2011

An Open Framework For Low-Latency Communications Across The Smart Grid Network

John Andrew Sturm
Indiana State University

AN OPEN FRAMEWORK FOR LOW-LATENCY COMMUNICATIONS ACROSS THE SMART GRID NETWORK

A Dissertation

Presented to

The College of Graduate and Professional Studies

College of Technology

Indiana State University

Terre Haute, Indiana

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy in Technology Management

---

by

John Andrew Sturm

December 2011

Keywords: Technology Management, Smart Grid, Automation Control, Digital Communications, Cryptography, OpenSSL, OpenVPN, OpenHIP, Open Grid

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

COMMITTEE MEMBERS

Committee Chair: Dr. Gerald Cockrell, Ed.D.

Professor, College of Technology

Indiana State University

Committee Member: Dr. George Maughan, Ed.D.

Professor and Program Director, College of Technology

Indiana State University

Committee Member: Dr. David Beach, PhD

Professor, College of Technology

Indiana State University

Committee Member: Dr. Yuetong Lin, PhD

Assistant Professor, College of Technology

Indiana State University

Committee Member: Dr. Roobik Gharabagi, PhD

Associate Professor, Electrical Engineering, Parks College of Engineering

St. Louis University

ABSTRACT

The recent White House (2011) policy paper for the Smart Grid that was released on June 13, 2011, *A Policy Framework for the 21st Century Grid: Enabling Our Secure Energy Future*, defines four major problems to be solved and the one that is addressed in this dissertation is Securing the Grid. Securing the Grid is referred to as one of the four pillars to be built on an open technology framework. The problem of securing the grid is further defined that cybersecurity practices must provide the special, low-latency communications needed for real-time automation control (White House, 2011, p. 49). The National Institute of Standards and Technology (NIST) is tasked with development of the cybersecurity communication standards through establishment of the NIST Cybersecurity Working Group (CSWG). NIST CSWG further states that low-latency is critical for automation control on the Smart Grid (NISTIR-Vol.3, 2010). The research and experimental planning for the solution tested in this dissertation provide low-latency through a system of open protocols that include HMAC keys (Hashed Message Authentication Code) and cryptographic identification for real-time control across the Smart Grid. It is serendipitous that HMAC keys (Hashed Message Authentication Code) can be processed very fast so there is little delay/latency added to the overall file transfer process (Goutis et al, 2005). In addition the research results offer guidance on the additional latency of AES versus Blowfish encryption algorithms for file transfers.

PREFACE

The development of the Smart Grid (SG) network can be considered as one of the greatest global engineering challenges. As a means for managing energy development and worldwide transmission, the system promises to touch everyone's life on a nearly continuous basis. The U.S. national labs and other countries have identified many of the problems and requirements that are described in the White House policy paper (2011). The solutions, however, promise to be elusive and transitive given the cultural, social and technology differences among societies. This dissertation highlights some of the critical Smart Grid communications issues, tests an initial set of open technology solutions (i.e. OpenSSL, OpenVPN, OpenHIP, Open Grid, etc.) that address one of the sub-problems, namely low-latency automation control file transfers, and recommends an open framework for the future that is capable of evolving over time as new demands and technologies become available. This paper builds on the research by Goutis et al (2005) in order to provide high-speed file transfer using HMAC keys (Hashed Message Authentication Code) to achieve low-latency. Ultimately the payoff of the SmartGrid should be the creation of global wealth that benefits all people.

ACKNOWLEDGMENTS

My wife and three daughters provide ongoing support and motivation to accomplish all the coursework, research and writing for this dissertation and are offered my deepest appreciation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

CHAPTER 1

INTRODUCTION

Securing the Grid is one of the four major pillars for the Smart Grid and it is further clarified that cybersecurity practices must provide special, "low-latency communications needed for real-time control" (White House, 2011, p. 49). Likewise the meaning and context of the research conducted for this dissertation is from the perspective of a utility industry engineer that needs to execute real-time control over the Smart Grid. The research and experimental planning described in this dissertation provides low-latency for control communications through cryptographic methods in order to meet the stated requirements. Chapter 5 summarizes the key recommendations to control engineers on the open software approach taken in this dissertation to reduce latency while maintaining strong security and reliable communications transport.

Background

The National Institute of Standards and Technology (NIST) was tasked with development of the appropriate cybersecurity communication standards through establishment of the NIST Cybersecurity Working Group (CSWG) to solve the problem. The NIST CSWG anticipated the tasking with a large research effort coordinated across government and commercial experts that is documented in a three volume set called the *Guidelines for Smart Grid Cyber Security* (NISTIR-Volumes 1-3, 2010). The NIST documents spell out the problems and requirements for the Smart Grid. Among the requirements for real-time automation control are high speed file

transfer capability to provide low-latency; strong security with Identity Management for trustworthy communications including Denial of Service (DoS) resistance to reduce delays/outages; and a reliable communications transport vehicle (i.e. protocol). Security is further defined as end-to-end trust (E2E trust) that implements cryptographic means of authentication at each end-point and also seamless security across all the protocol layers and routers, proxies, etc. between user interfaces and/or other devices. See the list of Acronyms in Appendix#2 per *Guidelines for Smart Grid Cyber Security: Vol. 3, Supportive Analyses and References* (NISTIR 7628-Vol. 3, 2010).

The first NIST requirement listed above is high speed file transfer capability to provide low-latency. Part of the innovativeness of the research in this dissertation is that the Identity Management technique proposed is to insert a Host Identity Tag (HIT), developed by the OpenHIP IETF project (Gurtov, 2008), within the secret pre-shared HMAC key of the OpenVPN transport protocol (OpenVPN, 2011). It is serendipitous that HMAC keys (Hashed Message Authentication Code) can be processed very fast so there is little delay/latency added to the overall file transfer process (Goutis et al, 2005). In their IEEE paper, *Efficient Small-Sized Implementation of the Keyed-Hash Message Authentication Code (HMAC)*, they tested a design approach to create a small-sized, high-speed HMAC processing model (Goutis et al, 2005, p. 1) and "The main contribution of the paper is the increase of the HMAC throughput to the required level to be used in modern telecommunication applications, such as VPN" tunnels (i.e. OpenVPN). This paper builds on the research by Goutis et al (2005) in order to provide high-speed file transfer using the File Transfer Protocol (FTP) over OpenVPN tunnels and to achieve low-latency.

The second requirement listed above is strong security with Identity Management for trustworthy communications that includes Denial of Service (DoS) resistance to reduce delays/latency. NIST states that (NIST Special Publication 1108, 2010, p. 39), "For the correct operation of IP (Internet Protocol) networks in Smart Grid environments, a suite of protocols must be identified and developed on the basis of standards defined by the Internet Engineering Task Force (IETF), commonly referred to as Request for Comments (RFCs)". The OpenHIP (Host Identity Protocol) is an on-going IETF research project that has been in development for over ten years, meets the NIST requirements since it is defined in numerous RFCs (IETF RFC#5201, 5202, 5204, 5205, 5206, 5338, 5770, 6078, 6079, etc.) and promises to provide secure identity with DoS resistance for all nodes on the internet (Gurtov, 2008).

The use of Host Identity Tags (HITs) from the OpenHIP technology project within the OpenVPN protocol provides strong Identity Management that enables Denial of Service (DoS) resistance and reduces latency in two ways: 1) because it authenticates the identity of each end-point in the network with a cryptographic name as part of setting up the communications path, and 2) because the OpenVPN transport protocol resists DoS attacks when it incorporates a secret key such as the proposed OpenHIP HIT Tag. In summary, the Identity Management technique proposed is to insert a HIT Tag within the secret pre-shared passphrase space allocated inside the TLS-Auth HMAC key of the OpenVPN transport protocol (OpenVPN, 2011).

Regarding the third requirement for a reliable communications transport vehicle, NIST proposes TLS (Transport Layer Security) as one possible protocol for consideration (NIST Special Publication 1108, 2010, p. 88). TLS is currently used across the internet in the widely accepted SSL/TLS (Secure Socket Layer/Transport Layer Security) protocol that was developed for reliable file transfers across the internet by Netscape (Hosner, 2004). OpenVPN is an open

software version of the SSL/TLS protocol that is built on the OpenSSL library. OpenVPN uses the OpenSSL cryptographic technology to create Virtual Private Networks (VPNs) that utilize SSL/TLS connections between and among various end-points such as the Smart Grid nodes (Yonan, 2004). In summary, OpenSSL is a public library of essential encryption techniques and cryptographic algorithms used to protect identity, information delivery and storage (OpenVPN, 2011). The combination of OpenVPN and OpenSSL creates a reliable transport vehicle on an open software technology framework that is required for Smart Grid communications.

Statement of the Purpose/Need

In summary, this dissertation combines the attributes and capabilities of existing technologies in a synergistic way to reduce the latency of control communications and meet the White House and NIST requirements for the Smart Grid. The solution tested in this dissertation utilizes cryptographic software (i.e. HMAC keys) and crypto-identities (i.e. HIT Tags) to achieve low-latency for automation control, while satisfying the need for end-to-end trust within an open technology framework for the Smart Grid in order to meet the stated requirements.

Statement of the Problem

The problem statement of this research was: the need for low-latency across local and remote SmartGrid network nodes in order to transmit automation control parameters that achieve acceptable levels of performance, security and reliability using an open technology framework. The major problem is divided into a number of research questions listed below.

Research Questions

Research Question #1

The first research question is: whether the 128-bit Host Identity Tag (HIT) cryptographic ID can be integrated inside the communication software protocols (i.e. OpenSSL, OpenVPN, and

OpenHIP) in order to create a common means of identity? Should the various open protocol software programs be able to function properly with the proposed 128-bit HIT crypto-ID, then the performance of the solution can be tested.

A cryptographic research testing lab was set up for this dissertation to examine file transfer speed/latency using HIT tags for identity within the well-known File Transfer Protocol (FTP). FTP packets were then conveyed within the OpenVPN protocol in order to add security as described in RFC 4217, *Securing FTP with TLS* (Ford-Hutchinson, 2005). The combination of the FTP, OpenSSL, OpenVPN and OpenHIP protocols act as a simple and convenient test vehicle. The speed (i.e. latency) of FTP file transfers represents the Dependent Variable (DV) to be tested and the Independent Variable (IV) is the size of the data payload. Should the open protocol software programs function properly with the HIT crypto-ID, then the performance can begin to be tested in a step-wise fashion: is the size of the payload related to the overhead/latency of file transfers?

Research Question #2

The second research question is: whether the 128-bit HIT crypto-ID adds significant overhead and latency to SSL/TLS file transfers for real-time control communications? The speed (i.e. latency) of FTP file transfers represents the Dependent Variable (DV) to be tested and the Independent Variable (IV) is the existence of the Host Identity (HIT) cryptographic ID tag within the file transfer packets. Does the presence of the HIT crypto-ID add significant overhead and thus latency to file transfers?

Research Question #3

The third question is: whether the type of encryption used (e.g. AES, Blowfish, etc.) affects the speed of file transfers? The Independent Variables (IV) are the type of encryption and

the length of the encryption keys. Do the type and key length of encryption add significant overhead and thus latency to file transfers?

Ultimately the dissertation provides a tested solution to the main research problem, namely: the need for low-latency across local and remote SmartGrid network nodes in order to transmit automation control parameters that achieve acceptable levels of performance, security and reliability using an open technology framework.

Type of Study

The following dissertation research is an empirical study performed through laboratory modeling and testing in order to support a statistical analysis and comparison of multiple file transfers that address the research questions above in order to form a set of recommendations (Leedy, 2010). Chapter 3 on Methodology describes the statistical approaches used.

Null Hypotheses

The Null Hypotheses are that there is no statistically significant relationship between each of the Independent Variables (file size payload, existence of the 128-bit HIT crypto-ID tag, type of encryption, and encryption key length) with the Dependent Variable (file transfer speed/latency).

1. Null Hypothesis Statement #1: Ho_1: $\beta_1 = 0$. There is no statistically significant relationship between the payload-size and the file transfer speed (latency).
2. Null Hypothesis Statement #2: Ho_2: $\beta_2 = 0$. There is no statistically significant relationship between the existence of the HIT crypto-ID tag within the file transfer packets and the file transfer speed (latency).

3. Null Hypothesis Statement #3: Ho_3: $\beta_3 = 0$. There is no statistically significant relationship between the type of encryption used (AES, Blowfish) to protect the file transfer and the file transfer speed (latency).
4. Null Hypothesis Statement #4: Ho_4: $\beta_4 = 0$. There is no statistically significant relationship between the key length of encryption used (128, 256) to protect the file transfer and the file transfer speed (latency).

The Alternative Hypotheses are that there is a statistically significant relationship between each of the Independent Variables (file size payload, existence of the 128-bit HIT ID tag, type of encryption, and encryption key length) with the Dependent Variable, file transfer speed, that cause the transfer speed (and associated latency/delay) to vary.

Hardware and Software Research Tools

The protocols were tested using standard PCs and IP-type network routing equipment to simulate the transmission lines that are typical among Utilities and Manufacturers for automation control. For testing and measurement purposes, file transfer protocol (FTP) servers and protocol/packet sniffers, such as Wireshark (formerly Ethereal), Traceroute, Ping, etc. were set up as illustrated in Figure 1. The File Transfer Protocol (FTP) was used to simulate, test and time file transfers for statistical analysis. Standard Statistical Analysis software (SPSS) was utilized to collect, validate and then analyze the data sets (Norusis, 2008).

The following image (Lamping, 2010) illustrates the shared Ethernet private test LAN (local area network) that was set up as the test lab environment. The hub router created a shared network, meaning all packets were received by all nodes on the network. Since the Ethernet adapter on the network was put into promiscuous mode, all packets on the network were seen by that adapter and thus captured for analysis. The shared media enabled the Wireshark network

sniffer (see laptop image below) to capture and record all file transfers with precise timing markers for later analysis.

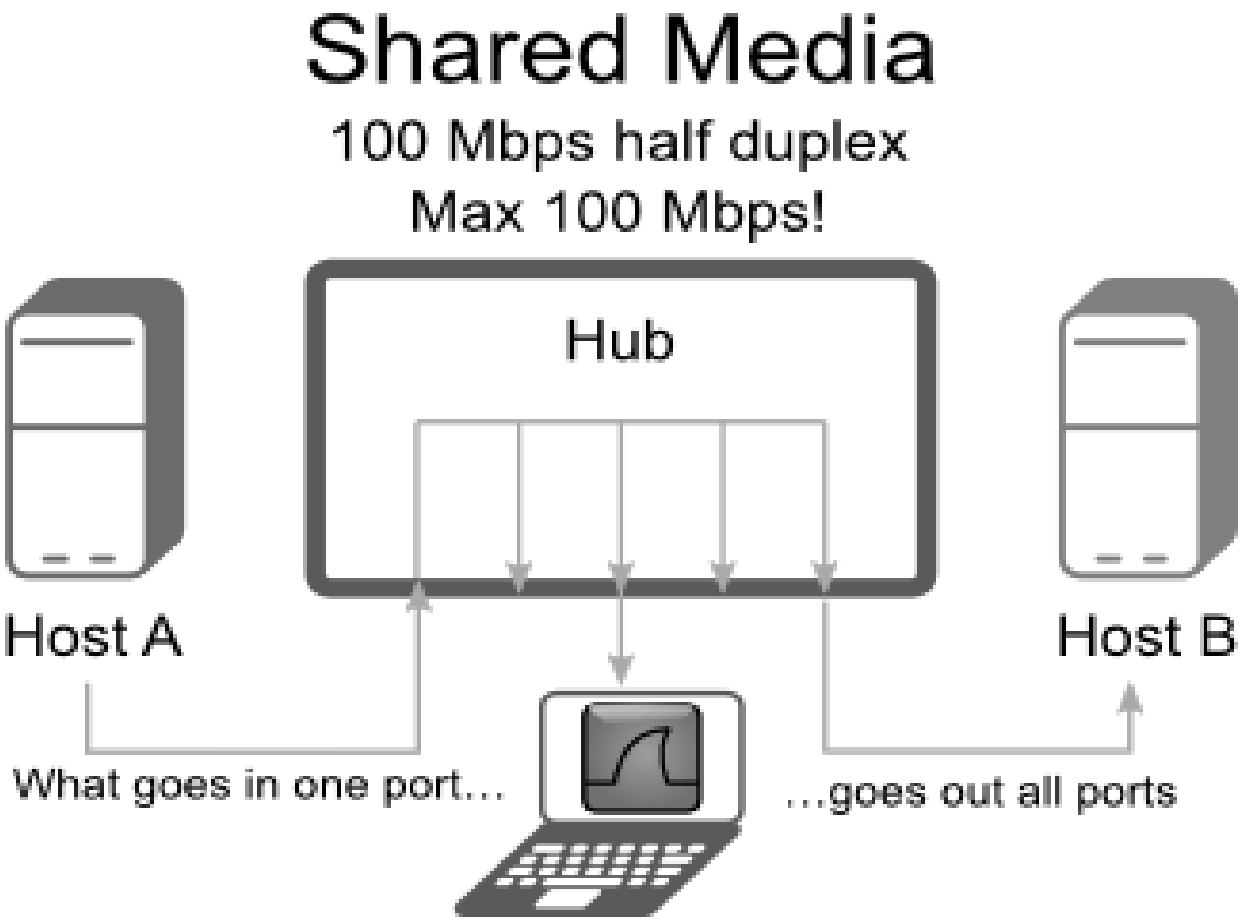


Figure 1 Laboratory Setup using Wireshark and SPSS Statistical Analysis on a laptop to collect and analyze the experimental data (Lamping, 2010).

Statement of Assumptions and Limitations

The major assumptions made during the experiments were that:

1. The OpenVPN software functioned as a research tool and enabled the connection of the various protocols such as OpenSSL and OpenHIP into a common software model for testing.

Specifically, the TLS-authority capability in OpenVPN functioned as a suitable authentication layer inside the communications protocol stack to incorporate and test the use of Host Identity Tags (HITs).

2. Another assumption was that the use of the FTP file transfer protocol served as an appropriate research communications vehicle to determine the latency.

Following are a list of limitations encountered:

1. A major limitation was that the research lab testing does not address scalability. The use of only a few computers does not simulate the tremendous volume of traffic expected on the Smart Grid that will potentially employ millions of routers, computers and automation control devices.
2. The Microsoft FTP server and client software only provided accuracy to a hundredth of a second and that limited the range of testing such that the minimum payload was 10MB (megabytes).
3. The Research LAN was isolated and not connected to any other computers or the Internet. Additional testing needs to be conducted on the impact of extraneous noise.
4. In addition, the amount of random CPU utilization due to background services in the operating system (Microsoft XP) caused variation in the processing power available for testing and had to be monitored closely during testing. All unnecessary services such as security, firewall, disk monitoring, I/O interrupts, etc. were turned off to minimize background task processing.
5. Wireshark could not be installed on the FTP server or client for timing because of additional load variations and interrupts on the network LAN adapter.

6. Network monitoring and logging had to be accomplished through Wireshark on a third, dedicated computer that listened quietly without consuming any resources and not affecting the processing on other nodes.
7. To maintain consistent CPU processor performance, utilization and availability for all tests, all experiments were conducted in a single day-long (12 hour) session that placed a limit on the number of tests that could be conducted.

Significance of the study

As mentioned, this topic is timely since the area of research supports the development of low-latency automation control communications that could securely and reliably operate across countries in order to support widely distributed Smart Grid communications in line with the recent White House (2011) policy paper. The Assumptions and Limitations are discussed further in Chapter 5 as part of the Conclusions and Recommendations.

Definition of Terms

The following list of terms and definitions explain important industry-related terminology that is used in this dissertation. The sources are the Smart Grid Interoperability Panel: A New, Open Forum for Standards Collaboration (2010) and the NISTIR 7628 Guidelines for Smart Grid Cyber Security v1.0, Vol. 1- 3. (2010):

CSWG	Cyber Security Working Group: identifies and analyzes security requirements and develops a risk mitigation strategy to ensure the security and integrity of the Smart Grid.
End-to-End Trust	(E2E Trust) Cryptographic means of authentication at each end-point and also seamless security across all the protocol layers and routers, proxies, etc. between user interfaces and/or other devices.

Low-Latency

Latency is a measure of the delay in the Availability of information on the Smart Grid and Availability is the most important security objective for power system reliability. Low-latency is defined as timely communications within the NIST guidelines expressed below in order to provide Availability of critical information along the Smart Grid network:

- ≤ 4 ms for protective relaying;
- Subseconds for transmission wide-area situational awareness monitoring;
- Seconds for substation and feeder SCADA data;
- Minutes for monitoring noncritical equipment and some market pricing information;
- Hours for meter reading and longer-term market pricing information; and
- Days/weeks/months for collecting long-term data such as power quality information.

Open Framework

Initiated by the National Institute of Standards and Technology (NIST), the Smart Grid Interoperability Panel (SGIP) plays a leadership role in facilitating and developing the Open Framework. The SGIP identifies and addresses standardization priorities. The starting point for this activity is the NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 1.0, which was issued in January 2010 for the transformation of the power system to the Smart Grid.

SGIP	Smart Grid Interoperability Panel is a new, open forum for standards collaboration . . . . Established in late 2009, the SGIP is a public‐private partnership dedicated to the interoperability of Smart Grid devices and systems—from home appliances to transmission substations to wind farms and other bulk power generators.
Smart Grid	An advanced electric power system that enables two-way flows of energy and information, promotes efficiency, and enables growing use of solar, wind, and other renewable energy sources. In the Energy Independence and Security Act of 2007 (EISA), the U.S. Congress established the development of a "smart" electric power grid as a national policy goal. Essential components of the Smart Grid, as conceived in the EISA legislation, include: standards, an information architecture, a cybersecurity strategy and an (open) framework for testing and certification.

CHAPTER 2

LITERATURE REVIEW

This chapter contains a summary of the available literature that addresses the Smart Grid communication requirements, related on-going research projects, and related efforts in building an open technology framework to support Smart Grid communications:

1. National Institute of Standards and Technology Requirements
    a. Smart Grid Cyber Security NIST Requirements, and
    b. Industrial Control Systems NIST Requirements.
2. Open Technology Framework Requirements and Related Research
    a. Open Cryptographic HMAC Standard from NIST for Low-Latency Communications,
    b. OpenHIP Host Identity Protocol for Identification and Authentication with Denial of Service (DoS) Resistance,
    c. OpenSSL and OpenVPN Protocol Models for Transport Layer Security (TLS) and DoS Resistance support, and
    d. Summary of the Open Framework Solution.

Taken together and viewed as one large system of systems (SoS) for automation control communications, the open technology components listed above also form the theoretical basis for the dissertation research and testing.

Background

The recent White House (2011) policy paper for the Smart Grid, *A Policy Framework for the 21st Century Grid: Enabling Our Secure Energy Future*, defines four pillars to be built on an open technology framework and the four pillars are:

1. Enabling cost-effective smart grid investments,
2. Unlocking the potential for innovation in the electric sector,
3. Empowering consumers and enabling them to make informed decisions, and
4. Securing the grid.

As stated, the problem that is addressed in this dissertation is Securing the Grid and it is further defined that cybersecurity practices must provide the special, low-latency communications needed for real-time control (White House, 2011). The National Institute of Standards and Technology (NIST) was tasked with development of the cybersecurity communication requirements and standards through establishment of the NIST Cybersecurity Working Group (CSWG).

National Institute of Standards and Technology Requirements

Smart Grid Cyber Security NIST Requirements

The NIST CSWG anticipated the tasking with a large research effort that is documented in a three volume set called the *Guidelines for Smart Grid Cyber Security* (NISTIR-Volumes 1-3, 2010). The NIST documents spell out the problems and requirements for the Smart Grid. NIST CSWG further stated that Identification and Authentication (i.e. Identity Management) is a critical element for trustworthy communications and an important part of the Smart Grid Security Requirements as illustrated in Table 1 (NISTIR-Vol.3, 2010):

NIST INTERAGENCY REPORT VOLUME 3 – APPENDIX H
MAPPINGS TO THE HIGH-LEVEL REQUIREMENTS

The following table is a High-Level mapping of research and development topics to the Smart Grid Security Requirements Families [See§8]:

- Identification and Authentication (SG.IA)
- Access Control (SG.AC)
- Awareness and Training (SG.AT)
- Audit and Accountability (SG.AU)
- Configuration Management (SG.CM)
- Continuity of Operations (SG.CP)
- Incident Response (SG.IR)
- Information and Document Management (SG.ID)
- Media Protection (SG.MP)
- Personnel Security (SG.PS)
- Physical and Environmental Security (SG.PE)
- Strategic Planning (SG.PL)
- Security Assessment and Authorization (SG.CA)
- Security Program Management (SG.PM)
- Planning (SG.PL)
- Smart Grid Information System and Communication Protection (SG.SC)
- Smart Grid Information System and Information Integrity (SG.SI)
- Smart Grid Information System and Services Acquisition (SG.SA)
- Smart Grid Information System Development and Maintenance (SG.MA)

Following are the individual members of the Identification and Authentication Requirement Family (SG.IA):

- Cryptographic Key Management for Identity
- Architecting Real-time security
- DoS/DDoS Resiliency
- Privacy and Access Control in Federated Systems
- Cloud Security
- Distributed versus Centralized security

Table 1 Smart Grid Security Requirements identified in the NIST Interagency Report (NISTIR-Vol. 3, 2010) Appendix H for Identification and Authentication (SG.IA).

The bottom section of Table 1 lists the six security requirements for Identification and Authentication (i.e. Identity Management) for Smart Grid communications. The first three

requirements: 1) Cryptographic Key Management for Identity, 2) Architecting Real-time security for low-latency, and 3) DoS/DDoS (Distributed Denial of Service) Resiliency, directly address the White House (2011) policies and form the core of the research requirements for secure, low-latency automation control communications in this dissertation. The other three requirements: 4) Privacy and Access Control in Federated Systems, 5) Cloud Security, and 6) Distributed versus Centralized security, are topics for future research that help to define additional interfaces, extensions of the transport vehicle (i.e. protocol), and levels of reliability and agility needed to convey the automation control messages (NISTIR-Vol. 3, 2010).

In summary, the principle NIST requirements for real-time automation control are: 1) a high-speed file transfer architecture to provide low-latency; 2) strong security through cryptographic identity management for trustworthy communications that includes Denial of Service (DoS) resistance; and 3) a reliable, real-time communications transport vehicle (i.e. the TLS protocol). Likewise, the solution tested in this dissertation provides low-latency across local and remote SmartGrid network nodes in order to transmit automation control parameters that achieve acceptable levels of performance, security and reliability using an open technology framework.

Industrial Control Systems NIST Requirements

The Smart Grid is also an Industrial Control System (ICS) that manages power plant generation and distribution/transmission operations so it is critical that the qualities of the open technology framework also meet the requirements identified by NIST for Industrial Control Systems (NIST, 2004). For instance, the security mechanisms (i.e. encryption, protocol, policies/firewalls, etc.) should not delay the sensor and automation control communications to the extent that dead-time (i.e. excessive latency) occurs in the communications path and causes

process instability (Goutis et al, 2005). The NIST Security Analysis (NIST, 2004) is summarized in the following table from the System Protection Profile analysis to indicate the scope of the System Target of Evaluation (STOE) for industrial control systems (ICS) applicable to the SmartGrid:

Feature	Description
Identity and Authentication	Identity and Authentication of the following: ▪ Financial and business critical information sent from the ICS to external systems ▪ Configuration change commands affecting core ICS functions (e.g. control algorithms, set points, limit points etc) ▪ Users and services accessing the protected assets (e.g. actuators, control systems, etc)
Confidentiality	Protection of business, financial and control data from unauthorized disclosure (as determined by risk assessment and approved by the data or system owner), including, but not limited to, appropriate segments within the ICS network.
Integrity	Protection against the unauthorized modification of the following: ▪ Information flows of a sensitive nature on exposed network segments ▪ Internal control data used throughout the ICS ▪ ICS operational system configuration
Availability (including DoS Resiliency)	Protection against the loss of availability of all critical and major ICS operational systems including, but not limited to, ▪ Control servers

Feature	Description
	▪ Primary communications channel (or network) ▪ ICS operational system configuration capability
Boundary Protection	Protection against unauthorized attempts to breech both the physical and the logical boundaries of the ICS.
Access control	Strict access control for the following: ▪ On-site and off-site remote access into the ICS network ▪ Externally-visible interfaces of the ICS ▪ System resources deemed by the owner(s) as requiring protection ▪ Those system functions capable of modifying ICS configuration ▪ Critical ICS processes based on state information relevant to that process (e.g. time of day, location, etc)
Backup / Recovery	Backup mechanisms for critical ICS data and control information to enable timely recovery from system compromises or damage.
Audit	Entries in the audit log of appropriate ICS components detailing the successful and unsuccessful security relevant activities of users and applications.
Monitoring	Monitoring and detection of unauthorized activity, unusual activity and attempts to defeat the security capabilities of the ICS, including the deployment of intrusion detection systems (IDS) at critical parts of the ICS infrastructure.

Feature	Description
Non-interference with safety functions	Non-interference of ICS security functions and safety-critical functions while maintaining ICS performance.
Self Verification	Self-tests to verify the configuration and integrity of the security functions of the ICS.
Emergency power	Emergency power sufficient to allow for graceful shutdown of the ICS in the event that primary and secondary power fail.
Security Plans, Policies & Procedures	Security plans, policies and procedures covering at least the following: ▪ Overarching security policy governing the access and necessary protection for all ICS components ▪ Security management of the ICS and associated infrastructure ▪ Security management roles and responsibilities ▪ Documentation of the organizational risk management process ▪ Business continuity and disaster recovery plans for the ICS ▪ Migration Strategy covering the identification, assessment and treatment of new or existing vulnerabilities ▪ Policies governing the roles, responsibilities and activities authorized for third parties interfacing with ICS components

Table 2 Detailed security and Information Assurance (IA) attributes indentified in the NIST System Target of Evaluation (STOE) used for security analysis of Industrial Control Systems (NIST, 2004).

Using the Internet for Industrial Control purposes such as the Smart Grid also exacerbates the existing security problems in the internet protocol (IP). The apparent ease of IP access to information anywhere at any time does not reveal the underlying security problems created by IP addressing that conflict with the NIST Guidelines in Table 2 for Identity and Authentication (NIST, 2004). The current, dual role of IP addresses as both end-point identifiers and the location of a network interface leads to confusion and a lack of assured identity (Jain, 2006). In particular, the need for end-to-end (E2E) trust requires not only cryptographic means of authentication at each end-point, but also seamless security across all the protocol layers and proxies between the user interface and other users/devices. Secure file transfers require a general purpose binding of user, data and transaction identity (Gurtov, 2008).

The adoption of the HIT cryptographic IDs from the IETF OpenHIP project for Identity Management across the Internet is analogous to obtaining a Driver's License in order to have the privilege of driving a car on the highway and is a strong deterrent to malicious conduct and hacking since it is then much more difficult to act anonymously (Gurtov, 2008). This dissertation takes one step further in applying the HIT crypto-IDs to objects such as Smart Grid automation control devices/sensors in order to meet the NIST Guidelines.

Open Technology Framework Requirements and Related Research

Viewed together, the open technologies of OpenSSL, OpenVPN, OpenHIP and Open Cryptography from NIST combine to form the solution presented in this dissertation and address the White House (2011) policies for an open technology framework. The open technology components of the solution are:

- Open Cryptographic HMAC Standard from NIST for Low-Latency Communications,

- OpenHIP Host Identity Protocol for Identification and Authentication with DoS Resistance, and
- OpenSSL and OpenVPN Protocol Models for TLS Support.

Viewed as one large system of systems (SoS) for automation control communications, the open technology components listed above also form the theoretical basis for the dissertation research and testing.

Open Cryptographic NIST HMAC for Low-Latency Communications

Part of the innovativeness of the research in this dissertation is to take advantage of the speed of HMAC (Hashed Message Authentication Code) cryptographic processing that was actually developed by NIST for the purpose of Identity Management (NIST-FIPS 198, 2002) inside the OpenVPN/OpenSSL protocol. The technique proposed is to insert a HIT identity tag within the pre-shared passphrase space allocated inside the TLS-Auth HMAC key of the OpenVPN protocol. It is serendipitous that HMAC keys (Hashed Message Authentication Code) can be processed very fast so there is little delay/latency added to the overall file transfer process (Goutis et al, 2005). In their IEEE paper, *Efficient Small-Sized Implementation of the Keyed-Hash Message Authentication Code (HMAC)*, they tested a design approach to create a small-sized, high-speed HMAC processing model (Goutis et al, 2005, p. 1) that can be translated into a small hardware device and as a result, "The main contribution of the paper is the increase of the HMAC throughput to the required level to be used in modern telecommunication applications, such as VPN" tunnels (i.e. OpenVPN). It is a direct logical step to build on the research of Goutis et al (2005) in order to achieve low-latency for Smart Grid control communications by using a similar HMAC algorithm from NIST (NIST-FIPS 198, 2002).

Per the NIST explanation of the HMAC standard (NIST-FIPS 198, 2002), NIST specifies the algorithm for applications requiring message authentication:

> Message authentication is achieved via the construction of a message authentication code (MAC). MACs based on cryptographic hash functions are known as HMACs. The purpose of a MAC is to authenticate both the source of a message and its integrity without the use of any additional mechanisms. HMACs have two functionally distinct parameters, a message input and a secret key known only to the message originator and intended receiver(s). . . . An HMAC function is used by the message sender to produce a value (the MAC) that is formed by condensing the secret key and the message input. The MAC is typically sent to the message receiver along with the message. The receiver computes the MAC on the received message using the same key and HMAC function as was used by the sender, and compares the result computed with the received MAC. If the two values match, the message has been correctly received and the receiver is assured that the sender is a member of the community of users that share the key.

The HMAC specification in this standard is a generalization of HMAC as specified in Internet RFC 2104, HMAC, Keyed-Hashing for Message Authentication, and ANSI X9.71, Keyed Hash Message Authentication Code. The HMAC specification provided by NIST computes a message authentication code (MAC) over the data 'text' using the HMAC function in the following operation (Equation 1):

$$\mathbf{MAC(text)_t = HMAC(K, text)_t = H((K_0 \oplus opad) || H((K_0 \oplus ipad) || text))_t} \qquad \mathbf{(1)}$$

Equation 1 HMAC specification provided by NIST computes a message authentication code (MAC) over the data 'text' using the HMAC function. Appendix #1 explains the same mathematical operations in a step-by-step process to calculate the HMAC.

In their research Goutis et al (2005) examined the SHA-1 hash function as an example of HMAC processing. It is an iterative algorithm that requires 80 transformation steps to generate the final hash value. The hash value resulting from the 80 iterations is a 160-bit Message Digest (MD). Goutis et al (2005) studied the efficiency of each step of the calculations and observed that some intermediate values can be pre-computed, stored in a register, and used without introducing any delay as illustrated in Figure 2:

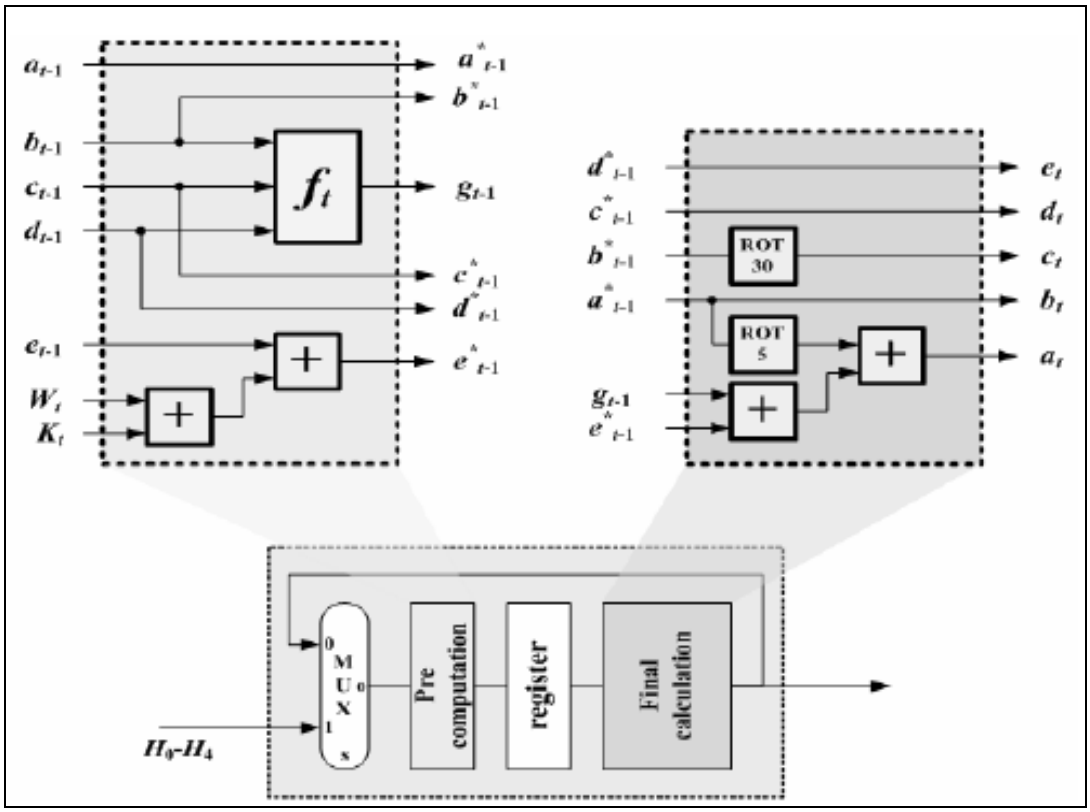


Figure 2 The modified SHA-1 operation block, separated in two calculation phases by Goutis et al (2005).

The two calculation phases consisting of the Pre-computation operations and the Final calculation phase can be processed intelligently to reduce the critical path and achieve significant performance advantages. Per Goutis et al (2005):

> The achieved throughput presents an increase compared to commercially available IP cores that range from 30%-390%. The main contribution of the paper is the increase of the HMAC throughput to the required level to be used in modern telecommunication applications, such as VPN (applications).

This is simply one example of the potential to speed up the HMAC processing time (and reduce latency) through innovative implementation into hardware that accelerates the processing steps.

OpenHIP Host Identity Protocol for Identification and Authentication

The Host Identity Protocol (HIP) from the IETF OpenHIP project (IETF RFCs 5201, 5202, 5204, 5205, 5206, 5338, 5770, 6078, and 6079) provides a significant step toward building a secure end-to-end (E2E) network for the Smart Grid control systems. HIP is a shim layer between the network and the transport layer to establish and manage secure connections between hosts and users/devices that are often mobile through strong Identity Management. Pekka Nikander (2004) is one of the original developers of OpenHIP and he states:

> HIP enhances the original Internet architecture by injecting a new thin (shim) layer between the IP layer and the transport protocols. This new layer introduces a new "Name Space" consisting of cryptographic identifiers, thereby implementing the so-called "identifier/locator" split. . . . The architectural enhancement implemented by HIP has profound consequences. A number of the previously hard problems become suddenly much easier. Mobility, multi-homing, and baseline end-to-end security integrate neatly

into the architecture. The use of cryptographic identifiers allows enhanced accountability, thereby providing a base for easier build up of trust (i.e. end-to-end trust). The OpenSSL, OpenVPN and Open Grid protocols are a natural fit with the Host Identity Protocol (HIP) to provide the essential binding capabilities needed to create E2E trust and thus the first research question described in Chapter 1 is to test whether the 128-bit Host Identity Tag (HIT) cryptographic ID can be integrated inside the communication software protocols (i.e. OpenSSL, OpenVPN, and OpenHIP) in order to create a common means of identity.

Figure 3 illustrates the basic architecture of the Host Identity Protocol (HIP). In the current TCP/IP architecture on the left, IP addresses represent both the location for routing purposes and the identity that is associated with port numbers and sockets for sessions. In the new HIP model on the right, there is a fifth layer inserted and the Host Identifiers, also called Host Identity Tags (HITs), translate into one or more IPV4 or IPV6 addresses (Nikander, 2004). The HIT serves as a 128-bit strong identifier for a host/user and is a cryptographic hash of the public key that was truncated (Nikander, 2004). The ease and speed of binding/rebinding of HITs and IP addresses (i.e. dynamic updating) enables strong identification to persist despite the fact that changing locations means that IP addresses often change rapidly (Gurtov, 2008).

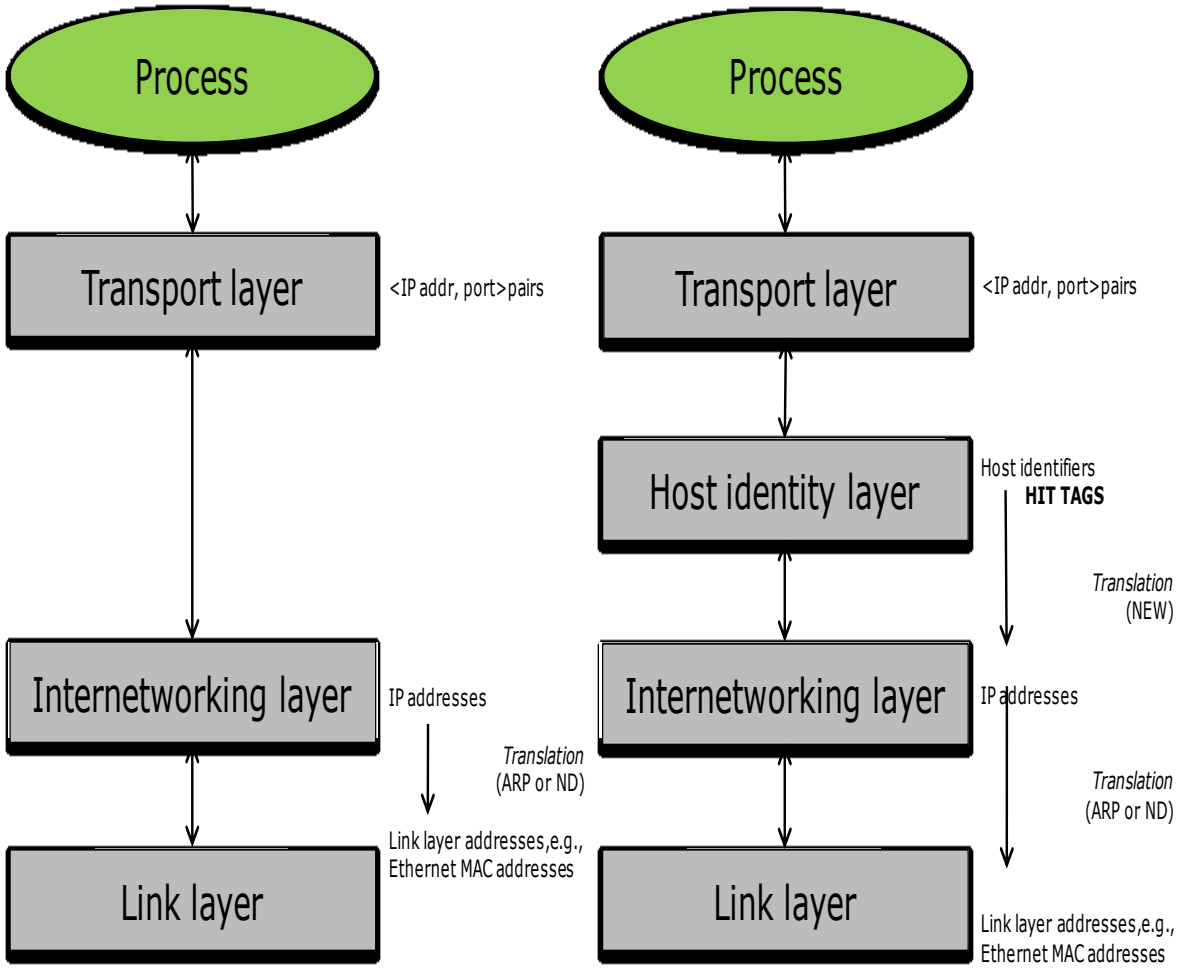


Figure 3 Current TCP/IP Framework on the left and the new proposed Internet Architecture on the right including the Host Identity Protocol (HIP) (Nikander, 2004).

OpenSSL and OpenVPN Protocol Models for TLS Support

Regarding the requirement for a reliable communications transport vehicle, NIST proposes TLS (Transport Layer Security) as one possible protocol for consideration (NIST Special Publication 1108, 2010, p. 88). TLS is currently used across the internet in the widely accepted SSL/TLS (Secure Socket Layer/Transport Layer Security) protocol that was developed for reliable file transfers across the internet by Netscape (Hosner, 2004). OpenVPN is an open software version of the SSL/TLS protocol that is built on the OpenSSL library. OpenVPN uses

the OpenSSL cryptographic technology to create Virtual Private Networks (VPNs) that utilize SSL/TLS connections between and among various end-points such as the Smart Grid nodes (Yonan, 2004). In summary, OpenSSL is a public library of essential encryption techniques and cryptographic algorithms used to protect identity, information delivery and storage (OpenVPN, 2011). The combination of OpenVPN and OpenSSL creates a reliable transport vehicle on an open software technology framework that is required for Smart Grid communications.

In addition, NIST requires Denial of Service (DoS) resistance in conjunction with the Identity Management specification as described in Appendix H of *Guidelines for Smart Grid Cyber Security: Vol. 3, Supportive Analyses and References* (NISTIR 7628-Vol. 3, 2010, p. H-4). An attacker to the Smart Grid could damage automation control communications simply by injecting a DoS attack that creates delays, and thus adds latency to the transmissions. OpenVPN resists DoS attacks when it incorporates a secret key (namely the OpenHIP HIT tag) as described in the OpenVPN 2.1 specifications (OpenVPN, 2011, p. 20):

> OpenVPN offers this special layer of authentication on top of the TLS control channel so that every packet on the control channel is authenticated by an HMAC signature and a unique ID for replay protection. This signature will also help protect against DoS (Denial of Service) attacks. An important rule of thumb in reducing vulnerability to DoS attacks is to minimize the amount of resources a potential, but as yet unauthenticated, client is able to consume. "TLS-Auth" does this by signing every TLS control channel packet with an HMAC signature (i.e. HIT Tag), including packets which are sent before the TLS level has had a chance to authenticate the peer. The result is that packets without the correct signature can be dropped immediately upon reception, before they have a chance to consume additional system resources such as by initiating a TLS handshake.

The TLS-Auth HMAC key described above uses the HIT Identity Tag as a secret key to sign and authenticate every packet on the control channel. A second set of HMAC keys is then derived and distributed to both ends under the protection of the HIT Tag-authenticated control channel. The second set of keys is used to protect the data channel using a similar HMAC algorithm inside the TLS tunnel (see Figure 4).

In addition to providing DoS resistance, the secret key (i.e. HIT Tag) adds end-to-end trust intrinsically to the protocol because it authenticates the identity of each end-point in the network with a cryptographic name as part of setting up the communications pathway. It is serendipitous that OpenVPN offers the additional layer of security called the Transport Layer Authority (TLS-Auth) that can be used to bind and test the union of user, control signaling, data and transaction identity so that the recipient can be assured of the provenance (i.e. who, when, what and how) that they received the data file (OpenVPN, 2011).

As illustrated in Figure 4, OpenVPN multiplexes the SSL/TLS session used for authentication and key exchange with the actual encrypted tunnel data stream. OpenVPN provides the SSL/TLS connection with a reliable transport layer as it is designed to operate over (OpenVPN, 2011). The actual IP packets, after being encrypted and signed with an HMAC, are tunneled over UDP.

This dissertation builds on similar research conducted by the OpenVPN Community Project that used FTP to test the speed of file transfers using the OpenVPN and OpenSSL protocols with and without encryption (OpenVPN, 2011). However, the previous research did not include testing OpenVPN and OpenSSL with and without the TLS-Auth capability for Denial of Service (DoS) resistance that is tested in this dissertation (Research Question #2).

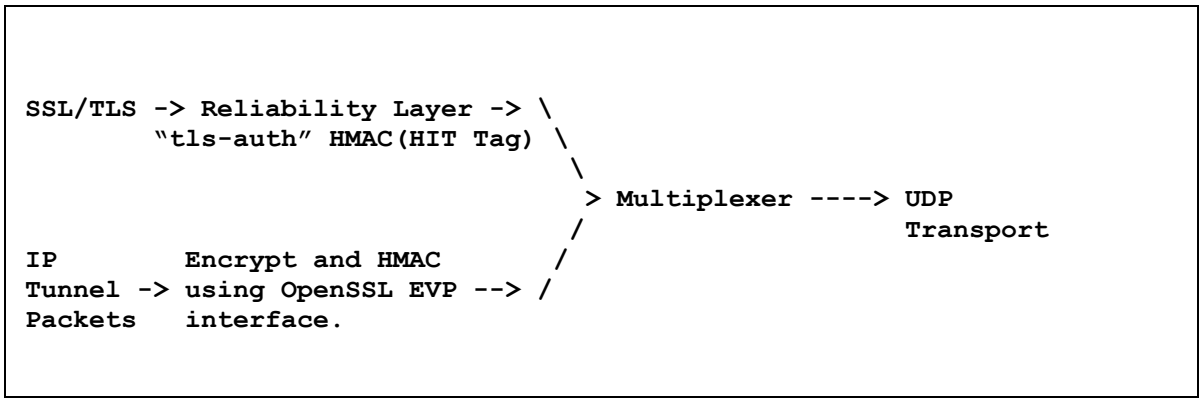


Figure 4 Overview of the OpenVPN and OpenSSL Security processing for SSL/TLS transmission and IP Tunnel Packets to carry the FTP file payload (OpenVPN, 2011). The tls-auth HMAC key shown above carries the HIT Identity Tags.

Summary of the Open Framework Solution

As reviewed above, this dissertation tests a set of open technology solutions (i.e. OpenSSL, OpenVPN, OpenHIP, Open HMAC Cryptography from NIST, etc.) and recommends an open framework for the Smart Grid that is capable of evolving over time as new demands and technologies become available. A common means of Identity Management is proposed that supports secure exchange of files with DoS resistance for reduction of latency and creates a linchpin/link across the various open systems. It is serendipitous that OpenVPN offers the additional layer of security called the Transport Layer Authority (TLS-Auth) that can be used to bind and test the union of user, control signaling, data and transaction identity over the TLS protocol so that the recipient can be assured of who, what and how (i.e. the provenance) that they received the data file. The IETF standard for OpenHIP is built on IPsec (IP Security), but the OpenHIP concept is flexible enough that it can also be built on the TLS protocol. Per Gurtov (2008), OpenHIP "encryption could be also implemented on the upper layer such as SSH or TLS. . . . a different transport mode than IPsec might be needed in HIP in some cases" (p. 48).

Currently there is IRTF research activity to develop a Lightweight HIP version called LHIP that uses the HMAC algorithm and Interactive Hash Chains (IHC).

It is also serendipitous that the HMAC (Hashed Message Authentication Code) standard is supported by NIST and can be processed very fast so there is little delay added to the overall file transfer process. In summary, this dissertation combines the attributes and capabilities of existing open technologies in a synergistic fashion to make them more practical, efficient and open for use.

CHAPTER 3

METHODOLOGY

The research and experimental planning described in this dissertation provides low-latency through a system of open protocols and cryptographic identification for real-time automation control across the Smart Grid. This dissertation proposes a common means of identity management, namely the use of Host Identity Tags (HITs) from the OpenHIP technology project to reduce latency and securely exchange automation control files across various open systems. The HIT identity management approach is being developed in the OpenHIP IETF research project as a cryptographic means to provide secure identity.

For the purpose of this dissertation, a research lab was set up to examine file transfers using HIT Tags for identity purposes within the well-known File Transfer Protocol (FTP). FTP packets were then conveyed within the secure OpenVPN protocol in order to reduce latency and add security. The combination of the FTP, OpenHIP, OpenSSL and OpenVPN protocols act as a simple and convenient test vehicle to evaluate the capabilities of the HIT Identity Management approach in the research lab.

Research Design and Procedures

Statement of the Problem

The problem statement of this research was: the need for low-latency across local and remote SmartGrid network nodes in order to transmit automation control parameters that achieve acceptable levels of performance, security and reliability using an open technology framework. The major problem is divided into a number of research questions listed below.

Research Questions

Research Question #1

The first research question is: whether the 128-bit Host Identity Tag (HIT) cryptographic ID can be integrated inside the communication software protocols (i.e. OpenSSL, OpenVPN, and OpenHIP) in order to create a common means of identity? Should the various open protocol software programs be able to function properly with the proposed 128-bit Host Identity Tag (HIT) cryptographic ID, then the performance of the solution can be tested.

A cryptographic research testing lab was set up for this dissertation to examine file transfer speed/latency using HIT tags for identity within the well-known File Transfer Protocol (FTP). FTP packets were then conveyed within the OpenVPN protocol in order to add security as described in RFC 4217, *Securing FTP with TLS* (Ford-Hutchinson, 2005). The combination of the FTP, OpenSSL, OpenVPN and OpenHIP protocols acted as a simple and convenient test vehicle. The speed (i.e. latency) of FTP file transfers represents the Dependent Variable (DV) to be tested and the Independent Variable (IV) is the size of the data payload. Should the open protocol software programs function properly with the HIT crypto-ID, then the performance can begin to be tested in a step-wise fashion: is the size of the payload related to the overhead/latency of file transfers?

Research Question #2

The second research question is: whether the 128-bit HIT crypto-ID adds significant overhead and latency to SSL/TLS file transfers for real-time control communications? The speed (i.e. latency) of FTP file transfers represents the Dependent Variable (DV) to be tested and the Independent Variable (IV) is the existence of the Host Identity (HIT) cryptographic ID tag within the file transfer packets. Does the presence of the HIT crypto-ID add significant overhead and thus latency to file transfers?

Research Question #3

The third question is: whether the type of encryption used (e.g. AES, Blowfish, etc.) affects the speed of file transfers? The Independent Variables (IV) are the type of encryption and the length of the encryption keys. Do the type and key length of encryption add significant overhead and thus latency to file transfers?

Ultimately the dissertation provides a tested solution to the main research problem, namely: the need for low-latency across local and remote SmartGrid network nodes in order to transmit automation control parameters that achieve acceptable levels of performance, security and reliability using an open technology framework.

Type of Study

The proposed dissertation research is an empirical study performed through laboratory modeling and testing in order to support a statistical analysis and comparison of multiple file transfers that address the research questions above in order to form a set of recommendations (Leedy, 2010).

Null Hypotheses

The Null Hypotheses are that there is no statistically significant relationship between each of the Independent Variables (file size payload, existence of the 128-bit HIT crypto-ID tag, type of encryption, and encryption key length) with the Dependent Variable (file transfer speed/latency).

1. Null Hypothesis Statement #1: Ho_1: $\beta_1 = 0$. There is no statistically significant relationship between the payload-size and the file transfer speed (latency).
2. Null Hypothesis Statement #2: Ho_2: $\beta_2 = 0$. There is no statistically significant relationship between the existence of the HIT crypto-ID tag within the file transfer packets and the file transfer speed (latency).
3. Null Hypothesis Statement #3: Ho_3: $\beta_3 = 0$. There is no statistically significant relationship between the type of encryption used (AES, Blowfish) to protect the file transfer and the file transfer speed (latency).
4. Null Hypothesis Statement #4: Ho_4: $\beta_4 = 0$. There is no statistically significant relationship between the key length of encryption used (128, 256) to protect the file transfer and the file transfer speed (latency).

The Alternative Hypotheses are that there is a statistically significant relationship between each of the Independent Variables (file size payload, existence of the 128-bit HIT ID tag, type of encryption, and encryption key length) with the Dependent Variable, file transfer speed, that causes the transfer speed (and associated latency/delay) to vary.

Laboratory Instrumentation

Hardware and Software Research Tools

The protocols were tested using standard PCs and IP-type network routing equipment to simulate the transmission lines that are typical among Utilities and Manufacturers for automation control. For testing and measurement purposes, file transfer protocol (FTP) servers and protocol/packet sniffers, such as Wireshark (formerly Ethereal), Traceroute, Ping, etc. were set up as illustrated below. The File Transfer Protocol (FTP) was used to simulate, test and time file transfers for statistical analysis. Standard Statistical Analysis software (SPSS) was utilized to collect, validate and then analyze the data sets (Norusis, 2008).

Laboratory Set Up

Figure 5 illustrates the shared Ethernet private test LAN (local area network) that was set up as the test lab environment. The hub router created a shared network, meaning all packets were received by all nodes on the network. Since the Ethernet adapter on the network was put into promiscuous mode, all packets on the network were seen by that adapter and thus captured for analysis. The shared media enabled the Wireshark network sniffer (see laptop image below) to capture and record all file transfers with precise timing markers for later analysis.

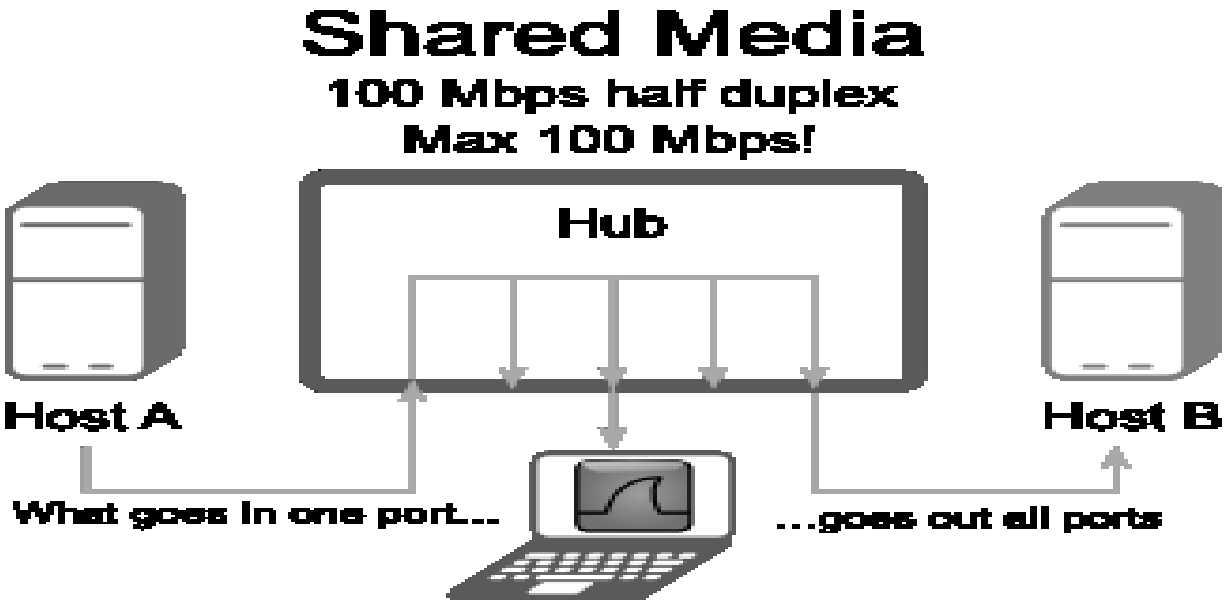


Figure 5 Laboratory Setup using Wireshark and SPSS Statistical Analysis on a laptop to collect and analyze the experimental data (Lamping, 2010).

Data Analysis Procedures

The test experiments were designed to understand the underlying relationships among the variables that impact file transfer speed and/or delays (latency) across the Smart Grid IP network. Multivariate Analysis of Variance (MANOVA) described later in this paper is used to highlight important interaction effects and to improve the associated predictive statistical model. Multivariate ANOVA is an effective way to calculate and examine the effects of factor(s) on several dependent variables at once using a general linear model in which the factors divide the cases into groups (Norusis, 2008). In addition, it enables the measurement of significant interaction effects, such as encryption type and key length. An additional research question is whether the relationships can be measured among the variables (and covariates) with the dependent variable, file transfer speed (in addition to the interaction effects among the variables) within a 5% level of significance using MANOVA.

Identification and Explanation of Variables

The predictive equation that describes the relationship among the variables is represented in Equation 2. It incorporates the dependent variable for file transfer speed (latency) represented as $\hat{Y}$; the four independent variables, payload-size (coefficient β_1), crypto-identifier type (coefficient β_2), encryption-type (coefficient β_3) and key-length (coefficient β_4); the constant C represents the value for $\hat{Y}$ when the β values are zero; the interaction effects among the variables; and finally the residuals are represented as $\acute{\varepsilon}$:

$$\hat{Y}=C+\beta_1 payloadsize+\beta_2 cryptoID+\beta_3 encryptype+\beta_4 keylength+interaction\ effects+\acute{\varepsilon} \quad (2)$$

Equation 2 Predictive equation for file transfer speed that describes the relationship among the Dependent and Independent variables.

Type I / Type II Error

Type I and II error rates are dependent on the accuracy of the measuring tools used for the data collection. The research lab local area network (LAN) is isolated from the internet to minimize disruptions and noise. Though every effort is made to prevent disruptions (power outages, surges, variations in CPU or LAN router speed, etc.) that affects the data timing/quality, there are some uncontrollable, extraneous noise problems due to the fact that the data collection laptop is not certified according to a standards body like NIST. Thus it is not be appropriate to apply a higher degree of significance to the results than the quality of the data being used for the analysis. At a 5% significance level, one expects only 5 out of 100 surveys to be able to reject the null hypotheses being tested. It appears that there might be more than a 5% error in the data,

therefore, it is not believed that the quality of the data deserves to be used to reveal any better accuracy than a 5% significance level.

In other words, one does not want to find Type I Error in the results by rejecting the Null Hypothesis when one should not have. At the same time, one does not want to fail to detect a true difference (i.e. between means) when it really does exist. Therefore a 5% significance level is the best that can be used given the available laboratory network quality of the data. Likewise, a 95% Confidence Level is used in the various analyses.

Threats to Validity

As mentioned Multivariate ANOVA (MANOVA) is an effective way to calculate and examine the effects of factor(s) on several dependent variables at once using a general linear model in which the factors divide the cases into groups (Norusis, 2008). In addition, MANOVA also enables one to measure the significance of Interaction Effects, such as encryption type and key length.

MANOVA analysis requires independence of observations, an interval or continuous dependent variable (namely file transfer speed), homogeneity of variance and normality (Norusis, 2008). The MANOVA analysis is not particularly sensitive to the normalcy of the data, so it represents a good alternative test to multiple linear regression (MLR) analysis that does require normally distributed variables.

CHAPTER 4

RESULTS

The results of the cryptographic experiments are presented in sequential order to address each of the Research Questions because each lab test adds more data in a step-wise fashion that builds on the previous results.

First Research Question

Integration of the HIT crypto-IDs in the Communication Software

Part of the first research question is whether the 128-bit Host Identity Tag (HIT) cryptographic IDs from OpenHIP can be integrated inside the communication software protocols (i.e. OpenSSL and OpenVPN) in order to create a common means of identity. Should the various open protocol software programs be able to function properly with the proposed 128-bit Host Identity Tag (HIT) cryptographic ID, then the performance of the solution can be tested.

The source and destination HIT Tag addresses and associated Local Scope Identifier (LSI) addresses were generated using the OpenHIP software module, called HITGEN, to create the cryptographic addresses illustrated in the XML output documents shown in Figures 6 and 7 (Gurtov, 2008). The two HIT Tags of both the source and destination computer nodes were then inserted into the 2048-bit OpenVPN Static Key (see Figure 8) for end-to-end authentication in order to create the communications tunnel. In addition the 32-bit LSI addresses that are displayed below the HIT Tags were derived from the 128-bit HIT Tags and were used as the IP

addresses for each of the computer nodes. LSIs are not globally unique and do not offer the strong authentication of HIT Tags, but they are useful on Local Area Networks such as this Dissertation Computer Test LAN (Gurtov, 2008).

```
<?xml version="1.0" encoding="UTF-8"?>
<my_host_identities>
  <host_identity alg="RSA" alg_id="5" length="128" anon="no"
incoming="yes" r1count="10">
    <name>INSIGNIA-D400A-1024</name>
    <N>
B285511522F5CDB4584AA0F33ABBF32F919060D41FF570F08C1203F8DB821F8182BC9D3
078D6ECE7AF4A30C12E64E3732EA78B8F0E3281992C6F640847F9D069D16A6A01FB41CB
4BDD14E2D7AD4ED759934EB7B7F8EAFB1830E9BAE788D883F7C7071FCAE5DD933EAF47A
A0E67341425DCE354DC556D9A847A05662D58522ECF</N>
    <E>010001</E>
    <D>
042C51A4E8D84E22B51DA97D8615F6AD59FDC205B3698D66521FE9AFDB91C322C7E798D
5153E10F3A98956726D9F3621EF29437DE89B0DA48301679939F58105AE3477C14941ED
DC3804FFD2B9EF8C16A99D3AD8D5B5A161133377F4660616449E3F0605458D4FADC6C83
BE306F038722051FBAD7D72415C6BDCF5249E7FFC41</D>
    <P>
E7D42B6E8105851CCE00333CD62EB8F55F6377C213B481A0A213F6EEDD7692B5CB8A318
10A7A1EA7DE8DFA4328C33937F65D80641EF8974C84D21F7387DB43E9</P>
    <Q>
C522484F2177E8EA9137782C425BEFE6ED15F36AC60DDA9A37D9EAC2DDC426473E637C9
C247B4FFAAA4A48FB90C7D9DE662B9866A5748B205BE793FF6369C1F7</Q>
    <dmp1>
E2BB3F3EF430D1DF3A1A380267F78A2D70FD7742F8C2B184C8FF7DA260367786156B32B
F61DFCDDBA06E7B34F3C8FB4D2046922B599F075A6F0C92760B890701</dmp1>
    <dmq1>
2306604A8EEBBB1A520AC4F33827158CB5FAFC70B017AE0B50790B58EC05F9B716C29E5
52FB62913A445E689ABC3965609591D8EBF3EE1A9322B07048D293129</dmq1>
    <iqmp>
881ACC077B05C11D04DF12057ED59760FE3C4682310CF1687676A5FDFEC2B8AA0B3E0C6
FA344DDC592574CC6641049E68F3D8EE8790CB5A28DC20FF64CE5758B</iqmp>
    <HIT>2001:10:1c13:6aa2:55bf:be70:1695:a15</HIT>
    <LSI>1.149.10.21</LSI>
  </host_identity>
</my_host_identities>
```

Figure 6 The source Host Identity Tag (HIT) address was generated using the OpenHIP software module, called HITGEN, to create the cryptographic address <HIT> illustrated in the fourth line from the bottom of the XML output document (Gurtov, 2008). In addition the 32-bit LSI address that is displayed below the HIT Tag was derived from the 128-bit HIT Tag and was used as the IP address for the computer node.

The destination Host Identity Tag (HIT) address was generated using the OpenHIP software module, called HITGEN, to create the cryptographic address <HIT> and associated

<LSI> IP address as illustrated in the XML output document in Figure 7:

```
<?xml version="1.0" encoding="UTF-8"?>
<my_host_identities>
  <host_identity alg="RSA" alg_id="5" length="128" anon="no"
incoming="yes" r1count="10">
    <name>TOSHIBAA75S229-1024</name>
    <N>
ABDB9F24145F62F39AB2BC50FD80E19EBA4C449CCD9B9D4F1368D56F527AAC011D24CA6
D5600F40EF7F86E08A2F2F53100871529249 6D858A6AA7063F8D43D398551CADAF17252
42E8D3C1A522F56489D479E4355E75749C7C18FA2D489C27E4EFA531A3FA3A73A48B9F5
36D5C3A42F23D583D354622CC22FEE83D756DEAF8F3</N>
    <E>010001</E>
    <D>
6D41722ABDC8E65F7839A8FED42ADB44CDAB2EF380C32D7ED8180D98781704C2B084732
C8F408BE7F83D37B6C5E12F7FB796291DBEE5272534CFA623E84D059E412F695160C698
F9D69E5F3B906765C9390950ACE7FFC358116A467F15E7343B5C83E23D3FE0E4252CB39
C6D74BEB361EAFE8C8C99679A4FD77E6080D70E1201</D>
    <P>
D35F665405CDECF2BD8FC3C04F4ADDDDA29EEF0FFDEFDA3A65D01AE87287A6978CEEFAC
9E51CE75B12F94DEA96D7EC78AB53160D6532FF33B7FFD5E887341EC5</P>
    <Q>
D0247765D381D4345984F05D42586620CC49016585364B96A76051D757389EF9EA62762
820BA3AD94B4603BC14277F9095C0C7EC441B21ED65E2F2C3386AB457</Q>
    <dmp1>
92B40DE3973BBB8F267E1790EBF7BC514DC31D8D6DE40104B31162FB9E32042FFF43069
10611AF89887BBBD66BE7655AC31E219A1E78ECA34ABBA80D81796D39</dmp1>
    <dmq1>
0C4C1B41C8DD42CC54FDA5B5DCD59C3313DEBC566328720ED494BC411CC61B9E685AA8E
0760E8AAB8BE6F711859F4FCA1B0EE8C0ECC52D9BF9090F8EB92694C1</dmq1>
    <iqmp>
6CED3EF003F2FD12C9A94B6C018E54C11C893DFB681CAABC81E27018076ACE8191EF6D4
A72273AD18316B81E917D709195F7012203847B14099A1AF16237C83F</iqmp>
    <HIT>2001:15:1a10:7537:7107:8783:1416:bc28</HIT>
    <LSI>1.149.10.22</LSI>
  </host_identity>
</my_host_identities>
```

Figure 7 The destination Host Identity Tag (HIT) address was generated using the OpenHIP software module, called HITGEN, to create the cryptographic address <HIT> illustrated in the fourth line from the bottom of the XML output document (Gurtov,2008). In addition the 32-bit LSI address that is displayed below the HIT Tag was derived from the 128-bit HIT Tag and was used as the IP address for the computer node.

After both the source and destination HIT Tags were created, they were inserted into the OpenVPN Static Key for end-to-end authentication of the source and destination computer nodes

as shown below in Figure 8:

```
#
# 2048 bit OpenVPN static key
#
-----BEGIN OpenVPN Static key V1-----
2001101c136aa255bfbe701695a15000
2001101c136aa255bfbe701695a15000
2001101c136aa255bfbe701695a15000
2001101c136aa255bfbe701695a15000
2001101c136aa255bfbe701695a15000
2001101c136aa255bfbe701695a15000
2001101c136aa255bfbe701695a15000
2001101c136aa255bfbe701695a15000
2001151a107537710787831416bc2800
2001151a107537710787831416bc2800
2001151a107537710787831416bc2800
2001151a107537710787831416bc2800
2001151a107537710787831416bc2800
2001151a107537710787831416bc2800
2001151a107537710787831416bc2800
2001151a107537710787831416bc2800
-----END OpenVPN Static key V1-----
```

Figure 8 View of the OpenVPN Static Key for end-to-end authentication that contains the Host Identity Tag (HIT) of both the source and destination computer nodes.

The OpenVPN and OpenSSL software programs performed normally after the Static Key (shown in Figure 8) that contains the Host Identity Tags (HITs) of both source and destination computer nodes enabled end-to-end authentication and creation of the OpenVPN tunnel. The communications tunnel then successfully supported the testing and data transmission that answered the remaining Research Questions two and three. Figure 9 illustrates successful FTP file transfers at the packet level during lab tests using the Wireshark network sniffer to validate the source and destination computer nodes on the test lab network. Thus the first part of Research Question #1 of incorporating HIT Tags into the OpenVPN/OpenSSL software protocol was successfully accomplished.

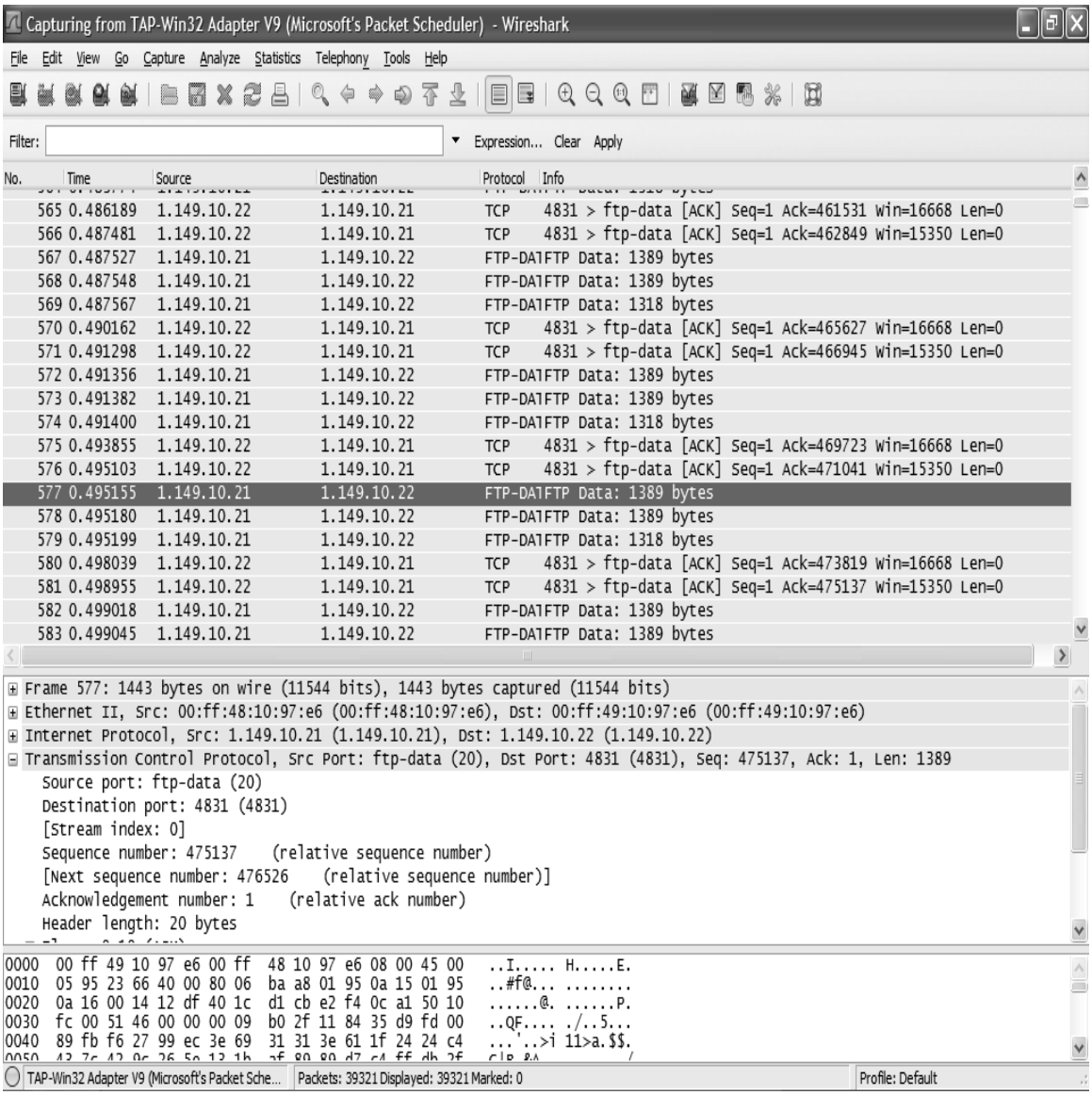


Figure 9 View of FTP file transfers at the packet level during lab testing with the Wireshark network sniffer to validate the source and destination computer nodes on the test lab network. Data was collected on the FTP server before OpenVPN tunnel encryption.

In addition the 32-bit LSI addresses that were derived from the HIT Tags were used as the IP addresses for each of the computer nodes (source IP address: 1.149.10.21 and destination

IP address: 1.149.10.22) as shown in the highlighted line in Figure 9 above from the Wireshark network sniffer on the computer test LAN.

Relationship between Latency and Payload Size

The second part of Research Question #1 is: whether the payload size adds significant overhead and latency to TLS file transfers for real-time control communications? The speed of FTP file transfers (latency) represents the Dependent Variable (DV) to be tested and the Independent Variable (IV) is the size of the data payload inside the FTP file transfer within the OpenVPN network test link. How does the size of the payload impact file transfer performance? Can the file transfer speed (latency) be predicted using an independent variable representing the payload-size as illustrated in Equation 3?

$$\hat{Y} = C + \beta_1 payloadsize + \varepsilon \quad (3)$$

Equation 3 Predictive equation that describes the relationship between latency, the dependent variable, and payloadsize, the independent variable.

Descriptive Statistics to Validate Data Files

The following statistical views validate the experimental lab data in order to match the requirements of the statistical methods to be used. The first two Research Questions #1 and #2 were analyzed together using Multivariate ANOVA in a separate analysis from Research Question #3. Figure 10 provides a summary of the experimental data collected on file transfer Latency using five different sizes of payloads: approximately 10MB (megabytes), 20MB,

30MB, 40MB and 50MB. The top of the chart displays latency without TLS-auth and the bottom half describes the latency data with TLS-auth (i.e. HIT Tags) enabled.

Case Processing Summary

	Payload Size	Cases					
		Valid		Missing		Total	
		N	Percent	N	Percent	N	Percent
Latency of File Xfr w/OpenVPN Tunnel	10.69	10	100.0%	0	.0%	10	100.0%
	21.65	10	100.0%	0	.0%	10	100.0%
	31.06	10	100.0%	0	.0%	10	100.0%
	41.40	10	100.0%	0	.0%	10	100.0%
	51.74	10	100.0%	0	.0%	10	100.0%
Latency w/TLSauth	10.69	10	100.0%	0	.0%	10	100.0%
	21.65	10	100.0%	0	.0%	10	100.0%
	31.06	10	100.0%	0	.0%	10	100.0%
	41.40	10	100.0%	0	.0%	10	100.0%
	51.74	10	100.0%	0	.0%	10	100.0%

Figure 10 Summary of the Experimental Data collected on file transfer Latency using five different sizes of payloads: approximately 10MB (megabytes), 20MB, 30MB, 40MB and 50MB. The top of the chart displays latency without TLS-auth and the bottom half describes the latency data with TLS-auth (i.e. HIT Tags) enabled.

The multivariate ANOVA (MANOVA) method requires linearity, normality, homogeneity of variances and independence across the data (Norusis, 2008). Linearity of the data is confirmed by observing the scatterplot of the residuals in Figure 16. In addition, the high degree of fit based on the R-squared values of 0.999 means that the data points fall very close to the fitted lines as shown in Figure 17.

The following test of normality indicates that the data appears normal for some payload sizes and is somewhat skewed for other sizes. As a result the overall histogram of the data is plotted in Figure 12 in order to detect any specific patterns to the data.

Tests of Normality

	Payload Size	Kolmogorov-Smirnov[a]			Shapiro-Wilk		
		Statistic	df	Sig.	Statistic	df	Sig.
Latency of File Xfr w/OpenVPN Tunnel	10.69	.209	10	.200*	.904	10	.241
	21.65	.154	10	.200*	.954	10	.712
	31.06	.430	10	.000	.529	10	.000
	41.40	.438	10	.000	.575	10	.000
	51.74	.400	10	.000	.597	10	.000
Latency w/TLSauth	10.69	.171	10	.200*	.956	10	.737
	21.65	.318	10	.005	.700	10	.001
	31.06	.174	10	.200*	.931	10	.462
	41.40	.334	10	.002	.747	10	.003
	51.74	.344	10	.001	.655	10	.000

a. Lilliefors Significance Correction

*. This is a lower bound of the true significance.

Figure 11 Summary of analysis of the experimental data for normality.

The overall histogram below in Figure 12 indicates that the data points for latency are highly concentrated about each of the payloads and have very low, homogeneous variance. The consistency of the test data is a valuable result and explains one reason why the predictive equation fits the data so well.

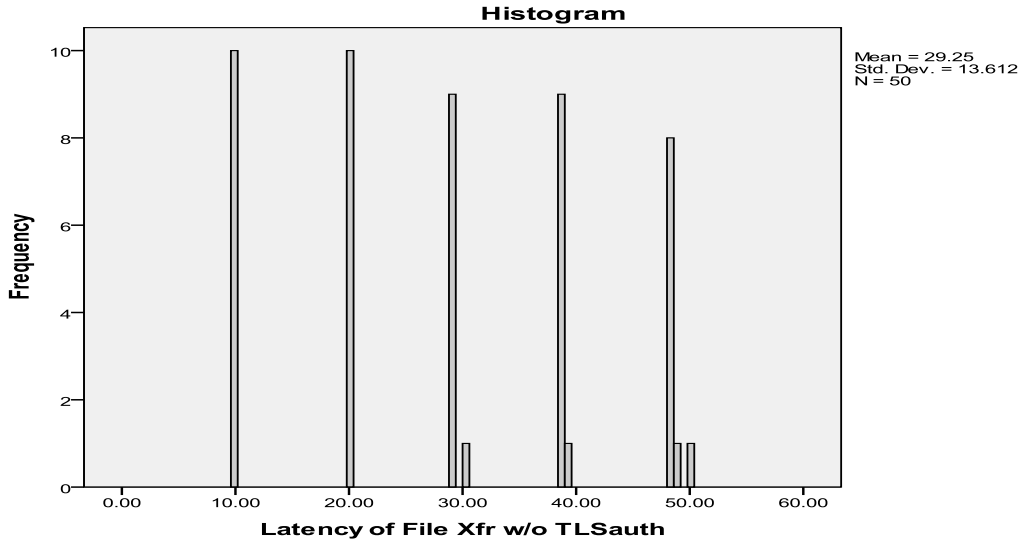
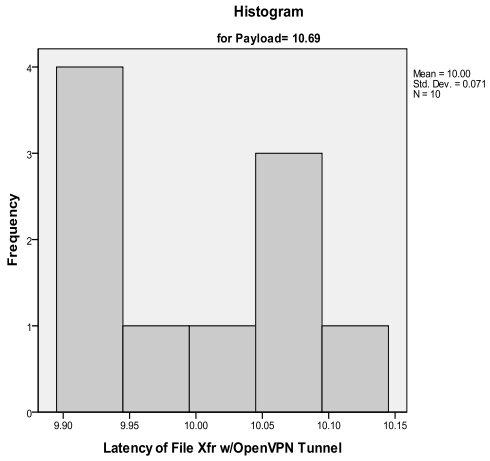


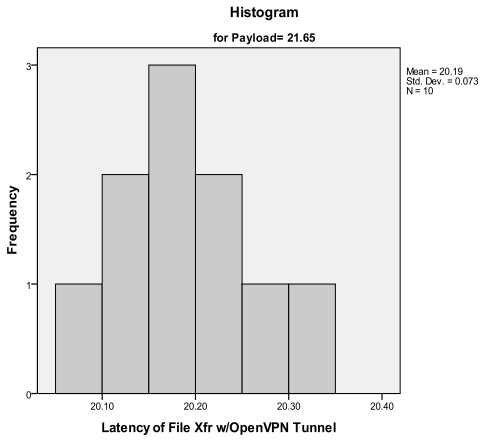
Figure 12 Overall Histogram of Latency vs. Payload Size for observation of normality.

Following are the detailed histograms for each of the payloads to understand why some distributions appeared non-normal in the K-S and S-W tests in Figure 11.

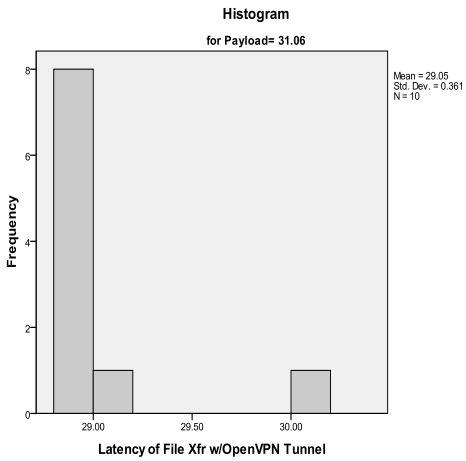
Figure 13 Individual histograms for each payload size beginning with 10MB:



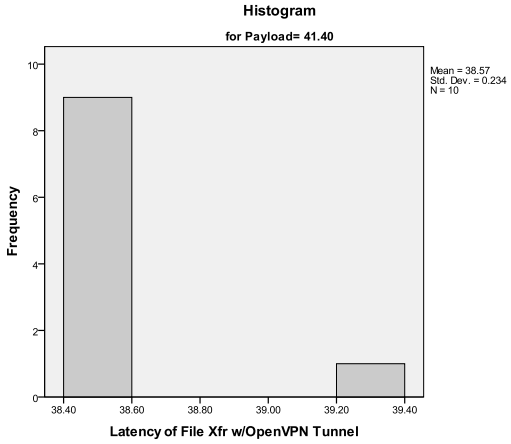
For 20MB:



For 30MB:



For 40MB:



For 50MB

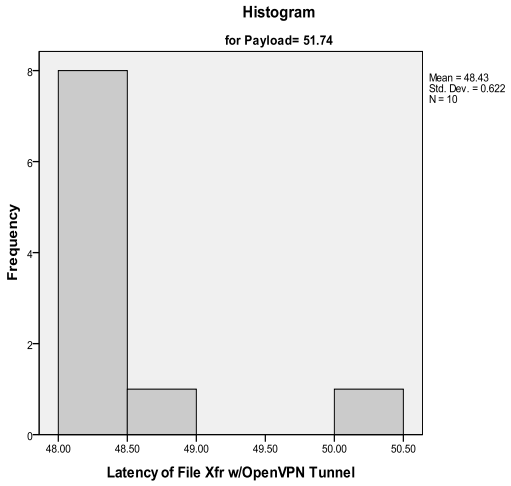


Figure 13 reveals that on close examination some of the data distributions (30, 40, and 50MB) are somewhat skewed even though the overall data histogram in Figure 12 indicates little concern. Fortunately the Multivariate Analysis of Variance (MANOVA) method is not particularly sensitive to the normalcy of data, so it represents a good alternative test to multiple linear regression (MLR) analysis that does require normally distributed variables (Norusis, 2008). In summary, the residuals are plotted later in Figure 16 for further observation of any unusual patterns and to confirm the validation of the data. Likewise the residuals are also plotted as a means to observe independence of the data elements.

Multivariate Analysis of Variance (MANOVA)

The MANOVA method provides the capability to simultaneously calculate the regression characteristics and parameter estimates for the latency of communications both with and without the TLS-auth (and associated HIT Tags) as shown in Figure 14. Thus the technique provides an accurate side-by-side comparison of the predictive equations for latency both with and without the HIT Tags as part of the file transfer process.

The first Research Question concerns, "How does the size of the payload impact file transfer performance? Can the file transfer speed (latency) be predicted using an independent variable representing the payload-size?". The parameter estimates from the MANOVA method in Figure 15 show that the payload sizes are highly significant but that the intercepts (C) are not.

$$\hat{Y}_{w/o\text{-}TLS} = (0.935)payloadsize + \varepsilon \quad (4)$$

Equation 4 Predictive equation for latency of file transfers using an OpenVPN tunnel without HIT Tag authentication.

The predictive equation (Equation 4) for latency of file transfer using an OpenVPN tunnel without HIT Tag authentication provides an R-Squared value of 0.999 meaning that 99.9% of the observed variation in the latency data is explained per Figure 14 below. Thus we can reject the null hypothesis that there is no statistically significant relationship between the payload-size and the file transfer speed (latency).

Tests of Between-Subjects Effects

Source	Dependent Variable	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared	Noncent. Parameter	Observed Power[b]
Corrected Model	Latency of File Xfr w/OpenVPN Tunnel	9073.388[a]	1	9073.388	79889.859	.000	.999	79889.859	1.000
	Latency w/TLSauth	9340.779[c]	1	9340.779	185769.445	.000	1.000	185769.445	1.000
Intercept	Latency of File Xfr w/OpenVPN Tunnel	.007	1	.007	.058	.810	.001	.058	.056
	Latency w/TLSauth	.071	1	.071	1.403	.242	.028	1.403	.213
Payload	Latency of File Xfr w/OpenVPN Tunnel	9073.388	1	9073.388	79889.859	.000	.999	79889.859	1.000
	Latency w/TLSauth	9340.779	1	9340.779	185769.445	.000	1.000	185769.445	1.000
Error	Latency of File Xfr w/OpenVPN Tunnel	5.452	48	.114					
	Latency w/TLSauth	2.414	48	.050					
Total	Latency of File Xfr w/OpenVPN Tunnel	51848.775	50						
	Latency w/TLSauth	53723.739	50						
Corrected Total	Latency of File Xfr w/OpenVPN Tunnel	9078.839	49						
	Latency w/TLSauth	9343.192	49						

a. R Squared = .999 (Adjusted R Squared = .999)

b. Computed using alpha = .05

c. R Squared = 1.000 (Adjusted R Squared = 1.000)

Figure 14 Summary of the Tests of Between-Subject Effects on file transfer Latency using five different sizes of payloads: approx. 10MB, 20MB, 30MB, 40MB and 50MB.

Parameter Estimates

Dependent Variable	Parameter	B	Std. Error	t	Sig.	95% Confidence Interval		Partial Eta Squared	Noncent. Parameter	Observed Power[a]
						Lower Bound	Upper Bound			
Latency of File Xfr w/OpenVPN Tunnel	Intercept	-.028	.114	-.241	.810	-.257	.202	.001	.241	.056
	Payload	.935	.003	282.648	.000	.928	.942	.999	282.648	1.000
Latency w/TLSauth	Intercept	.090	.076	1.185	.242	-.063	.242	.028	1.185	.213
	Payload	.949	.002	431.010	.000	.944	.953	1.000	431.010	1.000

a. Computed using alpha = .05

Figure 15 Parameter Estimates for the Predictive Equation on file transfer Latency using five different sizes of payloads: approx. 10MB, 20MB, 30MB, 40MB and 50MB.

Review of Residuals

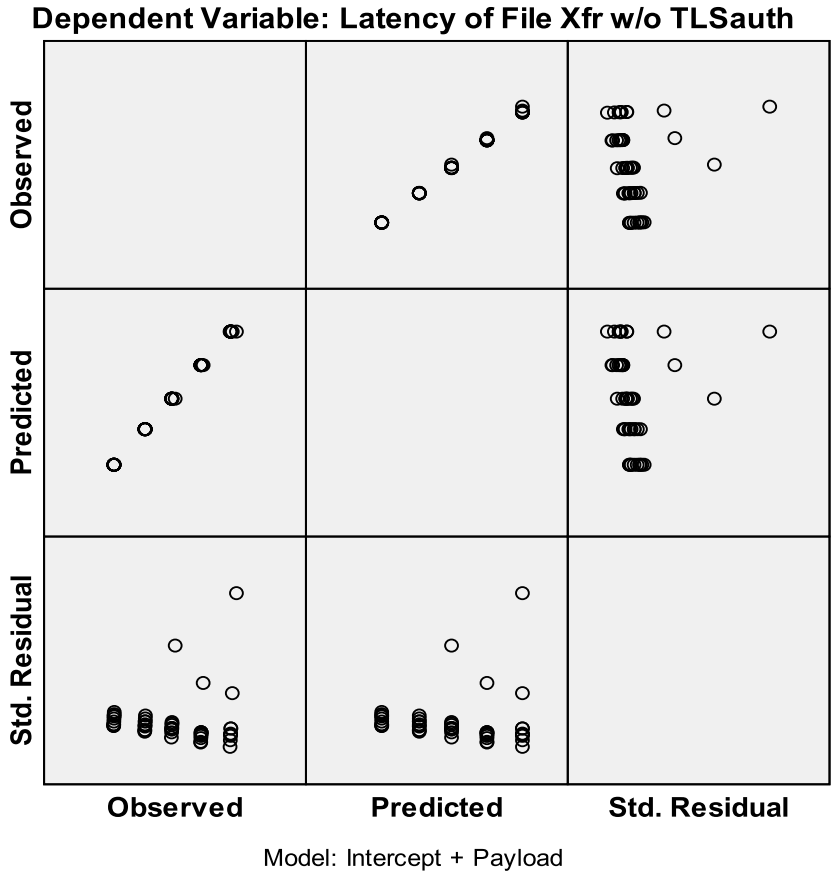


Figure 16 Residual Plots for the MANOVA method for the Latency of communications without the TLS-auth (and associated HIT Tag transmission).

The tight cluster of residuals in the plots in Figure 16 indicate that the values of the residuals are very small (except for a few outliers) due to the fact that the predictive equation has such a high degree of correlation with the observed data (the R-Squared value is 99.9% for both models). One also has to observe the very small variation in the residuals for each of the five payload sizes, not the overall plot, in order to see the true variation around the mean latency about each individual payload. Thus the data is very consistent in measuring the latency of file transfers without the TLS-auth (and associated HIT Tags).

Second Research Question

The second Research Question concerns the existence of the Host Identity (HIT) cryptographic ID tag within the file transfer packets. Specifically, "Does the presence of the crypto-ID add significant overhead and latency to SSL/TLS file transfers?".

As described earlier, the multivariate ANOVA (MANOVA) method provides the capability to simultaneously calculate the regression characteristics and parameter estimates for the latency of communications both with and without the TLS-auth (and associated HIT Tags). The parameter estimates from the multivariate ANOVA method in Figure 15 indicates that the payload size is highly significant but that the intercept (C) is not. The predictive equation for latency of file transfers using an OpenVPN tunnel with HIT Tag authentication is described as follows:

$$\hat{Y}_{w/TLS} = (0.949)\text{payloadsize} + \acute{\varepsilon} \quad (5)$$

Equation 5 Predictive equation for Latency of file transfers using an OpenVPN tunnel with HIT Tag authentication.

The predictive equation (Equation #5) for latency of file transfer using an OpenVPN tunnel with HIT Tag authentication provides an R-Squared value of 0.999+ meaning that 99.9+% of the observed variation in the latency data is explained. Thus we can reject the null hypothesis that there is no statistically significant relationship between the existence of the HIT Tag within the file transfer packets and the file transfer speed (latency).

The simplicity of Equation 4 and Equation 5 provides a valuable insight for further investigation and discussion in Chapter 5 on Conclusions. The ratio of the coefficients (1.0149732 or approximately 1.5%), means that the additional latency for transmission of HIT Tags is roughly 1.5% without considering the effects of encryption. In other words, meeting the NIST requirements for authentication and Denial of Service resistance using HIT Tags adds approximately 1.5% additional latency to FTP file transfers. Just as importantly, the amount of additional latency for using HIT Tags diminishes toward zero as the payload size becomes smaller and tends toward zero. Unfortunately the accuracy of the FTP file transfer timers in the laboratory test software did not permit testing transfer times below the 10MB payload size. Presumably more accurate computer systems and software can permit analysis of much smaller file transfer payloads and associated latencies.

Figure 17 provides a plot of both predictive equations #4 and 5. The lighter gray line on the left illustrates the latency with the HIT Tags and the dark line on the right shows the latency without. Both lines reflect a high R-Squared value of at least 0.999 per the multivariate ANOVA analysis (see Figure 14) and mean that the equations explain over 99.9% of the variation in the data collected in the experiments. Thus the difference between the two fitted lines displayed in Figure 17 represents the small, but statistically significant, amount of additional latency used to carry the HIT Tags for authentication.

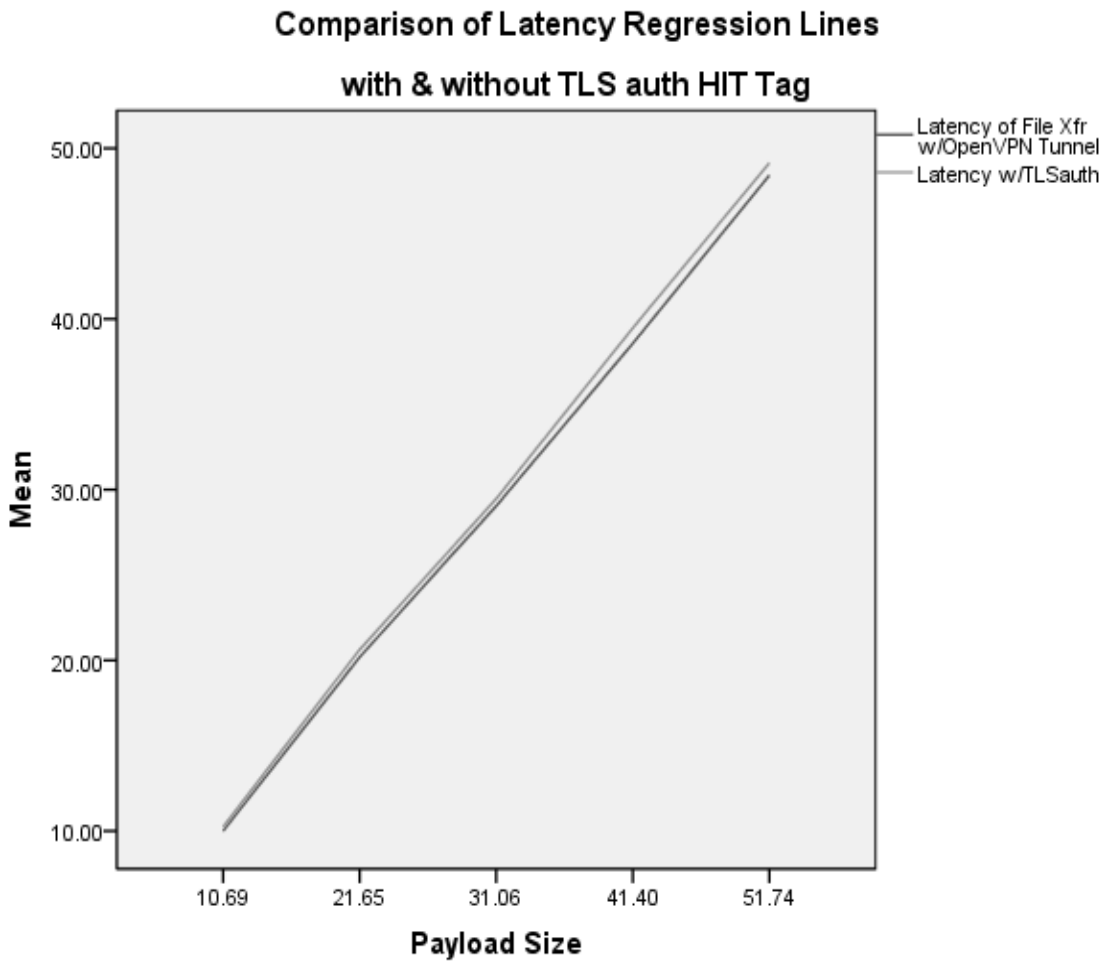


Figure 17 Fitted lines from the predictive equations on file transfer latency both with and without the use of HIT Tags (and without encryption) for five different sizes of payloads: approximately 10MB (megabytes), 20MB, 30MB, 40MB and 50MB. The lighter gray line on the left illustrates the latency with the HIT Tags and the dark line on the right shows the latency without HIT Tags.

The chart of parameter estimates in Figure 15 also shows the 95% confidence interval for **β** coefficients both with and without the TLS-auth (HIT Tag) capability. The range of **β** values (0.928 to 0.942 and 0.944 to 0.953) do not overlap, confirming that at least to a level of 95% confidence, the difference in latency both with and without TLS-auth (and associated HIT Tag) is significant, though very small.

Review of Residuals

The cluster of residuals in the plots in Figure 18 for the latency of file transfers with TLS-auth (and associated HIT Tags) indicate that the values of the residuals are not quite as small as previously shown in Figure 16, however the residuals are still very small (except for a few outliers) due to the fact that the predictive equation has such a high degree of correlation with the observed data (the R-Squared value is 99.9% for both models).

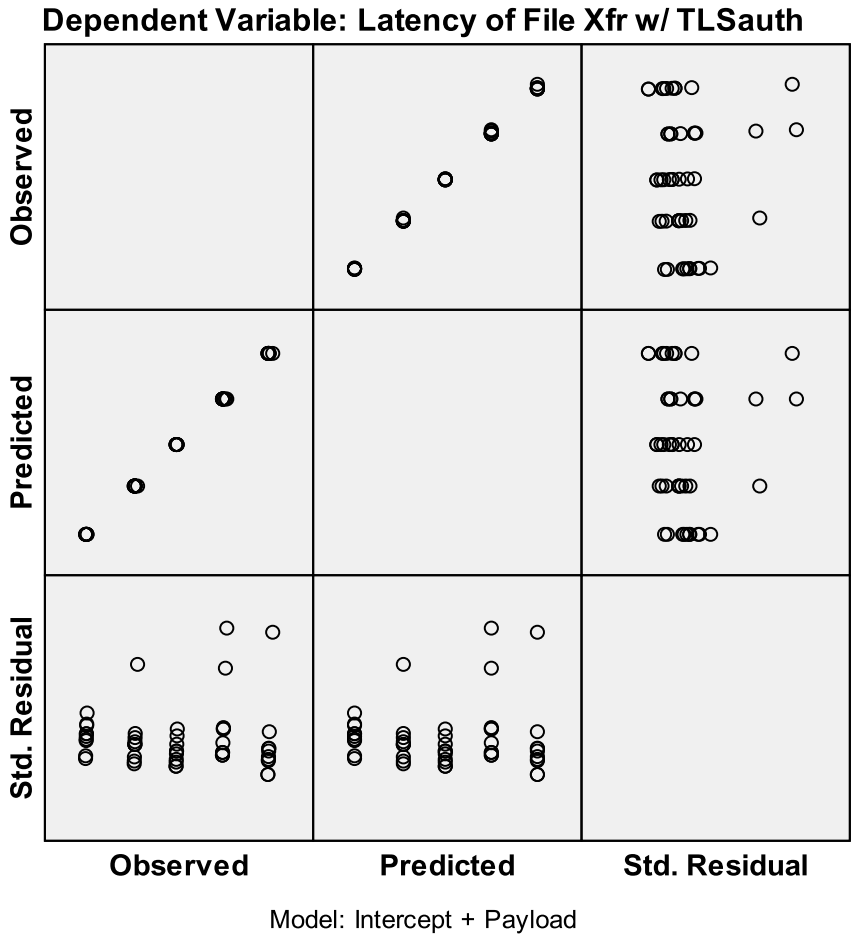


Figure 18 Residual Plots for the MANOVA analysis of the latency of file transfers with the TLS-auth (and associated HIT Tag transmission).

One also has to observe the variation for each of the five payload sizes, not just the overall plot in order to see the true variation around the mean latency about each individual payload. The data is very consistent in measuring the latency with TLS-auth.

Third Research Question

The third Research Question concerns the type of encryption used (AES, Blowfish) to protect the file transfer and the length of the associated keys (128, 256 bits). For an accurate side-by-side comparison of the results it was valuable to extend the previous MANOVA analysis for Research Questions #1 and #2 to include the additional test data on encryption type and key lengths.

Research Question #3 concerns, "Do different types of encryption affect the speed of file transfers? ". Likewise Research Question #3 also investigates, "Do different lengths of encryption keys affect the speed of file transfers?". Both encryption type and key length can be studied simultaneously in Multivariate ANOVA. As described earlier, MANOVA is an effective way to calculate and examine the effects of factor(s) on several variables at once using a general linear model in which the factors divide the cases into groups (Norusis, 2008). In addition, it enables the measurement of any significant interaction effects, such as encryption type and key length.

Just as the MANOVA method calculated the regression and parameter estimates for the latency of communications both with and without the TLS-auth (and associated HIT Tags) as shown earlier in Figure 14 and Figure 15, the same method was extended to analyze the impact of data encryption and encryption key length on latency as illustrated in Figures #19 and 20.

Tests of Between-Subjects Effects

Source	Dependent Variable	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared	Noncent. Parameter	Observed Power[b]
Corrected Model	Latency of File Xfr w/o TLSauth	9073.388[a]	1	9073.388	79889.859	.000	.999	79889.859	1.000
	Latency of File Xfr w/ TLSauth	9340.779[c]	1	9340.779	185769.445	.000	1.000	185769.445	1.000
	Latency of File Xfr w/ TLSauth & AES128	10596.661[d]	1	10596.661	322953.674	.000	1.000	322953.674	1.000
	Latency of File Xfr w/ TLSauth & AES256	10713.649[e]	1	10713.649	73572.277	.000	.999	73572.277	1.000
	Latency of File Xfr w/ TLSauth & BF128	9564.913[f]	1	9564.913	121281.645	.000	1.000	121281.645	1.000
	Latency of File Xfr w/ TLSauth & BF256	9576.988[g]	1	9576.988	84238.259	.000	.999	84238.259	1.000
Intercept	Latency of File Xfr w/o TLSauth	.007	1	.007	.058	.810	.001	.058	.056
	Latency of File Xfr w/ TLSauth	.071	1	.071	1.403	.242	.028	1.403	.213
	Latency of File Xfr w/ TLSauth & AES128	.161	1	.161	4.897	.032	.093	4.897	.583
	Latency of File Xfr w/ TLSauth & AES256	.014	1	.014	.096	.758	.002	.096	.061
	Latency of File Xfr w/ TLSauth & BF128	.223	1	.223	2.822	.099	.056	2.822	.377
	Latency of File Xfr w/ TLSauth & BF256	.197	1	.197	1.730	.195	.035	1.730	.252
Payload	Latency of File Xfr w/o TLSauth	9073.388	1	9073.388	79889.859	.000	.999	79889.859	1.000
	Latency of File Xfr w/ TLSauth	9340.779	1	9340.779	185769.445	.000	1.000	185769.445	1.000
	Latency of File Xfr w/ TLSauth & AES128	10596.661	1	10596.661	322953.674	.000	1.000	322953.674	1.000
	Latency of File Xfr w/ TLSauth & AES256	10713.649	1	10713.649	73572.277	.000	.999	73572.277	1.000
	Latency of File Xfr w/ TLSauth & BF128	9564.913	1	9564.913	121281.645	.000	1.000	121281.645	1.000
	Latency of File Xfr w/ TLSauth & BF256	9576.988	1	9576.988	84238.259	.000	.999	84238.259	1.000
Error	Latency of File Xfr w/o TLSauth	5.452	48	.114					
	Latency of File Xfr w/ TLSauth	2.414	48	.050					
	Latency of File Xfr w/ TLSauth & AES128	1.575	48	.033					
	Latency of File Xfr w/ TLSauth & AES256	6.990	48	.146					
	Latency of File Xfr w/ TLSauth & BF128	3.786	48	.079					
	Latency of File Xfr w/ TLSauth & BF256	5.457	48	.114					
Total	Latency of File Xfr w/o TLSauth	51848.775	50						
	Latency of File Xfr w/ TLSauth	53723.739	50						
	Latency of File Xfr w/ TLSauth & AES128	61072.506	50						
	Latency of File Xfr w/ TLSauth & AES256	61444.949	50						
	Latency of File Xfr w/ TLSauth & BF128	55221.448	50						
	Latency of File Xfr w/ TLSauth & BF256	55263.603	50						
Corrected Total	Latency of File Xfr w/o TLSauth	9078.839	49						
	Latency of File Xfr w/ TLSauth	9343.192	49						
	Latency of File Xfr w/ TLSauth & AES128	10598.236	49						
	Latency of File Xfr w/ TLSauth & AES256	10720.639	49						
	Latency of File Xfr w/ TLSauth & BF128	9568.698	49						
	Latency of File Xfr w/ TLSauth & BF256	9582.445	49						

a. R Squared = .999 (Adjusted R Squared = .999)

b. Computed using alpha = .05

c. R Squared = 1.000 (Adjusted R Squared = 1.000)

d. R Squared = 1.000 (Adjusted R Squared = 1.000)

e. R Squared = .999 (Adjusted R Squared = .999)

f. R Squared = 1.000 (Adjusted R Squared = 1.000)

g. R Squared = .999 (Adjusted R Squared = .999)

Figure 19 Test of Between-Subject Effects for the multivariate ANOVA analysis of the latency of file transfers including TLS-auth (HIT Tags), encryption type and key length.

Parameter Estimates

Dependent Variable	Parameter	B	Std. Error	t	Sig.	95% Confidence Interval		Partial Eta Squared	Noncent. Parameter	Observed Power[a]
						Lower Bound	Upper Bound			
Latency of File Xfr w/o TLSauth	Intercept	-.028	.114	-.241	.810	-.257	.202	.001	.241	.056
	Payload	.935	.003	282.648	.000	.928	.942	.999	282.648	1.000
Latency of File Xfr w/ TLSauth	Intercept	.090	.076	1.185	.242	-.063	.242	.028	1.185	.213
	Payload	.949	.002	431.010	.000	.944	.953	1.000	431.010	1.000
Latency of File Xfr w/ TLSauth & AES128	Intercept	.136	.061	2.213	.032	.012	.259	.093	2.213	.583
	Payload	1.011	.002	568.290	.000	1.007	1.014	1.000	568.290	1.000
Latency of File Xfr w/ TLSauth & AES256	Intercept	.040	.129	.310	.758	-.220	.300	.002	.310	.061
	Payload	1.016	.004	271.242	.000	1.009	1.024	.999	271.242	1.000
Latency of File Xfr w/ TLSauth & BF128	Intercept	.160	.095	1.680	.099	-.031	.351	.056	1.680	.377
	Payload	.960	.003	348.255	.000	.955	.966	1.000	348.255	1.000
Latency of File Xfr w/ TLSauth & BF256	Intercept	.150	.114	1.315	.195	-.079	.379	.035	1.315	.252
	Payload	.961	.003	290.238	.000	.954	.967	.999	290.238	1.000

a. Computed using alpha = .05

Figure 20 Parameter Estimates for the multivariate ANOVA analysis of the latency of file transfers including TLS-auth (HIT Tags), encryption type and key length.

The following predictive equations are repeated from the previous analysis for Research Questions #1 and #2 that determined the additional latency for HIT Tag authentication:

$$\hat{Y}_{w/o\text{-}TLS} = (0.935)payloadsize + \acute{\varepsilon} \quad (4)$$

Equation 4 Predictive equation for latency of file transfers using an OpenVPN tunnel without HIT Tag authentication.

and

$$\hat{Y}_{w/TLS} = (0.949)payloadsize + \acute{\varepsilon} \quad (5)$$

Equation 5 Predictive equation for Latency of file transfers using an OpenVPN tunnel with HIT Tag authentication.

The parameter estimates from the MANOVA analysis in Figure 20 indicate that the payload sizes are highly significant but that the intercepts (C) are not (except for the single case of AES encryption with 128-bit key length where C = 0.136). The MANOVA analysis confirms the previous two equations (#4 and #5) again and also provides the predictive equations for AES and Blowfish encryption using both 128 and 256 key lengths as follows:

$$\hat{Y}_{\text{w/TLS \& AES-128}} = 0.136 + (1.011)\ \text{payloadsize} + \acute{\varepsilon} \quad (6)$$

Equation 6 Predictive equation that describes the relationship between latency and the AES type of encryption with 128-bit key length.

$$\hat{Y}_{\text{w/TLS \& AES-256}} = (1.016)\ \text{payloadsize} + \acute{\varepsilon} \quad (7)$$

Equation 7 Predictive equation that describes the relationship between latency and the AES type of encryption with 256-bit key length.

$$\hat{Y}_{\text{w/TLS \& BF-128}} = (0.960)\ \text{payloadsize} + \acute{\varepsilon} \quad (8)$$

Equation 8 Predictive equation that describes the relationship between latency and the Blowfish type of encryption with 128-bit key length.

$$\hat{Y}_{\text{w/TLS \& BF-256}} = (0.961)\ \text{payloadsize} + \acute{\varepsilon} \quad (9)$$

Equation 9 Predictive equation that describes the relationship between latency and the Blowfish type of encryption with 256-bit key length.

All the predictive equations shown above (Equations #6, 7, 8 and 9) for latency of file transfer using an OpenVPN tunnel with HIT Tag authentication provide an R-Squared value of at

least 0.999 meaning that 99.9% of the observed variation in the latency data is explained. Thus we can reject the null hypotheses that there is no statistically significant relationship between the existence of the HIT Tag within the file transfer packets and the file transfer speed (latency) regardless of the type of encryption or key length employed.

The simplicity of Equations #6, 7, 8, and 9 provides a valuable insight for further investigation and discussion in Chapter 5 on Conclusions. Just as in the earlier analysis of Equations #3 and 4, the ratio of the coefficients is a measure of the additional latency for file transfers as shown below in Figure 21:

File Transfer w/OpenVPN	Approx. Addl Latency vs. FTP only w/o HIT Tag
FTP file transfer with HIT Tag only	1.5%
FTP file transfer with HIT Tag and AES w/128-bit key (plus intercept constant, C = 0.136)	8.1%
FTP file transfer with HIT Tag and AES w/256-bit key	8.7%
FTP file transfer with HIT Tag and Blowfish w/128-bit key	2.7%
FTP file transfer with HIT Tag and Blowfish w/256-bit key	2.8%

Figure 21 Additional Latency predicted by the multivariate ANOVA analysis for FTP file transfers that include TLS-auth (HIT Tags) and various encryption types and lengths.

Figure 22 provides a plot of both predictive equations for the case of AES encryption. The dark lines on the left illustrate the latency with the HIT Tags and AES encryption (the fitted lines with AES 128 and 256-bit key lengths fall on top of one another and are indistinguishable) versus the lighter gray line on the right that shows the latency with HIT Tags but without encryption. All fitted lines reflect a high R-Squared value of at least 0.999 per the multivariate

ANOVA analysis (Figure 19) and mean that the equations explain over 99.9% of the variation in the data collected in the experiments. Thus the difference between the three fitted lines displayed in Figure 22 represents the small, but statistically significant amount of additional latency used to carry the HIT Tags for authentication using AES encryption.

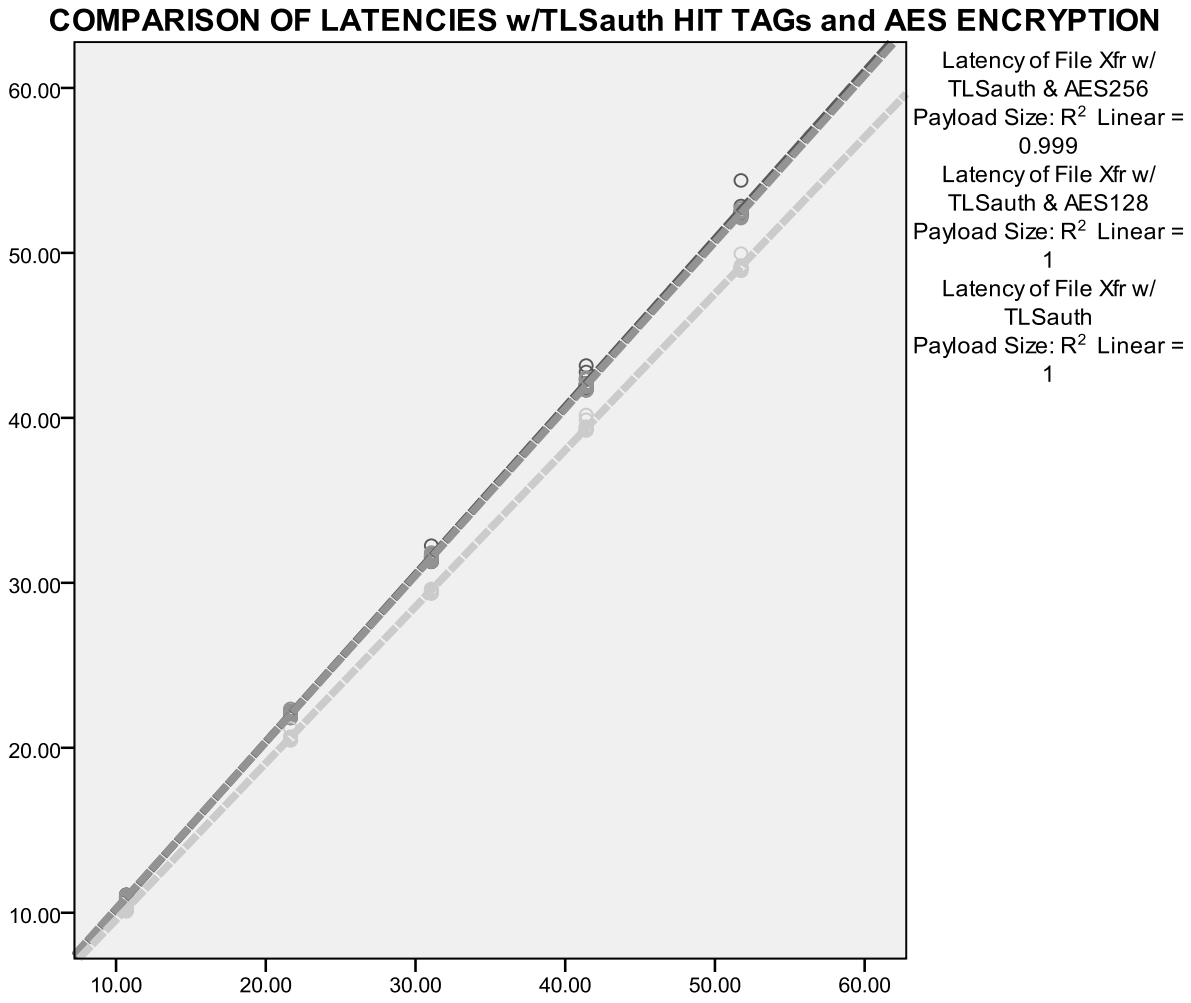


Figure 22 Fitted lines from the predictive equations on file transfer latency for five different sizes of payloads: approximately 10MB (megabytes), 20MB, 30MB, 40MB and 50MB. The dark lines on the left illustrate the latency with the HIT Tags and AES encryption (the fitted lines with AES 128 and 256-bit key lengths fall on top of one another and are indistinguishable) versus the lighter gray line on the right that shows the latency with HIT Tags but without encryption.

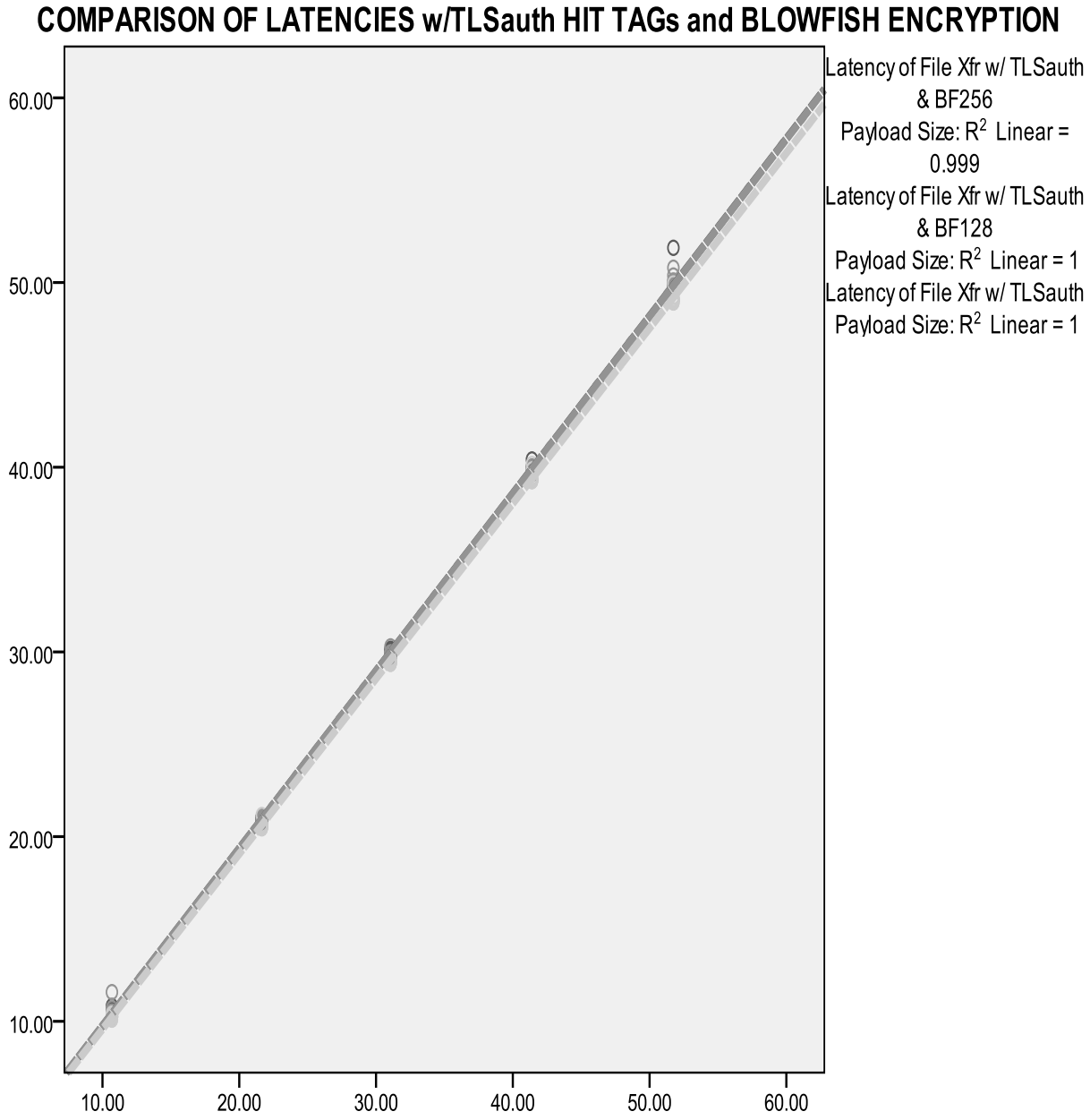


Figure 23 Fitted lines from the predictive equations on file transfer latency for five different sizes of payloads: approximately 10MB (megabytes), 20MB, 30MB, 40MB and 50MB. The dark lines on the left illustrate the latency with the HIT Tags and Blowfish encryption (the fitted lines with Blowfish 128 and 256-bit key lengths fall on top of one another and are indistinguishable) versus the lighter gray line on the right that shows the latency with HIT Tags but without encryption.

Figure 23 provides a plot of both predictive equations for the case of Blowfish encryption. The dark lines on the left illustrate the latency with the HIT Tags and Blowfish encryption (the fitted lines with Blowfish 128 and 256-bit key lengths fall on top of one another and are indistinguishable) versus the lighter gray line on the right that shows the latency with HIT Tags but without encryption. All fitted lines reflect a high R-Squared value of at least 0.999 per the multivariate ANOVA analysis (Figure 19) and mean that the equations explain over 99.9% of the variation in the data collected in the experiments. Thus the difference between the three fitted lines displayed in Figure 23 represents the small, but statistically significant, amount of additional latency used to carry the HIT Tags for authentication using Blowfish encryption.

Review of Residuals for AES and Blowfish Encryption

As described in the earlier cases, the tight cluster of residuals in the following plots in Figures 24, 25, 26 and 27 indicate that the values of the residuals are very small (except for a few outliers) due to the fact that the predictive equations have such a high degree of correlation with the observed data (the R-Squared value is 99.9% for all models). One also has to observe the small variation for each of the five payload sizes within each of the figures, not just the overall plots in order to see the true variation around the mean latency about each individual payload. Figure 24 for AES and 128-bit key length shows a larger variation of the data since the residuals are slightly larger than the other cases, but the overall plot of the residuals indicates good randomness and independence with a lack of any specific patterns. Thus the data is very consistent in measuring the latency of file transfers in the transmissions.

Plots of Residuals for AES and Blowfish Encryption

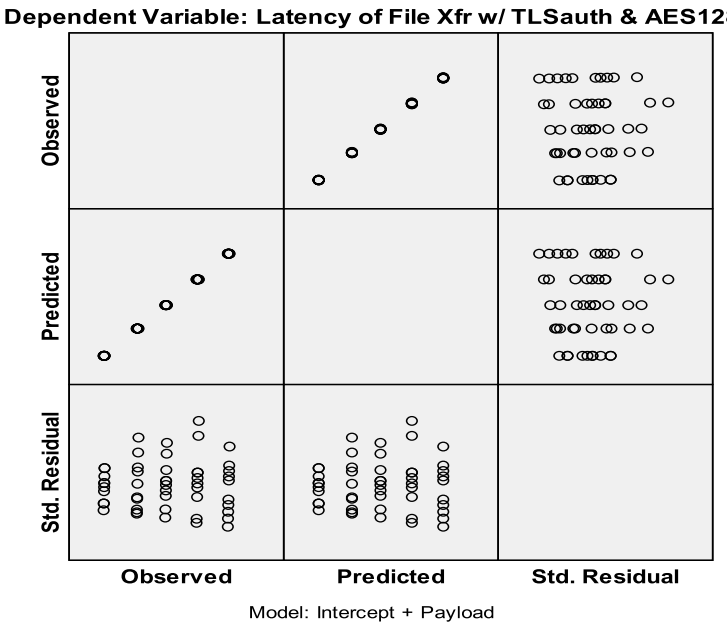


Figure 24 Review of Residuals for AES Encryption and 128-bit Key Length

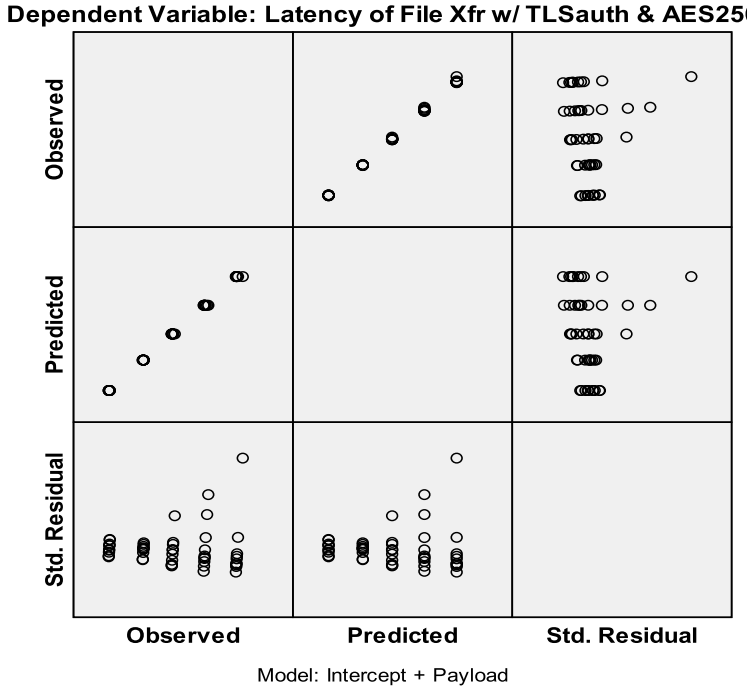


Figure 25 Review of Residuals for AES Encryption and 256-bit Key Length

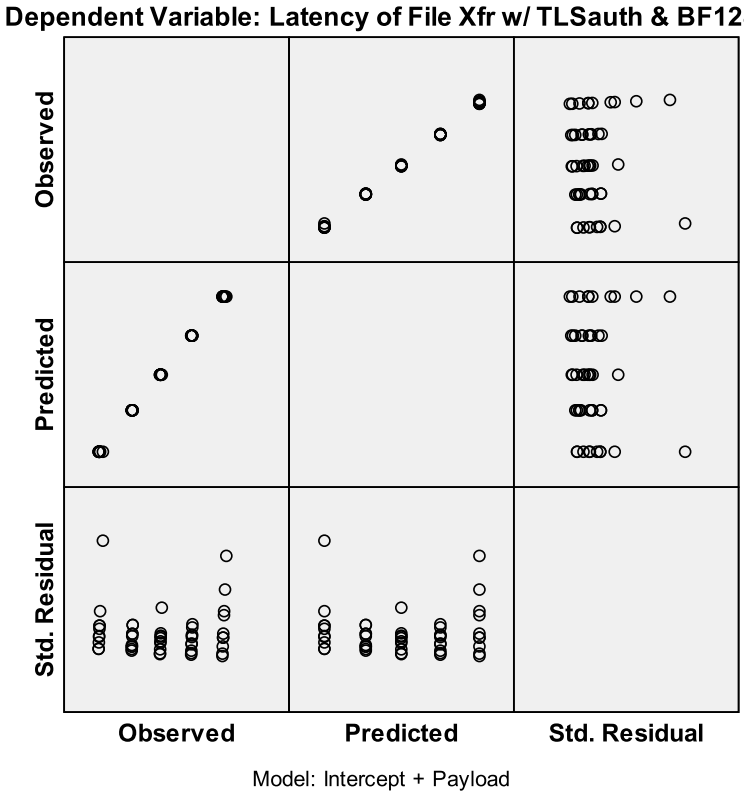


Figure 26 Review of Residuals for Blowfish Encryption and 128-bit Key Length

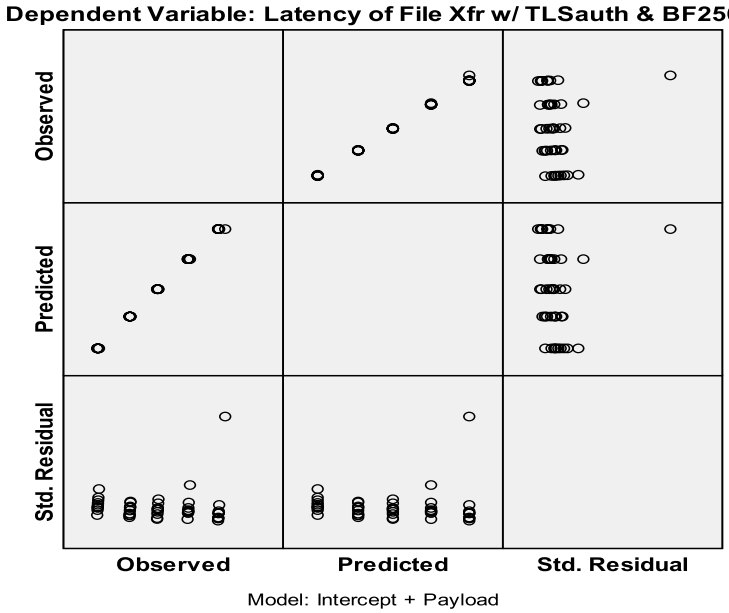


Figure 27 Review of Residuals for Blowfish Encryption and 256-bit Key Length

Summary of Results

Finally, if all combinations of FTP file transfers with/without the HIT Tags and with/without the different types of encryption and key lengths are entered into the same multivariate ANOVA analysis, the results are as follows:

Parameter Estimates

Dependent Variable	Parameter	B	Std. Error	t	Sig.	95% Confidence Interval		Partial Eta Squared	Noncent. Parameter	Observed Power[a]
						Lower Bound	Upper Bound			
Latency of File Xfr w/ TLSauth	Intercept	.090	.076	1.185	.242	-.063	.242	.028	1.185	.213
	Payload	.949	.002	431.010	.000	.944	.953	1.000	431.010	1.000
Latency of File Xfr w/o TLSauth	Intercept	-.028	.114	-.241	.810	-.257	.202	.001	.241	.056
	Payload	.935	.003	282.648	.000	.928	.942	.999	282.648	1.000
Latency of File Xfr w/ TLSauth & AES128	Intercept	.136	.061	2.213	.032	.012	.259	.093	2.213	.583
	Payload	1.011	.002	568.290	.000	1.007	1.014	1.000	568.290	1.000
Latency of File Xfr wo/ TLSauth & w/AES128	Intercept	.087	.083	1.056	.296	-.079	.253	.023	1.056	.179
	Payload	.957	.002	399.072	.000	.952	.962	1.000	399.072	1.000
Latency of File Xfr w/ TLSauth & AES256	Intercept	.040	.129	.310	.758	-.220	.300	.002	.310	.061
	Payload	1.016	.004	271.242	.000	1.009	1.024	.999	271.242	1.000
Latency of File Xfr wo/ TLSauth & w/AES256	Intercept	.001	.053	.010	.992	-.105	.106	.000	.010	.050
	Payload	.956	.002	626.614	.000	.953	.959	1.000	626.614	1.000
Latency of File Xfr w/ TLSauth & BF128	Intercept	.160	.095	1.680	.099	-.031	.351	.056	1.680	.377
	Payload	.960	.003	348.255	.000	.955	.966	1.000	348.255	1.000
Latency of File Xfr w/o TLSauth & w/BF128	Intercept	.033	.090	.372	.712	-.147	.214	.003	.372	.065
	Payload	.948	.003	363.739	.000	.942	.953	1.000	363.739	1.000
Latency of File Xfr w/ TLSauth & BF256	Intercept	.150	.114	1.315	.195	-.079	.379	.035	1.315	.252
	Payload	.961	.003	290.238	.000	.954	.967	.999	290.238	1.000
Latency of File Xfr w/o TLSauth & w/BF256	Intercept	.024	.052	.462	.646	-.080	.128	.004	.462	.074
	Payload	.944	.001	629.801	.000	.941	.947	1.000	629.801	1.000

a. Computed using alpha = .05

Figure 28 Overall MANOVA Parameter Estimates for all cases with/without TLS-auth (i.e. HIT Tags) and with/without all AES and Blowfish Encryption and Key Lengths.

The plots of the residuals are included in Appendix #3 to support the Parameter Estimates in Figure 28 and indicate similar good randomness and independence with a lack of specific patterns as observed before. The data is very consistent in measuring the latency of file transfers in the transmissions.

A summary of the additional latency for all cases with/without TLS-auth (i.e. HIT Tags) for all AES and Blowfish Encryption and Key Lengths is displayed in Figure 29 below:

File Transfer w/OpenVPN Tunnel	Approx. Addl Latency with vs. w/o HIT Tag
FTP file transfer only	1.5%
FTP file transfer with AES w/128-bit key (not including intercept constant, C = 0.136)	5.6%
FTP file transfer with AES w/256-bit key	6.2%
FTP file transfer with Blowfish w/128-bit key	1.3%
FTP file transfer with Blowfish w/256-bit key	1.8%

Figure 29 Additional Latency predicted by the MANOVA analysis for FTP file transfers for all cases with/without TLS-auth (i.e. HIT Tags).

In summary, the dissertation research and experimental planning provides a tested solution to the main research problem, namely: the need for low-latency across local and remote SmartGrid network nodes in order to transmit automation control parameters that achieve acceptable levels of performance, security and reliability using an open technology framework. Based on the results shown in Figure 29, the Blowfish encryption method offers less latency than AES encryption when HIT Tags are utilized; however, additional research and experimentation is warranted before stating any conclusions. The results of this dissertation indicate that use of the OpenVPN TLS-auth capability with HIT Tags is one possible means for the Smart Grid to securely and reliably transmit automation control parameters with relatively low-latency.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

The research and experimental results described in this dissertation provide guidance for control engineers to achieve low-latency file transfers through a system of open protocols that incorporate HMAC key processing and cryptographic identification for real-time communications across the Smart Grid. Among the requirements for real-time control are a reliable and consistent communications transport vehicle (such as the TLS protocol); high speed file transfer capability to provide low-latency with Identity Management for Denial of Service (DoS) resistance to reduce delays/outages; and strong authentication/security through encryption per the *Guidelines for Smart Grid Cyber Security* (NISTIR-Volumes 1-3, 2010). Security is further defined as end-to-end trust (E2E trust) that implements cryptographic means of authentication (i.e. HIT Tags) at each end-point and also seamless security across all the protocol layers and routers, proxies, etc. between user interfaces and/or other devices. The following discussion summarizes the key recommendations for automation control using the open software approach proposed in this dissertation to reduce latency while maintaining reliable communications transport and strong security.

Conclusions

Research Question#1 – Achieving Reliable FTP-TLS File Transfers

The testing described in Chapter 4 on latency versus payload size answered Research Question #1 by demonstrating that the OpenVPN TLS software implementation provided extremely consistent and reliable FTP file transfers while utilizing the 128-bit Host Identity Tag (HIT) cryptographic ID for end-to-end authentication. The statistical analysis provided a set of predictive equations that explained over 99.9% of the observed variation in the sample times (R-Squared values are >0.999 in all tests of the experimental models). In summary, the combination of OpenVPN and OpenSSL with the 128-bit Host Identity Tag (HIT) cryptographic ID created a reliable transport vehicle on an open software technology framework that is required for Smart Grid communications.

Research Question#2 – Achieving Low-Latency and DoS Resistance

The Identity Management technique proposed is to insert a Host Identity Tag (HIT), developed by the OpenHIP IETF project (Gurtov, 2008), within the secret pre-shared HMAC key of the OpenVPN transport protocol (OpenVPN, 2011). The results presented in Chapter 4 showed how typical automation file transfers (i.e. FTP applications) conducted with full end-to-end TLS-authentication, consumed only a nominal amount of additional latency. The added latency for a typical FTP file transfer through an OpenVPN TLS tunnel with end-to-end authentication but without encryption averaged only 1.5% of the existing transfer time for the same transfer without authentication and without encryption.

Research Question#3 – Impact of Encryption and Key Length

The third research question concerned the type of encryption used (e.g. AES, Blowfish, etc.) and the affect on the speed of file transfers and thus latency. The final experimental tests described in Chapter 4 investigated the impact of encryption and the length of the encryption keys on latency. The testing indicated that the type and key length of encryption added significant overhead and latency to file transfers. The incremental latency (with and without HIT Tags) using Blowfish encryption with 128/256-bit keys averaged 1.3% /1.8% compared with 5.6%/6.2% using AES.

Discussion on Conclusions and Recommendations

NIST proposed TLS (Transport Layer Security) as one possible protocol for consideration (NIST Special Publication 1108, 2010, p. 88) to achieve reliable real-time communications. The 128-bit Host Identity Tag (HIT) cryptographic ID was successfully integrated inside the OpenVPN version of the TLS communication software protocol in order to create a common means of identity and the approach functioned reliably.

As described in Chapter 4, the results also showed how typical automation file transfers (i.e. FTP applications) conducted with full end-to-end TLS-authentication, consumed only a nominal amount of additional latency. It is serendipitous that HMAC keys (Hashed Message Authentication Code) which are extensively used in the OpenVPN protocol can be processed very fast so there is little delay/latency added to the overall file transfer process (Goutis et al, 2005). This paper built on the research into HMAC processing by Goutis et al (2005) in order to provide high-speed file transfer using the File Transfer Protocol (FTP) over OpenVPN tunnels and achieve low-latency.

The use of a Host Identity Tag (HIT) from the OpenHIP technology project within the OpenVPN protocol provides strong Identity Management that enables Denial of Service (DoS) resistance (Gurtov, 2008) and reduces latency in two ways: 1) because it authenticates the identity of each end-point in the network with a cryptographic name as part of setting up the communications path to block man-in-the-middle attacks, and 2) because the OpenVPN transport protocol resists DoS attacks when it incorporates a secret key such as the proposed OpenHIP HIT Tag. In summary, the Identity Management technique proposed is to insert a HIT Tag within the secret pre-shared passphrase space allocated inside the TLS-Auth HMAC key of the OpenVPN transport protocol (OpenVPN, 2011). The testing described in Chapter 4 documents hundreds of successful FTP file transfers that utilized HIT Tags for successful end-to-end authentication.

As mentioned, the experiments also indicated that the type and key length of encryption added significant overhead and latency to file transfers. In fact, the results of the statistical analysis revealed interesting differences in latency for Blowfish encryption versus the Advanced Encryption Standard (AES). Further study of the nature of AES and Blowfish encryption for control automation communications is needed to understand the large variation in latencies. There are significant differences in the mathematical algorithms between AES which uses a 128-bit data block and the Blowfish encryption algorithm which utilizes a 64-bit data block. There may be significant reasons to be uncovered in future research to select a faster algorithm for purposes of control automation that still meets the NIST security requirements.

Discussion on Assumptions and Limitations

As described earlier, the major assumptions made and then utilized during the experiments were that:

1. The OpenVPN software worked well as a research tool and enabled the connection of the various protocols such as OpenSSL and OpenHIP into a common software model during all the testing as described in Chapter 4. Specifically, the TLS-authority capability in OpenVPN functioned as a suitable authentication layer inside the communications protocol stack to incorporate and test the use of Host Identity Tags (HITs) in order to achieve end-to-end authentication using cryptographic IDs.
2. In addition, the use of the FTP file transfer protocol worked well as an appropriate research communications vehicle to determine the latency of the test transfers. The Microsoft client and server FTP software successfully provided the timing services used for the experiments.

Following are a list of limitations encountered during the experiments:

1. A major limitation was that the research lab testing does not address scalability. The use of only a few computers does not simulate the tremendous volume of traffic expected on the Smart Grid that will potentially employ millions of routers, computers and automation control devices.
2. The Microsoft FTP server and client software worked well, but only provided accuracy for timing file transfers down to a hundredth of a second and that limited the range of testing such that the minimum payload was 10MB (megabytes). Custom development of FTP research software could enable far more accurate timing and thus permit much smaller payloads to be tested for latency of transmission.

3. The Research LAN was isolated and not connected to any other computers or the Internet. Additional testing needs to be conducted on the impact of extraneous noise on file transfers.
4. In addition the amount of random CPU utilization due to background services in the operating system (Microsoft XP) caused variation in the processing power available for testing and had to be monitored closely during testing. All unnecessary services such as security, firewall, disk monitoring, I/O interrupts, etc. were turned off to minimize background task processing for the experimental software. Custom software could be written to automatically monitor and regulate the variation in CPU availability to minimize testing error even further.
5. Wireshark could not be installed on the FTP server or client for timing because it induced additional load variations and interrupts on the network LAN adapter that affected file transfer times.
6. Network monitoring and logging had to be accomplished through Wireshark on a third, dedicated computer that listened quietly without consuming any resources and not affecting the processing on other nodes.
7. To maintain consistent CPU processor monitoring, performance, utilization and availability for all tests, all experiments were conducted in a single day-long (12 hour) session that placed a limit on the number of tests that could be conducted.

Recommendations for Future Research

As mentioned, there are significant implications for additional research. While the research in this dissertation is limited to a single HMAC Key authentication process using HIT Tags, the use of multiple, interactive HMAC Key exchanges (i.e. Interactive Hash Chains) provides very robust security and approaches the level of high assurance (Perrig, 2005).

Likewise HMAC Key processing can extend the Smart Grid firewalls to the network end-points for improved end-to-end security and further minimization of latency. The expanded use of HMAC Keys, HIT Tags, Message Digests and Digital Signatures as an extended firewall adds significant security, reliability, speed and accuracy for all file transfers, not just real-time automation control. Some additional areas for further research include:

- Further study of the nature of AES and Blowfish, as well as other potential types of encryption, for control automation communications is needed to understand the large variation in latency. The incremental latency (with and without HIT Tags) using Blowfish encryption with 128/256-bit keys averaged only 1.3% /1.8% compared with 5.6%/6.2% using AES. There are significant differences in the mathematical algorithms between AES which uses a 128-bit data block for instance and the Blowfish encryption algorithm which utilizes a 64-bit data block. There may be significant reasons to be uncovered in future research to select a faster algorithm for purposes of control automation that still meets the NIST security requirements.
- Implementation of DNS Secure (DNSSEC) for trusted access and retrieval of the HIT Tags for users/devices;
- Encapsulation of the provenance (who, what, when, where, and why about the transmission of automation control system updates) within the HMAC so that all parties/nodes communicate trust by understanding each other's attributes without resorting to multiple additional layers of proprietary protocols;
- Self-management and load-balancing capabilities within the protocol to minimize energy consumption across the Smart Grid;

- Remote programming of routers/nodes to reconfigure the network in order to meet new communication demands;
- Automatic renegotiation of security keys for non-stop, trustworthy communications that could utilize Interactive Hash Chains (IHC);
- Definition of additional interfaces for Privacy and Access Control in Federated Systems; and
- Extension of the Host Identity Tag-enabled TLS transport layer for Cloud Security.

Interactive Hash Chains (IHC)

Interactive Hash Chains could be used to distribute HIT Tags to users and sensor/control devices on the SmartGrid. Likewise the use of Interactive Hash Chains could be of significant value to low compute-power devices because they are orders of magnitude more compact and efficient than public key (asymmetric) algorithms (Perrig, 2005). IETF RFC 4082 (2005) describes the IHC technology and also describes the difficulty of provisioning a secret key to both end-points. There is often significant additional expense and operational maintenance if utilities provision a sensor/control device with a secret key and then ship it to a customer. However, Interactive Hash Chains enable the secure deployment of secret keys after deployment and thus provide the benefits and lower cost of symmetric keys with the greater security associated with asymmetric keys (Perrig, 2005).

Basically, the IHC method transmits information (i.e. control parameters, sensor feedback, etc.) in an HMAC transmission computed with a secret key that is not disclosed until later (Gurtov, 2008). The HMAC is buffered and then a subsequent HMAC is sent that discloses the key to be used to unwrap the protected information. The primary requirement for use of IHC is precise timing on the network and the SmartGrid already has a need for accurate timing

synchronization to satisfy other requirements. One of the most widely known versions of the technology is called TESLA (Timed Efficient Stream Loss-tolerant Authentication) by Adrian Perrig (2005) at Carnegie Mellon University.

REFERENCES

American Gas Association. (2006). AGA Report No. 12 - Cryptographic Protection of SCADA Communications Part 1: Background, Policies and Test Plan. Retrieved March 1, 2010, from http://www.aga.org

Atwood, J. W. and Xueyong Zhu (2007). A Web Database Security Model Using the Host Identity Protocol. Hefei, Anhui, China: Network Information Center University of Science and Technology of China.

Baugher, M., McGrew, D., Naslund, M., Carrara, E. and Norrman, K. (2004). The Secure Real-time Transport Protocol (SRTP). Internet Engineering Task Force, RFC 3711, Mar. 2004. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3711.txt

Blake, M. (2003). Coordinating multiple agents for workflow-oriented process orchestration. Department of Computer Science, Georgetown University. [Online]. Available: http://www.cs.georgetown.edu/~blakeb/pubs/blake_ISEB2003.pdf

Camarillo, G., I. M´as and Nikander, P. (2008). A Framework to Combine the Session Initiation Protocol and the Host Identity Protocol. Ericsson Research. IEEE Communications Society WCNC 2008 proceedings.

Clark, D., Wroclawski, J., Sollins, K., and Braden, R. (2005). Tussle in Cyberspace: Defining Tomorrow's Internet", IEEE Transaction on Networking, Vol 13, No 3, June 2005.

Courtois, Nicholas T (2004). Is AES a Secure Cipher?

Available from: http://www.cryptosystem.net/aes/

Dumon, P. (1997). OS Kernels: A Little Overview and Comparison.

Available from: http://tunes.org/~unios/oskernels.html#rings

Emerson Process Management. (November 3, 2009). Emerson introduces FOUNDATION™ Fieldbus Interfaces for remote oil, gas and water applications. Retrieved March 1, 2010, from http://www2.emersonprocess.com/en-US/news/pr/Pages/911-ROC-fieldbus.aspx

Fieldbus Foundation ISA (2010). ISA standards and related information.

Retrieved March 1, 2010, from www.isa.org/standards

Ford-Hutchinson, P. (2005). RFC 4217, Securing FTP with TLS.

Internet Engineering Task Force, RFC 4217, Oct. 2005. Retrieved March 10, 2010, from http://www.rfc-editor.org/rfc/rfc4217.txt

Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A. and Stewart, L. (1999). HTTP Authentication: Basic and Digest Access Authentication. Internet Engineering Task Force, RFC 2617, Jun. 1999. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2617.txt

Glanzer, D. (2005). FOUNDATION Fieldbus HSE: An Open, High Speed Solution.

Retrieved March 1, 2010, from http://www.fieldbus.org/images/stories/enduserresources/technicalreferences/documents/HSE%20Brazil%20articlefinal.pdf

Goutis, C., Kakarountas, A., Michail, H., Papadonikolakis, M., Yiakoumis, I. (2005). Efficient Small-Sized Implementation of the Keyed-Hash Message Authentication Code. EUROCON 2005, Serbia and Montenegro, Belgrade, November 22-24, 2005. Retrieved March 15, 2011, from http://www.stanford.edu/~yiannisy/cgi-bin/docs/eurocon.pdf

Gurtov, A. (2008). Host Identity Protocol (HIP): Towards the Secure Mobile Internet. United Kingdom: John Wiley and Sons Ltd.

Handley, M. , Jacobson, V. and Perkins, C. (2006). SDP: Session Description Protocol. Internet Engineering Task Force, RFC 4566, Jul. 2006. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4566.txt

Henderson, T. (2008). RFC 5338 – Using the Host Identity Protocol with Legacy Applications, Internet Engineering Task Force, RFC 5338, Sep. 2008. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5338.txt

Henderson, T. (2008). Boeing HIP Secure Mobile Architecture - Update to IETF 73 Host Identity Protocol Research Group (HIPRG), Meeting (November 21, 2008), thomas.r.henderson@boeing.com Seattle: Boeing Aircraft

Hosner, C. (2004). OpenVPN and the SSL VPN Revolution. GSEC v.1.4b. SANS Institute. Retrieved April 1, 2011 from: http://openvpn.net/index.php/open-source/articles.html

Jain, R. (2006). Internet 3.0: Ten Problems with Current Internet Architecture and Solutions for the Next Generation. Proceedings of the Military Communications Conference (MILCOM 2006), Washington DC, October 23-25, 2006. Retrieved Feb 1, 2010, from: http://www1.cse.wustl.edu/~jain/papers/ftp/gina.pdf

Khalil, T. (2000). Management of Technology: The Key to Competitiveness and Wealth Creation. University of Miami. Boston: McGraw-Hill,

Kowalenko, K. and Weiss, J. (14 April 2010). FEATURE STORY - The Cyberhacker's Next Victim: Industrial Infrastructures. THE INSTITUTE - IEEE Home » Featured This Month » Article, Retrieved March 1, 2010, from http://www.theinstitute.ieee.org/portal/site/tionline/menuitem.130a3558587d56e8fb2275875bac26c8/index.jsp?&pName=institute_level1_article&TheCat=2201&article=tionline/legacy/inst2010/apr10/featuretechnology.xml&

Lamping, U. (2010). Wireshark Developer's Guide for Wireshark 1.4. Retrieved April 1, 2011 from http://www.wireshark.org/docs/wsdg_html_chunked/

Law Y., Doumen J., Hartel P. (2003). Survey and Benchmark of Block Ciphers for Wireless Sensor Networks. Available from: http://www.ub.utwente.nl/webdocs/ctit/1/000000eb.pdf

Leedy, P. D. (2010). Practical Research Planning and Design. Ninth Edition. New Jersey: Pearson Education

Mahy, R., Biggs, B. and Dean, R. (2004). The Session Initiation Protocol (SIP) Replaces Header. Internet Engineering Task Force, RFC 3891, Sep. 2004. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3891.txt

Marsan, C. (2009, October). Will Smart Grid Power IPv6? Network World, Retrieved February 1, 2010, from http://www.networkworld.com/news/2009/102909-smart-grid-ipv6.html

Moskowitz, R. (2010). Re: [HIPsec] HIP Usage in a Large Smart Grid/SCADA Environment. Internet Engineering Task Force, work in progress. Retrieved February 15, 2011, from http://www.ietf.org/mail-archive/web/hipsec/current/msg02892.html

Nikander, P. and Sarela, M. (2004). Applying Host Identity Protocol to Tactical Networks. Helsinki University of Technology and Ericsson Research IP Networks.

Nikander, P., Laganier, J. and Dupont, F. (2007). An IPv6 Prefix for Overlay Routable Cryptographic Hash Identifiers (ORCHID). Internet Engineering Task Force, RFC 4843, Apr. 2007. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4843.txt

NIST-FIPS-140-2. (2001). National Institute of Standards and Technology, Security Requirements for Cryptographic Modules, Federal Information Processing Standards Publication 140-2, May 25, 2001. Gaithersburg, MD: NIST

NIST-FIPS-198. (2002). National Institute of Standards and Technology, The Keyed-Hash Message Authentication Code (HMAC), Federal Information Processing Standards Publication 198, March 6, 2002. Gaithersburg, MD: NIST

NIST. (2004). System Protection Profile - Industrial Control Systems Version 1.0. NIST, Gaithersburg, MD Retrieved March 1, 2010, from http://www.isd.mel.nist.gov/projects

NIST. (2010). Smart Grid Interoperability Panel: A New, Open Forum for Standards Collaboration [Online]. Available: http://collaborate.nist.gov/twiki-sggrid/pub/SmartGrid/CMEWG/Whatis_SGIP_final.pdf

NIST Special Publication 1108. (2010). NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 1.0. Office of the National Coordinator for Smart Grid Interoperability. Gaithersburg, MD: NIST

NISTIR-Vol.1. (2010). Guidelines for Smart Grid Cyber Security: Vol. 1, Smart Grid Cyber Security Strategy, Architecture, and High-Level Requirements. National Institute of Standards and Technology Interagency Report. Gaithersburg, MD: NIST

NISTIR-Vol.2. (2010). Guidelines for Smart Grid Cyber Security: Vol. 2, Privacy and the Smart Grid. National Institute of Standards and Technology Interagency Report. Gaithersburg, MD: NIST

NISTIR-Vol.3. (2010). Guidelines for Smart Grid Cyber Security: Vol. 3, Supportive Analyses and References. National Institute of Standards and Technology Interagency Report. Gaithersburg, MD: NIST

Norusis, M. J. (2008). SPSS Statistics 17.0: Statistical Procedures Companion. New Jersey: Prentice Hall

Open Grid (2008). Secure Communication Profile 1.0. Jun 13, 2008. Retrieved April 1, 2011 from http://www.ggf.org/documents/GFD.132.pdf

OpenVPN (2011). OpenVPN Community Project. Retrieved April 1, 2011 from http://openvpn.net

Paine, R. (2007, May). Secure Mobile Architecture SMA Basics for IEEE 802.21. Secure Mobile Architecture (SMA) Demo Team. Seattle: Boeing Aircraft, Math and Computing Technologies

Perez, S. (2008). 2008 Proceedings of the Privacy and Security Working Group of the Communications Futures Program. Boston: Massachusetts Institute of Technology

Perrig, A. (2005). Timed Efficient Stream Loss-tolerant Authentication (TESLA). Carnegie Mellon U. IETF RFC 4082. Retrieved April 1, 2011 from: http://www.ietf.org/rfc/rfc4082.txt

Phifer, Lisa (2003). VPN: Tunnel Vision. Information Security Magazine Online. Available from: http://infosecuritymag.techtarget.com/ss/0,295796,sid6_iss21_art83,00.html

Rescorla, E. (2001). SSL and TLS: Designing and Building Secure Systems. Indianapolis, IN: Addison-Wesley.

Robinson, C. (February, 2010). ISA99: Charting a security standards roadmap into a risky new decade. ISA | InTech, Retrieved March 21, 2010, from http://www.isa.org/InTechTemplate.cfm?Section=Standards_Update1&template=/ContentManagement/ContentDisplay.cfm&ContentID=81089RESOURCES

Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and Schooler, E. (2002). SIP: Session Initiation Protocol. Internet Engineering Task Force, RFC 3261, Jun. 2002. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3261.txt

Rosenberg J. and Schulzrinne, H. (2002). An Offer/Answer Model with Session Description Protocol (SDP). Internet Engineering Task Force, RFC 3264, Jun. 2002. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3264.txt

Rosenberg, J. and Schulzrinne, H. (2002). Session Initiation Protocol (SIP): Locating SIP Servers. Internet Engineering Task Force, RFC 3263, Jun. 2002. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3263.txt

RSA Laboratories. (1996). The Status of MD5 After a Recent Attack. CryptoBytes. Available from: ftp://ftp.rsasecurity.com/pub/cryptobytes/crypto2n2.pdf

Schneier, B Ferguson, N. (1999). A Cryptographic Evaluation of IPSec. [Online]. Available from: http://www.schneier.com/

Snyder, J. (2004). SSL VPN Gateways.

Network World Fusion. Online. Available from:

http://www.nwfusion.com/reviews/2004/0112revmain.html

So, J. Y., Wang, J. and Jones, D. (2005). SHIP mobility management hybrid SIPHIP scheme.

Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2005 and First ACIS International Workshop on Self-Assembling Wireless Networks. SNPD/SAWN 2005, 2005.

Sturm, J. A. (2008). The Internet Is Growing Logarithmically.

Terre Haute, IN: Indiana State University

The Economist. (2004, March). The Future of Technology - Energy.

The Economist, Retrieved February 1, 2010, from

http://media.economist.com/theworldin/index.cfm?d=2004

Weiss, J. (2010). Protecting Industrial Control Systems from Electronic Threats. Highland Park, New Jersey: Momentum Press.

White House. (2011). *A Policy Framework for the 21st Century Grid: Enabling Our Secure Energy Future*. Released June 13, 2011, Washington D.C.: Executive Office of the President. Retrieved June 13, 2011, from

http://www.whitehouse.gov/sites/default/files/microsites/ostp/nstc-smart-grid-june2011.pdf

Yonan, J. (2004). OpenVPN Source Forge home page.

Retrieved April 1, 2011 from http://openvpn.net

APPENDIX #1: FIPS PUB 198 – NIST HMAC STANDARD

FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION

The Keyed-Hash Message Authentication Code (HMAC)

CATEGORY: COMPUTER SECURITY SUBCATEGORY: CRYPTOGRAPHY
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8900
Issued March 6, 2002

U.S. Department of Commerce
Donald L. Evans, Secretary
Technology Administration
Philip J. Bond, Under Secretary
National Institute of Standards and Technology
Arden L. Bement, Jr., Director

Foreword

The Federal Information Processing Standards Publication Series of the National Institute of Standards and Technology (NIST) is the official series of publications relating to standards and guidelines adopted and promulgated under the provisions of Section 5131 of the Information Technology Management Reform Act of 1996 (Public Law 104-106) and the Computer Security Act of 1987 (Public Law 100-235). These mandates have given the Secretary of Commerce and NIST important responsibilities for improving the utilization and management of computer and related telecommunications systems in the Federal government. The NIST, through its Information Technology Laboratory, provides leadership, technical guidance, and coordination of government efforts in the development of standards and guidelines in these areas.

Comments concerning Federal Information Processing Standards Publications are welcomed and should be addressed to the Director, Information Technology Laboratory, National Institute of Standards and Technology, 100 Bureau Drive, Stop 8900, Gaithersburg, MD 20899-8900.
William Mehuron, Director
Information Technology Laboratory

Abstract

This standard describes a keyed-hash message authentication code (HMAC), a mechanism for message authentication using cryptographic hash functions. HMAC can be used with any iterative Approved cryptographic hash function, in combination with a shared secret key. The cryptographic strength of HMAC depends on the properties of the underlying hash function. The HMAC specification in this standard is a generalization of Internet RFC 2104, *HMAC, Keyed-Hashing for Message Authentication*, and ANSI X9.71, *Keyed Hash Message Authentication Code.*

Keywords: computer security, cryptography, HMAC, MAC, message authentication, Federal Information Processing Standard (FIPS).

Federal Information Processing Standards Publication 198
2002 March 6
Announcing the Standard for

The Keyed-Hash Message Authentication Code (HMAC)

Federal Information Processing Standards Publications (FIPS PUBS) are issued by the National Institute of Standards and Technology (NIST) after approval by the Secretary of Commerce pursuant to Section 5131 of the Information Technology Management Reform Act of 1996 (Public Law 104-106) and the Computer Security Act of 1987 (Public Law 100-235).

1. Name of Standard. Keyed-Hash Message Authentication Code (HMAC) (FIPS PUB 198).

2. Category of Standard. Computer Security Standard. **Subcategory.** Cryptography.

3. Explanation. This standard specifies an algorithm for applications requiring message authentication. Message authentication is achieved via the construction of a message authentication code (MAC). MACs based on cryptographic hash functions are known as HMACs.

The purpose of a MAC is to authenticate both the source of a message and its integrity without the use of any additional mechanisms. HMACs have two functionally distinct parameters, a message input and a secret key known only to the message originator and intended receiver(s). Additional applications of keyed-hash functions include their use in challenge-response identification protocols for computing responses, which are a function of both a secret key and a challenge message.

An HMAC function is used by the message sender to produce a value (the MAC) that is formed by condensing the secret key and the message input. The MAC is typically sent to the message receiver along with the message. The receiver computes the MAC on the received message using the same key and HMAC function as was used by the sender, and compares the result computed with the received MAC. If the two values match, the message has been correctly received, and the receiver is assured that the sender is a member of the community of users that share the key.
The HMAC specification in this standard is a generalization of HMAC as specified in Internet RFC 2104, *HMAC, Keyed-Hashing for Message Authentication*, and ANSI X9.71, *Keyed Hash Message Authentication Code.*

4. Approving Authority. Secretary of Commerce.

5. Maintenance Agency. Department of Commerce, National Institute of Standards and Technology, Information Technology Laboratory (ITL).

6. Applicability. This standard is applicable to all Federal departments and agencies for the protection of sensitive unclassified information that is not subject to section 2315 of Title 10, United States Code, or section 3502(2) of Title 44, United States Code. This

standard shall be used in designing, acquiring and implementing keyed-hash message authentication techniques in systems that Federal departments and agencies operate or which are operated for them under contract. The adoption and use of this standard is available on a voluntary basis to private and commercial organizations.

7. Specifications. Federal Information Processing Standard (FIPS) 198, Keyed-Hash Message Authentication Code (HMAC) (affixed).

8. Implementations. The authentication mechanism described in this standard may be implemented in software, firmware, hardware, or any combination thereof. NIST has developed a Cryptographic Module Validation Program that will test implementations for conformance with this HMAC standard. Information on this program is available at http://csrc.nist.gov/cryptval/.

Agencies are advised that keys used for HMAC applications should not be used for other purposes.

9. Other Approved Security Functions. HMAC implementations that comply with this standard shall employ cryptographic algorithms, cryptographic key generation algorithms and key management techniques that have been approved for protecting Federal government sensitive information. Approved cryptographic algorithms and techniques include those that are either:

a. specified in a Federal Information Processing Standard (FIPS),
b. adopted in a FIPS or NIST Recommendation and specified either in an appendix to the FIPS or NIST Recommendation or in a document referenced by the FIPS or NIST Recommendation, or
c. specified in the list of Approved security functions for FIPS 140-2.

10. Export Control. Certain cryptographic devices and technical data regarding them are subject to Federal export controls. Exports of cryptographic modules implementing this standard and technical data regarding them must comply with these Federal regulations and be licensed by the Bureau of Export Administration of the U.S. Department of Commerce. Applicable Federal government export controls are specified in Title 15, Code of Federal Regulations (CFR) Part 740.17; Title 15, CFR Part 742; and Title 15, CFR Part 774, Category 5, Part 2.

11. Implementation Schedule. This standard becomes effective on September 6, 2002.

12. Qualifications. The security afforded by the HMAC function is dependent on maintaining the secrecy of the key. Therefore, users must guard against disclosure of these keys. While it is the intent of this standard to specify a mechanism to provide message authentication, conformance to this standard does not assure that a particular implementation is secure. It is the responsibility of the implementer to ensure that any module containing an HMAC implementation is designed and built in a secure manner.

Similarly, the use of a product containing an implementation that conforms to this standard does not guarantee the security of the overall system in which the product is used. The responsible authority in each agency shall assure that an overall system provides an acceptable level of security.

Since a standard of this nature must be flexible enough to adapt to advancements and innovations in science and technology, this standard will be reviewed every five years in order to assess its adequacy.

13. Waiver Procedure. Under certain exceptional circumstances, the heads of Federal agencies, or their delegates, may approve waivers to Federal Information Processing Standards (FIPS). The heads of such agencies may redelegate such authority only to a senior official designated pursuant to Section 3506(b) of Title 44, U.S. Code. Waivers shall be granted only when compliance with this standard would

a. adversely affect the accomplishment of the mission of an operator of Federal computer system or

b. cause a major adverse financial impact on the operator that is not offset by government-wide savings.

Agency heads may act upon a written waiver request containing the information detailed above. Agency heads may also act without a written waiver request when they determine that conditions for meeting the standard cannot be met. Agency heads may approve waivers only by a written decision that explains the basis on which the agency head made the required finding(s). A copy of each such decision, with procurement sensitive or classified portions clearly identified, shall be sent to: National Institute of Standards and Technology; ATTN: FIPS Waiver Decision, Information Technology Laboratory, 100 Bureau Drive, Stop 8900, Gaithersburg, MD 20899-8900.

In addition, notice of each waiver granted and each delegation of authority to approve waivers shall be sent promptly to the Committee on Government Operations of the House of Representatives and the Committee on Government Affairs of the Senate and shall be published promptly in the Federal Register.
When the determination on a waiver applies to the procurement of equipment and/or services, a notice of the waiver determination must be published in the Commerce Business Daily as a part of the notice of solicitation for offers of an acquisition or, if the waiver determination is made after that notice is published, by amendment to such notice.

A copy of the waiver, any supporting documents, the document approving the waiver and any supporting and accompanying documents, with such deletions as the agency is authorized and decides to make under Section 552(b) of Title 5, U.S. Code, shall be part of the procurement documentation and retained by the agency.

14. Where to obtain copies. This publication is available by accessing http://csrc.nist.gov/publications/. A list of other available computer security publications,

including ordering information, can be obtained from NIST Publications List 91, which is available at the same web site. Alternatively, copies of NIST computer security publications are available from: National Technical Information Service (NTIS), 5285 Port Royal Road, Springfield, VA 22161.

Federal Information Processing Standards Publication 198
2002 March 6
Specifications for

The Keyed-Hash Message Authentication Code

TABLE OF CONTENTS

1. INTRODUCTION

Providing a way to check the integrity of information transmitted over or stored in an unreliable medium is a prime necessity in the world of open computing and communications. Mechanisms that provide such integrity checks based on a secret key are usually called message authentication codes (MACs). Typically, message authentication codes are used between two parties that share a secret key in order to authenticate information transmitted between these parties. This standard defines a MAC that uses a cryptographic hash function in conjunction with a secret key. This mechanism is called HMAC and is a generalization of HMAC as specified in [1] and [3].

HMAC shall be used in combination with an Approved cryptographic hash function. HMAC uses a secret key for the calculation and verification of the MACs. The main goals behind the HMAC construction [3] are:

- To use available hash functions without modifications; in particular, hash functions that perform well in software, and for which code is freely and widely available,

- To preserve the original performance of the hash function without incurring a significant degradation,

- To use and handle keys in a simple way,

- To have a well-understood cryptographic analysis of the strength of the authentication mechanism based on reasonable assumptions on the underlying hash function, and

- To allow for easy replaceability of the underlying hash function in the event that faster or more secure hash functions are later available.

2. GLOSSARY OF TERMS AND ACRONYMS

2.1 Glossary of Terms

The following definitions are used throughout this standard:

Approved: FIPS-approved or NIST recommended. An algorithm or technique that is either 1) specified in a FIPS or NIST Recommendation, or 2) adopted in a FIPS or NIST Recommendation and specified either the FIPS or NIST Recommendation, or in a document referenced by the FIPS or NIST Recommendation.

Cryptographic key (key): a parameter used in conjunction with a cryptographic algorithm that determines the specific operation of that algorithm. In this standard, the cryptographic key is used by the HMAC algorithm to produce a MAC on the data.

Hash function: an Approved mathematical function that maps a string of arbitrary length (up to a pre-determined maximum size) to a fixed length string. It may be used to produce a checksum, called a hash value or message digest, for a potentially long string or message.

Keyed-hash based message authentication code (HMAC): a message authentication code that uses a cryptographic key in conjunction with a hash function.

Message Authentication Code (MAC): a cryptographic checksum that results from passing data through a message authentication algorithm. In this standard, the message authentication algorithm is called HMAC, while the result of applying HMAC is called the MAC.

Secret key: a cryptographic key that is uniquely associated with one or more entities. The use of the term "secret" in this context does not imply a classification level; rather the term implies the need to protect the key from disclosure or substitution.

2.2 Acronyms

The following acronyms and abbreviations are used throughout this standard:

FIPS Federal Information Processing Standard

FIPS PUB FIPS Publication

HMAC Keyed-Hash Message Authentication Code

MAC Message Authentication Code

NIST National Institute of Standards and Technology

2.3 HMAC Parameters and Symbols

HMAC uses the following parameters:

B Block size (in bytes) of the input to the Approved hash function.

H An Approved hash function.

ipad Inner pad; the byte x’36’ repeated *B* times.

K Secret key shared between the originator and the intended receiver(s).

K_0 The key *K* after any necessary pre-processing to form a *B* byte key.

L Block size (in bytes) of the output of the Approved hash function.

opad Outer pad; the byte x'5c' repeated B times.

t The number of bytes of MAC.

text The data on which the HMAC is calculated; *text* does **not** include the padded key. The length of *text* is n bits, where $0 \le n < 2^B - 8B$.

x'N' Hexadecimal notation, where each symbol in the string 'N' represents 4 binary bits.

|| Concatenation

⊕ Exclusive-Or operation.

3. CRYPTOGRAPHIC KEYS

The size of the key, K, shall be equal to or greater than $L/2$, where L is the size of the hash function output. Note that keys greater than L bytes do not significantly increase the function strength. Applications that use keys longer than B-bytes shall first hash the key using H and then use the resultant L-byte string as the HMAC key, K. Keys shall be chosen at random using an Approved key generation method and shall be changed periodically. Note that the keys should be protected in a manner that is consistent with the value of the data that is to be protected (i.e., the *text* that is authenticated using the HMAC function).

4. TRUNCATED OUTPUT

A well-known practice with MACs is to truncate their output (i.e., the length of the MAC used is less than the length of the output of the MAC function L). Applications of this standard may truncate the output of HMAC. When a truncated HMAC is used, the t leftmost bytes of the HMAC computation shall be used as the MAC. The output length, t, shall be no less than four bytes (i.e., $4 \le t \le L$). However, t shall be at least

$L / 2$ bytes (i.e. $L / 2 \le t \le L$)

unless an application or protocol makes numerous trials impractical. For example, a low bandwidth channel might prevent numerous trials on a 4 byte MAC, or a protocol might allow only a small number of invalid MAC attempts. See Appendix B.

5. HMAC SPECIFICATION

To compute a MAC over the data '*text*' using the HMAC function, the following operation is performed:

$$\boldsymbol{MAC(text)_t = HMAC(K, text)_t = H((K_0 \oplus opad) \,||\, H((K_0 \oplus ipad) \,||\, text))_t}$$

Table 1 illustrates the step by step process in the HMAC algorithm, which is depicted in Figure 1.

Table 1: The HMAC Algorithm

STEPS STEP-BY-STEP DESCRIPTION

Step 1 If the length of $K = B$: set $K_0 = K$. Go to step 4.

Step 2 If the length of $K > B$: hash K to obtain an L byte string, then append $(B\text{-}L)$ zeros to create a B-byte string K_0 (i.e., $K_0 = H(K) \,||\, 00...00$). Go to step 4.

Step 3 If the length of $K < B$: append zeros to the end of K to create a B-byte string K_0 (e.g., if K is 20 bytes in length and $B = 64$, then K will be appended with 44 zero bytes 0x00).

Step 4 Exclusive-Or K_0 with *ipad* to produce a B-byte string: $\boldsymbol{K_0 \oplus ipad}$.

Step 5 Append the stream of data '*text*' to the string resulting from step 4: $\boldsymbol{(K_0 \oplus ipad) \,||\, text}$.

Step 6 Apply H to the stream generated in step 5: $\boldsymbol{H((K_0 \oplus ipad) \,||\, text)}$.

Step 7 Exclusive-Or K_0 with *opad*: $\boldsymbol{K_0 \oplus opad}$.

Step 8 Append the result from step 6 to step 7: $\boldsymbol{(K_0 \oplus opad) \,||\, H((K_0 \oplus ipad) \,||\, text)}$.

Step 9 Apply H to the result from step 8: $\boldsymbol{H((K_0 \oplus opad) \,||\, H((K_0 \oplus ipad) \,||\, text))}$.

Step 10 Select the leftmost t bytes of the result of step 9 as the MAC.

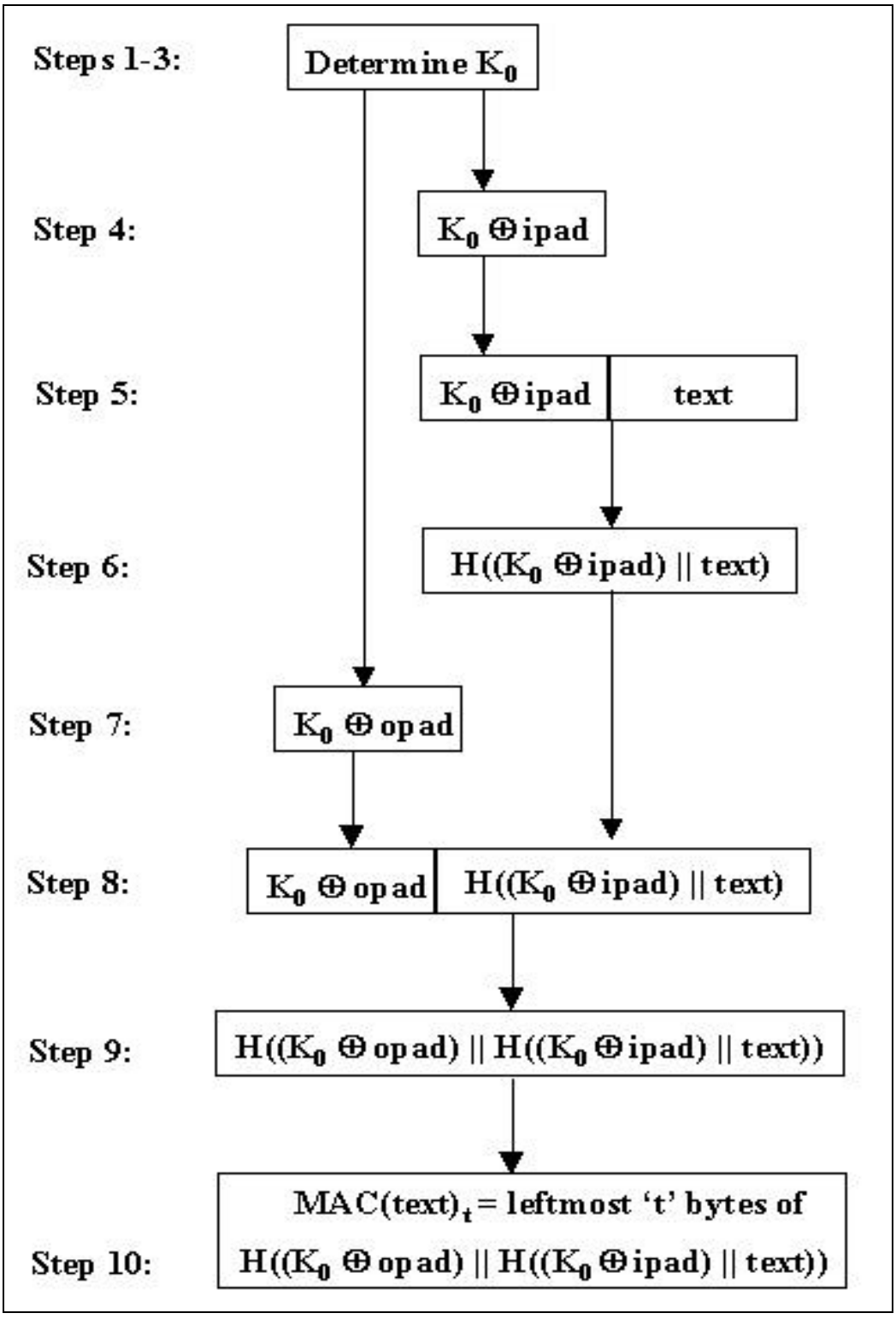


Figure 1: Illustration of the HMAC Construction

6. IMPLEMENTATION NOTE

The HMAC algorithm is specified for an arbitrary Approved cryptographic hash function, *H*. With minor modifications, an HMAC implementation can easily replace one hash function, *H*, with another hash function, *H'*.

Conceptually, the intermediate results of the compression function on the *B*-byte blocks ($K_0 \oplus ipad$) and ($K_0 \oplus opad$) can be precomputed once, at the time of generation of the key *K*, or before its first use. These intermediate results can be stored and then used to initialize *H* each time that a message needs to be authenticated using the same key. For each authenticated message using the key *K*, this method saves the application of the hash function of *H* on two *B*-byte blocks (i.e., on ($K \oplus ipad$) and ($K \oplus opad$)). This saving may be significant when authenticating short streams of data. **These stored intermediate values shall be treated and protected in the same manner as secret keys.**

Choosing to implement HMAC in this manner has no effect on interoperability.

Object identifiers (OIDs) for HMAC are posted at http://csrc.nist.gov/csor, along with procedures for adding new OIDs.

APPENDIX A: HMAC EXAMPLES

These examples are provided in order to promote correct implementations of HMAC.

The SHA-1 hash function used in these examples is specified in [4].

A.1 SHA-1 with 64-Byte Key

Text: "Sample #1"

Key: 00010203 04050607 08090a0b 0c0d0e0f
10111213 14151617 18191a1b 1c1d1e1f
20212223 24252627 28292a2b 2c2d2e2f
30313233 34353637 38393a3b 3c3d3e3f

K_0: 00010203 04050607 08090a0b 0c0d0e0f
10111213 14151617 18191a1b 1c1d1e1f
20212223 24252627 28292a2b 2c2d2e2f
30313233 34353637 38393a3b 3c3d3e3f

$K_0 \oplus$ ipad:
36373435 32333031 3e3f3c3d 3a3b3839
26272425 22232021 2e2f2c2d 2a2b2829
16171415 12131011 1e1f1c1d 1a1b1819
06070405 02030001 0e0f0c0d 0a0b0809

(Key $\oplus$ ipad)||text:
36373435 32333031 3e3f3c3d 3a3b3839
26272425 22232021 2e2f2c2d 2a2b2829
16171415 12131011 1e1f1c1d 1a1b1819
06070405 02030001 0e0f0c0d 0a0b0809
53616d70 6c652023 31

Hash((Key $\oplus$ ipad)||text):
bcc2c68c abbbf1c3 f5b05d8e 7e73a4d2
7b7e1b20

$K_0 \oplus$ opad:
5c5d5e5f 58595a5b 54555657 50515253
4c4d4e4f 48494a4b 44454647 40414243
7c7d7e7f 78797a7b 74757677 70717273
6c6d6e6f 68696a6b 64656667 60616263

($K_0 \oplus$ opad) || Hash(($K_0 \oplus$ ipad)||text):
```
5c5d5e5f 58595a5b 54555657 50515253
4c4d4e4f 48494a4b 44454647 40414243

7c7d7e7f 78797a7b 74757677 70717273
6c6d6e6f 68696a6b 64656667 60616263
bcc2c68c abbbf1c3 f5b05d8e 7e73a4d2
7b7e1b20
```

HMAC(Key, Text) = Hash(($K_0 \oplus$ opad) || Hash(($K_0 \oplus$ ipad)||text)):
```
4f4ca3d5 d68ba7cc 0a1208c9 c61e9c5d
a0403c0a
```

20-byte HMAC(Key, Text):
```
4f4ca3d5 d68ba7cc 0a1208c9 c61e9c5d
a0403c0a
```

A.2 SHA-1 with 20-Byte Key

Text: "Sample #2"

Key: `30313233 34353637 38393a3b 3c3d3e3f`
`40414243`

K_0: `30313233 34353637 38393a3b 3c3d3e3f`
`40414243 00000000 00000000 00000000`
`00000000 00000000 00000000 00000000`
`00000000 00000000 00000000 00000000`

$K_0 \oplus$ ipad:
```
06070405 02030001 0e0f0c0d 0a0b0809
76777475 36363636 36363636 36363636
36363636 36363636 36363636 36363636
36363636 36363636 36363636 36363636
```

(Key $\oplus$ ipad)||text:
```
06070405 02030001 0e0f0c0d 0a0b0809
76777475 36363636 36363636 36363636
36363636 36363636 36363636 36363636
36363636 36363636 36363636 36363636
53616d70 6c652023 32800000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000248
```

Hash((Key ⊕ ipad)||text):
74766e5f 6913e8cb 6f7f108a 11298b15
010c353a

K_0 ⊕ opad:
6c6d6e6f 68696a6b 64656667 60616263
1c1d1e1f 5c5c5c5c 5c5c5c5c 5c5c5c5c
5c5c5c5c 5c5c5c5c 5c5c5c5c 5c5c5c5c
5c5c5c5c 5c5c5c5c 5c5c5c5c 5c5c5c5c

(K_0 ⊕ opad) || Hash((Key ⊕ ipad)||text):
6c6d6e6f 68696a6b 64656667 60616263
1c1d1e1f 5c5c5c5c 5c5c5c5c 5c5c5c5c
5c5c5c5c 5c5c5c5c 5c5c5c5c 5c5c5c5c
5c5c5c5c 5c5c5c5c 5c5c5c5c 5c5c5c5c
74766e5f 6913e8cb 6f7f108a 11298b15
010c353a

HMAC(Key, Text) = Hash((K_0 ⊕ opad) || Hash((Key ⊕ ipad)||text)):
0922d340 5faa3d19 4f82a458 30737d5c
c6c75d24

20-byte HMAC(Key, Text):
0922d340 5faa3d19 4f82a458 30737d5c
c6c75d24

A.3 SHA-1 with 100-Byte Key

Text: "Sample #3"

Key: 50515253 54555657 58595a5b 5c5d5e5f
60616263 64656667 68696a6b 6c6d6e6f
70717273 74757677 78797a7b 7c7d7e7f
80818283 84858687 88898a8b 8c8d8e8f
90919293 94959697 98999a9b 9c9d9e9f
a0a1a2a3 a4a5a6a7 a8a9aaab acadaeaf
b0b1b2b3

Hash(Key):
a4aabe16 54e78da4 40d2a403 015636bf
4bb2f329

K_0: a4aabe16 54e78da4 40d2a403 015636bf
4bb2f329 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000

$K_0 \oplus$ ipad:
929c8820 62d1bb92 76e49235 37600089
7d84c51f 36363636 36363636 36363636
36363636 36363636 36363636 36363636
36363636 36363636 36363636 36363636

(Key $\oplus$ ipad)||text:
929c8820 62d1bb92 76e49235 37600089
7d84c51f 36363636 36363636 36363636
36363636 36363636 36363636 36363636
36363636 36363636 36363636 36363636
53616d70 6c652023 33

Hash((Key $\oplus$ ipad)||text):
d98315c4 2152bea0 d057de97 84427676
2a1a5576

$K_0 \oplus$ opad:
f8f6e24a 08bbd1f8 1c8ef85f 5d0a6ae3
17eeaf75 5c5c5c5c 5c5c5c5c 5c5c5c5c
5c5c5c5c 5c5c5c5c 5c5c5c5c 5c5c5c5c
5c5c5c5c 5c5c5c5c 5c5c5c5c 5c5c5c5c

($K_0 \oplus$ opad) || Hash((Key $\oplus$ ipad)||text):
f8f6e24a 08bbd1f8 1c8ef85f 5d0a6ae3
17eeaf75 5c5c5c5c 5c5c5c5c 5c5c5c5c
5c5c5c5c 5c5c5c5c 5c5c5c5c 5c5c5c5c
5c5c5c5c 5c5c5c5c 5c5c5c5c 5c5c5c5c
d98315c4 2152bea0 d057de97 84427676
2a1a5576

HMAC(Key, Text) = Hash(($K_0 \oplus$ opad) || Hash((Key $\oplus$ ipad)||text)):
bcf41eab 8bb2d802 f3d05caf 7cb092ec
f8d1a3aa
20-byte HMAC(Key, Text):
bcf41eab 8bb2d802 f3d05caf 7cb092ec
f8d1a3aa

A.4 SHA-1 with 49-Byte Key, Truncated to 12-Byte HMAC

Text: "Sample #4"

Key: 70717273 74757677 78797a7b 7c7d7e7f
80818283 84858687 88898a8b 8c8d8e8f
90919293 94959697 98999a9b 9c9d9e9f

a0
K_0: 70717273 74757677 78797a7b 7c7d7e7f
80818283 84858687 88898a8b 8c8d8e8f
90919293 94959697 98999a9b 9c9d9e9f
a0000000 00000000 00000000 00000000

$K_0 \oplus$ ipad:
46474445 42434041 4e4f4c4d 4a4b4849
b6b7b4b5 b2b3b0b1 bebfbcbd babbb8b9
a6a7a4a5 a2a3a0a1 aeafacad aaaba8a9
96363636 36363636 36363636 36363636

(Key $\oplus$ ipad)||text:
46474445 42434041 4e4f4c4d 4a4b4849
b6b7b4b5 b2b3b0b1 bebfbcbd babbb8b9
a6a7a4a5 a2a3a0a1 aeafacad aaaba8a9
96363636 36363636 36363636 36363636
53616d70 6c652023 34

Hash((Key $\oplus$ ipad)||text):
bf1e889d 876c34b7 bef3496e d998c8d1
16673a2e

$K_0 \oplus$ opad:
2c2d2e2f 28292a2b 24252627 20212223
dcdddedf d8d9dadb d4d5d6d7 d0d1d2d3
cccdcecf c8c9cacb c4c5c6c7 c0c1c2c3
fc5c5c5c 5c5c5c5c 5c5c5c5c 5c5c5c5c

($K_0 \oplus$ opad) || Hash((Key $\oplus$ ipad)||text):
2c2d2e2f 28292a2b 24252627 20212223
dcdddedf d8d9dadb d4d5d6d7 d0d1d2d3
cccdcecf c8c9cacb c4c5c6c7 c0c1c2c3
fc5c5c5c 5c5c5c5c 5c5c5c5c 5c5c5c5c
bf1e889d 876c34b7 bef3496e d998c8d1
16673a2e

HMAC(Key, Text) = Hash((K0 ⊕ opad) || Hash((Key ⊕ ipad)||text)):
9ea886ef e268dbec ce420c75 24df32e0
751a2a26

12-byte HMAC(Key, Text):
9ea886ef e268dbec ce420c75

APPENDIX B: A LIMITATION OF MAC ALGORITHMS

The successful verification of a MAC does not completely guarantee that the accompanying message is authentic: there is a chance that a source with no knowledge of the key can present a purported MAC on the plaintext message that will pass the verification procedure. For example, an arbitrary purported MAC of t bits on an arbitrary plaintext message may be successfully verified with an expected probability of $(1/2)_t$. This limitation is inherent in any MAC algorithm.

The limitation is magnified if an application permits a given non-authentic message to be repeatedly presented for verification with different purported MACs. Each individual trial succeeds only with a small probability, $(1/2)_t$; however, for repeated trials, the probability increases that, eventually, one of the MACs will be successfully verified. Similarly, if an application permits a given purported MAC to be presented with different non-authentic messages, then the probability increases that, eventually, the MAC will be successfully verified for one of the messages.

Therefore, in general, if the MAC is truncated, then its length, t, should be chosen as large as is practical, with at least half as many bits as the output block size, L. The minimum value for t is relaxed to 32 bits for applications in which the two types of repeated trials that are described in the previous paragraph are sufficiently restricted. For example, the application, or the protocol that controls the application, may monitor all of the plaintext messages and MACs that are presented for verification, and permanently reject any plaintext message or any MAC that is included in too many unsuccessful trials. Another example occurs when the bandwidth of the communications channel is low enough to preclude too many trials, of either type. In both cases, the maximum number of allowed unsuccessful trails must be pre-determined based on the risks associated with the sensitivity of the data, the length of t and the MAC algorithm used.

APPENDIX C: REFERENCES

[1] American Bankers Association, *Keyed Hash Message Authentication Code*, ANSI X9.71, Washington, D.C., 2000.

[2] National Institute of Standards and Technology, *Security Requirements for Cryptographic Modules*, Federal Information Processing Standards Publication 140-2, May 25, 2001.

[3] H. Krawczyk, M. Bellare, and R. Canetti, *HMAC: Keyed-Hashing for Message Authentication*, Internet Engineering Task Force, Request for Comments (RFC) 2104, February 1997.

[4] National Institute of Standards and Technology, *Secure Hash Standard (SHS)*, Federal Information processing Standards Publication 180-1, 17 April 1995.

APPENDIX #2: LIST OF ACRONYMS

Source: NISTIR 7628 Guidelines for Smart Grid Cyber Security v1.0, Vol. 3.– Aug 2010

3DES	Triple Data Encryption Standard (168 Bit)
AAA	Authentication, Authorization, and Accounting
Active Directory:	A technology created by Microsoft that provides a variety of network services and is a central component of the Windows Server platform. The directory service provides the means to manage the identities and relationships that make up network environments.
ADEPT	Agile Delivery of Electrical Power Technology
AEAD	Authenticated Encryption with Associated Data
AEP	American Electric Power
AES	Advanced Encryption Standard
AGA	American Gas Association
AGC	Automatic Generation Control. A standalone subsystem that regulates the power output of electric generators within a prescribed area in response to changes in system frequency, tie-line loading, and the relation of these to each other. This maintains the scheduled system frequency and established interchange with other areas within predetermined limits.

Aggregation	Practice of summarizing certain data and presenting it as a total without any PII identifiers
AICPA	American Institute of Certified Public Accountants. The national, professional organization for all Certified Public Accountants.
AMI	Advanced Metering Infrastructure
AMI-SEC	AMI Security [Task Force]
AMR	Automatic Meter Reading
Anonymize	To organize data in such a way as to preserve the anonymity or hide the personal identity of the individual(s) to whom the data pertains and also a process of transformation or elimination of PII for purposes of sharing data
ANSI	American National Standards Institute
API	Application Programming Interface
ASAP-SG	Advanced Security Acceleration Project – Smart Grid
ASTM	American Society for Testing and Materials
Asymmetric cipher	Cryptography solution in which separate keys are used for encryption and decryption, where one key is public and the other is private.
ATR	Attribute
B2B	Business to Business
BAN	Building Area Network
BEM	Building Energy Management

Block cipher	A symmetric key cipher operating on fixed-length groups of bits, called blocks, with an unvarying transformation—in contrast to a stream cipher, which operates on individual digits one at a time and whose transformation varies during the encryption. A block cipher, however, can effectively act as a stream cipher when used in certain modes of operation.
Botnet	Robot Network. A large number of compromised computers also called a "zombie army," that can be used to flood a network with messages as a denial of service attack. A thriving botnet business consists in selling lists of compromised computers to hackers and spammers.
C&I	Commercial and Industrial
CA	Certificate Authority
CALEA	Communications Assistance for Law Enforcement Act
CAN-SPAM	Controlling the Assault of Non-Solicited Pornography and Marketing
CBC	Cipher Block Chaining
CEC	California Energy Commission
CEO	Chief Executive Officer
CFO	Chief Financial Officer
CHP	Combined Heat and Power
CI&A	Confidentiality, Integrity, and Availability
CIM	Common Information Model. A structured set of definitions that allow different Smart Grid domain representatives to communicate important concepts and exchange information easily and effectively.

CIMA	Chartered Institute of Management Accountants
CIO	Chief Information Officer
CIP	Critical Infrastructure Protection
CIPA	Children's Internet Protection Act
CIS	Cryptographic Interoperability Strategy
CIS	Customer Information System
CISO	Chief Information Security Officer
CMMS	Computer-based Maintenance Management Systems
COTS	Commercial Off-the-Shelf
CPU	Central Processing Unit
CRL	Certificate Revocation List
CSCTG	Cyber Security Coordination Task Group
CSO	Chief Security Officer
CSP	Critical Security Parameters
CSR	Certificate Signing Request
CSR	Customer Service Representative
CSSWG	Control Systems Security Working Group
CSWG	Cyber Security Working Group
CTR mode	Counter mode. A block cipher mode of operation also known as Integer Counter Mode (ICM) and Segmented Integer Counter (SIC) mode.
CVE	Common Vulnerabilities and Exposures
CWE	Common Weakness Enumeration
DA	Distribution Automation

DARPA	Defense Advanced Research Projects Agency
DCS	Distributed Control System. A computer-based control system where several sections within the plants have their own processors, linked together to provide both information dissemination and manufacturing coordination.
DDoS	Distributed Denial of Service
De-identify	A form of anonymization that does not attempt to control the data once it has had PII identifiers removed, so it is at risk of re-identification.
DER	Distributed Energy Resources
DES	Data Encryption Standard
DEWG	Domain Expert Working Group
DFR	Digital Fault Recorder
DGM	Distribution Grid Management
DHS	Department of Homeland Security
Diffie-Hellman	A cryptographic key exchange protocol first published by Whitfield Diffie and Martin Hellman in 1976. It allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel.
Distinguished names	String representations that uniquely identify users, systems, and organizations.
DMS	Distribution Management System
DN	Distinguished Name
DNP	Distributed Network Protocol

DNS	Domain Name Service
DoD	Department of Defense
DOE	Department of Energy
DoS	Denial of Service
DR	Demand Response
DRBG	Deterministic Random Bit Generators
DRM	Digital Rights Management. A generic term for access control technologies used by standards providers, publishers, copyright holders, manufacturers, etc. to impose limitations on the usage of digital content and devices. The term is used to describe any technology that inhibits the use of digital content in a manner not desired or intended by the content provider.
DRMS	Distribution Resource Management System
DSL	Digital Subscriber Line
DSPF	Distribution System Power Flow
DSS	Digital Signature Standard
EAP	Extensible Authentication Protocol
EAX mode	A mode of operation for cryptographic block ciphers. It is an AEAD algorithm designed to simultaneously provide both authentication and privacy of the message with a two-pass scheme, one pass for achieving privacy and one for authenticity for each block; and also a mixed authenticated encryption mode of operation of a block cipher in order to reduce the area overhead required by traditional authentication schemes.

EAX’	A modification of the EAX mode used in the ANSI C12.22 standard for transport of meter-based data over a network.
ECC	Elliptic Curve Cryptography (encryption)
ECDH	Elliptic Curve Diffie-Hellman. A key agreement protocol that allows two parties, each having an elliptic curve public-private key pair, to establish a shared secret over an insecure channel.
ECDSA	Elliptic Curve Digital Signature Algorithm
ECPA	Electronic Communications Privacy Act
EEPROM	Electrically Erasable Programmable Read-Only Memory
EISA	Energy Independence and Security Act
EKU	Extended Key Usage
EMS	Energy Management System
EMSK	Extended Master Session Key
End-to-End Trust	(E2E Trust) Cryptographic means of authentication at each end-point and also seamless security across all the protocol layers and routers, proxies, etc. between user interfaces and/or other devices.
Entropy	In the case of transmitted messages, a measure of the amount of information that is missing before reception.
Ephemeral Unified Model	A ECDH scheme where each party generates an ephemeral key pair to be used in the computation of the shared secret.
EPIC	Electronic Privacy Information Center
EPRI	Electric Power Research Institute

EPSA	Electric Power Supply Association
ES	Electric Storage
ESI	Energy Services Interface
ESP	Energy Service Provider
ET	Electric Transportation
EUMD	End Use Measurement Device
EV	Electric Vehicle
EV/PHEV	Electric Vehicle/Plug-in Hybrid Electric Vehicles. Cars or other vehicles that draw electricity from batteries to power an electric motor. PHEVs also contain an internal combustion engine.
EvDO	Evolution Data Optimized
EVSE	Electric Vehicle Service Element
FACTA	Fair and Accurate Credit Transactions Act
FAQ	Frequently Asked Questions
FERC	Federal Energy Regulatory Commission
FERPA	Family Educational Rights and Privacy Act
FIPS	Federal Information Processing Standards
FIPS 140-2	Publication 140-2 is a U.S. government computer security standard used to accredit cryptographic modules. NIST issued the FIPS 140 Publication Series to coordinate the requirements and standards for cryptography modules that include both hardware and software components.
FLIR	Fault Location, Isolation, Restoration
FTP	File Transfer Protocol

G&T	Generations and Transmission
GAPP	Generally Accepted Privacy Principles. Privacy principles and criteria developed and updated by the AICPA and Canadian Institute of Chartered Accountants to assist organizations in the design and implementation of sound privacy practices and policies.
GIC	Group Insurance Commission
GIS	Geographic Information System
GLBA	Gramm-Leach Bliley Act
GPRS	General Packet Radio Service
GPSK	Generalized Pre-Shared Key
Granularity	The extent to which a system contains separate components, e.g., the fineness or coarseness with which data fields are subdivided in data collection, transmission, and storage systems. The more components in a system, the more flexible it is. In more general terms, the degree to which a volume of information is finely detailed.
GRC	Governance, Risk, and Compliance
GRIDS	Grid-Scale Rampable Intermittent Dispatchable Storage
GWAC	GridWise Architecture Council

Hacker	In common usage, a hacker is a person who breaks into computers and/or computer networks, usually by gaining access to administrative controls. Proponents may be motivated by diverse objectives from the shear entertainment value they find in the challenge of circumventing computer/network security to political or other ends. Hackers are often unconcerned about the use of illegal means to achieve their ends. Out-and-out cyber-criminal hackers are often referred to as "crackers."
HAN	Home Area Network. A network of energy management devices, digital consumer electronics, signal-controlled or -enabled appliances, and applications within a home environment that is on the home side of the electric meter.
Hash	Any well-defined procedure or mathematical function that converts a large, possibly variable-sized amount of data into a small datum, usually a single integer that may serve as an index to an array. The values returned by a hash function are called hash values, hash codes, hash sums, checksums, or simply hashes.
HIPAA	Health Insurance Portability and Accountability Act
HITECH	Health Information Technology for Economic and Clinical Health
HMAC	Hash Message Authentication Code
HSM	Hardware Security Module
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
Hz	hertz

IBE	Identity-Based Encryption
ICS	Industrial Control Systems
ID	Identification
IDS	Intrusion Detection System
IEC	International Electrotechnical Commission
IED	Intelligent Electronic Device
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IFAC	International Federation of Accountants
IKE	Internet Key Exchange. Protocol used to set up a security association in the IPsec protocol suite.
INL	Idaho National Laboratory
IP	Internet Protocol
IPP	Independent Power Producer
IPR	Intellectual Property Rights
IPS	Intrusion Prevention System
IPSec	Internet Protocol Security
IRTF	Internet Research Task Force
IS	Information Security
ISA	International Society of Automation
ISAKMP	Internet Security Association and Key Management Protocol
ISMS	Information Security Management System
ISO	International Organization for Standardization

ISO	Independent System Operator
ISO/IEC27001	International Organization for Standardization/International Electrotechnical Commission Standard 27001. A auditable international standard that specifies the requirements for establishing, implementing, operating, monitoring, reviewing, maintaining and improving a documented Information Security Management System within the context of the organization's overall business risks. It uses a process approach for protection of critical information.
IT	Information Technology
ITGI	IT Governance Institute
ITL	Information Technology Laboratory
IVR	Interactive Voice Response
JNI	Java Native Interface
JTC	Joint Technical Committee
KDC	Key Distribution Center
KEK	Key Encryption Key
Kerberos	A computer network authentication protocol, developed by the Massachusetts Institute of Technology, which allows nodes communicating over a nonsecure network to prove their identity to one another in a secure manner. It is also a suite of free software published by MIT that implements this protocol.
LAN	Local Area Network

LATENCY	Average service execution time, for example, the time duration to complete an FTP file transfer (Blake, 2003)
LDAP	Lightweight Directory Access Protocol
LMS	Load Management System
LTC	Load Tap Changer
MAC	Message Authentication Code
MAC address	Media Access Control address. The unique serial number burned into Ethernet and Token Ring adapters that identifies that network card from all others.
MAC protection	Message Authentication Code protection. In cryptography, a short piece of information used to authenticate a message. The MAC value protects data integrity and authenticity of the tagged message by allowing verifiers (who also possess the secret key used to generate the value) to detect any changes to the message content.
MDMS	Meter Data Management System
min	minute
MIT	Massachusetts Institute of Technology
MITM	"Man in the Middle" type of hacker attack
ms	millisecond (10-3 second)
MTBF	Mean Time Before Failure
MW	megawatt
NAN	Neighborhood Area Network
NERC	North American Electric Reliability Corporation

NIPP	National Infrastructure Protection Plan
NIST	National Institute of Standards and Technology
NISTIR	NIST Interagency Report
NMAP	Networked Messaging Application Protocol
NRECA	National Rural Electric Cooperative Association
NSA	National Security Agency
NSA Suite B	A set of cryptographic algorithms promulgated by the National Security Agency to serve as an interoperable cryptographic base for both unclassified information and most classified information.
NSF	National Science Foundation
NSTC	National Science and Technology Council
NVD	National Vulnerability Database
OCSP	Online Certificate Status Protocol
OE	Office of Electricity Delivery and Energy Reliability
OECD	Organization for Economic Cooperation and Development. A global governmental forum of 30+ market democracies for comparison of policy experiences, good practices, and coordination of domestic and international policies. It is one of the world's largest and most reliable sources of comparable statistical, economic and social data.
OID	Object Identifier
OMS	Outage Management System

One-Pass Diffie-Hellman:	A key-agreement scheme in which an ephemeral key pair generated by one party is used together with the other party's static key pair in the computation of the shared secret.
OWASP	Open Web Application Security Project
PANA	Protocol for carrying Authentication for Network Access
PAP	Priority Action Plan
PC	Personal Computer
PDA	Personal Digital Assistant
PDC	Phasor Data Concentrator
PE	Protocol Encryption
PE mode	An encryption mode combining CTR mode and ECB mode developed for streaming SCADA messages. It relies on the SCADA protocol's ability to detect incorrect SCADA messages; and also Position Embedding mode: A cryptographic mode designed specifically for low-latency integrity protection on low-speed serial links.
Personal Information	Information that reveals details, either explicitly or implicitly, about a specific individual's household dwelling or other type of premises. This is expanded beyond the normal "individual" component because there are serious privacy impacts for all individuals living in one dwelling or premise. This can include items such as energy use patterns or other types of activities. The pattern can become unique to a household or premises just as a fingerprint or DNA is unique to an individual.

PEV	Plug-In Electric Vehicle
PFS	Perfect Forward Secrecy
PHEV	Plug In Hybrid Electric Vehicle
PIA	Privacy Impact Assessment. A process used to evaluate the possible privacy risks to personal information, in all forms, collected, transmitted, shared, stored, disposed of, and accessed in any other way, along with the mitigation of those risks at the beginning of and throughout the life cycle of the associated process, program or system.
PII	Personally Identifiable Information
PKCS	Public-Key Cryptography Standards
PKI	Public Key Infrastructure
PKMv2	Privacy Key Management version 2
PLC	Programmable Logic Controller
PMU	Phasor Measurement Unit
POTS	Plain Old Telephone Service
PPP	Point-to-Point Protocol
PQ	Power Quality
Public-key cryptography:	A cryptographic approach that involves the use of asymmetric key algorithms instead of or in addition to symmetric key algorithms. Unlike symmetric key algorithms, it does not require a secure initial exchange of one or more secret keys to both sender and receiver.
PUC	Public Utilities Commission

QoS	Quality of Service
R&D	Research and Development
RA	Registration Authority
RADIUS	Remote Authentication Dial In User Service
RAM	Random Access Memory
RBAC	Role-Based Access Control
Retail Access	Competitive retail or market-based pricing offered by energy services companies or utilities to some or all of their customers under the approval/regulation of state public utilities departments.
RF	Radio Frequency
RFC	Request for Comments
RNG	Random Number Generator
RP	Relying Party
RSA	Widely used in electronic commerce protocols, this algorithm for public-key cryptography is named for Rivest, Shamir, and Adleman who were first to publicly describe it. This was the first algorithm known to be suitable for signing as well as encryption and represents a great advance in public key cryptography.
RSA algorithm	RSA is public key cryptography algorithm named for its co-inventors: Ron Rivest, Adi Shamir, and Len Adleman.
RTO	Regional Transmission Operator
RTP	Real-Time Pricing
RTU	Remote Terminal Unit

s	second
S/MIME	Secure/Multipurpose Internet Mail Extensions
SA	Security Association
SAM	Security Authentication Module
SCADA	Supervisory Control and Data Acquisition
SCE	Southern California Edison
SDLC	Software Development Life Cycle
SDO	Standard Developing Organization
SEL	Schweitzer Engineering Laboratories
SEM	Security Event Management
SEP	Smart Energy Profile
SGIP	Smart Grid Interoperability Panel
SGIP TWiki	An open collaboration site for the Smart Grid community to work with NIST in developing a framework that includes protocols and model standards for information management to achieve interoperability of Smart Grid devices and systems and is part of a robust process for continued development and implementation of standards as needs and opportunities arise and as technology advances.
SGIP-CSWG	SGIP – Cyber Security Working Group
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard

Single sign-on	A property of access control of multiple, related, but independent software systems. With this property a user/device logs in once and gains access to all related systems without being prompted to log in again at each of them.
SNMP	Simple Network Management Protocol
Social Engineering	The act of manipulating people into performing actions or divulging confidential information. The term typically applies to trickery or deception being used for purposes of information gathering, fraud, or computer system access.
SP	Special Publication
SPOF	Signal Point of Failure
SSH	Secure Shell. A protocol for secure remote login and other secure network services over an insecure network.
SSID	Service Set Identifier
SSL	Secure Socket Layer
SSL/TLS	Secure Socket Layer / Transport Layer Security
SSN	Social Security Number
SSO	Single Sign-On
SSP	Sector-specific Plans

Strong Authentication	A receiver of a message should be able to determine the origin of the message. This implies that no attacker should be able to send a message with forged source information (Gurtov, 2008). The advantage of using the HIT (Host Identity Tag) versus an IP address in the application is the concept of *channel binding*. The calling application is bound to the cryptographic host name and the ESP (or TLS) tunnel created by HIP. Therefore, either an application connect() call connects to a host owning the private key corresponding to the HIT, or the call fails.
Symmetric cipher	Cryptography solution in which both parties use the same key for encryption and decryption, hence the encryption key must be shared between the two parties before any messages can be decrypted.
T&D	Transmission and Distribution
T&D DEWG	T&D Domain Expert Working Group
TA	Trust Anchor
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol / Internet Protocol
TCPA	Telephone Consumer Protection Act
TCS	Trouble Call System
Telnet	Teletype network. A network protocol used on the Internet or local area networks to provide a bidirectional interactive communications facility. The term telnet may also refer to the software that implements the client part of the protocol.

TEMPEST	A codename referring to investigations and studies of conducted emissions. Compromising emanations are defined as unintentional intelligence-bearing signals which, if intercepted and analyzed, may disclose the information transmitted, received, handled, or otherwise processed by any information-processing equipment.
TLS	Transport Layer Security
TNC	Trusted Network Connect
TOCTOU	Time of Check, Time of Use
TPI	Two-Person Integrity
TRSM	Tamper Resistant Security Modules
Trust anchor	In cryptography, an authoritative entity represented via a public key and associated data. When there is a chain of trust, usually the top entity to be trusted becomes the trust anchor. The public key (of the trust anchor) is used to verify digital signatures and the associated data.
TWiki	A flexible, open source collaboration and Web application platform (i.e., a structured Wiki) typically used to run a project development space, a document management system, a knowledge base, or any other groupware tool on an intranet, extranet, or the Internet to foster information flow between members of a distributed work group.
UCAIug	UtiliSec Working Group
UDP/IP	User Datagram Protocol/Internet Protocol
Upsell	Marketing term for the practice of suggesting higher priced products or services to a customer who is considering a purchase.

URL	Universal Resource Locator
USRK	Usage-Specific Root Key
Van Eck phreaking	Named after Dutch computer researcher Wim van Eck, phreaking is the process of eavesdropping on the contents of a CRT and LCD display by detecting its electromagnetic emissions. Because of its connection to eavesdropping, the term is also applied to exploiting telephone networks.
VAR	Volts-Amps-Reactive
VLAN	Virtual Local Area Network
VPN	Virtual Private Network
WAMS	Wide Area Measurement System
WAN	Wide Area Network
WASA	Wide Area Situational Awareness
WG	Working Group
Wi-Fi	Term often used as a synonym for IEEE 802.11 technology. Wi-Fi is a trademark of the Wi-Fi Alliance that may be used with certified products that belong to a class of WLAN devices based on the IEEE 802.11 standards.
WiMAX	Worldwide Interoperability for Microwave Access. A telecommunications protocol that provides fixed and fully mobile Internet access; and also a Wireless digital communications system, also known as IEEE 802.16, which is intended for wireless "metropolitan area networks."
WLAN	Wireless Local Area Network
WMS	Work Management System

XML	Extensible Markup Language

APPENDIX #3: ADDITIONAL STATISTICAL ANALYSIS OUTPUT

Summary of Results

As stated earlier in Chapter 4, if all combinations of FTP file transfers with/without the HIT Tags and with/without the different types of encryption and key lengths are entered into the multivariate ANOVA analysis, the results are as follows:

Parameter Estimates

Dependent Variable	Parameter	B	Std. Error	t	Sig.	95% Confidence Interval		Partial Eta Squared	Noncent. Parameter	Observed Power[a]
						Lower Bound	Upper Bound			
Latency of File Xfr w/ TLSauth	Intercept	.090	.076	1.185	.242	-.063	.242	.028	1.185	.213
	Payload	.949	.002	431.010	.000	.944	.953	1.000	431.010	1.000
Latency of File Xfr w/o TLSauth	Intercept	-.028	.114	-.241	.810	-.257	.202	.001	.241	.056
	Payload	.935	.003	282.648	.000	.928	.942	.999	282.648	1.000
Latency of File Xfr w/ TLSauth & AES128	Intercept	.136	.061	2.213	.032	.012	.259	.093	2.213	.583
	Payload	1.011	.002	568.290	.000	1.007	1.014	1.000	568.290	1.000
Latency of File Xfr wo/ TLSauth & w/AES128	Intercept	.087	.083	1.056	.296	-.079	.253	.023	1.056	.179
	Payload	.957	.002	399.072	.000	.952	.962	1.000	399.072	1.000
Latency of File Xfr w/ TLSauth & AES256	Intercept	.040	.129	.310	.758	-.220	.300	.002	.310	.061
	Payload	1.016	.004	271.242	.000	1.009	1.024	.999	271.242	1.000
Latency of File Xfr wo/ TLSauth & w/AES256	Intercept	.001	.053	.010	.992	-.105	.106	.000	.010	.050
	Payload	.956	.002	626.614	.000	.953	.959	1.000	626.614	1.000
Latency of File Xfr w/ TLSauth & BF128	Intercept	.160	.095	1.680	.099	-.031	.351	.056	1.680	.377
	Payload	.960	.003	348.255	.000	.955	.966	1.000	348.255	1.000
Latency of File Xfr w/o TLSauth & w/BF128	Intercept	.033	.090	.372	.712	-.147	.214	.003	.372	.065
	Payload	.948	.003	363.739	.000	.942	.953	1.000	363.739	1.000
Latency of File Xfr w/ TLSauth & BF256	Intercept	.150	.114	1.315	.195	-.079	.379	.035	1.315	.252
	Payload	.961	.003	290.238	.000	.954	.967	.999	290.238	1.000
Latency of File Xfr w/o TLSauth & w/BF256	Intercept	.024	.052	.462	.646	-.080	.128	.004	.462	.074
	Payload	.944	.001	629.801	.000	.941	.947	1.000	629.801	1.000

a. Computed using alpha = .05

Repeat of Figure 28 Overall Multivariate ANOVA Parameter Estimates for all cases with/without TLS-auth (i.e. HIT Tags) and with/without all AES and Blowfish Encryption and Key Lengths.

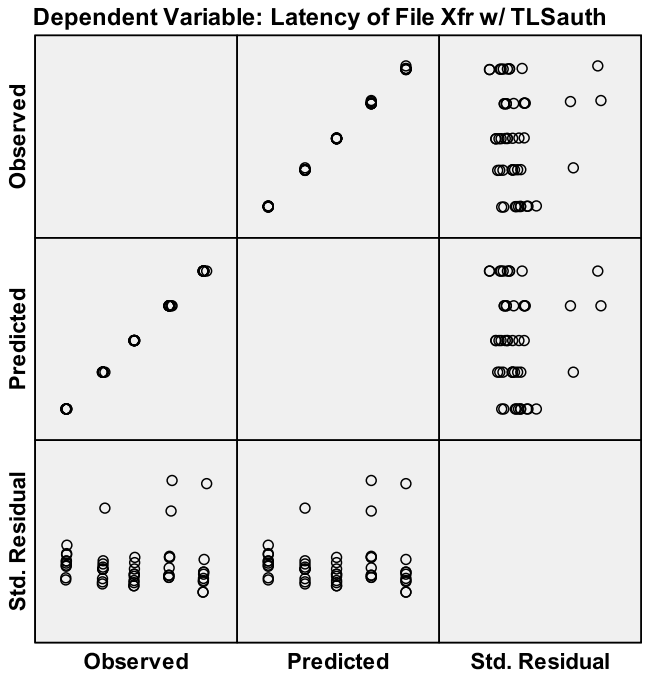
A summary of the additional latency for all cases with/without TLS-auth (i.e. HIT Tags) for all AES and Blowfish Encryption and Key Lengths is displayed again in Figure 29 below:

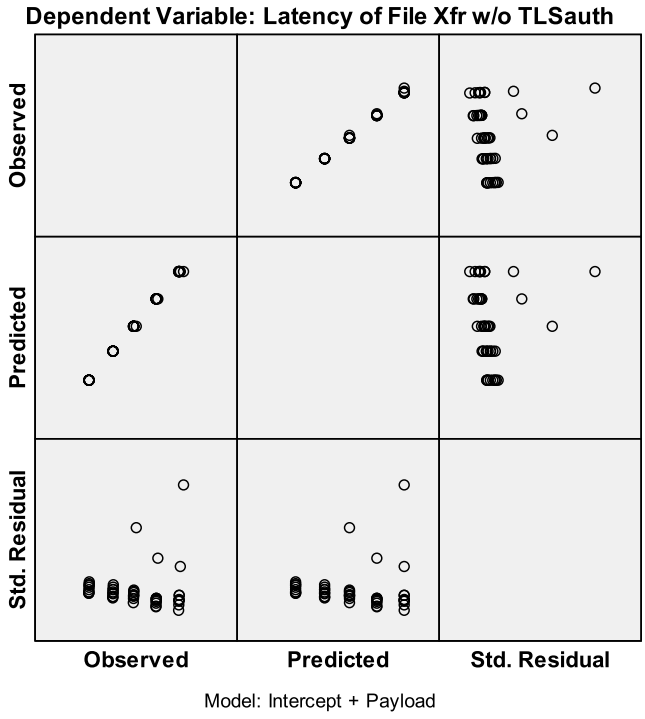
File Transfer w/OpenVPN Tunnel	Approx. Addl Latency with vs. w/o HIT Tag
FTP file transfer only	1.5%
FTP file transfer with AES w/128-bit key (not including intercept constant, C = 0.136)	5.6%
FTP file transfer with AES w/256-bit key	6.2%
FTP file transfer with Blowfish w/128-bit key	1.3%
FTP file transfer with Blowfish w/256-bit key	1.8%

Repeat of Figure 29 Additional Latency predicted by the multivariate ANOVA analysis for FTP file transfers for all cases with/without TLS-auth (i.e. HIT Tags).

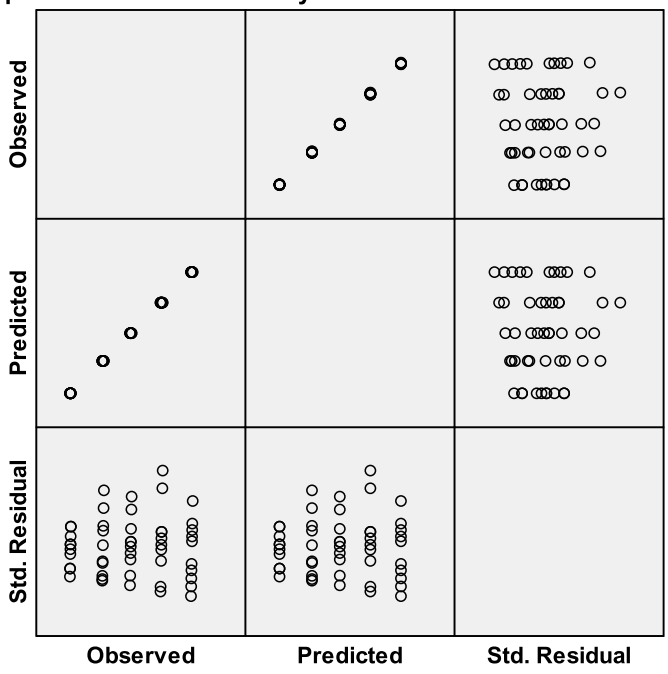
In summary, the dissertation provides a tested solution and recommendations to the main research problem, namely: the need for low-latency across local and remote SmartGrid network nodes in order to transmit automation control parameters that achieve acceptable levels of performance, security and reliability using an open technology framework. This dissertation supports the use of the OpenVPN TLS-auth capability with HIT Tags as one possible means for the Smart Grid to securely and reliably transmit automation control parameters with relatively low-latency.

The plots of the residuals are included below and indicate similar good randomness and independence with a lack of specific patterns as observed before in Chapter 4. The data is very consistent in measuring the latency of file transfers in the transmissions.


Dependent Variable: Latency of File Xfr w/ TLSauth
Observed
Predicted
Std. Residual
Observed
Predicted
Std. Residual
Model: Intercept + Payload

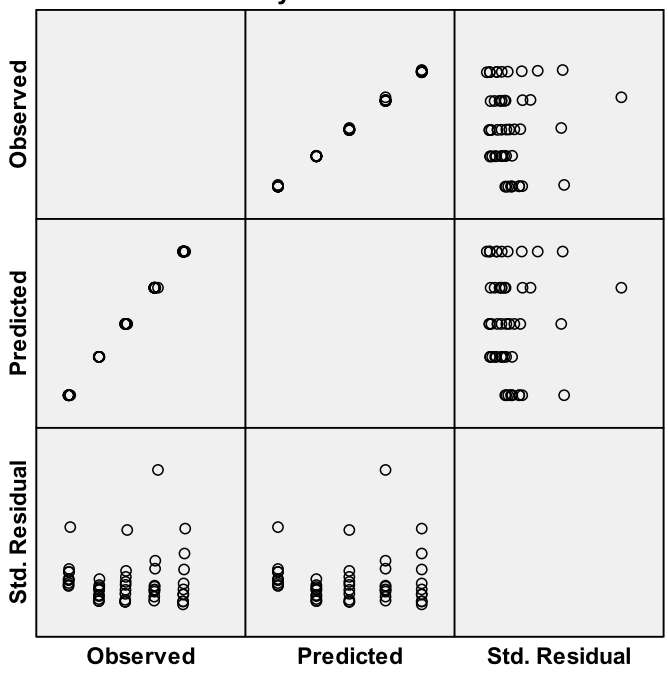

Dependent Variable: Latency of File Xfr w/o TLSauth
Observed
Predicted
Std. Residual
Observed
Predicted
Std. Residual
Model: Intercept + Payload

Dependent Variable: Latency of File Xfr w/ TLSauth & AES128



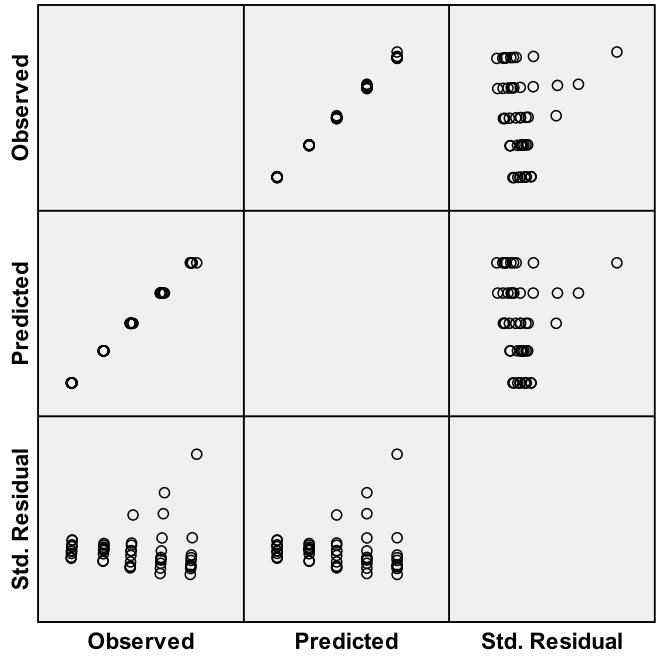
Model: Intercept + Payload

Dependent Variable: Latency of File Xfr wo/ TLSauth & w/AES128



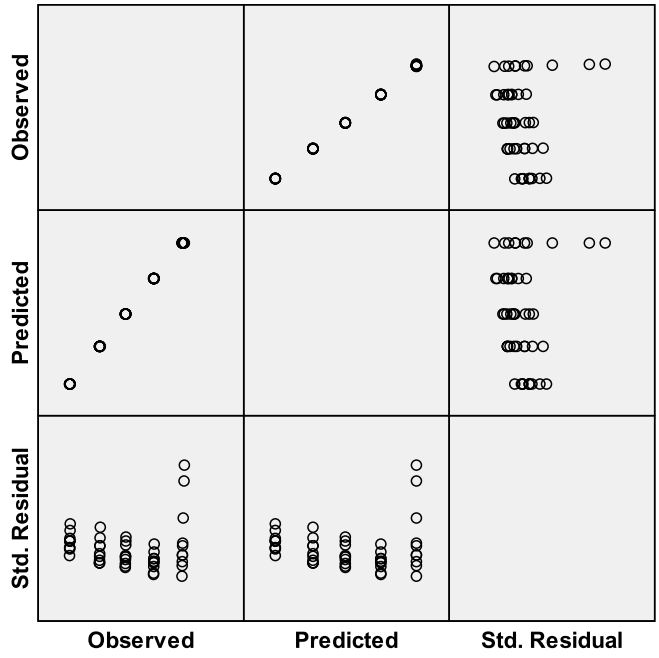
Model: Intercept + Payload

Dependent Variable: Latency of File Xfr w/ TLSauth & AES256



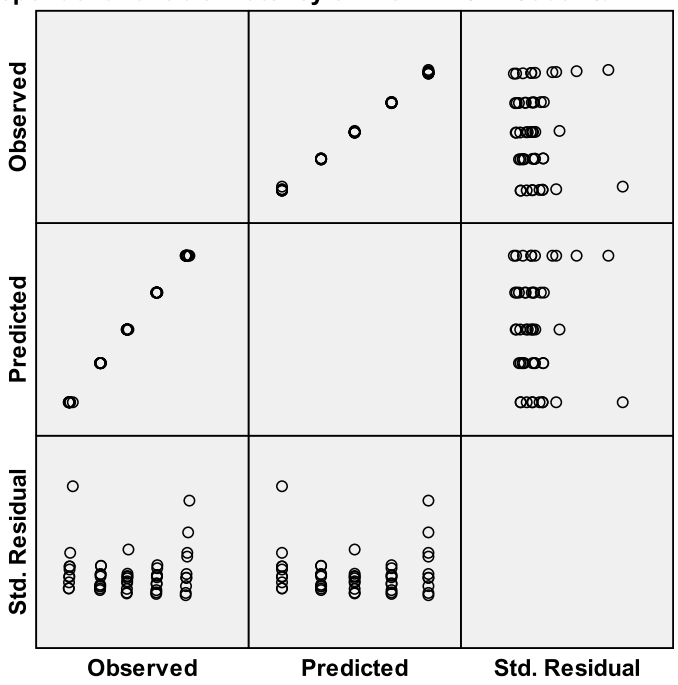
Model: Intercept + Payload

Dependent Variable: Latency of File Xfr wo/ TLSauth & w/AES256



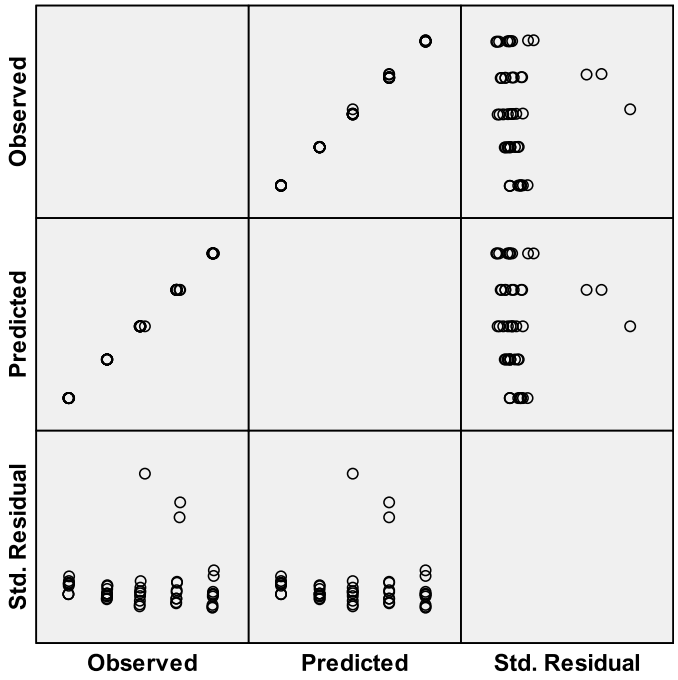
Model: Intercept + Payload

Dependent Variable: Latency of File Xfr w/ TLSauth & BF128



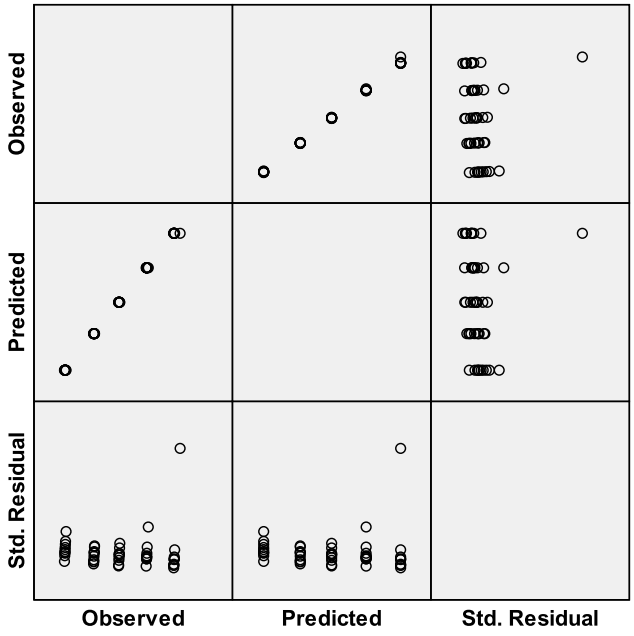
Model: Intercept + Payload

Dependent Variable: Latency of File Xfr w/o TLSauth & w/BF128



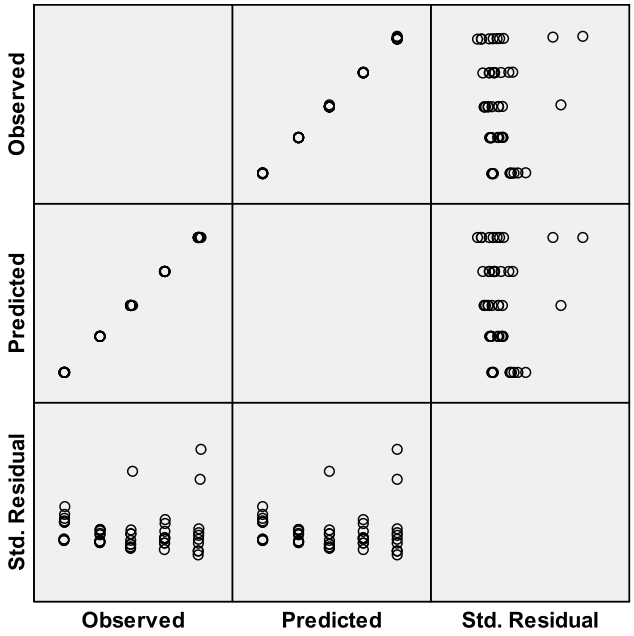
Model: Intercept + Payload

Dependent Variable: Latency of File Xfr w/ TLSauth & BF256



Model: Intercept + Payload

Dependent Variable: Latency of File Xfr w/o TLSauth & w/BF256



Model: Intercept + Payload

Experimental Data

Payload	LatencywTLSauth	LatencywoTLSauth	LatencywTLSauthandAES128	LatencywoTLSauthandwAES128
51.74	49.05	48.75	52.52	49.73
51.74	49.02	48.02	52.22	49.48
51.74	49.95	48.2	52.17	49.59
51.74	49.09	48.27	52.31	50.14
51.74	49.11	48.27	52.45	49.38
51.74	48.92	48.17	52.11	49.41
51.74	49.23	48.19	52.7	49.48
51.74	48.92	48.19	52.48	49.41
51.74	49.03	50.11	52.27	49.53
51.74	49.11	48.11	52.56	49.89
41.4	39.34	38.55	42.06	39.84
41.4	39.27	38.42	42.33	40.83
41.4	39.44	38.55	41.72	39.67
41.4	39.25	38.5	42.06	39.92
41.4	40.17	38.41	41.69	39.66
41.4	39.25	38.55	42.02	39.63
41.4	39.44	38.53	41.95	39.56
41.4	39.88	39.22	41.88	39.52
41.4	39.45	38.52	41.98	39.63
41.4	39.27	38.47	42.44	39.61
31.06	29.47	28.92	31.52	29.92
31.06	29.36	28.88	31.44	29.63
31.06	29.36	28.92	31.55	29.69
31.06	29.63	29	31.28	29.81
31.06	29.39	30.06	31.83	29.61
31.06	29.52	29.02	31.63	30.33
31.06	29.41	28.98	31.55	29.78
31.06	29.58	28.81	31.75	29.86
31.06	29.45	28.94	31.34	29.73
31.06	29.47	28.94	31.48	29.61
21.65	20.67	20.27	22.25	20.75
21.65	20.59	20.09	21.81	20.61
21.65	21.17	20.19	22.14	20.77
21.65	20.59	20.23	21.83	20.83
21.65	20.45	20.31	21.92	20.63
21.65	20.59	20.16	21.91	20.67
21.65	20.64	20.17	22.02	20.72
21.65	20.5	20.11	22.11	20.66
21.65	20.61	20.23	22.36	20.73
21.65	20.47	20.11	21.8	20.61
10.69	10.23	9.98	11.06	10.86
10.69	10.11	10.06	10.8	10.3
10.69	10.22	9.94	11	10.27
10.69	10.27	10.05	11.06	10.33
10.69	10.42	9.92	10.8	10.33
10.69	10.09	10.02	10.92	10.28
10.69	10.33	9.94	10.89	10.34
10.69	10.34	10.08	10.95	10.41
10.69	10.25	9.92	10.75	10.41
10.69	10.27	10.11	10.95	10.44