                      …
        response_1                      no
//end message data
…
```

Figure 30. UDP Response Identifying Source Did Not Send the Packet

In this case the source IP address in the attacker-generated packet was spoofed, so the **trust agent** of the system at that IP address responded to its **network trust system** that it had not sent the packet. Note that it is also possible in a network where logging of all transactions occurs to a historical database, for the **network-level trust system** to simply query this database without having to create unnecessary traffic to be processed by individual nodes and their trust agents. Now **trust systems** in both companies realized that malicious activity was occurring and began the process of tracking down the originating node for the traffic in order to block it closest to the source.

No *trust level* change was required because the real node was performing properly and existing **trust system** rules would block unencrypted traffic. Obviously, security responses that lower the *trust level* for any IP address or user could be leveraged by the attacker as a DoS against the adjacent company, by sending further spoofed packets to lower the *trust level* for a legitimate IP address or user. In this manner, it might be

expected that the **trust system** would eventually block all traffic, even legitimate packets, from the adjacent company IP address.

As an aside, note that this is where the advantage of multiple, collaborative **trust-enabled routers** can be brought to bear in increasing the intelligence of the overall **trust system**. The **trust system** realized that the actual system in the adjacent company configured with the source IP address it was seeing in the spoofed packets, was not actually creating and sending the spoofed packets it was seeing and correlated these events with the first attempts using the adjacent company's IP addresses.

The next step would be to track down the source of the spoofed packets. By sending a *track_source* packet out the interface from which the spoofed packets were being received, the MPL and adjacent company **trust system's** would query other **trust systems** (i.e. **trust routers**, **systems**, and **agents**) it was aware of down-the-line, to determine which other **trust systems** had also seen the packet and on which interface (i.e. link or links) it had arrived. The *track_source* would also inform them to block (i.e. update specific *firewall rules* to not allow the unauthorized traffic to a particular granularity) and initiate their own *track_source* for any further traffic of this type.

As the next **trust system** down the line received the *track_source* packet, it would check to see if that packet had crossed its path. Recognizing the packet and incoming interface, it would then send a *track_source* on that link to the next **trust system** or **systems**, which would in turn check to see if they had processed the same packet and, if so, track the source. Eventually, a **trust system** would respond back to the previous **trust system** that queried it. In the best case, it would state it had found the originating source, blocked the traffic, and alerted to the activity. In the worst case, it would indicate it had

no more reachable **trust systems** to query or those it could query all responded negatively and the trail had run cold. Even in this case, the updated *firewall rules* would block further similar activity at the level closest to the source and a better picture of the incoming avenue of attack could be determined. The event detail was logged to the historical database.

In addition to the log entry, a suspicious event was initiated, generating a security alert to the screen of security analysts and network administrators as depicted in Figure 31. Further event detail could be accessed and drilled into from the analyst GUI to the security database and historical databases.

```
SECURITY ALERT:
   SEID-13:30:34.1756-30Jun07

   INFRACTION/S          1) Encryption error—unencrypted connection attempt.
                         2) Attempted logon from external IP not allowed for
                         username smwashingt, role MPL_SCADA_administrator.
                         3) Malicious packet—packet-listed source_IP encryption
                         current and did not send packet.

   ACTION/S              Denied by MPL Firewall Rules.
                         Queried adjacent1 trust system.
                         Tracked source to 2001:DC99:31AC:9AAD:FC29:6C1A:80EA:5231.
                         //adjacent1_office_workstation
                         Adjacent trust system generated alert.

   PACKET DETAIL
   type                  control (SYN)
   time_message_created  …
   source_address        2001:DC98:5634:2110:BD1C:BA89:7325:3905
                         //MPL_SCADA_admin_workstation1
   source_port           9593
   dest_address          2001:A344:4DD1:F76F:D2CB:3B09:5629:1000
                         //MPL_logon_server
   dest_port
   protocol              TCP
```

Figure 31. Security Alert (Failed Remote Logon Event)

4.7.2 Scenario 5 - Encrypted Remote Logon Attempts with Compromised Key.

Realizing this would not be as easy as he thought, the attacker began capturing and analyzing network traffic outgoing from and incoming to the adjacent utility company's network. He recognized the communications between the two companies were all encrypted, so the only way he would be able to read packet data or connect to the MPL network would be to crack the encryption or get inside the network itself. The traffic captures previously reported by the installed sniffer on MPL SCADA_admin_workstation1 had also showed encrypted port 500 interactions between systems on MPL's network, indicating the use of IPsec. He began work to crack the key.

After considerable time he was able to crack the private encryption key for external communications and recognize the signature of key update packets. He also optimized the algorithms for encryption cracking so that shortly after a key change, the new key could be cracked in a matter of minutes. As he sniffed, decrypted, and studied traffic between the two companies, the attacker began to learn typical utility message types, node names, addresses, and common status values of MPL's equipment. After even more work he was able to crack the private key needed to spoof and encrypt packets that would be interpreted as either coming from the MPL or adjacent company networks.

Shortly after MPL had installed **trust systems** and conducted a complete security policy review, just prior to the attacker's latest attempts, the company had locked down unnecessary FTP services on its systems and denied external FTP connections. All systems were scanned to remove viruses and rootkits, including the attacker's Trojan. As a result, the attacker could no longer receive reports from the now deleted program that had been installed on MPL's SCADA_admin_workstation1.

Fortunately for the attacker, he still had the keystroke captures and encrypted password cache from an earlier time when Sally had logged on to the MPL network. He was able to run a cracking program against the hashed password and keystroke dumps he had captured when the Trojan was still active. After a few minutes, he was able to extract the decrypted password and waited for the opportunity to try again.

In the attacker's fourth attempt, the same packets were sent, this time properly encrypted. The *encryption check* passed but the *firewall rules module* noted a rules mismatch in its scorekeeper, because logon attempts from an external IP address (i.e. outside of the MPL network), indicated by the incoming interface and the source IP address in the packet, were not allowed by the MPL security policy**.** The activity was blocked at the MPL **trust system** and the adjacent company **trust system** was notified to update its *firewall rules* to block further logon attempts from its network into the MPL network. The *firewall_rules_update* request might have initiated an alert to the screen of the adjacent company's network security analysts to either approve or deny the requested rules change, in this manner providing a human-in-the loop review, instead of completely automated inter-company security configuration changes.

Finally, in a fifth attempt, the spoofed source IP was changed to reflect a legitimate MPL address from which user *smwashingt* might realistically attempt to logon internally to the MPL logon server. In this case, the packet was not received on the proper internal interface for that IP address (i.e. received on an external interface when MPL logon traffic should have been all internal) and was again rejected by the MPL **trust system** *firewall rules* at trust_router3.

Consider if trust router3 had not been there. The **network trust system** at trust_router2, receiving TCP control packets and a *logon_request* from an MPL corporate office IP address, might have been tempted to assume the activity to be legitimate (by its IP address and interface) and allowed the packets into the network, routing them to the logon server.

A quick **trust system** comparison of the actual traversal time of the packet (from send to receive timestamps) to the estimated travel time for a packet from the corporate office to reach the trust_router2 (based on distance and last congestion measurement) would have indicated the packet likely originated a much further distance away and would have been watched as suspicious.

Without the attacker being able to insert himself in the middle of the conversations, the logon server responses were routed to the source IP listed in the packets, an MPL node, which would have dropped them because it was not expecting them (i.e. it had no active connection with the logon server and had not sent a *logon_request*). A **trust agent** at that node would have recognized this activity as suspicious and alerted the **network trust system**.

Even if the attacker had gained physical or virtual access to MPL switches or links, and could perform a man-in-the middle attack, he needed the correct credentials for the logon to be approved.

What he did not know yet was that even with the correct username and password, he would not be granted a high enough ACCN to gain root-level access. Had MPL's security policy allowed Sally to simply logon with a separate root account password for higher level privileges, the attacker's captures and password cracking program would

have provided the tools necessary to steal her password and gain root access. In contrast, with the **trust system**'s credential credibility based access control, the quantity and credibility of logon credentials is used in determining a user's *access level*. The attacker had no easy way to spoof Sally's biometric or smart card credentials and could not use these to gain root-level access.

4.7.3 Scenario 6 – False Status Update.

Having failed at a logon attempt, with the intent to still exhibit remote control of the MPL network, the attacker turned his attention to studying SCADA protocol documentation gleaned from numerous technical papers and vendor websites on the Internet. From his review, and after sniffing and cracking MPL's inter-company traffic, the attacker recognized the communications protocols for MPL's SCADA updates and other operational messages.

The attacker crafted a false *status* message, Figure 32, to test his newly found expertise and attempt to direct emergency actions on the MPL SCADA network. He spoofed the adjacent company's master control station IP address and sent the message using TCP with the emergency flag set.

```
MESSAGE TCP
//begin IPv6 outer header for tunnel mode
        IP2_traffic_class             11A0
        IP2_flow_label                124C7
        IP2_source_address            2001:DC99:31AC:9AAD:FC29:6C1A:80EA:4057
                                      //network_TS@adjacent1_trust_router7
        IP2_destination_address       2001:DC98:5634:2110:BD1C:BA89:7325:4050
                                      //network_TS@MPL_trust_router2
//end IPv6 outer header for tunnel mode
…
//begin IPv6 inner header for tunnel mode
        IP1_traffic_class             32EF
        IP1_flow_label                AA89C
        IP1_payload_length            101
        IP1_source_address            2001:DC99:31AC:9AAD:FC29:6C1A:80EA:3200
                                      //adjacent1_SCADA_master_station
        IP1_destination_address       2001:DC98:5634:2110:BD1C:BA89:7325:3200
                                      //MPL_SCADA_master_station
//end IPv6 inner header for tunnel mode
//begin TCP header
        …
        TCP_source_port               500
        TCP_destination_port          500
        …
//end TCP header
//begin message data
        message_type                  status
        time_message_created          09:00:00.0000-1Jul07
        busNumber                     3378
        busName                       PARKVIEW
        CAname                        CA1
        companyName                   C
        nominalVoltageKV              +0211.000
        busVoltPu                     +0001.084
        VoltKV                        +0137.581
        busAngleDeg                   +0013.790
        loadMW                        +0017.610
        loadMvar                      +0320.740
        gen_MW                        -0236.740
        genMvar                       +0234.020
        switchedShuntsMvar            +0200.000
        actGshuntMW                   +0009.110
        actBshuntMvar                 -0006.760
//end message data
…
```

Figure 32. Packet 6-1 (Status Message with Spoofed Adjacent Source IP)

The packet looked legitimate and passed all **trust system** checks except the time check. Status updates from that company were normally forwarded every 4 secs. This one was early. For reliability, the input was matched to the last status from the company, which indicated a tremendous jump in values. Input from CA1 was also expected. The last *area_status* from CA1 indicated no such emergency conditions. The final straw was an actual *status* from the adjacent company, on-time, indicating no emergency situation.

The MPL **trust system** then queried the adjacent company's **trust system** to verify its master station had sent the first emergency *status* message. It replied negatively and together they began tracking and blocking the source.

The attacker could see this interaction and, though not successful in causing the reaction he'd hoped, now he had the means to initiate connections to the MPL network, but had to re-crack the key after each daily key update. The **trust systems** were also blocking his activity now from the compromised adjacent company workstation.

4.7.4 Scenario 7 - Work Schedule Mismatch.

The next evening, the attacker traveled to a nearby remote, unattended substation owned by MPL. MPL had purchased security cameras and motion detectors to monitor for break-ins to the substation yard but had not yet installed them. The attacker was able to climb the fence into the substation with his laptop and found an unlocked door through which he could access one of the company's IEDs and the data concentrator. MPL was still in the process of implementing its security policy and the SCADA administrators were currently disabling all dial-up connections and logons previously allowed through terminal ports on substation equipment. In this manner, vendor representatives or MPL administrators were now required to either physically log on to a computer within the MPL network or be granted access (i.e. after a terminal port was re-activated) by an administrator from the operations center, to allow a direct computer connection to IEDs and other SCADA nodes.

After attempting the first terminal port, which had been disabled by the SCADA administrator, the attacker found another port that had not been disabled. He was able to

connect his laptop to the IED but found he was not allowed to logon directly to the system, instead the IED **nodal trust agent** displayed a banner explaining that he was required to first logon to the network to gain access to the data and code on the IED. An IED **trust agent** was capable of forwarding *logon_requests*, on behalf of a connected user, to the network logon server for authentication and accountability of actions. The attacker's first attempt failed because MPL was now using a different key for internal communications than it used for external communications between MPL and other utility organizations; however, it was using the same authentication and encryption protocols and mode.

The attacker was prepared, and had the encryption cracking program loaded on his laptop. After a few minutes of effort in cracking the internal encryption key, the attacker crafted connection requests and a *logon_request* message, Packet 7-4, displayed in Figure 33. He then encrypted it, and forwarded it to the MPL network logon server in an attempt to logon with Sally's username and password. He did not have any other credentials he could supply.

```
MESSAGE TCP
/begin IPv6 tunnel mode outer header
    …
    IP2_source_address            2001:DC98:5634:2110:BD1C:BA89:7325:4239
                                  //nodal_TS@MPL_IED-239
    IP2_destination_address       2001:DC98:5634:2110:BD1C:BA89:7325:4050
                                  //network_TS@MPL_trust_router2
    …
//end IPv6 tunnel mode outer header
//begin AH header
    …
//end AH header
//begin ESP header
    …
//end ESP header
//begin IPv6 tunnel mode inner header
    …
    IP1_source_address            2001:DC98:5634:2110:BD1C:BA89:7325:3908
                                  //attacker_laptop_with_spoofed_MPL_IP_and_MAC
    IP1_destination_address       2001:A344:4DD1:F76F:D2CB:3B09:5629:1000
                                  //MPL_logon_server
    …
//end IPv6 tunnel mode inner header
    …
//begin message data
    message_type                  logon_request
    time_message_created          20:00:00.000-2Jul07
    username                      smwashingt
    number_of_credentials         4
    credential_1_type             PASS
    credential_1                  !#V8k12g4x
//end message data
//begin ESP trailer
    …
//end ESP trailer
//begin ESP trailer
    …
//end ESP trailer
```

Figure 33. Packet 7-4 (After Hours Logon Request from Substation IED)

The logon server validated the two credentials and notified the **trust system**. The **trust system** checked the logon time against Sally's work schedule, depicted in Table 25, and identified that she was not scheduled to work during that shift. As a result, a suspicious event was initiated and a security alert generated, as depicted in Figure 34.

Table 25. Trust System Work Schedule File Entry.

Username	**Date**	**Start**	**Stop**	**(+/-)**
smwashingt	2Jul07	08:00	18:00	00:35

```
SECURITY ALERT:
          SEID-20:00:03.0207-2Jul07
          INFRACTION/S                  Logon not authorized—work schedule mismatch.
          ACTION/s                      Denied by NTS ACM.
          PACKET DETAIL
                 type                   logon_request
                 time_message_created   20:00:00.0000-2Jul07
                 source_address         2001:DC98:5634:2110:BD1C:BA89:7325:0239
                 source_port            500(ISAKMP)
                 dest_address           2001:A344:4DD1:F76F:D2CB:3B09:5629:1000
                 dest_port              500(ISAKMP)
                 protocol               TCP

SUSPICIOUS EVENT LOG:
   ------------SEID-20:00:03.0207-2Jul07(NETWORK TRUST SYSTEM)-------------
   UPDATE-20:00:03.5341-2Jul07 (All times in seconds)
          tracker/s                     2001:DC98:5634:2110:BD1C:BA89:7325:0239
                                        smwashingt
          message_type                  logon_request
          ____________________________________________________________________
          ACTIONS
          Logon denied by ACM
          ____________________________________________________________________
          TIME CHECK

          sent                          …
          received                      …
          incoming delay                …                              [PASSED]
          ____________________________________________________________________
          FIREWALL RULES CHECK

          rule_matched    …
          source_IP       2001:DC98:5634:2110:BD1C:BA89:7325:0239      [PASSED]
          dest_IP         2001:A344:4DD1:F76F:D2CB:3B09:5629:1000      [PASSED]
          destPort        500                                          [PASSED]
          ____________________________________________________________________
          CHECKSUM CHECK

          checksum                      010111010101111                [PASSED]
          ____________________________________________________________________
          FORMAT CHECK

          message_type                  logon_request                  [PASSED]
          time_message_created          20:00:00.000-2Jul07            [PASSED]
          username                      smwashingt                     [PASSED]
          number_of_credentials         4                              [PASSED]
          credential_1_type             PASS                           [PASSED]
          credential_1                  **********                     [PASSED]
          ____________________________________________________________________
          LOGON CREDENTIALS CHECK

          time_logon_attempted          20:00:00.000-2Jul07
          username                      smwashingt                     [PASSED]
          password                      **********                     [PASSED]
          logon_ACCN                    2
          ____________________________________________________________________
          ACM CHECK

          work_schedule_day             2Jul07                         [PASSED]
          work_schedule_time            07:25-18:35                    [FAILED]
          trust level                   -0                             [PASSED]
          effective_ACCN_assigned       0
          ____________________________________________________________________
          SANITIZATION          Not required.
```

Figure 34. Work Schedule Mismatch Warning and Denied Logon

During the day, a message would have been sent to a logged on network administrator, requesting approval or denial of the logon; however, with no logged on administrators (or if a timely response was not received), the logon was denied by the **trust system** with an effective ACCN of 0 assigned. The attacker would have to try again when Sally was scheduled to work.

The security alert and log results would be sufficient to indicate malicious activity, especially if Sally were questioned the next day to verify if she had tried to logon to the substation IED after hours. Recognizing the malicious attempt would prompt the network administrator to require an immediate password change for her account before allowing her to logon again, further complicating the attacker's attempt to use Sally's account. However, let's assume that the event was not caught or reacted to quickly enough.

The attacker left the substation in a fury and stormed through the empty parking lot toward his car. Noting something on the ground, he picked it up, and squinted reading the small print of the card, depicted in Figure 5.

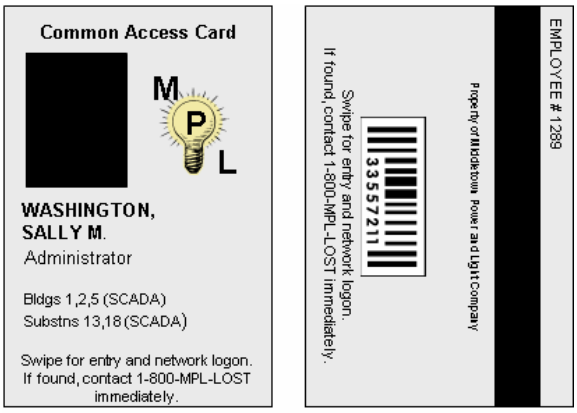


Figure 35. Front and Back, Respectively, of Administrator Smart Card

4.7.5 Scenario 8 - Malicious Simultaneous Logon.

The next morning the attacker attempted access from the substation once again. Sally was working that day and the attacker's logon occurred after Sally had already logged onto the network from SCADA_admin_workstation1 with the correct password and fingerprint scan. With these credentials, the **trust system** had assigned her an effective ACCN of 4, root-level access, as a SCADA_administrator. For some unknown reason, she hadn't been able to find her smart card that morning and assumed she may have accidentally left it at work or dropped it during her trip to the substation the afternoon before.

That morning the attacker again supplied the correct username and password credentials, which were validated by the logon server as depicted in Figure 36.

```
MESSAGE TCP
//begin IPv6 header
    …
    IP1_source_address            2001:A344:4DD1:F76F:D2CB:3B09:5629:1000
                                  //MPL_logon_server
    IP1_destination_address       2001:DC98:5634:2110:BD1C:BA89:7325:0239
                                  //MPL_IED-239
    …
//end IPv6 header
//begin AH header
    …
//end AH header
//begin ESP header
    …
//end ESP header
//begin message data
    message_type                  logon_evaluated
    time_message_created          08:45:00.000-3Jul07
    username                      smwashingt
    number_of_credentials         1
    credential_1_type             PASS
    credential_1_pass             YES
//end message data
//begin ESP trailer
    …
 //end ESP trailer
 //begin ESP ICV
    …
 //end ESP ICV
```

Figure 36. Packet 8-4 (Credentials Evaluation for Second IED Logon Attempt)

In addition, the **trust system** recognized a previous and still active logon by the same username at another IP address, SCADA_admin_workstation1. The simultaneous logon attempt prompted the initiation of a *suspicious event* and a *query_simultaneous_logon* message (Figure 37) forwarded to SCADA_admin_workstation1, where the same username had logged on previously.

```
MESSAGE UDP
//begin IPv6 header
    …
    IP1_source_address            2001:DC98:5634:2110:BD1C:BA89:7325:3210
                                  //network_TS@MPL_trust_router2
    IP1_destination_address       2001:DC98:5634:2110:BD1C:BA89:7325:4051
                                  //nodal_TS@MPL_SCADA_Workstation1
//end IPv6 header
//begin AH header
    …
//end AH header
//begin ESP header
    …
//end ESP header
//begin UDP header
    …
    destination_port              500
    protocol                      UDP
    …
//end UDP header
//begin message data
    message_type                  query_simultaneous_logon
    time_message_created          …
    username                      smwashingt
    logon_IP                      2001:DC98:5634:2110:BD1C:BA89:7325:0239
                                  //MPL_IED-239
    on_behalf_of                  2001:DC98:5634:2110:BD1C:BA89:7325:3908
                                  //attacker_laptop_with spoofed_MPL_IP_and_MAC
    effective_ACCN_assigned       2
//end message data
//begin ESP trailer
    …
//end ESP trailer
//begin ESP ICV
    …
//end ESP ICV
```

Figure 37. Simultaneous Logon Query Message to First Logged-on User

The alert illustrated in Figure 38 was displayed on the screen of SCADA_admin_ workstation1.

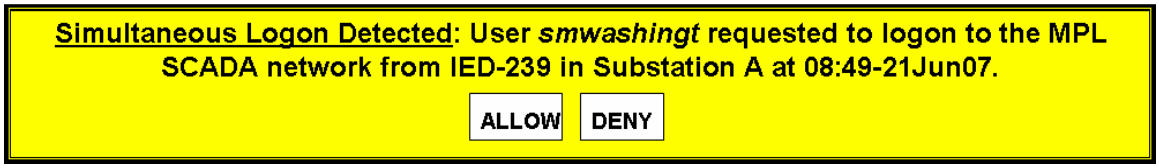


Figure 38. Simultaneous Logon Alert Displayed at SCADA_admin_workstation1

At that instant, Sally was away from her desk and did not see the message. When no response was received within the **trust system's** time-to-wait threshold (set to 15 seconds), the simultaneous logon was allowed and a *logon_approved* message was sent to the source IP address from which the attacker's request was initiated.

However, with only username and password supplied, the **trust system** assigned an *effective ACCN* of 2 to this second logon attempt by username *smwashingt*, granting only basic user access and not the SCADA administrator role the attacker desired.

The IED-239 **trust agent** updated its *nodal ACM* with the approved username and *effective ACCN*, then granted access to the attacker. The attacker was thrilled when his logon was approved but soon found he was only granted full access to office automation tools and e-mail and read-only access to web pages, as a basic user, but no administrative rights. Furthermore, as he perused directories he could view network common drives, but he was prevented from viewing any of the node's operational data and code files or any

other SCADA network systems or tools, the folders of which were all denied read access (including viewing their existence) due to his low *effective ACCN*.

Even if the attacker had attempted to use the smart card before Sally notified the network administrator she had lost hers (and the network administrator disabled smart card logon credentials until it was found or replaced), without the PIN, the attacker could only be assigned a *logon ACCN* of 3 with the correct username, password, and card, which, according to the MPL **trust system** *ACM* of Table 14, would allow him read-only access to operational and emergency data and code and execute access to tools, but not the ability to modify, copy, or delete either data type. Although this limited administrator privilege is primarily to allow a legitimate non-elevated employee to perform basic administrator/operator functions quickly in emergency situations, it does not allow full administrative privileges which would be much more devastating in the hands of an attacker.

Denied his desired administrative privilege on the SCADA network, the attacker quickly perused the MPL intranet pages and discovered the name of another MPL SCADA administrator. He sent an elevation request (i.e. a *query_elevation* message) with the parameters depicted in Figure 39.

```
MESSAGE UDP
//begin IPv6 tunnel mode outer header
    …
    IP2_source_address            2001:DC98:5634:2110:BD1C:BA89:7325:4239
                                  //nodal_TS@MPL_IED-239
    IP2_destination_address       2001:DC98:5634:2110:BD1C:BA89:7325:4050
                                  //network_TS@MPL_trust_router2
    …
//end IPv6 tunnel mode outer header
//begin AH header
    …
//end AH header
//begin ESP header
    …
//end ESP header
 //begin IPv6 tunnel mode inner header
    …
    IP1_source_address            2001:DC98:5634:2110:BD1C:BA89:7325:0239
                                  //MPL_IED-239
    IP1_destination_address       2001:DC98:5634:2110:BD1C:BA89:7325:3905
                                  //MPL_SCADA_admin_workstation5
//end IPv6 tunnel mode inner header
//begin UDP header

    …
    destination_port              500
    protocol                      UDP
    …
 //end UDP header
 //begin message data
    message_type                  query_elevation
    time_message_created          …
    username                      smwashingt
    effective_ACCN_assigned       2
    requested_ACCN                4
    note                          I forgot my card today. Thanks.
 //end message data
 //begin ESP trailer
    …
 //end ESP trailer
 //begin ESP ICV
     …
 //end ESP ICV
```

Figure 39. Elevation Request Message from the Attacker to a SCADA Administrator

When the second SCADA administrator received the *elevation_request* message on his desktop, he had not visually identified Sally. Since she was in a different building but he could recognize her voice, he called her desk to make sure it was really her attempting to elevate her privileges without supplying all of the required credentials.

When the phone rang, Sally was just returning and picked up the call. She confirmed she had not initiated the request. The second SCADA administrator promptly denied the elevation request as depicted in Figure 40.

```
MESSAGE UDP
…
//begin message data
        message_type                query_response
        time_message_created        …
        query_type                  query_elevation
        query_time                  …
        response1                   no
//end message data
…
```

Figure 40. Message Denying Attacker's Elevation Request

The **trust system** initiated a *suspicious event* and continued monitoring for any more related suspicious activity. Meanwhile, Sally noticed the simultaneous logon message still displayed on her screen. She immediately selected DENY, which sent the message shown in Figure 410 to the **trust system**.

```
MESSAGE UDP
…
//begin message data
        message_type                query_response
        time_message_created        …
        query_type                  query_sim_logon
        query_time                  …
        response1                   no
//end message data
…
```

Figure 41. Denial of Simultaneous Logon by the True User

The **trust system** notified the logon server to logoff the second logon by username smwashingt, disconnecting the attacker and updating the network-level **trust system** *ACM* and that of IED-239 not to allow further logon attempts by that username

from that source IP address. A *security alert*, Figure 42, was then generated by the **trust system.**

```
SECURITY ALERT-…
   SEID                 …
   ACSE                 Simultaneous logon denied.
   INFRACTION           Username smwashingt at SCADA_admin_workstation1 denied
                        simultaneous logon at IED-239.
                        Simultaneous logon active for … sec.
   ACTION               Second logon disconnected by network_TS@trust_router2.
                        Automated simultaneous logon by smwashingt denied in ACM
                        until reinstated.
   PACKET DETAIL
     type               query_response
     source_address     2001:DC98:5634:2110:BD1C:BA89:7325:3901
                        //MPL_SCADA_admin_workstation1
     source_port        500(ISAKMP)
     dest_address       2001:DC98:5634:2110:BD1C:BA89:7325:4050
     dest_port          500(ISAKMP)
     protocol           UDP
     sim_logon_IP       2001:DC98:5634:2110:BD1C:BA89:7325:0239
                        //MPL_IED-239
     on_behalf_of       2001:DC98:5634:2110:BD1C:BA89:7325:3908
                        //attacker_laptop_with spoofed_MPL_IP_and_MAC
```

Figure 42. Security Alert for Malicious Simultaneous Logon

Failing at all attempts over the last few days, the attacker abandoned his plot to disrupt MPL's operations and turned his attention to easier targets in other companies that had not implemented such comprehensive rings of defense.

4.7.6 Scenario 9 - Disgruntled Employees.

Installation of IEDs, high-speed fiber optic links, and the **trust system's** additional security measures had increased MPL's efficiency and security, reducing the need for as many SCADA administrators. As a result, two employees with poor performance records (who were only kept around because of their close-held knowledge about the legacy systems) were notified in advance that they would be let go. This was to be their last day. Angry, they had been plotting together, over the last week, to steal company-sensitive financial and network configuration data they could sell for profit.

They also planned to sabotage the SCADA network with false data, hoping to cause a local blackout that might cost MPL hundreds of thousands of dollars in revenue this month. The individuals were aware that the company security policy required their accounts to be immediately disabled the very afternoon of their last day, just after leaving the building. As administrators, still in possession of a smart card and able to provide biometric credentials in addition to a PIN, on their final day they were still authorized to logon with full root-level privileges.

One individual, an IT_network_administrator, attempted to steal a particularly negative quarterly financial forecast and overdue maintenance records, which if made public, might hurt the company's reputation and potential value of company stocks. He also planned to download network diagrams, password files, and configuration settings that would be valuable to US or international hackers seeking to exploit utility networks.

He searched the common drives and found the quarterly report data which was viewable to his *role*. He then attempted to copy it to a thumbdrive, whereby his workstation sent the packet shown in Figure 43, a *copy* request message, to the common drive server hosting the file.

```
MESSAGE TCP
…
//begin message data
    message_type           operation_request
    time_message_created   …
    username               bearnold
    operation_type         copy
    file                   L:\Finance\QuarterlyReports\Jul-Sep\FinancialForecast.ppt
//end message data
…
```

Figure 43. Insider's Request to Copy File FinancialForecast.ppt

The **trust system,** checked the administrator's *role* and *effective ACCN* against the *ACM* and found he was authorized to read but not copy this data. In addition, as an IT network administrator, he did not have permissions to change the **trust system** *ACM* settings, as a *security administrator role* would have had. Figure 44 illustrates the denial message displayed to the disgruntled employee's screen.

Operation Request Denied: Username *bearnold* not authorized to copy company-sensitive financial data. If you believe this restriction is in error, contact net-security-team@mpl.com.

Figure 44. Denial Message for Copy Attempt

Next he found the folders containing network diagrams and the logon server's password cache. He did have authority, by his *role*, to copy these and was able to download them to his thumbdrive using the requests depicted in Figures 45 and 46.

```
MESSAGE TCP
…
//begin message data
        message_type                    operation_request
        time_message_created            …
        username                        bearnold
        operation_type                  copy
        from_file                       L:\IT\Diagrams\LAN_Diagram(current).vsd
                                        //common drive
        from_file_data_type             ND
                                        //network data
        from_file_caveat                company-sensitive
//end message data

MESSAGE TCP
…
//begin message data
        message_type                    operation_request
        time_message_created            …
        username                        bearnold
        operation_type                  paste
        from_file                       L:\IT\Diagrams\LAN_Diagram(current).vsd
                                        //common drive
        to_file                         F:\Copy of LAN_Diagram(current).vsd
                                        //removable drive
//end message data
```

Figure 45. Insider's Copy and Paste of the Network Diagram File

```
MESSAGE TCP
…
//begin message data
        message_type                    operation_request
        time_message_ceated             …
        username                        bearnold
        operation_type                  copy
        from_file                       C:/etc/passwd\MPLpw.txt
        from_file_data_type             ND
                                        //network data
        from_file_caveat                restricted-release
//end message data


MESSAGE TCP
…
//begin message data
        message_type                    operation_request
        time_message_created            …
        username                        bearnold
        operation_type                  paste
        from_file                       C:\etc\passwd\MPLpw.txt
                                        //logon server password file
        from_file_data_type             ND
        from_file_caveat                restricted-release
        to_file                         F:\Copy of C:\etc\passwd\MPLpw.txt
                                        //removable drive
//end message data
```

Figure 46. Insider's Copy and Paste of the Password File

However, all actions were logged to the historical database for which he did not have permissions to modify. Next he attempted to e-mail them to his home e-mail account, sending the Packet depicted in Figure 47.

```
MESSAGE TCP
…
//begin message data
        message_type                    e-mail
        time_message_created            …
        username                        bearnold
        To                              hackersblog@yoohoo.com
        Cc                              ihatemycompany@snotmail.com
        Bcc                             jwboothe@homenetwork.net
        Text                            m@dH@k3r, I got the initial $5000 check, so
                                        here's  the LAN diagram and password file for
                                        MPL as promised! I expect 50% of the highest
                                        bid when this gets posted on your site.
                                        -benedict
        number_of_attachments           2
        attachment_1                    F:\Copy of  LAN_Diagram(current).vsd
                                        //copied network diagram
        attachment_1_dataType           ND
        attachment_2_caveat             company-sensitive
        attachment_2                    F:\Copy of  C:\etc\passwd\MPLpw.txt
                                        //copied password file
        attachment_2_dataType           ND
        attachment_2_caveat             restricted-release
        e-mail_dataTypes                ND
        e-mail_caveat                   restricted-release
//end message data
…
```

Figure 47. Disgruntled Employee's First E-mail Attempt

The **trust system** inspected the message and found the attachments. When it checked the *data type* against usernames associated with the sender and receiver e-mail accounts it determined that these files contained company-sensitive data not authorized for release outside the company network, so the e-mail was blocked. The log entry and security alert pictures in Figure 48 were generated.

```
SECURITY ALERT-…
   SEID                          …
   ACSE                          Release-restricted data not authorized to leave
                                 company network.
   INFRACTION                    User bearnold attempted to e-mail release-
                                 restricted, ND attachment to unauthorized
                                 recipient/s.
   ACTION                        Attachment stripped from e-mail by ACM.
   PACKET DETAIL
         type                    e-mail
          source_address         2001:DC98:5634:2110:BD1C:BA89:7325:3901
                                 //MPL_network_admin_workstation1
          source_port            500(ISAKMP)
          dest_address1          hackersblog@yoohoo.com
          dest_address2          ihatemycompany@snotmail.com
          dest_address3          jwboothe@homenetwork.net
          dest_port              500(ISAKMP)
          protocol               TCP
          number_of_attachments  2
          attachment_1           F:\Copy of LAN_Diagram(current).vsd
          attachment_1_dataType  ND
          attachment_2_caveat    company-sensitive
          attachment_2           F:\Copy of C:\etc\passwd\MPLpw.txt
          attachment_2_dataType  ND
          attachment_2_caveat    restricted-release
          e-mail_dataTypes       ND
          overall_e-mail_caveat  restricted-release
```

Figure 48. Security Alert and Log Entry for Blocked E-mail

The administrator then changed the names of the files, re-attached them, and attempted to resend them as shown in Figures 49 and 50.

```
MESSAGE TCP
…
//begin message data
    message type           operation_request
    time_message_created   …
    username               bearnold
    operation_type         modify
    attribute              filename
    from                   F:\Copy of  LAN_Diagram(current).vsd
                           //copy of network diagram (original name)
    to                     F:\picture.vsd
                           //new name of file
    data type              ND
    caveat                 company-sensitive
//end message data
…
```

```
MESSAGE TCP
…
//begin message data
    message type           operation_request
    time_message_created   …
    username               bearnold
    operation_type         modify
    attribute              filename
    from                   F:\Copy of C:\etc\passwd\MPLpw.txt
                           //copy of password file (original name)
    to                     F:\moneymaker.txt      //new name of file
    data_type              ND
                           //same file, so data_type remains (un-editable)
    caveat                 restricted-release
                           //same file, so data type remains (un-editable)
//end message data
…
```

Figure 49. File Name Changes on Files Copied to Thumbdrive

```
MESSAGE TCP
…
//begin message data
    message_type           e-mail
    time_message_created   …
    username               bearnold
    To                     ihatemycompany@snotmail.com
    Cc                     hackersblog@yoohoo.com
    Bcc                    jwboothe@homenetwork.net
    text                   m@dH@k3r, I got the initial $5000 check, so here's the LAN
                           diagram and password file for Company A as promised! I
                           expect 50% of the highest bid when this gets posted on
                           your site.  -benedict
    number_of_attachments  2
    attachment_1           F:\picture.vsd
                           //copied network diagram
    attachment_1_dataType  ND
    attachment_2_caveat    company-sensitive
    attachment_2           F:\moneymaker.txt
                           //copied password file
    attachment_2_dataType  ND
    attachment_2_caveat    restricted-release
```

Figure 50. Insider's Second Outgoing E-mail Attempt with File Names Changed

At this time the **trust system** was only configured to prevent inadvertent disclosures, however, simply changing the filename of a copy of an existing file, that had already been assigned a *data type*, did not change the file's assigned *data type*. Again the e-mail was blocked. The administrator then removed his thumb drive. A workstation-level **trust agent** might have generated a message to the administrator's screen asking if

he meant to download company-sensitive data or generating a *security_alert*. In such a case, even clicking yes and proceeding with the theft or modifying the contents slightly and renaming, the file download would still be logged to the historical database.

In this case the **trust system,** updated the *suspicious event,* generated a new *security_alert* with the second failed attempt details, and lowered the *trust level* for the username**.** An analyst seeing the *security_ alert* event might have immediately recognized the potential harm and stopped the theft right away. Let's assume this did not happen immediately, but all actions were recorded and viewable after-the-fact.

The second disgruntled employee, a SCADA administrator, was authorized to successfully download current SCADA configuration files. Had he been assigned any other *role*, he would not have had these privileges. In this case, the **trust system** was not configured to alert for *copy* actions on sensitive-data by an employee on his last day, which would have alerted security analysts of suspicious activity, however, his actions were also logged by the historical database.

The next morning, reviews of the previous day's logs indicated the activity by the administrators and they were greeted by law enforcement at their residences.

4.8 Chapter Summary

Chapter 4 has demonstrated **trust system** functionality and security enhancement in the face of various benign and malicious scenarios. In each case, the **trust system** and encryption simulations performed within acceptable time threshold requirements indicating the potential for general implementation of the **trust system** on near real-time utility communication architectures and carefully application to some real-time utility

networks. Congestion was assumed to be prevented through bandwidth management but was not simulated. A successful implementation will require bandwidth and QoS guarantees, which will add additional overhead and delay to the scenarios.

V. Conclusions and Recommendations

5.1 Chapter Overview

This chapter summarizes the research findings and applicability of employing collaborative, situationally-aware **trust agents** to manage security mechanisms such as IPsec encryption, format inspection, and trust-based access control over time-constrained Utility Intranet communications, both in local and wide-area interactions. It concludes by recommending areas for follow-on research.

5.2 Conclusions of Research

This thesis indicates that the implementation of the proposed **trust system** inspections add minimal overhead to communications and can reasonably be applied to near real-time requirements. These mechanisms were shown to perform well in the face of determined attacks. It also shows that a mix of UDP and TCP traffic can deliver notifications that meet the majority of expected utility SCADA and wide-area protection systems. In ideal, uncongested cases, they can even meet hard real-time response thresholds, but must be augmented by strict bandwidth guarantees and maintain the state of on-going events to prevent the negative effects of TCP congestion control and UDP unreliable delivery on critical communications. While the implications of the proposed **trust system** hold great promise for the electric power grid and other utilities, they are certainly even more appropriate for many other networks that can afford less-strict delivery requirements.

5.3 Significance of Research

In a very difficult and not well understood area of communications, where security solutions are still in their infancy, this research is a step forward in defining a unique, defense-in-depth capability for an industry that has been slow to understand and accept their increasing vulnerability to digital avenues of disruption. The community has been even slower to learn new concepts and embrace the greater priority and corporate-dedication required to keep operations running smoothly and prevent potential catastrophic consequences from network attacks in the coming years.

This research is important in debunking the myth that security mechanisms cannot be applied to SCADA systems, yet it does reveal the added complexity of such endeavors, where mistakes are unforgiving and can cripple industrial processes and risk human life. Nevertheless, the old paradigm of ignoring network security in order to keep process control and emergency reaction simple must be left behind and will require a great degree of corporate and utility community investment in technologies, unique network administration skillsets, network planning, testing, and routine training programs (covering topics such as network technologies, attacker capabilities, and security essentials) to continuously assess and refine the security posture of utility organizations.

This research also points to the increased safety that can result across the grid through the secure sharing of information, facilitated by the **trust system**.

5.4 Recommendations for Action

It is the opinion of this author, that immediate action should be taken to develop an ideal security architecture for the national power grid and that a national level agency

should gather and manage detailed system and infrastructure requirements at levels higher than individual companies, enforcing both reliability and security standards at the same high level for all manufacturers. A national utility communications simulator should be constructed to thoroughly test new configurations and industry patches before deploying them to the national power grid.

An incentive must also be provided for developers of power equipment to gain the appropriate network security expertise and for utilities to incorporate security into their architectures. One way is to enforce a certification program for utilities that ranks them according to their performance, efficiency, security posture, incident response and prevention, innovation, and environmental impact. This program would serve to increase healthy competition in the areas that will benefit the country in its security, energy independence, and health for new generations. A certification program should also give consumers a choice in their providers, increasing the incentive for companies to transform their operations.

5.5 Recommendations for Future Research

Follow-on research is required in four main areas. First, IP-based security mechanisms are highly dependent on bandwidth and QoS guarantees, which will also add additional negotiations and processing not accounted for in this thesis. A study and incorporation of bandwidth management capabilities like Multi-Protocol Layer Switching (MPLS) into this scheme is required.

Second, additional testing with encryption schemes at IP network, and application layers is needed to define which combination of software, hardware, and protocol-level encryption and key distribution schemes are most appropriate.

Third, a more detailed integration of IEC61850 protocol message formats is needed to accurately test exact time delays of future implementations over electric utility communications networks.

Finally, the **trust system** simulator code should be optimized and integrated into a robust network simulator such as Network Simulator 2 (NS/2) or OPNET, with more robust scenarios of power events and network penetration attempts incorporated into **trust system** scenarios. As a starting point, Hopkinson, et al., have already designed a simulation engine known as Electric Power and Communication synCHronizing Simulator (EPOCHS) that integrates NS/2 with a power system simulator.

An initial goal of the simulator development for this paper was to provide a generic tool into which the specific communication parameters of each individual system or network device could be entered and modified, depending on the actual or proposed topology of a company's network. The simulator code provided generic constant-parameter place-holders for each device and medium (i.e. quantity of devices, link length and signal speed in medium, message size, parameters for propagation and transmission delay calculations, router queue size and processing delays) which could be modified in the future to accurately reflect vendor-provided performance statistics of that company's network as newer technologies become available. The implementation for these calculations was rudimentary (generic formulas and delay summation). Greater performance granularity could be achieved with a network simulator such as OPNet or

NS/2, which can simulate real-world performance of specific vendor routers and other network devices and introduce varying background traffic loads.

Realistic scenarios could easily be implemented using the actual New York Power Pool (NYPP) bus data that has been incorporated into EPOCHS files. A cascading blackout scenario, similar to what occurred in the Northeast US in 2003, could be recreated to show that the **trust system** is able to prevent the cascade and to measure the time required for breakers to trip with the security mechanisms in place, a mix of TCP and UDP trip messages, link drops, and varying background traffic loads. Also measured should be the time required to notify and receive responses from control areas, regional coordinators, and regional control centers.

To provide a sample of realistic data, the NYPP file could be used to simulate multiple, interconnected SCADA networks. Specifically the busses and loads might be arbitrarily divided into eight different zones (A-H) of roughly the same size, as depicted in Figure 51. Each area, in this case would represent a different utility company and two or three utility companies would comprise a single control area (CA) with a reliability coordinator and ISO, for a total of three CAs. Although, the actual NYPP is organized under a single ISO, the NYPP is unique in this regard. This selection is realistic since most other states of comparable size are comprised of multiple ISOs overseeing one, two, or three utility companies.

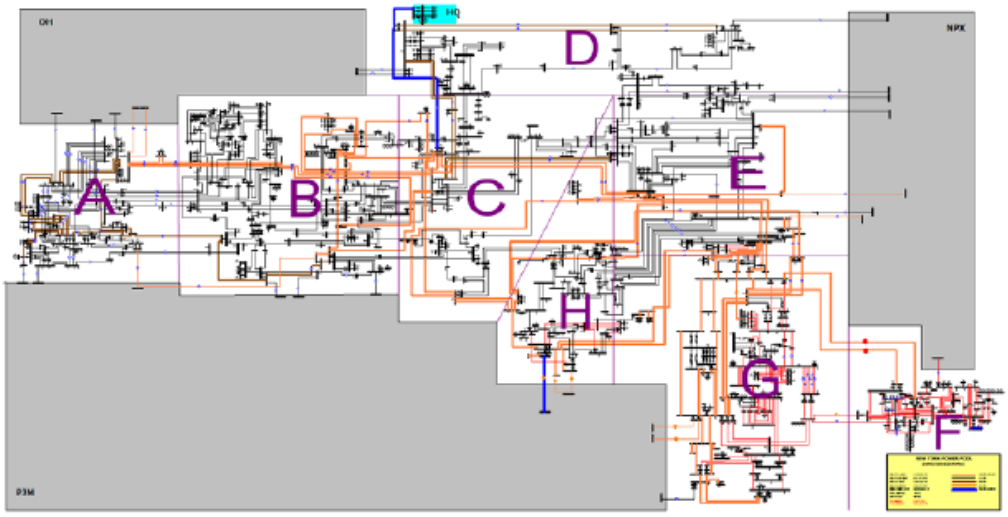


Figure 51. New York Power Pool Subdivided Into Utility Companies

These eight subdivisions (i.e. A,B,C,D,E,F,G, and H) would represent eight separate electric utility companies with responsibilities for generation, transmission, and distribution. Each company has Utility Intranet connections to others in its vicinity. These data connection edges could be modeled to parallel the actual point-to-point flows of electric power that currently exist between power system nodes (generation plants, substations, etc). The eight companies might comprise three different control areas, where, for example, CA1 is comprised of three companies (A,B,C), CA2 is comprised of three more companies (D,E,F), and the final control area, CA3, is comprised of only two companies (G,H).

Generators, step-down transformers, and other power system entities would be replaced in the Intranet communication model with communication nodes representing either an IED/switch/router combination in a substation (or generation facility) or, at the supervisory level, either the EMS/SCADA master station and its switch and router, the

company control center facility, the area control and engineering centers, or the reliability coordination offices.

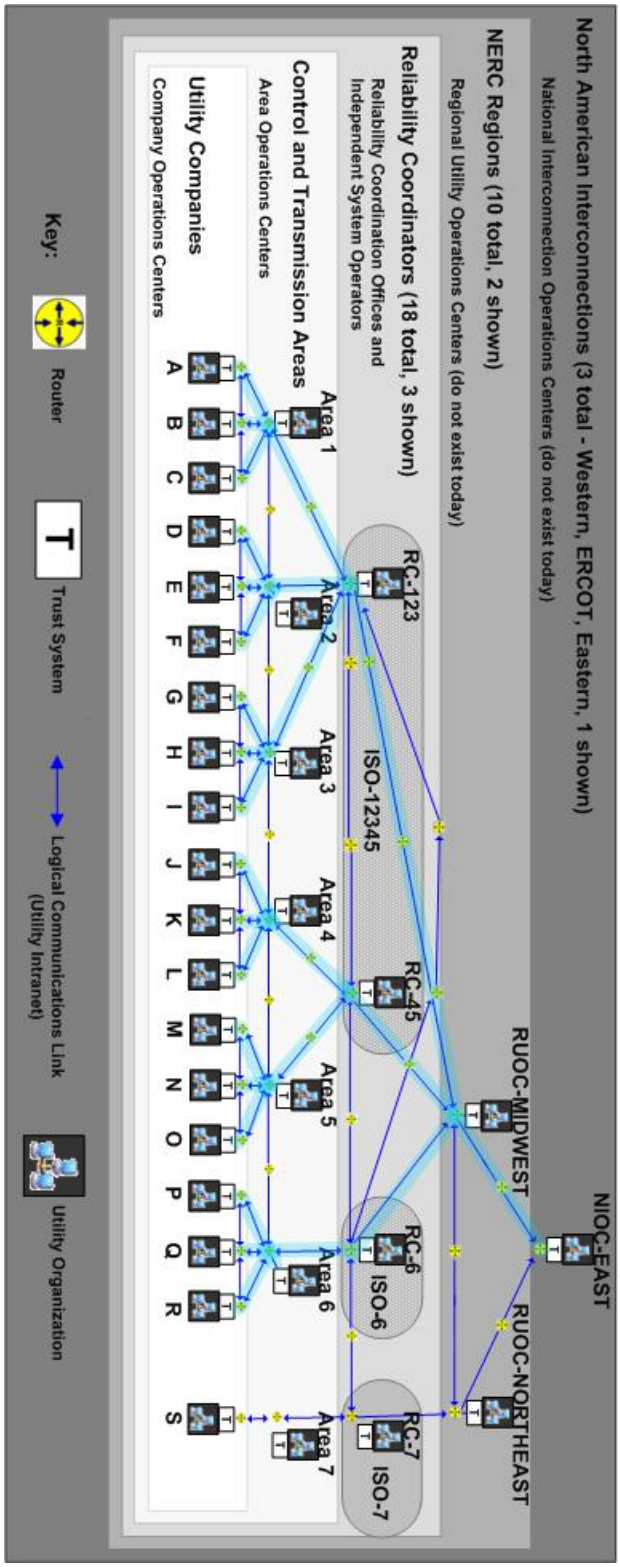
5.6 Summary

A variant of the **trust system** concept will enhance security and safety within the US Utility Intranet by the unique traffic authorization, packet inspection, access control, encryption, collaboration, and information sharing it enables.

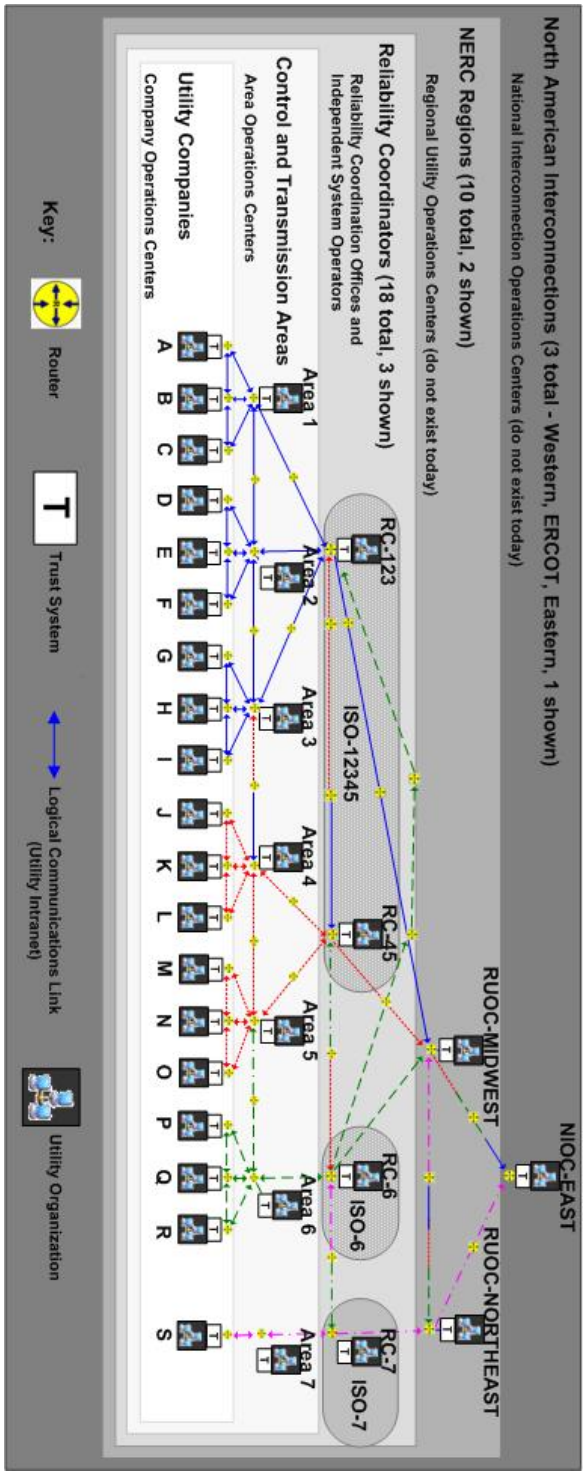
The **trust system** can provide these capabilities (or a subset thereof) within the strict time constraints of many SCADA and protection communications and is flexibly configurable and modular, making it customizable and financially advantageous to any corporation's specific needs and budget.

It is this author's opinion that a comprehensive, collaborative, and intelligence-gathering approach like the **trust system** concept, will be the wave of the future in automated network security implementations.

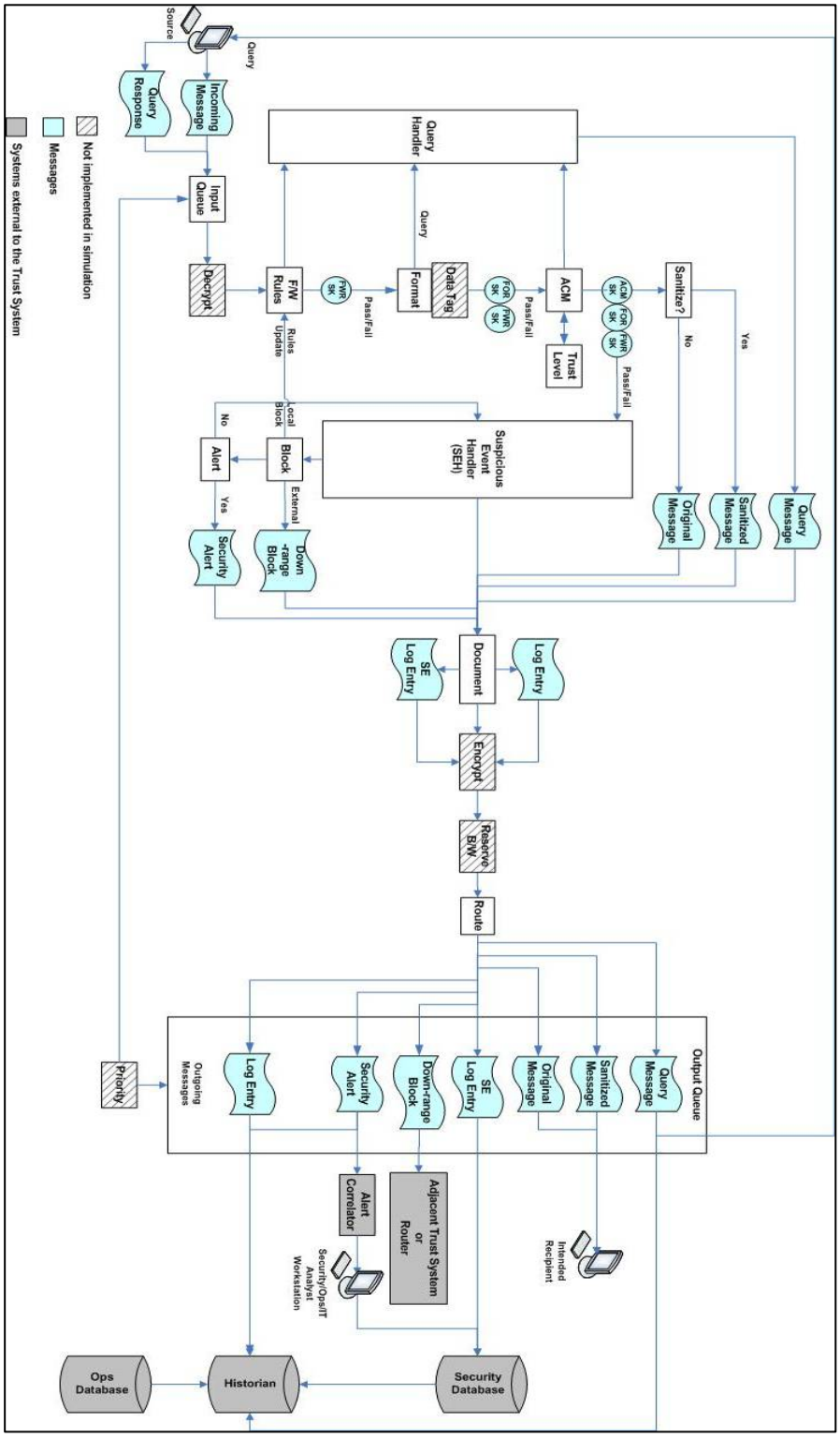
Appendix A: Proposed Electric Utility Organizational Structure



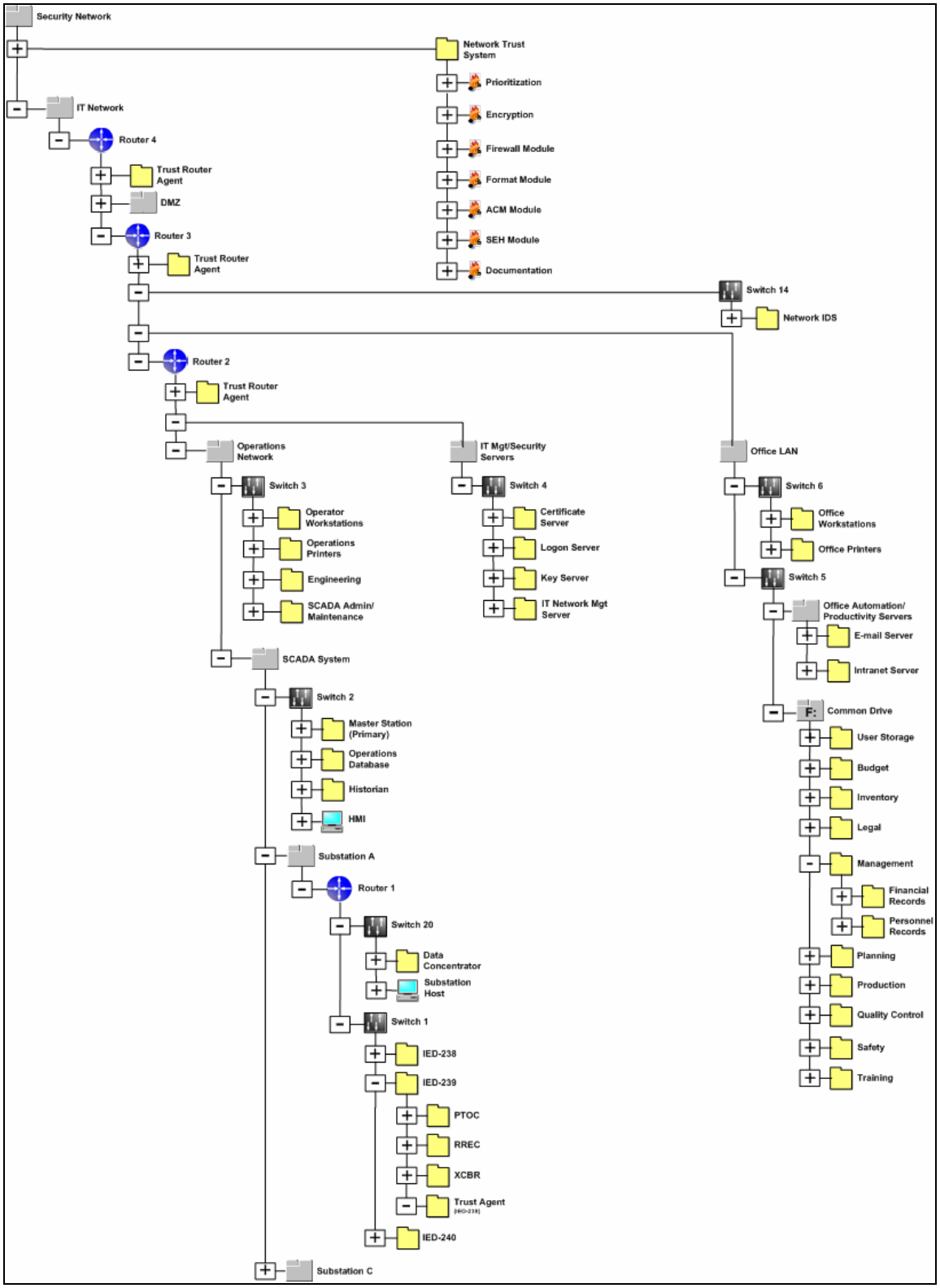
Appendix B: Information Sharing Possible Between Enclaves in the Utility Intranet



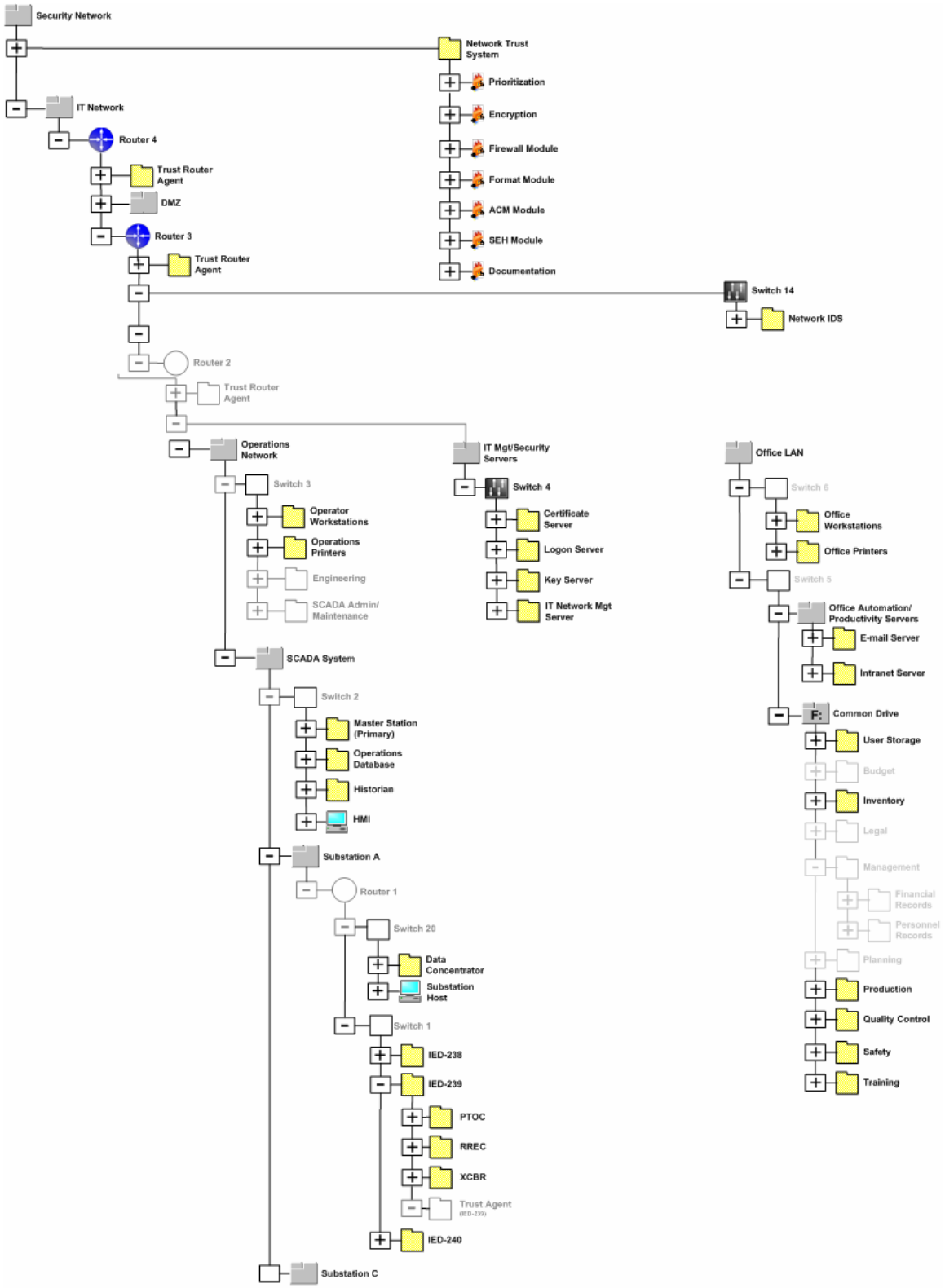
Appendix C: Trust System Functions and Output



Appendix D: Example File Structure for a Company's Operational Network



Appendix E: Operator's Network Views on Operations LAN vs. Office LAN



Appendix F: Measured Trust System Check Delay per Message Type

Message Type	Protocol	Trust System Delay (ms)											
		FW Rules Check			Format Check			ACM Check			Total		
		Min	Ave	Max	Min	Ave	Max	Min	Ave	Max	Min	Ave	Max
status	UDP	0.1806	0.1887	0.1957	0.0026	0.0047	0.0066	0.0358	0.0423	0.0477	0.2190	0.2358	0.2500
	TCP	0.1851	0.1955	0.2072	0.0032	0.0047	0.0072	0.0358	0.0434	0.0485	0.2241	0.2436	0.2629
logon_evaluated	UDP	0.1374	0.1508	0.1590	0.0030	0.0048	0.0060	0.0332	0.0399	0.0456	0.1773	0.1954	0.2123
	TCP	0.1491	0.1589	0.1717	0.0030	0.0048	0.0064	0.0368	0.0406	0.0473	0.1853	0.2042	0.2237
set	UDP	0.1458	0.1544	0.1642	0.0034	0.0053	0.0068	0.0331	0.0377	0.0415	0.1829	0.1974	0.2125
	TCP	0.1499	0.1622	0.1693	0.0048	0.0064	0.0086	0.0337	0.0382	0.0445	0.1878	0.2069	0.2224
control	TCP	0.1451	0.1575	0.1663	0.0020	0.0045	0.0058	0.0226	0.0288	0.0346	0.1697	0.1908	0.2067
get_status	UDP	0.0472	0.0527	0.0579	0.0026	0.0040	0.0060	0.0174	0.0233	0.0268	0.0672	0.0800	0.0907
	TCP	0.0481	0.0548	0.0594	0.0032	0.0049	0.0068	0.0200	0.0235	0.0286	0.0713	0.0832	0.0949
logon_request	TCP	0.0852	0.0935	0.0997	0.0028	0.0045	0.0058	0.0370	0.0424	0.0514	0.1250	0.1404	0.1570
query_encryption	UDP	0.0385	0.0421	0.0448	0.0028	0.0044	0.0058	0.0160	0.0196	0.0231	0.0573	0.0661	0.0737
	TCP	0.0429	0.0482	0.0532	0.0030	0.0054	0.0060	0.0164	0.0197	0.0232	0.0623	0.0734	0.0825
query_response	UDP	0.0513	0.0576	0.0661	0.0034	0.0060	0.0070	0.0174	0.0193	0.0214	0.0727	0.0816	0.0945
	TCP	0.0587	0.0642	0.0696	0.0040	0.0048	0.0072	0.0176	0.0199	0.0230	0.0797	0.0901	0.0998
query_packet	UDP	0.0541	0.0544	0.0480	0.0480	0.0492	0.0635	0.0042	0.0063	0.0092	0.1063	0.1100	0.1207
	TCP	0.0551	0.0632	0.0673	0.0541	0.0544	0.0680	0.0148	0.0199	0.0248	0.1240	0.1375	0.1601
logon_approved	UDP	0.0541	0.0632	0.0673	0.0024	0.0046	0.0062	0.0148	0.0199	0.0248	0.0719	0.0877	0.0983
	TCP	0.0789	0.0871	0.0907	0.0030	0.0049	0.0068	0.0244	0.0277	0.0317	0.1057	0.1197	0.1292
logon_denied	UDP	0.0755	0.0858	0.0945	0.0028	0.0040	0.0058	0.0209	0.0280	0.0244	0.0992	0.1178	0.1247
	TCP	0.0941	0.0929	0.0974	0.0054	0.0045	0.0066	0.0230	0.0287	0.0353	0.1225	0.1261	0.1393

Appendix G: Calculated Encryption/Authentication Delay per Message Type

			Encryption/Decryption + Authentication Delay (ms)																			
			IPSec Transport Mode										IPSec Tunnel Mode									
			AES-128 SHA-1			Blowfish-192 SHA-2-256			3DES SHA-2-256			AES-128 SHA-1			Blowfish-192 SHA-2-256			3DES SHA-2-256		
Message Type	Protocol	Payload Length (bits)	3GHz	4GHz	8GHz	12GHz	3GHz	4GHz	8GHz	12GHz	3GHz	4GHz	8GHz	12GHz	3GHz	4GHz	8GHz	12GHz	3GHz	4GHz	8GHz	12GHz
control	TCP	855	0.0094	0.0070	0.0035	0.0023	0.0126	0.0094	0.0047	0.0031	0.0301	0.0226	0.0113	0.0075	0.0096	0.0072	0.0036	0.0024	0.0128	0.0096	0.0048	0.0032	0.0308	0.0231	0.0115	0.0077
	UDP	769	0.0091	0.0068	0.0034	0.0023	0.0122	0.0091	0.0046	0.0030	0.0292	0.0219	0.0109	0.0073	0.0093	0.0069	0.0035	0.0023	0.0124	0.0093	0.0047	0.0031	0.0298	0.0224	0.0112	0.0075
query_response	TCP	863	0.0094	0.0070	0.0035	0.0023	0.0126	0.0094	0.0047	0.0031	0.0302	0.0227	0.0113	0.0076	0.0096	0.0072	0.0036	0.0024	0.0129	0.0096	0.0048	0.0032	0.0309	0.0231	0.0116	0.0077
	UDP	801	0.0092	0.0069	0.0034	0.0023	0.0123	0.0092	0.0046	0.0031	0.0295	0.0222	0.0111	0.0074	0.0094	0.0070	0.0035	0.0023	0.0126	0.0094	0.0047	0.0031	0.0302	0.0226	0.0113	0.0075
set	TCP	895	0.0095	0.0071	0.0036	0.0024	0.0127	0.0096	0.0048	0.0032	0.0306	0.0229	0.0115	0.0076	0.0097	0.0073	0.0036	0.0024	0.0130	0.0098	0.0049	0.0033	0.0312	0.0234	0.0117	0.0078
	UDP	801	0.0092	0.0069	0.0034	0.0023	0.0123	0.0092	0.0046	0.0031	0.0295	0.0222	0.0111	0.0074	0.0094	0.0070	0.0035	0.0023	0.0126	0.0094	0.0047	0.0031	0.0302	0.0226	0.0113	0.0075
get_status	TCP	895	0.0095	0.0071	0.0036	0.0024	0.0127	0.0096	0.0048	0.0032	0.0306	0.0229	0.0115	0.0076	0.0097	0.0073	0.0036	0.0024	0.0130	0.0098	0.0049	0.0033	0.0312	0.0234	0.0117	0.0078
	UDP	813	0.0092	0.0069	0.0035	0.0023	0.0124	0.0093	0.0046	0.0031	0.0297	0.0223	0.0111	0.0074	0.0094	0.0071	0.0035	0.0024	0.0126	0.0095	0.0047	0.0032	0.0303	0.0227	0.0114	0.0076
logon_approved	TCP	907	0.0095	0.0072	0.0036	0.0024	0.0128	0.0096	0.0048	0.0032	0.0307	0.0230	0.0115	0.0077	0.0097	0.0073	0.0037	0.0024	0.0131	0.0098	0.0049	0.0033	0.0314	0.0235	0.0118	0.0078
	UDP	813	0.0092	0.0069	0.0035	0.0023	0.0124	0.0093	0.0046	0.0031	0.0297	0.0223	0.0111	0.0074	0.0094	0.0071	0.0035	0.0024	0.0126	0.0095	0.0047	0.0032	0.0303	0.0227	0.0114	0.0076
logon_denied	TCP	907	0.0095	0.0072	0.0036	0.0024	0.0128	0.0096	0.0048	0.0032	0.0307	0.0230	0.0115	0.0077	0.0097	0.0073	0.0037	0.0024	0.0131	0.0098	0.0049	0.0033	0.0314	0.0235	0.0118	0.0078
	UDP	825	0.0093	0.0069	0.0035	0.0023	0.0124	0.0093	0.0047	0.0031	0.0298	0.0224	0.0112	0.0075	0.0095	0.0071	0.0035	0.0024	0.0127	0.0095	0.0048	0.0032	0.0304	0.0228	0.0114	0.0076
query_encryption	TCP	919	0.0096	0.0072	0.0036	0.0024	0.0129	0.0096	0.0048	0.0032	0.0308	0.0231	0.0116	0.0077	0.0098	0.0073	0.0037	0.0024	0.0131	0.0098	0.0049	0.0033	0.0315	0.0236	0.0118	0.0079
	UDP	829	0.0093	0.0070	0.0035	0.0023	0.0124	0.0093	0.0047	0.0031	0.0299	0.0224	0.0112	0.0075	0.0095	0.0071	0.0036	0.0024	0.0127	0.0095	0.0048	0.0032	0.0305	0.0229	0.0114	0.0076
logon_request	TCP	923	0.0096	0.0072	0.0036	0.0024	0.0129	0.0097	0.0048	0.0032	0.0309	0.0232	0.0116	0.0077	0.0098	0.0074	0.0037	0.0025	0.0131	0.0099	0.0049	0.0033	0.0315	0.0237	0.0118	0.0079
	UDP	961	0.0097	0.0073	0.0036	0.0024	0.0130	0.0098	0.0049	0.0033	0.0313	0.0235	0.0117	0.0078	0.0099	0.0075	0.0037	0.0025	0.0133	0.0100	0.0050	0.0033	0.0320	0.0240	0.0120	0.0080
logon_evaluated	TCP	1055	0.0101	0.0075	0.0038	0.0025	0.0135	0.0101	0.0051	0.0034	0.0323	0.0243	0.0121	0.0081	0.0103	0.0077	0.0038	0.0026	0.0138	0.0103	0.0052	0.0034	0.0330	0.0248	0.0124	0.0083
query_ simultaneous_ logon	UDP	1001	0.0099	0.0074	0.0037	0.0025	0.0132	0.0099	0.0050	0.0033	0.0317	0.0238	0.0119	0.0079	0.0101	0.0076	0.0038	0.0025	0.0135	0.0101	0.0051	0.0034	0.0324	0.0243	0.0122	0.0081
	TCP	1095	0.0102	0.0076	0.0038	0.0025	0.0137	0.0102	0.0051	0.0034	0.0328	0.0246	0.0123	0.0082	0.0104	0.0078	0.0039	0.0026	0.0140	0.0105	0.0052	0.0035	0.0335	0.0251	0.0126	0.0084
query_packet	UDP	1021	0.0099	0.0075	0.0037	0.0025	0.0133	0.0100	0.0050	0.0033	0.0320	0.0240	0.0120	0.0080	0.0101	0.0076	0.0038	0.0025	0.0136	0.0102	0.0051	0.0034	0.0326	0.0245	0.0122	0.0082
	TCP	1115	0.0103	0.0077	0.0038	0.0026	0.0138	0.0103	0.0052	0.0034	0.0330	0.0248	0.0124	0.0083	0.0105	0.0079	0.0039	0.0026	0.0140	0.0105	0.0053	0.0035	0.0337	0.0253	0.0126	0.0084
query_elevation	UDP	1089	0.0102	0.0076	0.0038	0.0025	0.0136	0.0102	0.0051	0.0034	0.0327	0.0245	0.0123	0.0082	0.0104	0.0078	0.0039	0.0026	0.0139	0.0104	0.0052	0.0035	0.0334	0.0251	0.0125	0.0084
	TCP	1183	0.0105	0.0079	0.0039	0.0026	0.0141	0.0105	0.0053	0.0035	0.0337	0.0253	0.0127	0.0084	0.0107	0.0080	0.0040	0.0027	0.0144	0.0108	0.0054	0.0036	0.0345	0.0259	0.0129	0.0086
status	UDP	1425	0.0113	0.0085	0.0042	0.0028	0.0152	0.0114	0.0057	0.0038	0.0364	0.0273	0.0137	0.0091	0.0116	0.0087	0.0043	0.0029	0.0155	0.0116	0.0058	0.0039	0.0372	0.0279	0.0140	0.0093
	TCP	1519	0.0116	0.0087	0.0044	0.0029	0.0156	0.0117	0.0059	0.0039	0.0374	0.0281	0.0140	0.0094	0.0119	0.0089	0.0045	0.0030	0.0159	0.0120	0.0060	0.0040	0.0383	0.0287	0.0143	0.0096

Appendix H: Scenario 2 Delay Results

Packet	Type	Protocol	TCP Flags	Source	Destination	Packet Size (bits)	Payload Size (bits)	IPSec Mode	Queue Size (B)	d_{proc} (ms)	TS Delay (ms)	Link Delays (ms)	Rtr/Sw Delays (ms)	Per Packet Delay (ms) Regular TCP 3GHz	4GHz	8GHz	12GHz	Abbreviated TCP 3GHz	4GHz	8GHz	12GHz
2-1	control	TCP	SYN	CA1_master_station	IED-239_my_SCADA_master	834	258	none	300	0.09	0.6451	2.2010	1.1577	4.0122	3.8509	3.6090	3.5284	4.0122	3.8509	3.6090	3.5284
						1431	855	transport	300	0.09	0.7279	2.2542	1.2777	4.2958	4.1138	3.8409	3.7499	4.2958	4.1138	3.8409	3.7499
						1431	855	tunnel	300	0.09	0.7292	2.2542	1.2777	4.2976	4.1153	3.8418	3.7507	4.2976	4.1153	3.8418	3.7507
2-2	control	TCP	SYN ACK	IED-239_CA1_master_station	IED-239_my_SCADA_master	1431	855	tunnel	1500	2.00	0.7292	15.7334	22.8091	39.3081	39.1258	38.8524	38.7612	39.3081	39.1258	38.8524	38.7612
						834	258	none	300	0.09	0.6451	2.1777	1.1577	3.9889	3.8276	3.5857	3.5051	3.9889	3.8276	3.5857	3.5051
						1431	855	transport	300	0.09	0.7279	2.2642	1.2777	4.3059	4.1239	3.8510	3.7600	4.3059	4.1239	3.8510	3.7600
						1431	855	tunnel	300	0.09	0.7292	2.2644	1.2777	4.3077	4.1254	3.8520	3.7608	4.3077	4.1254	3.8520	3.7608
2-3	control	TCP	ACK	CA1_master_station	IED-239_my_SCADA_master	834	258	none	300	0.09	0.6451	2.2010	1.1577	4.0122	3.8509	3.6090	3.5284	4.0122	3.8509	3.6090	3.5284
						1431	855	transport	300	0.09	0.7279	2.2542	1.2777	4.2958	4.1138	3.8409	3.7499	4.2958	4.1138	3.8409	3.7499
						1431	855	tunnel	300	0.09	0.7292	2.2542	1.2777	4.2976	4.1153	3.8418	3.7507	4.2976	4.1153	3.8418	3.7507
						1431	855	tunnel	1500	2.00	0.7292	15.7334	22.8091	39.3081	39.1258	38.8524	38.7612	39.3081	39.1258	38.8524	38.7612
2-4	trip	TCP	ACK	CA1_master_station	IED-239_my_SCADA_master	874	298	none	300	0.09	0.6834	2.2046	1.1657	4.0725	3.8992	3.6391	3.5524	4.0725	3.8992	3.6391	3.5524
						1471	895	transport	300	0.09	0.7771	2.2578	1.2858	4.3574	4.1631	3.8716	3.7745	4.3574	4.1631	3.8716	3.7745
						2001	1425	tunnel	300	0.09	0.8073	2.3050	1.3924	4.5513	4.3495	4.0468	3.9459	4.5513	4.3495	4.0468	3.9459
						2001	1425	tunnel	1500	2.00	0.8073	15.8277	23.1314	39.8131	39.6112	39.3085	39.2076	39.8131	39.6112	39.3085	39.2076
2-5	control	TCP	ACK	IED-239_CA1_master_station		834	258	none	300	0.09	0.6451	2.1777	1.1577	3.9889	3.8276	3.5857	3.5051				
						1431	855	transport	300	0.09	0.7279	2.2644	1.2777	4.3059	4.1239	3.8510	3.7600				
						1431	855	tunnel	300	0.09	0.7292	2.2644	1.2777	4.3077	4.1254	3.8520	3.7608				
						1431	855	tunnel	1500	2.00	0.7292	19.7696	22.8091	43.3443	43.1620	42.8886	42.7974				
SCENARIO COMPLETE (trip executed)				**TOTALS:**	**(transport mode, 300B queue, 0.9ms d_{proc})**				3.6887	11.2950	6.3868			21.5608	20.6386	19.2553	18.7942	12.9591	12.4008	11.5635	11.2844
					(tunnel mode, 300B queue, 0.9ms d_{proc})				3.7243	11.3422	6.5038			21.7620	20.8309	19.4343	18.9688	13.1567	12.5902	11.7406	11.4573
					(tunnel mode, 1500B queue, 2.0ms d_{proc})				3.7243	86.8338	114.3677			205.1181	204.1870	202.7904	202.3249	122.4656	121.8991	121.0495	120.7662
2-6	status	TCP	ACK	IED-239	CA1_master_station_my_SCADA_master	1498	922	none	300	0.09	0.8336	2.2741	1.2912	4.4139	4.2055	3.8929	3.7887	4.4139	4.2055	3.8929	3.7887
						2095	1519	transport	300	0.09	0.9321	2.3807	1.4113	4.7519	4.5188	4.1693	4.0528	4.7519	4.5188	4.1693	4.0528
						2095	1519	tunnel	300	0.09	0.9339	2.3807	1.4113	4.7542	4.5208	4.1706	4.0538	4.7542	4.5208	4.1706	4.0538
						2095	1519	tunnel	1500	2.00	0.9339	20.0123	23.1845	44.1791	43.9456	43.5954	43.4786	44.1791	43.9456	43.5954	43.4786
2-7	control	TCP	ACK	CA1_master_station	IED-239_	834	258	none	300	0.09	0.6451	2.2010	1.1577	4.0122	3.8509	3.6090	3.5284				
						1431	855	transport	300	0.09	0.7279	2.2542	1.2777	4.2958	4.1138	3.8409	3.7499				
						1431	855	tunnel	300	0.09	0.7292	2.2542	1.2777	4.2976	4.1153	3.8418	3.7507				
						1431	855	tunnel	1500	2.00	0.7292	15.7334	22.8091	39.3081	39.1258	38.8524	38.7612				
2-8	control	TCP	FIN ACK	CA1_master_station	IED-239_	834	258	none	300	0.09	0.6451	2.2010	1.1577	4.0122	3.8509	3.6090	3.5284	4.0122	3.8509	3.6090	3.5284
						1431	855	transport	300	0.09	0.7279	2.2542	1.2777	4.2958	4.1138	3.8409	3.7499	4.2958	4.1138	3.8409	3.7499
						1431	855	tunnel	300	0.09	0.7292	2.2542	1.2777	4.2976	4.1153	3.8418	3.7507	4.2976	4.1153	3.8418	3.7507
						1431	855	tunnel	1500	2.00	0.7292	15.7334	22.8091	39.3081	39.1258	38.8524	38.7612	39.3081	39.1258	38.8524	38.7612
2-9	control	TCP	ACK	IED-239	CA1_master_station	834	258	none	300	0.09	0.6451	2.1777	1.1577	3.9889	3.8276	3.5857	3.5051				
						1431	855	transport	300	0.09	0.7279	2.2644	1.2777	4.3059	4.1239	3.8510	3.7600				
						1431	855	tunnel	300	0.09	0.7292	2.2644	1.2777	4.3077	4.1254	3.8520	3.7608				
						1431	855	tunnel	1500	2.00	0.7292	19.7696	22.8091	43.3443	43.1620	42.8886	42.7974				
2-10	control	TCP	FIN ACK	IED-239	CA1_master_station	834	258	none	300	0.09	0.6451	2.1777	1.1577	3.9889	3.8276	3.5857	3.5051	3.9889	3.8276	3.5857	3.5051
						1431	855	transport	300	0.09	0.7279	2.2644	1.2777	4.3059	4.1239	3.8510	3.7600	4.3059	4.1239	3.8510	3.7600
						1431	855	tunnel	300	0.09	0.7292	2.2644	1.2777	4.3077	4.1254	3.8520	3.7608	4.3077	4.1254	3.8520	3.7608
						1431	855	tunnel	1500	2.00	0.7292	19.7696	22.8091	43.3443	43.1620	42.8886	42.7974	43.3443	43.1620	42.8886	42.7974
2-11	control	TCP	ACK	CA1_master_station	IED-239	834	258	none	300	0.09	0.6451	2.2010	1.1577	4.0122	3.8509	3.6090	3.5284	4.0122	3.8509	3.6090	3.5284
						1431	855	transport	300	0.09	0.7279	2.2542	1.2777	4.2958	4.1138	3.8409	3.7499	4.2958	4.1138	3.8409	3.7499
						1431	855	tunnel	300	0.09	0.7292	2.2542	1.2777	4.2976	4.1153	3.8418	3.7507	4.2976	4.1153	3.8418	3.7507
						1431	855	tunnel	1500	2.00	0.7292	15.7334	22.8091	39.3081	39.1258	38.8524	38.7612	39.3081	39.1258	38.8524	38.7612
CONVERSATION COMPLETE (TCP connection closed)				**TOTALS:**	**(transport mode, 300B queue, 0.9ms d_{proc})**				3.8394	11.2914	6.3887			46.0245	45.9484	42.8342	41.7962	30.8138	29.4670	27.4467	26.7733
					(tunnel mode, 300B queue, 0.9ms d_{proc})				3.8462	11.2914	6.3887			46.0245	45.9484	42.8342	41.7962	30.8138	29.4670	27.4467	26.7733
					(tunnel mode, 1500B queue, 2.0ms d_{proc})				3.8462	86.7396	114.0454			453.9103	451.8342	448.7200	447.6820	288.6053	287.2584	285.2381	284.5647

Appendix I: Scenario 3 Delay Results

Pckt	Msg Type	Prot	TCP Control Flags		Source	Destination	IPSec Mode	Queue Size (B)	d_{proc} (ms)	# TS	Trust System Delay (ms)	Link Delays (ms)	Router/ Switch Delays (ms)	Per Packet Delay (ms)							
														Regular TCP				Abbreviated TCP			
														3GHz	4GHz	8GHz	12GHz	3GHz	4GHz	8GHz	12GHz
3-1	control	TCP	SYN	URG	SCADA_admin_workstation1	logon_server	tunnel	300	0.09	1	0.2411	0.3912	0.3902	1.0589	0.9986	0.9082	0.8781	1.0589	0.9986	0.9082	0.8781
							tunnel	1500	2.00	1	0.2411	6.4303	6.4293	13.1371	13.0768	12.9864	12.9563	13.1371	13.0768	12.9864	12.9563
3-2	control	TCP	SYN-ACK	URG	logon_server	SCADA_admin_workstation1	tunnel	300	0.09	1	0.2411	0.3912	0.3902	1.0589	0.9986	0.9082	0.8781	1.0589	0.9986	0.9082	0.8781
							tunnel	1500	2.00	1	0.2411	6.4303	6.4293	13.1371	13.0768	12.9864	12.9563	13.1371	13.0768	12.9864	12.9563
3-3	control	TCP	ACK	URG	SCADA_admin_workstation1	logon_server	tunnel	300	0.09	1	0.2411	0.3912	0.3902	1.0589	0.9986	0.9082	0.8781				
							tunnel	1500	2.00	1	0.2411	6.4303	6.4293	13.1371	13.0768	12.9864	12.9563				
3-4	logon_request	TCP	PSH-ACK	URG	SCADA_admin_workstation1	logon_server	tunnel	300	0.09	1	0.1945	0.3969	0.3959	1.0250	0.9763	0.9034	0.8791	1.0250	0.9763	0.9034	0.8791
							tunnel	1500	2.00	1	0.1945	6.4507	6.4497	13.1325	13.0839	13.0110	12.9866	13.1325	13.0839	13.0110	12.9866
3-5	control	TCP	ACK	URG	logon_server	SCADA_admin_workstation1	tunnel	300	0.09	1	0.2411	0.3912	0.3902	1.0589	0.9986	0.9082	0.8781	1.0589	0.9986	0.9082	0.8781
							tunnel	1500	2.00	1	0.2411	6.4303	6.4293	13.1371	13.0768	12.9864	12.9563	13.1371	13.0768	12.9864	12.9563
3-6	control	TCP	SYN	URG	logon_server	NTS	tunnel	300	0.09	1	0.2411	0.1308	0.1301	0.5383	0.4781	0.3877	0.3575	0.5383	0.4781	0.3877	0.3575
							tunnel	1500	2.00	1	0.2411	2.1439	2.1431	4.5644	4.5041	4.4137	4.3836	4.5644	4.5041	4.4137	4.3836
3-7	control	TCP	SYN-ACK	URG	NTS	logon_server	tunnel	300	0.09	1	0.2411	0.1308	0.1301	0.5383	0.4781	0.3877	0.3575	0.5383	0.4781	0.3877	0.3575
							tunnel	1500	2.00	1	0.2411	2.1439	2.1431	4.5644	4.5041	4.4137	4.3836	4.5644	4.5041	4.4137	4.3836
3-8	control	TCP	ACK	URG	logon_server	NTS	tunnel	300	0.09	1	0.2411	0.1308	0.1301	0.5383	0.4781	0.3877	0.3575				
							tunnel	1500	2.00	1	0.2411	2.1439	2.1431	4.5644	4.5041	4.4137	4.3836				
3-9	logon_evaluated	TCP	PSH-ACK	URG	logon_server	NTS	tunnel	300	0.09	1	0.2646	0.1364	0.1357	0.5760	0.5098	0.4106	0.3775	0.5760	0.5098	0.4106	0.3775
							tunnel	1500	2.00	1	0.2646	2.1639	2.1631	4.6308	4.5647	4.4654	4.4323	4.6308	4.5647	4.4654	4.4323
3-10	control	TCP	ACK	URG	NTS	logon_server	tunnel	300	0.09	1	0.2411	0.1308	0.1301	0.5383	0.4781	0.3877	0.3575				
							tunnel	1500	2.00	1	0.2411	2.1439	2.1431	4.5644	4.5041	4.4137	4.3836				
3-11	logon_denied	TCP	PSH-ACK	URG	NTS	logon_server	tunnel	300	0.09	1	0.1766	0.1323	0.1315	0.4777	0.4336	0.3674	0.3453	0.4777	0.4336	0.3674	0.3453
							tunnel	1500	2.00	1	0.1766	2.1491	2.1483	4.5113	4.4671	4.4009	4.3788	4.5113	4.4671	4.4009	4.3788
3-12	control	TCP	ACK	URG	logon_server	NTS	tunnel	300	0.09	1	0.2411	0.1308	0.1301	0.5383	0.4781	0.3877	0.3575	0.5383	0.4781	0.3877	0.3575
							tunnel	1500	2.00	1	0.2411	2.1439	2.1431	4.5644	4.5041	4.4137	4.3836	4.5644	4.5041	4.4137	4.3836
3-13	logon_denied	TCP	PSH-ACK	URG	logon_server	SCADA_admin_workstation1	tunnel	300	0.09	1	0.1766	0.3956	0.3946	1.0041	0.9599	0.8937	0.8716	1.0041	0.9599	0.8937	0.8716
							tunnel	1500	2.00	1	0.1766	6.4459	6.4449	13.1047	13.0606	12.9944	12.9723	13.1047	13.0606	12.9944	12.9723
3-14	control	TCP	ACK	URG	SCADA_admin_workstation1	logon_server	tunnel	300	0.09	1	0.2411	0.3912	0.3902	1.0589	0.9986	0.9082	0.8781	1.0589	0.9986	0.9082	0.8781
							tunnel	1500	2.00	1	0.2411	6.4303	6.4293	13.1371	13.0768	12.9864	12.9563	13.1371	13.0768	12.9864	12.9563
3-15	logon_request	TCP	PSH-ACK	URG	SCADA_admin_workstation1	logon_server	tunnel	300	0.09	1	0.1945	0.3969	0.3959	1.0250	0.9763	0.9034	0.8791	1.0250	0.9763	0.9034	0.8791
							tunnel	1500	2.00	1	0.1945	6.4507	6.4497	13.1325	13.0839	13.0110	12.9866	13.1325	13.0839	13.0110	12.9866
3-16	control	TCP	ACK	URG	logon_server	SCADA_admin_workstation1	tunnel	300	0.09	1	0.2411	0.3912	0.3902	1.0589	0.9986	0.9082	0.8781	1.0589	0.9986	0.9082	0.8781
							tunnel	1500	2.00	1	0.2411	6.4303	6.4293	13.1371	13.0768	12.9864	12.9563	13.1371	13.0768	12.9864	12.9563
3-17	logon_evaluated	TCP	PSH-ACK	URG	logon_server	NTS	tunnel	300	0.09	1	0.2646	0.1364	0.1357	0.5760	0.5098	0.4106	0.3775	0.5760	0.5098	0.4106	0.3775
							tunnel	1500	2.00	1	0.2646	2.1639	2.1631	4.6308	4.5647	4.4654	4.4323	4.6308	4.5647	4.4654	4.4323
3-18	control	TCP	ACK	URG	NTS	logon_server	tunnel	300	0.09	1	0.2411	0.1308	0.1301	0.5383	0.4781	0.3877	0.3575				
							tunnel	1500	2.00	1	0.2411	2.1439	2.1431	4.5644	4.5041	4.4137	4.3836				
3-19	logon_denied	TCP	PSH-ACK	URG	NTS	logon_server	tunnel	300	0.09	1	0.1766	0.1323	0.1315	0.4777	0.4336	0.3674	0.3453	0.4777	0.4336	0.3674	0.3453
							tunnel	1500	2.00	1	0.1766	2.1491	2.1483	4.5113	4.4671	4.4009	4.3788	4.5113	4.4671	4.4009	4.3788
3-20	control	TCP	ACK	URG	logon_server	NTS	tunnel	300	0.09	1	0.2411	0.1308	0.1301	0.5383	0.4781	0.3877	0.3575	0.5383	0.4781	0.3877	0.3575
							tunnel	1500	2.00	1	0.2411	2.1439	2.1431	4.5644	4.5041	4.4137	4.3836	4.5644	4.5041	4.4137	4.3836
3-21	logon_denied	TCP	PSH-ACK	URG	logon_server	SCADA_admin_workstation1	tunnel	300	0.09	1	0.1766	0.3956	0.3946	1.0041	0.9599	0.8937	0.8716	1.0041	0.9599	0.8937	0.8716
							tunnel	1500	2.00	1	0.1766	6.4459	6.4449	13.1047	13.0606	12.9944	12.9723	13.1047	13.0606	12.9944	12.9723
3-22	control	TCP	ACK	URG	SCADA_admin_workstation1	logon_server	tunnel	300	0.09	1	0.2411	0.3912	0.3902	1.0589	0.9986	0.9082	0.8781	1.0589	0.9986	0.9082	0.8781
							tunnel	1500	2.00	1	0.2411	6.4303	6.4293	13.1371	13.0768	12.9864	12.9563	13.1371	13.0768	12.9864	12.9563
3-23	logon_request	TCP	PSH-ACK	URG	SCADA_admin_workstation1	logon_server	tunnel	300	0.09	1	0.1945	0.3969	0.3959	1.0250	0.9763	0.9034	0.8791	1.0250	0.9763	0.9034	0.8791
							tunnel	1500	2.00	1	0.1945	6.4507	6.4497	13.1325	13.0839	13.0110	12.9866	13.1325	13.0839	13.0110	12.9866
3-24	control	TCP	ACK	URG	logon_server	SCADA_admin_workstation1	tunnel	300	0.09	1	0.2411	0.3912	0.3902	1.0589	0.9986	0.9082	0.8781	1.0589	0.9986	0.9082	0.8781
							tunnel	1500	2.00	1	0.2411	6.4303	6.4293	13.1371	13.0768	12.9864	12.9563	13.1371	13.0768	12.9864	12.9563
3-25	logon_evaluated	TCP	PSH-ACK	URG	logon_server	NTS	tunnel	300	0.09	1	0.2646	0.1364	0.1357	0.5760	0.5098	0.4106	0.3775	0.5760	0.5098	0.4106	0.3775
							tunnel	1500	2.00	1	0.2646	2.1639	2.1631	4.6308	4.5647	4.4654	4.4323	4.6308	4.5647	4.4654	4.4323
3-26	control	TCP	ACK	URG	NTS	logon_server	tunnel	300	0.09	1	0.2411	0.1308	0.1301	0.5383	0.4781	0.3877	0.3575				
							tunnel	1500	2.00	1	0.2411	2.1439	2.1431	4.5644	4.5041	4.4137	4.3836				
3-27	logon_denied	TCP	PSH-ACK	URG	NTS	logon_server	tunnel	300	0.09	1	0.1766	0.1323	0.1315	0.4777	0.4336	0.3674	0.3453	0.4777	0.4336	0.3674	0.3453
							tunnel	1500	2.00	1	0.1766	2.1491	2.1483	4.5113	4.4671	4.4009	4.3788	4.5113	4.4671	4.4009	4.3788
3-28	control	TCP	ACK	URG	logon_server	NTS	tunnel	300	0.09	1	0.2411	0.1308	0.1301	0.5383	0.4781	0.3877	0.3575	0.5383	0.4781	0.3877	0.3575
							tunnel	1500	2.00	1	0.2411	2.1439	2.1431	4.5644	4.5041	4.4137	4.3836	4.5644	4.5041	4.4137	4.3836
3-29	logon_denied	TCP	PSH-ACK	URG	logon_server	SCADA_admin_workstation1	tunnel	300	0.09	1	0.1766	0.3956	0.3946	1.0041	0.9599	0.8937	0.8716	1.0041	0.9599	0.8937	0.8716
							tunnel	1500	2.00	1	0.1766	6.4459	6.4449	13.1047	13.0606	12.9944	12.9723	13.1047	13.0606	12.9944	12.9723
3-30	control	TCP	ACK	URG	SCADA_admin_workstation1	logon_server	tunnel	300	0.09	1	0.2411	0.3912	0.3902	1.0589	0.9986	0.9082	0.8781	1.0589	0.9986	0.9082	0.8781
							tunnel	1500	2.00	1	0.2411	6.4303	6.4293	13.1371	13.0768	12.9864	12.9563	13.1371	13.0768	12.9864	12.9563
3-31	logon_request	TCP	PSH-ACK	URG	SCADA_admin_workstation1	logon_server	tunnel	300	0.09	1	0.1945	0.1327	0.1320	0.4968	0.4482	0.3752	0.3509	0.4968	0.4482	0.3752	0.3509
							tunnel	1500	2.00	1	0.1945	2.1507	2.1499	4.5327	4.4840	4.4111	4.3868	4.5327	4.4840	4.4111	4.3868
3-32	control	TCP	ACK	URG	logon_server	SCADA_admin_workstation1	tunnel	300	0.09	1	0.2411	0.3912	0.3902	1.0589	0.9986	0.9082	0.8781	1.0589	0.9986	0.9082	0.8781
							tunnel	1500	2.00	1	0.2411	6.4303	6.4293	13.1371	13.0768	12.9864	12.9563	13.1371	13.0768	12.9864	12.9563
3-33	logon_evaluated	TCP	PSH-ACK	URG	logon_server	NTS	tunnel	300	0.09	1	0.2646	0.1364	0.1357	0.5760	0.5098	0.4106	0.3775	0.5760	0.5098	0.4106	0.3775
							tunnel	1500	2.00	1	0.2646	2.1639	2.1631	4.6308	4.5647	4.4654	4.4323	4.6308	4.5647	4.4654	4.4323
3-34	control	TCP	ACK	URG	NTS	logon_server	tunnel	300	0.09	1	0.2411	0.1308	0.1301	0.5383	0.4781	0.3877	0.3575				
							tunnel	1500	2.00	1	0.2411	2.1439	2.1431	4.5644	4.5041	4.4137	4.3836				
3-35	logon_approved	TCP	PSH-ACK	URG	NTS	logon_server	tunnel	300	0.09	1	0.2565	0.1323	0.1315	0.5576	0.4935	0.3973	0.3653	0.5576	0.4935	0.3973	0.3653
							tunnel	1500	2.00	1	0.2565	2.1491	2.1483	4.5912	4.5271	4.4309	4.3988	4.5912	4.5271	4.4309	4.3988
3-36	control	TCP	ACK	URG	logon_server	NTS	tunnel	300	0.09	1	0.2411	0.1308	0.1301	0.5383	0.4781	0.3877	0.3575	0.5383	0.4781	0.3877	0.3575
							tunnel	1500	2.00	1	0.2411	2.1439	2.1431	4.5644	4.5041	4.4137	4.3836	4.5644	4.5041	4.4137	4.3836
3-37	logon_approved	TCP	PSH-ACK	URG	logon_server	SCADA_admin_workstation1	tunnel	300	0.09	1	0.2565	0.3956	0.3946	1.0840	1.0199	0.9237	0.8916	1.0840	1.0199	0.9237	0.8916
							tunnel	1500	2.00	1	0.2565	6.4459	6.4449	13.1846	13.1205	13.0243	12.9923	13.1846	13.1205	13.0243	12.9923
3-38	control	TCP	ACK	URG	SCADA_admin_workstation1	logon_server	tunnel	300	0.09	1	0.2411	0.3912	0.3902	1.0589	0.9986	0.9082	0.8781	1.0589	0.9986	0.9082	0.8781
							tunnel	1500	2.00	1	0.2411	6.4303	6.4293	13.1371	13.0768	12.9864	12.9563	13.1371	13.0768	12.9864	12.9563
3-39	control	TCP	FIN-ACK	URG	SCADA_admin_workstation1	logon_server	tunnel	300	0.09	1	0.2411	0.3912	0.3902	1.0589	0.9986	0.9082	0.8781	1.0589	0.9986	0.9082	0.8781
							tunnel	1500	2.00	1	0.2411	6.4303	6.4293	13.1371	13.0768	12.9864	12.9563	13.1371	13.0768	12.9864	12.9563
3-40	control	TCP	ACK	URG	logon_server	SCADA_admin_workstation1	tunnel	300	0.09	1	0.2411	0.3912	0.3902	1.0589	0.9986	0.9082	0.8781				
							tunnel	1500	2.00	1	0.2411	6.4303	6.4293	13.1371	13.0768	12.9864	12.9563				
3-41	control	TCP	FIN-ACK	URG	logon_server	SCADA_admin_workstation1	tunnel	300	0.09	1	0.2411	0.3912	0.3902	1.0589	0.9986	0.9082	0.8781	1.0589	0.9986	0.9082	0.8781
							tunnel	1500	2.00	1	0.2411	6.4303	6.4293	13.1371	13.0768	12.9864	12.9563	13.1371	13.0768	12.9864	12.9563
3-42	control	TCP	ACK	URG	SCADA_admin_workstation1	logon_server	tunnel	300	0.09	1	0.2411	0.3912	0.3902	1.0589	0.9986	0.9082	0.8781	1.0589	0.9986	0.9082	0.8781
							tunnel	1500	2.00	1	0.2411	6.4303	6.4293	13.1371	13.0768	12.9864	12.9563	13.1371	13.0768	12.9864	12.9563
SCENARIO COMPLETE (user logged on to the network)											9.6775	11.2678	11.2506	33.7675	31.3482	27.7191	26.5094	28.9581	26.9606	23.9644	22.9656
											9.6775	184.5748	184.5378	380.3420	377.9226	374.2935	373.0839	331.2458	329.2483	326.2521	325.2534
3-43	control	TCP	FIN-ACK	URG	logon_server	NTS	tunnel	300	0.09	1	0.2411	0.1308	0.1301	0.5383	0.4781	0.3877	0.3575	0.5383	0.4781	0.3877	0.3575
							tunnel	1500	2.00	1	0.2411	2.1439	2.1431	4.5644	4.5041	4.4137	4.3836	4.5644	4.5041	4.4137	4.3836
3-44	control	TCP	ACK	URG	NTS	logon_server	tunnel	300	0.09	1	0.2411	0.1308	0.1301	0.5383	0.4781	0.3877	0.3575				
							tunnel	1500	2.00	1	0.2411	2.1439	2.1431	4.5644	4.5041	4.4137	4.3836				
3-45	control	TCP	FIN-ACK	URG	NTS	logon_server	tunnel	300	0.09	1	0.2411	0.1308	0.1301	0.5383	0.4781	0.3877	0.3575	0.5383	0.4781	0.3877	0.3575
							tunnel	1500	2.00	1	0.2411	2.1439	2.1431	4.5644	4.5041	4.4137	4.3836	4.5644	4.5041	4.4137	4.3836
3-46	control	TCP	ACK	URG	logon_server	NTS	tunnel	300	0.09	1	0.2411	0.1308	0.1301	0.5383	0.4781	0.3877	0.3575	0.5383	0.4781	0.3877	0.3575
							tunnel	1500	2.00	1	0.2411	2.1439	2.1431	4.5644	4.5041	4.4137	4.3836	4.5644	4.5041	4.4137	4.3836
CONVERSATIONS COMPLETE (all logon-related connections closed)											10.6418	11.8109	11.7709	35.9209	33.2605	29.2698	27.9396	30.5731	28.3948	25.1274	24.0382
											10.6418	193.1502	193.1102	398.5998	395.9392	391.9485	390.6183	344.9390	342.7607	339.4933	338.4042

Bibliography

1. Krutz, R.L., PhD, *Securing SCADA systems*. Indianapolis, IN: Wiley Publishing, Inc., 2006.

2. ----. "SCADA," updated 24 July 2007. http://en.wikipedia.org/ wiki/SCADA. Accessed 26 Jul 2007.

3. Bowen, Calvert L., III, Timothy K. Buennemeyer, and Ryan W. Thomas. "Next Generation SCADA Security: Best Practices and Client Puzzles." *Proceedings of the 2005 IEEE,* presented at the Workshop on Information Assurance and Security, United States Military Academy, West Point NY (2005).

4. Graham, R. and D. Maynor. "SCADA Security and Terrorism: We're not crying wolf." MS PowerPoint presentation for Internet Security Systems.

5. Louisiana Department of Health and Hospitals. "Reports of Missing and Deceased," updated Aug. 2, 2006. http://www.dhh.louisiana.gov/offices/ page.asp?ID=192&Detail=5248. Accessed 23 Jun 2007.

6. Grimes, Mark. "SCADA Exposed." MS PowerPoint presentation for SAIC.

7. Graham, R. and D. Maynor, "SCADA Security and Terrorism: We're not crying wolf." MS PowerPoint presentation for Internet Security Systems. Presented at the Blackhat Conference, 2006.

8. Pollet, Jonathon. "Developing a Solid SCADA Security Strategy," Sensors for Industry Conference (SICON), Houston TX (19-21 Nov 2002).

9. Hopkinson, Kenneth M., et al., "Evaluating the Effects of Background Traffic on a Utility Intranet." Submission to the *IEEE Transactions on Power Systems* (2005).

10. Birman, K., et al., "Overcoming Communications Challenges in Software for Monitoring and Controlling Power Systems," *Proceedings of the2005 IEEE,* Vol. 93, No. 5 (2005).

11. Bailey, David and Edwin Wright, *Practical SCADA for Industry*. Oxford UK: Newnes Publications, 2003.

12. ----. "RTU vs. PLC." MSE-Tetragenics, Butte MT. http://www.tetragenics.com/ Articles/ RTUvsPLC.htm. Accessed 5 Jul 2007.

13. ----. "Intelligent Electronic Device," updated 5 Sep 2006. http://en.wikipedia.org/wiki/Intelligent_electronic_device. Accessed 10 Jul 2007.

14. Proudfoot, Douglas. "UCA and 61850 for Dummies." MS PowerPoint presentation for Siemens Power Transmission and Distribution (2002).

15. ----. "SCADA Systems." Enercon Engineering, Inc., East Peoria IL. http://www.enercon-eng.com/scada.htm (2002). Accessed 1 Jul 2007.

16. Adamiak, Mark, et al. "Wide Area Protection-Technology and Infrastructures," *IEEE Transactions on Power Delivery,* Vol. 21, No. 2 (2006).

17. ----. "Optical Fiber to the Desktop," http://www.lascomm.com /articles/fib-art5.htm. Lascomm, Westlake Village CA. Accessed 5 Jul 2007.

18. McDonald, John D. *Electric Power Substations Engineering*. Boca Raton FL: CRC Press LLC, 2003.

19. West, Andrew. "SCADA and Substation Control Communications." Presented at the Southern African SCADA and MES Conference: Johannesburg, South Africa (11, 12 May 2005).

20. Giovanini, Renan, et al., "A Primary and Backup Cooperative Protection System Based on Wide Area Agents," *IEEE Transactions on Power Delivery,* 2005.

21. Hsi, Pao-Hsiang. and Shi-Lin Chen. "Distribution Automation Communication Infrastructure," *IEEE Transactions on Power Delivery,* Vol. 13, Issue 3 (July 1998).

22. Adamiak, Mark. "IntelliGrid Architecture...a System with a View." General Electric Multilin MS PowerPoint presentation (39 slides). Presented at the University of Illinois (15 November 2005).

23. ----. *Mitigating the Top 10 Network Security Risks in SCADA and Process Control Systems: A McAfee IntruShield Solution Guide.* Santa Clara CA: McAfee, Inc, April 2007.

24. Leyden, John. "Why power plants need anti-virus: Symantec's cyber-security power pitch," *The Register,* http://www.theregister.co.uk/2005/03/17/ industrial_cyber-security/ (17 March 2005).

25. Falco, Joe, Michael Lochner, and David Teumim, "Guidance and Performance Impact Testing to Support the use of Antivirus Software on SCADA and Industrial Control Systems," Instrumentation, Systems, and Automation Society Expo 2005, Chicago IL (25-27 October 2005).

26. ----. "Denial-of-service attack." Updated 8 July 2007. http://en.wikipedia.org/ wiki/ Denial-of-service_attack. Accessed 9 July 2007.

27. Bilderback, Nathan, Account Executive, ASI Computer Technologies, Inc., Duluth GA. E-mail quote. 16 May 2007.

28. U.S.-Canada Power System Outage Task Force. *Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations*. Washington DC and Ottawa ON, April 2004.

29. ----. "Northeast Blackout of 2003," updated 29 July 2007. http://en.wikipedia.org/wiki/ 2003_North_America_blackout. Accessed 30 July 2007.

30. Bishop, M. *Computer Security: Art and Science*. Boston MA: Pearson Education, Inc., 2003.

31. Kurose, J.F. and K.W. Ross. *Computer Networking: A Top-down Approach Featuring the Internet, 3rd ed.* Boston MA: Addison-Wesley, 2004.

32. Niedermayer, Heiko, Andreas Klenk, and Georg Carle, University of Tübingen, Germany. "The Networking Perspective of Security Performance - a Measurement Study." Presented at the 13th GI/ITG Conference on Measurement, Modeling, and Evaluation of Computer and Communication Systems, Friedrich-Alexander-Universitat, Erlangen-Nürnberg, Germany (27-29 March 2006).

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
31 Jul 2007	Master's Thesis	June 2006 – July 2007

4. TITLE AND SUBTITLE
COLLABORATIVE, TRUST-BASED SECURITY MECHANISMS FOR A NATIONAL UTILITY INTRANET

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Coates, Gregory M., Major, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way, Building 640
WPAFB OH 45433-8865

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GIA/ENG/07-05

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This thesis investigates security mechanisms for utility control and protection networks using IP-based protocol interaction. It proposes flexible, cost-effective solutions in strategic locations to protect transitioning legacy and full IP-standards architectures. It also demonstrates how operational signatures can be defined to enact organizationally-unique standard operating procedures for zero failure in environments with varying levels of uncertainty and trust. The research evaluates layering encryption, authentication, traffic filtering, content checks, and event correlation mechanisms over time-critical primary and backup control/protection signaling to prevent disruption by internal and external malicious activity or errors. Finally, it shows how a regional/national implementation can protect private communities of interest and foster a mix of both centralized and distributed emergency prediction, mitigation, detection, and response with secure, automatic peer-to-peer notifications that share situational awareness across control, transmission, and reliability boundaries and prevent wide-spread, catastrophic power outages.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Kenneth M. Hopkinson, PhD
U	U	U	UU	240	19b. TELEPHONE NUMBER (937) 255-3636, ext 4579 (kenneth.hopkinson@afit.edu)

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18