

https://dspace.org/

Whether using encryption in SCADA systems, the services performance requirements are still met in OT IT environment over an MPLS core network?

Chego, Lloyd

2016

http://hdl.handle.net/10539/21050

Chego, Lloyd (2016) Whether using encryption in SCADA systems, the services performance requirements are still met in OT IT environment over an MPLS core network?, University of Witwatersrand, Johannesburg, <http://wiredspace.wits.ac.za/handle/10539/21050>

Downloaded from a DSpace repository. https://dspace.org/

Whether using encryption in SCADA systems, the services performance requirements are still met in OT IT environment over an MPLS core network?

A Research Project Thesis Final Submission

submitted in fulfillment of the requirements

for

Master of Science in Engineering [Electrical]: Telecommunications

at the

University Of The Witwatersrand, Johannesburg

by

Lloyd Chego: 592504



07 June 2016

Supervisor: Prof. Rex van Olst

Declaration

I Lloyd Ntoeng Chego, hereby declare that this Thesis is my own, unaided work. It is being submitted for the degree of Masters of Science at the University of Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other university. Where applicable I have provided the necessary references according to the University of Witwatersrand's policies.



Lloyd N. Chego

07 June 2016

Abstract

Utilities use Supervisory Control and Data Acquisition systems as their industrial control system. The architecture of these systems in the past was based on them being isolated from other networks. Now with recent ever changing requirements of capabilities from these systems there is a need to converge with information technology systems and with the need to have these industrial networks communicating on packet switched networks there are cyber security concerns that come up.

This research project looks at the whether using encryption in an IP/MPLS core network for SCADA in an OT IT environment has an effect on the performance requirements. This was done through an experimental simulation with the results recorded. The research project also looks at the key literature study considerations.

Dedication

This thesis is dedicated to my family, friends and South Africans at large to increase the knowledge footprint of our nation for the advancement of our country.

Acknowledgements

First and foremost I would like acknowledge God the father of our Lord Jesus our savior whom without I am nothing. I would like to also acknowledge my supervisor, Professor Rex van Olst for dedicating his time guiding me throughout this research. I would like to acknowledge and dedicate a special thanks to my wife, Vanessa Chego for the continued support and encouragement throughout this work. Furthermore acknowledging the motivation brought about by my son, Ata Chego. I would like to thank my parents, Collie and Onica Chego and my siblings, Lucia Letoaba, Lorraine, Letlhogonolo and Kabelo Chego for the continued support.

Lastly I would like to thank my friends and colleagues, particularly Mr. Tatana Mabasa, Eric Mabotja, who assisted me greatly with the experimental setup.

Table of Contents

Declaration i
Abstract ii
Dedication iii
Acknowledgements iv
List of Figures x
List of Tables xiv
Abbreviations xv
CHAPTER 1 14
1. Introduction 14
1.1. Background 14
1.2. Key Research Question 14
1.3. Research Objective 14
1.4. Methodology 14
1.5. SCADA and RTU setting 15
1.6. IP/MPLS Network 15
1.7. SCADA Encryption and Decryption Implementation 16
1.8. Final Test bench setup 16
1.9. Expected Outcomes 17
1.10. Conclusion 17
CHAPTER 2 19
2. Supervisory Control And Data Acquisition Systems (SCADA) 19
2.1. Introduction 19
2.2. What is SCADA 19
2.3. How SCADA works 20
2.4. Supervisory Control And Data Acquisition (SCADA) Main Components 20
2.4.1. Human Machine Interface 20
2.4.2. Supervisory Control And Data Acquisition Protocols 21
2.4.3. Field Devices 21
2.4.4. Remote Terminal Unit 22
2.4.5. Supervisory Control And Data Acquisition Communications Network 22
2.5. Performance requirements of SCADA systems in OT/IT 22
2.5.1. Supervisory Control And Data Acquisition Requirements 22
2.5.2. Management Station (DMS) 23

2.5.3. Telecommunications Management Centre and Security Centre 23
2.5.4. Future SCADA links will be configured to eliminate sub-multiplexing. 23
2.6. Conclusion 24
CHAPTER 3 25
3. Operational Technology Information Technology (OT IT) 25
3.1. What is OT/IT? 25
3.2. Governing Principles of OT/IT convergence 26
3.3. OT/IT Convergence Characteristics 26
3.4. Conclusion 28
CHAPTER 4 29
4. Multiprotocol Label Switching Core Network for SCADA 29
4.1. Introduction 29
4.2. Multiprotocol Label Switching core Network Architecture 30
4.3. Basic Cyber Security Requirements for SCADA MPLS Core Network 31
4.4. Cyber Security Techniques for SCADA MPLS core Network 32
4.4.1. Routing Authentication 32
4.4.2. Access Control List 32
4.4.3. Customer Edge and Provider Edge Routers 33
4.4.4. Label Distribution Protocol Authentication 33
4.4.5. Firewalling 33
4.4.6. Trusted Devices 33
4.4.7. IPSec in MPLS networks 33
4.5. Conclusion 34
CHAPTER 5 35
5. Encryption and Decryption Techniques 35
5.1. Introduction 35
5.2. Cryptography History 35
5.3. Cryptography Basics 36
5.4. Cryptography Systems 37
5.5. Basic Ciphers 38
5.5.1. Substitution Cipher 38
5.5.2. Permutation Cipher 38
5.5.3. Concealment Cipher 38

5.6. Keying Information 39
5.7. Types of Encryption Schemes 40
5.8. Symmetric Encryption 41
5.8.1. Stream Cipher: Confidentiality 41
5.8.2. Block Cipher: Confidentiality 42
5.8.3. Information Integrity 46
5.8.4. Cryptographic Hashing 47
5.9. Asymmetric Encryption 48
5.9.1. Introduction 48
5.9.2. Public Key Infrastructure 49
5.10. Authentication Systems 49
5.10.1. Secure Socket Layer 49
5.10.2. Secure Shell (SSH) 50
5.11. Virtual Private Networks (VPN) 50
5.11.1. Introduction 50
5.11.2. Point to Point Tunneling Protocol (PPTP) 51
5.11.3. Layer 2 Tunneling Protocol (L2TP) 51
5.11.4. Internet Protocol Security (IPSec) 51
5.11.5. Open Virtual Private Network (OpenVPN) 51
5.12. Conclusion 51
CHAPTER 6 52
6. SCADA Systems Ethical Hacking and Vulnerability Assessment 52
6.1. Ethical Hacking Overview 52
6.2. SCADA Vulnerability Assessment Process 53
6.2.1. Assessment Planning 53
6.2.2. Actual Testing and Assessment Process 53
6.3. SCADA Vulnerabilities in a OT/IT Environment 54
6.3.1. Network Attacks 54
6.3.2. Attacks on Encryption Systems 54
6.3.3. Physical Access Attacks 55
6.3.4. Social Engineering Attacks 56
6.3.5. Operating Systems Vulnerabilities 57
6.3.6. Legislative constraints 57

6.4. Conclusion 58
CHAPTER 7 59
7. Lab experiment and simulation setup and configuration 59
7.1. Experiment Objective 59
7.2. Lab experiment equipment and tools used 59
7.3. Description of simulated SCADA services 60
7.3.1. Telecontrol 60
7.3.2. Substation Condition 60
7.3.3. Remote Monitoring/Metering 60
7.3.4. RTU Error Statistics 61
7.4. Lab experiment and simulation setup overview 61
7.5. MPLS/IP core network design and configuration 65
7.5.1. Basic Network design and Router Configuration 66
7.5.2. AES 256 Encryption in the MPLS/IP network 68
7.6. SCADA and RTU simulation set-up calibration 68
7.6.1. Routers configuration calibration 69
7.6.2. SCADA (FieldComm) Configuration 69
7.6.3. RTU Configuration 70
7.7. Conclusion 71
CHAPTER 8 72
8. Experimental and Simulation Results 72
8.1. Introduction 72
8.2. IP/MPLS Core network Routers Configuration Results 72
8.3. IP/MPLS Core network Encryption configuration Results 75
8.4. SCADA and RTU Connection Results 76
8.4.1. SCADA (FieldComm) and RTU connection confirmation without going through the IP/MPLS Core network 76
8.4.2. Emulator TCP/IP Serial Confirmation 78
8.4.3. SCADA (FieldComm) and RTU connection confirmation through the Encrypted IP/MPLS Core network 81
8.5. Results of SCADA Services Tested over an encrypted IP/MPLS core network 82
8.5.1. Telecontrol 82
8.5.2. Substation Condition 88
8.5.3. Remote Monitoring/Metering 92

8.6. RTU Error Statistics 96
8.7. Summary of Results 97
CHAPTER 9 99
9. Conclusions and Recommendations for Future Work 99
9.1. Research Project Conclusion 99
9.2. Recommendations for Future work 100
9.2.1. Proposed experiment for VHF (Voice) Services over IP/MPLS with encryption 100
9.2.2. Proposed experiment for Teleprotection over IP/MPLS with encryption 100
9.2.3. The Role for Digital Forensics in SCADA systems 101
Bibliography 103
Appendix A: A Typical Utility IP/MPLS Core Network III
Appendix B: Code for Configuration IV

List of Figures

Figure 1.1 FieldComm, RTU and protection relay setup (Chego, 2014)..... 15
Figure 1.2 SCADA services through an MPLS network setup (Chego, 2014)..... 16
Figure 1.3 IP/MPLS network with encryption performed within the network routers (Chego, 2014) 17
Figure 2.1 An illustration of a typical electricity utility SCADA system (Anon., 2011) 19
Figure 2.2 An Illustration of a HMI screen for a utilities SCADA system (E&A Engineering Solutions, 2014)..... 21
Figure 2.3 An illustration of a typical SCADA MPLS/IP core network for a utility (Huawei, 2016) 22
Figure 3.1 An illustration of OT/IT integration (Garg, 2013). 26
Figure 4.1 A demonstration of the basic functioning of a MPLS network (Rouse, 2015) 29
Figure 4.2 An illustration of an Energy utility IP/MPLS core ring network with several divisional services access networks (Alcatel Lucent, 2012) 30
Figure 4.3 An illustration of the circuit emulation implemented in MPLS for legacy services (Alcatel-Lucent, 2011) 31
Figure 5.1 Ancient Greek's cryptographic Scytale (Leemburg, 1995) 35
Figure 5.2 Illustration of Caesar Cipher (Adrian, 2014)..... 36
Figure 5.3 An Illustration of the internal operation of the Enigma Machine (Dade, 2006).... 36
Figure 5.4 An illustration of a substitution cipher (Mat, 2009)..... 38
Figure 5.5 A Permutation Cipher example (Shelton, 2014)..... 38
Figure 5.6 An illustration of a concealment cipher (Dr Ellefsen & Blauw, 2014)..... 39
Figure 5.7 A key derivation function from a passcode illustration (Jeff, 2012)..... 40
Figure 5.8 An illustration of a stream cipher (Lemke, 2014)..... 42
Figure 5.9 A Ron's Cipher implementation..... 42
Figure 5.10 Electronic Codebook mode encryption (Brandsma, 2012)..... 43
Figure 5.11 Electronic Codebook mode decryption (Brandsma, 2012)..... 43
Figure 5.12 Cipher Block Chaining (CBC) mode encryption (Brandsma, 2012) 43
Figure 5.13 Cipher Block Chaining (CBC) mode decryption (Brandsma, 2012) 44
Figure 5.14 DES encryption algorithm illustration (Van Tilborg & Jajodia, 2011)..... 45
Figure 5.15 AES overall structure (Secretary of Commerce, 2001)..... 46
Figure 5.16 Message Integrity Code illustration (Oracle, 2010)..... 47
Figure 5.17 Message Authentication Code illustration (Bernstein, n.d.)..... 47
Figure 5.18 RSA encryption scheme illustration (Lib4U, 2013)..... 49
Figure 5.19 A Secure Socket Layer session example (IdenTrust SSL, 2015) 50
Figure 7.1 Experiment and Simulation block diagram 62
Figure 7.2 Illustration of a D20 RTU and the IST ERTU used for the simulation and experiment..... 62
Figure 7.3 The Toshiba laptop running the MPLS/IP core network on GNS3 simulator, TCP/IP-RS232 conversion and it is connected to the RTU via RS232 63

Figure 7.4 The HP laptop running the SCADA simulator and RS232-TCP/IP conversion which transmits the data to the Toshiba laptop's GNS3 network via a network switch 63
Figure 7.5 An illustration of the switch and the MacBook Laptop connected to generate traffic and to try and send messages to the network 63
Figure 7.6 An overview of the overall simulation and experiment setup 64
Figure 7.7 Functional block diagram for the lab experiment and simulation 65
Figure 7.8 An illustration of an IP/MPLS core network connected to two access networks (LAN and Loopback) 67
Figure 7.9 The GNS3 network connected to SCADA and the RTU through the access networks (LAN and Loopback) illustration 67
Figure 7.10 An illustration of SCADA (FieldComm) configuration for serial communication 69
Figure 7.11 An illustration of SCADA settings with the Address of the SCADA..... 70
Figure 7.12 SCADA settings configuration continued 70
Figure 7.13 An Illustration of RTU settings configuration 71
Figure 7.14 An illustration of various messages which can be sent to the RTU for execution 71
Figure 8.1 An illustration of Router 1 interfaces configuration 73
Figure 8.2 An illustration of Router 2 interfaces configuration 73
Figure 8.3 Router 1 to 192.168.1.2 (interface facing the LAN network) interface successful ping..... 74
Figure 8.4 Router 1 to 192.168.2.1 interface successful ping 74
Figure 8.5 Router 1 to 192.168.1.5 interface successful ping 74
Figure 8.6 Router 1 to 192.168.2.2 interface successful ping 74
Figure 8.7 Router 1 to 192.168.1.6 (Loopback network) interface successful ping..... 74
Figure 8.8 Router 2 to 192.168.1.5 interface successful ping 75
Figure 8.9 Router 2 to 192.168.2.2 interface successful ping 75
Figure 8.10 Router 2 to 192.168.1.6 interface successful pings..... 75
Figure 8.11 Router 2 to 192.168.2.1 interface successful pings..... 75
Figure 8.12 Router 2 to 192.168.1.2 (interface facing LAN) interface successful ping..... 75
Figure 8.13 An illustration of the correct application of crypto engine to Router 1 76
Figure 8.14 An illustration of the correct application of crypto engine to Router 2 76
Figure 8.15. The figure shows no connection between the SCADA and RTU. 77
Figure 8.16. The figure shows the confirmation of SCADA connection to the RTU. 78
Figure 8.17. Emulator used for Serial to TCP/IP conversion of SCADA information sent over the IP/MPLS core network. 79
Figure 8.18 Confirmation of emulator connection to destination 192.168.1.6 (Loopback) ... 79
Figure 8.19 Emulator used for TCP/IP to Serial conversion of SCADA information to be executed by the RTU after propagation through encrypted IP/MPLS core network and also confirmation of connectivity between the two emulators through the network 80
Figure 8.20 The figure confirms the communication between SCADA and RTU through the IP/MPLS core network with data conversion from Serial to TCP and back from TCP to Serial 81
Figure 8.21 An illustration of SCADA service message; "Direct Operate" dashboard..... 82

Figure 8.22 Frame details of null function by SCADA to the RTU for Direct Operate service function 83
Figure 8.23 Frame details of a reply by the RTU for the null function sent by SCADA to the RTU for Direct Operate service function 83
Figure 8.24 Frame details of trip function by SCADA to the RTU for Direct Operate service function 84
Figure 8.25 Frame details of a reply by the RTU for the trip function sent by SCADA to the RTU for Direct Operate service function 84
Figure 8.26 Frame details of trip function by SCADA to the RTU for Direct Operate service function 85
Figure 8.27 Frame details of a reply by the RTU for the close function sent by SCADA to the RTU for Direct Operate service function 85
Figure 8.28 The Operate message dashboard illustration 86
Figure 8.29 Frame details of trip function by SCADA to the RTU for Operate service function 86
Figure 8.30 Frame details of a reply by the RTU for the trip function sent by SCADA to the RTU for Operate service function 87
Figure 8.31 Frame details of close function by SCADA to the RTU for Operate service function 87
Figure 8.32 Frame details of a reply by the RTU for the close function sent by SCADA to the RTU for Operate service function 88
Figure 8.33 An Illustration of an integrity poll message 88
Figure 8.34 Frame details of Integrity Poll function by SCADA to the RTU 89
Figure 8.35 Frame details of a reply by the RTU for the Integrity Poll function sent by SCADA to the RTU 89
Figure 8.36 Frame details of Cold Restart function by SCADA to the RTU 90
Figure 8.37 Frame details of a reply by the RTU for the Event Poll function sent by SCADA to the RTU 90
Figure 8.38 Frame details of Cold Restart function by SCADA to the RTU 91
Figure 8.39 Frame details of a reply by the RTU for the Warm Restart function sent by SCADA to the RTU 91
Figure 8.40 An Illustration of a read date and time message 92
Figure 8.41 Frame details of Read Date and Time function by SCADA to the RTU 92
Figure 8.42 Frame details of a reply by the RTU for the Read Date and Time function sent by SCADA to the RTU 93
Figure 8.43 An Illustration of a Write date and time message 94
Figure 8.44 An Illustration of a Write date and time message with a date change 94
Figure 8.45 Frame details of a reply by the RTU for the Write Date and Time function sent by SCADA to the RTU 95
Figure 8.46 The RTU Errors Statistics for the first iteration of SCADA communication with a RTU over an encrypted IP/MPLS core network 96
Figure 8.47 The RTU Errors Statistics for the second iterations of SCADA communication with a RTU over an encrypted IP/MPLS core network 97
Figure 8.48 A demonstration of an encryption introduced latency 99

Figure 9.1 Testing of voice services over IP network..... 101
Figure 9.2 An illustration of the Locard's Exchange Principle (Casey, 2004)..... 102
Figure 0.1: An illustration of a typical utility IP/MPLS core network (Anon., 2014)..... III

List of Tables

Table 2.1 SCADA services performance's minimum requirements (Maritz & Bronkhorst, 2010) 24
Table 3.1 Characteristics of OT, IT and converged OT/IT (Groenewald, et al., 2012)..... 28
Table 5.1 A comparative analysis summary of symmetric and asymmetric encryption schemes. 41
Table 8.1 Summary of results of SCADA services over an encrypted IP/MPLS Core Network 98

Abbreviations

Abbreviation/Acronym	Description
ACL	Access Control List
ADSS	All-Dielectric Self Supporting
AES	Advanced Encryption Standard
ASCII	American Standard Code for Information Interchange
BME	Bandwidth Management Equipment
BME	Bandwidth Management Equipment
CEF	CISCO Express Forwarding
CISSP	Certified Information Systems Security Professional
DCS	Distributed Control System
DDoS	Distributed Denial of Service
DES	Data Encryption Standard
DMS	Distribution Management System
DMS	Distribution Management System
DNP	Distributed Network Protocol
DoS	Denial of Service
ECB	Electronic Codebook
EMS	Energy Management System
FAT	File Allocation Table
FM	Frequency Modulation
GB	Giga Bytes
GSM	Global System for Mobile
GUI	Graphical User Interface
GUI	Graphical User Interface
HMI	Human Machine Interface
HP	Hewlett-Packard
HTTPS	Hyptext Transfer Protocol Security
ICMP	Internet Control Message Protocol
IEC	International Electrotechnical Commission
IED	Intelligent Electronic Device
IEEE	Institute of Electrical, Electronic Engineers
IP	Internet Protocol
IPsec	Internet Protocol Security
IPVPN	Internet Protocol Virtual Private Network
ISO	International Standards Organization
IT/OT	Information Technology Operational Technology
L2TP	Layer 2 Tunnelling Protocol
LAN	Local Area Network
LSP	Label Switched Paths

LTE	Long Term Evolution
MAC	Media Access
Mbps	Megabits per second
MD5	Message Digest 5
MIC	Message Integrity Code
MITM	Man-in-the-Middle
MPLS	Multiprotocol Label Switching
NCC	National Control Centre
NERC-CIP	North American Electric Reliability Corporation- Critical Infrastructure Protection
NIST	National Institute of Standards and Technology
NTFS	New Technology File System
OpenVPN	Open Virtual Private Network
OPGW	Optical Ground Wire
OSI	Open Standard Interconnection
PKI	Public Key Infrastructure
PPTP	Point to Point Tunnelling Protocol
PSK	Pre-Shared Key
RAM	Random Access Memory
RS232	Recommended Standard 232
RSA	Rivest, Shamir, Adleman
RTU	Remote Terminal Unit
SCADA	Supervisory Control And Data Acquisition
SCC	System Control Computer
SHA	Secure Hash Algorithm
SQL	Structured Query Language
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TLS	Transport Layer Security
UHF	Ultra High Frequency
USB	Universal Serial Bus
VHF	Very High Frequency
VPLS	Virtual Private Local Area Network Services
VPN	Virtual Private Network
WAN	Wide Area Network
XOR	Exclusive OR

CHAPTER 1

1. Introduction

1.1. Background

The background of this research is based on the fact that industrial networks including Supervisory Control and Data Acquisition (SCADA) systems were initially designed and developed as isolated networks and with minimal or no security at all. The purposes of this research project is to determine if encryption as part of the whole Cybersecurity value chain, can be used to secure communication in SCADA system and still allow SCADA service commands to be executed as per minimum SCADA performance requirements. This research is of extreme importance because SCADA systems are deployed in various industries including critical infrastructure networks such as in utilities. This means an attack on these SCADA systems can be very catastrophic including the loss of human lives. Thus this research project seeks to determine the most suitable encryption standard for SCADA systems in utilities.

1.2. Key Research Question

The key research question for this project is *"whether using encryption in SCADA systems, the services performance requirements are still met in OT/ IT environment over an MPLS core network"?* The research project will determine if SCADA services still perform as per minimum requirements over an encrypted MPLS/IP network in an OT/IT environment.

1.3. Research Objective

The primary objective for the research project is thus is to research and demonstrate that encryption is essential for secure SCADA communication over a MPLS/IP core network. Encryption forms an essential part of the Cyber Security value chain which has to achieve the following cyber security objectives

Confidentiality: ensuring that the information source is really from that source.
Integrity: ensuring that the information has not been altered in any way.
Availability: ensuring that system is not comprised but that it is available.

These objectives of encryption should be met with SCADA service performance requirements not violated which is the objective of the research project. Thus the key research question focuses only on the encryption part of the Cyber Security value chain of SCADA over an encrypted MPLS/IP network and no other aspects or areas.

1.4. Methodology

The research project had two components, the technical component and the non-technical component as part of completing the identified tasks. For the non-technical aspect, a detailed literature survey was conducted. The literature survey discusses an overview of SCADA, its components and its service requirements in an OT IT environment. The literature survey also discusses the OT IT environment and its operating principles and convergence between IT

and OT. Furthermore a discussion is given for IP/MPLS core networks, including its design consideration and methodology for SCADA in OT IT environment. The literature survey also discusses encryption techniques of which together with the IP/MPLS core network forms part of a critical part of this research project. A vulnerability assessment and ethical hacking techniques of SCADA systems is discussed to give typical SCADA vulnerabilities. This is to give an understanding of SCADA vulnerabilities and show to what extend can encryption assist in the whole SCADA cybersecurity value chain and also assist with preparing with regards to those vulnerabilities. The non-technical component sets to achieve or serve as a guide in terms of the best practice with regards to the role of encryption in the whole value chain and also what is the best design approach regarding the IP/MPLS core network with security and OT/IT convergence in mind. Thus the non-technical component sets the theoretical foundation for the simulation experiment.

The technical component involved implementing and testing the encryption and decryption for various SCADA services over an IP/MPLS core network. The overview is that the encryption and decryption was implemented using the Advanced Encryption Standard (AES-256) on periphery routers of the IP/MPLS core network. Then various SCADA system services commands were simulated into the network using a SCADA simulator called FieldComm. FieldComm is a communications tool for use with DNP3.0 protocol networks and devices (TruData Consulting, 2008). These commands will be encrypted and propagated through the virtually developed IP/MPLS network then decrypted and passed through to the remote terminal unit (RTU) to be executed.

1.5. SCADA and RTU setting

The initial test bench setup would involve setting up and configuring the remote terminal with the SCADA simulator. FieldComm will be used to generate and simulate various SCADA services. The purpose for this stage was to ensure correct operation of SCADA commands and the execution by the RTU. This will ensure that FieldComm and the RTU. This is illustrated in the figure below.

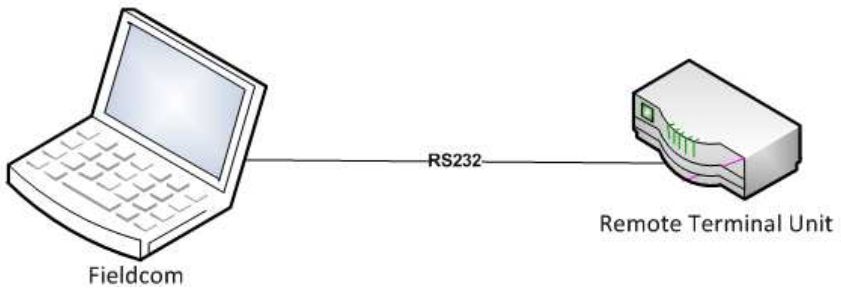


Figure 1.1 FieldComm, RTU and protection relay setup (Chego, 2014)

1.6. IP/MPLS Network

The next stage of the test bench would be the creation and the design of the MPLS core network. The MPLS network was designed by using GNS3. GNS3 is an open source telecommunications network design platform. It allows an accurate network design

mimicking real life conditions. Most networks can be designed firstly on GNS3 and then be implemented practically. The ideal MPLS core network for a very large utility is shown in Appendix A as example. This type of network requires high levels of computational and processing power, therefore for the purposes of this research only a scaled version of the network was implemented as per the processing capabilities of the computer used. The network is an IP-based network using routers. Once the network was build and configured, at its periphery a machine running FieldComm will be interfaced to the network with the RTU interface on the other end. The implementation of this setup is given in more details in chapter 8. The above overview is illustrated in the following figure.

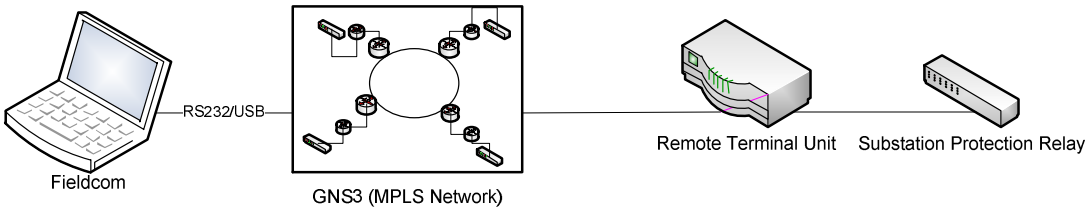


Figure 1.2 SCADA services through an MPLS network setup (Chego, 2014)

1.7. SCADA Encryption and Decryption Implementation

The next stage would involve the implementation of encryption/decryption. Both the encryption and decryption algorithms will be implemented using the National Institute of Standards and Technology (NIST) approved and tested AES-256 encryption. Further details of the implementation are given in chapter 8.

1.8. Final Test bench setup

The following figure gives an overview of the high level final test bench setup which is given in more details in chapter 8. In this test bench various SCADA service commands were simulated and issued by FieldComm. These were encrypted and passed to the IP/MPLS core network to the desired destination router. The service commands will be decrypted and will be processed by the RTU.

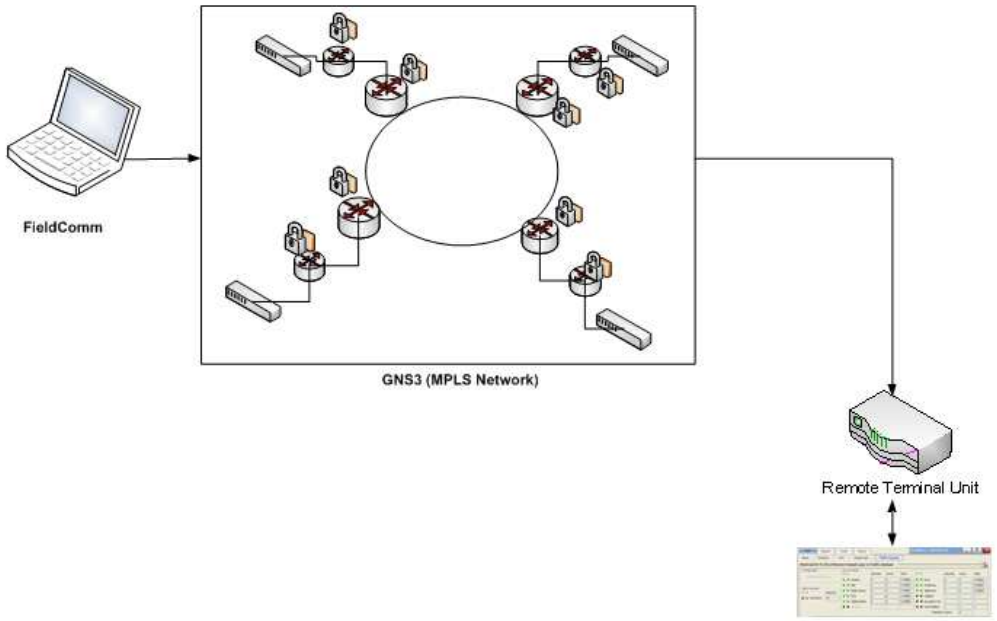


Figure 1.3 IP/MPLS network with encryption performed within the network routers (Chego, 2014)

1.9. Expected Outcomes

Upon completion of the project, the expected outcome technically is that SCADA encryption would be implemented successfully; meaning information would be encrypted and decrypted successfully over a MPLS/IP core network. This success in encryption and decryption process should be achieved with minimal effect on the current SCADA services performance requirements. The key milestones undergone in order to achieve were as follows; complete a literature survey, compile a non-technical report section for the literature survey, then conduct the technical part (experiment) and complete the technical part of the report including results and analysis thereof.

1.10.Conclusion

The research project sought to determine whether using encryption in SCADA systems, the services performance requirements are still met in an OT IT environment over an IP/MPLS core network. The conclusion of this research question was achieved through various stages as discussed in the previous sections and this was compiled into the various chapters of this research project report. Chapter 2 will give a brief discussion on a typical SCADA system, its basic architecture and service requirements in Operational Technology Information Technology (OT/IT). Chapter 3 discusses OT/IT in general and the convergence principles thereof. Chapter 4 introduces the Multiprotocol Label Switching (MPLS) Core networks. Chapter 5 discusses the encryption/decryption techniques. Chapter 6 comments on ethical hacking and vulnerability assessment in determining SCADA vulnerabilities and lists various SCADA vulnerabilities into several categories. From Chapter 7 onwards the focus is on the simulation experiment. Chapter 7 discusses the simulation experiment setup and

configuration. Chapter 8 gives the results of the simulation experiment and discusses them. Chapter 9 gives a summary of the results and also the research project with a brief discussion on recommendations for future work which can be continued further in this research field. The research report concludes with several appendices as supportive information for the research project.

CHAPTER 2

2. Supervisory Control And Data Acquisition Systems (SCADA)

2.1. Introduction

This chapter gives an introduction of what SCADA is, how it works and briefly discusses the main components that make up the SCADA system particularly in utilities. The chapter also discusses the various SCADA performance requirements in an OT IT environment of which some of those services will be tested during the experiment work in Chapter 7 and Chapter 8.

2.2. What is SCADA

SCADA systems are deployed in several industries including energy, transportation, traffic agencies, petroleum, telecommunications, and other processing facilities such as in food processing. They are the underlining control system for most of critical national infrastructures. The following figure 2.1 illustrates a typical SCADA system in utilities.

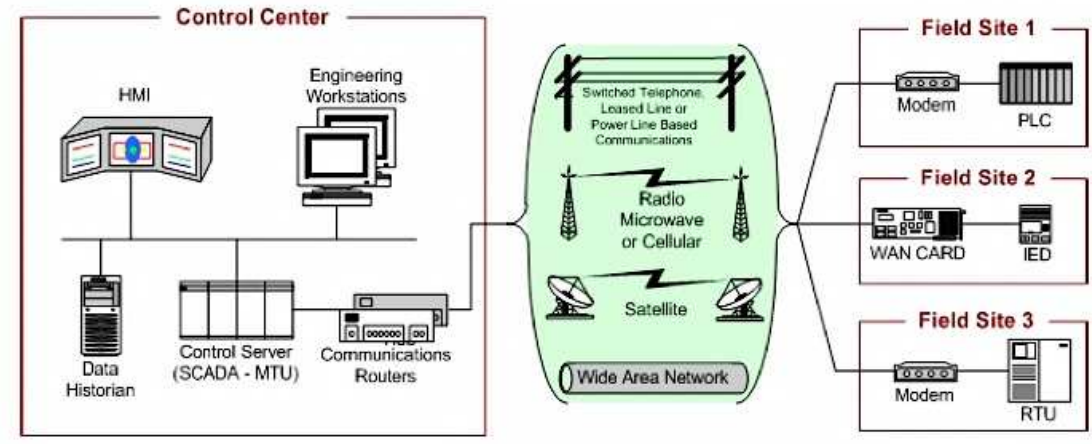


Figure 2.1 An illustration of a typical electricity utility SCADA system (Anon., 2011)

Initially when SCADA systems were implemented, security was not a major concern as the architecture was based on the fact that the network was totally isolated from other networks as most industrial networks were (Systems & Network Analysis Center, n.d.). SCADA systems have gone through an evolutionary process, from one generation to the next including from proprietary protocols to more open standard protocols. However the security of SCADA systems has not evolved at the rate that is acceptable to ensure acceptable levels of effective cyber security. This is alarming as SCADA systems control some of the critical infrastructure in a country and in industry. Many organizations, which have SCADA systems deployed, are not aware of SCADA vulnerabilities and most assume since SCADA systems are operating on an industrial network which until now operates independently from the

corporate network. The implications for SCADA failure are profound for an industry and a country. As aforementioned SCADA systems are deployed in utilities to control a country's electricity infrastructure and if it fails the consequences are profound which can result in economic losses on proportional scales and loss of human life to mention a few. SCADA failure can also be used in modern warfare against a country where systematic attacks are executed against a nation's critical infrastructure before the actual military strike commences. An enemy can sabotage the electricity infrastructure, the telecommunications infrastructure, banking, and water, basically everything controlled by SCADA to gain warfare advantage. This implies that SCADA failure has profound consequences for any country.

There are many motivations that would fuel SCADA systems attacks, which include, industrial and international espionage, terrorism, strategic military attack on infrastructure, former employees seeking revenge, etc. (Thales, 2013).. This means for an organization or even a country, a form of security measure and strategy needs to be put in place to protect SCADA systems and consequently critical infrastructure. Cyber security for SCADA has to be viewed as a value chain and not only consider one aspect of it. The most common mistake is that encryption or any other part of the cyber security value chain can be treated in isolation and as an ultimate solution. However organizations sometimes fail to understand that encryption plays a critical part within the whole value chain but it is not the ultimate. The cyber security for SCADA must be considered as parts of a value chain with each link in the value chain to be robust as it is required to be. However for the purposes of this research project, the focus was on encryption part of the value chain.

2.3. How SCADA works

In utilities a SCADA system is a software application whereby data or information is acquired the electrical distribution systems with much of the information coming from substations (Terezinho, n.d.). Typically a substation will have a certain number of intelligent electronic devices and programmable logic controllers. Amongst other uses, these are used to measure, monitor and control a number of equipment within the substation in real time. The information from these devices is transmitted to the SCADA system and the SCADA system is also able to transmitted instructions to these substation based devices. This is done over a telecommunication's transmission network. This telecommunication transmission network can be one of a number of various mediums, such as fibre optic, microwave, GPRS, etc.

2.4. Supervisory Control And Data Acquisition (SCADA) Main Components

The following section gives a brief description of the key components of SCADA system used in utilities.

2.4.1. Human Machine Interface

The Human Machine Interface (HMI) is the user interface which users use when operating the SCADA system. Data is processed by the SCADA software and it is displayed on the HMI graphical user interface screen which is then interpreted by the user to command appropriate actions. This means that the HMI also has controls, be it physical and software

based that the user or operator can manipulate. The HMI is also connected to the SCADA's database systems and other servers, and it is also connected to Intelligent Electronic Devices (IED's), Programmable Logic Controllers (PLC's), the access network, the core network and other peripherals of the SCADA systems. This is so that the user can be able to provide statistics such as trends, be part of scheduled maintenance, diagnostic data, etc. The advantage of the HMI system is that the information about the SCADA is presented graphically in a manner which depicts the actual physical SCADA topology. The following figure gives an illustration of the SCADA HMI.

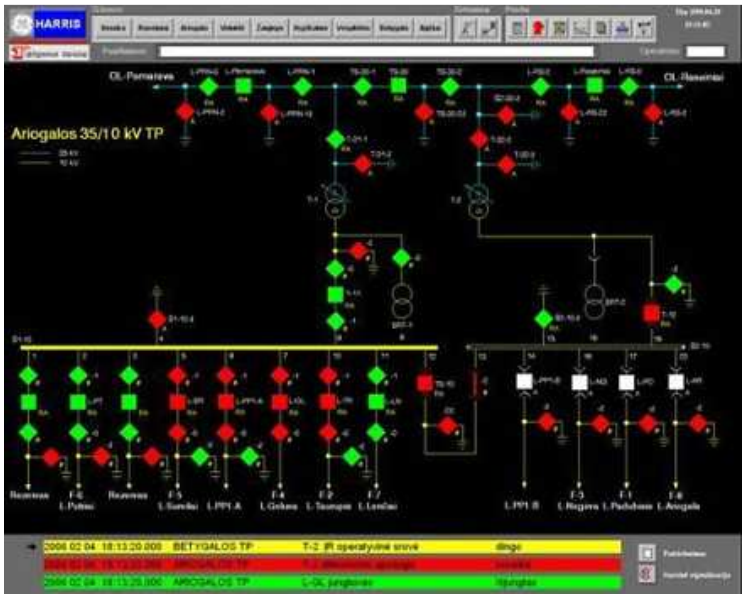


Figure 2.2 An Illustration of a HMI screen for a utilities SCADA system (E&A Engineering Solutions, 2014)

2.4.2. Supervisory Control And Data Acquisition Protocols

SCADA systems consist of protocols which are basically languages which allow the SCADA to be able to communicate with other components. These protocols in general are based on the International Standards Organization (ISO) standard which has several layers, typically seven layers. This is referred to as the Open Systems Interconnection (OSI) standard. There are proprietary and non-proprietary protocols which SCADA uses and in recent times industry moved away from proprietary SCADA protocols to more open standard protocols (Kalapatapu, 2004). The typically open standard protocols used in SCADA are MODBUS, MODBUS X, Distributed Network Protocol (DNP), ASCII, IEC 61850 to mention a few. Some of these protocols are legacy protocols and are not preferred. Currently in utilities 'distribution environment DNP3 is commonly used with a move towards IEC61850. For the purposes of this experiment the SCADA simulator that is used, uses DNP3 protocol.

2.4.3. Field Devices

Field Devices include Intelligent Electronic Devices (IED's), Programmable Logic Controllers (PLC's), relays, etc. These are normally devices deployed in distributed fashion in different locations, or substations in the case of energy utilities.

2.4.4. Remote Terminal Unit

Technically the Remote Terminal Unit (RTU) is a field device but because of its role and importance it is often discussed separately from other field devices. Several of the field devices such as the PLC's IED's and relays connect to the RTU which acts a gateway to facilitate communication between these devices and the SCADA system.

2.4.5. Supervisory Control And Data Acquisition Communications Network

A SCADA system usually comprises of a transport network, which can either be a switched telephone line, power line communication, radio or microwave, satellite, or even a packet switched network. In the case of the packet switched (IP/MPLS) network as the core, the SCADA system can be connected to it via an access network. The IP/MPLS core network is the focus of this research project and it is discussed in more detail in Chapter 4 with the implementation done in chapter 8. The following figure gives an illustration of a SCADA MPLS/IP network and its components. The figure illustrates the access network, aggregation and the core network.

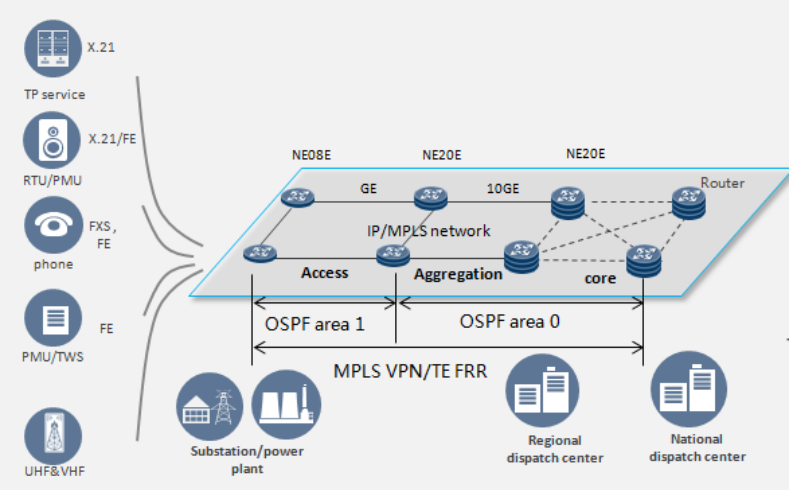


Figure 2.3 An illustration of a typical SCADA MPLS/IP core network for a utility (Huawei, 2016)

2.5. Performance requirements of SCADA systems in OT/IT

Operational Technology Information Technology (OT/IT) SCADA Telecommunication Services requirements are listed below for various SCADA services.

2.5.1. Supervisory Control And Data Acquisition Requirements

SCADA link requirements are summarised as follows (Maritz & Bronkhorst, 2010):

- A two 64 kbps SCADA link is required between each substation and the Energy Management System (EMS) at the National Control Centre (NCC)
- A 64 kbps SCADA link is required between each substation and the System Control Computer (SCC)

- A 64 kbps SCADA link is required between each substation and the regional Distribution

2.5.2. Management Station (DMS)

- A DMS is a type of a SCADA system
- DMSs require a 1 Mbps data link to the SCC
- A 10 Mbps circuit is required for data replication between the NCC (EMS) the SCC
- 1 Mbps links are required between the NCC and the regional TOCs, the Eskom

2.5.3. Telecommunications Management Centre and Security Centre

- SCADA links from substations to the DMS, EMS and SCC have a real requirement of 64 kbps and an aspiration requirement of 1 Mbps
- The SCADA links must be deterministic (Maritz & Bronkhorst, 2010).

2.5.4. Future SCADA links will be configured to eliminate sub-multiplexing.

The 10 Mbps circuit between the NCC and the SCC will be IP-based. The aspiration requirement is to implement a 20 Mbps circuit. This could be through either two 10 Mbps or four 5 Mbps circuits. The individual circuits will terminate on different floors of the SCC. SCADA links will have a dual port requirement as per Figure below. The purpose is to provide redundancy at Bandwidth Management Equipment (BME) port level.

The following Table 2.1 illustrates the service requirements for SCADA services which will be IP based on an MPLS network for a Distribution business. The table outlines or summarizes the various SCADA services and gives the various services' bandwidth requirements, minimum latency and loss requirements. The table also gives the availability requirement of each service and also the level of priority each service is allocated. This means services with high or critical levels of priority, their performance requirements cannot be compromised at any cost. The table also gives an indication of whether encryption or security is required for each of the service or if security is vital.

SCADA Service	**Bandwidth, Channel**	**Latency, Loss**	**Availability**	**Service Priority**	**Security**
Control Centre Voice services (Strategic Voice)	64 kbps, IP	Latency <150ms, Loss <1%	98%	Critical	Optional
Substation Condition Monitoring and post analysis	512 kbps, IP	Latency <200ms, Loss < 1%	98%	High	Required
Substation and Security Video Surveillances	128 kbps per stream, IP	Latency <150ms, Loss < 1%	98,33%	Medium	Optional
Telecontrol (11kv, 22kv, 33kv, substations RTU's)	9 kbps, IP	Latency <300ms, Loss < 0.05%	98,33%	Critical	Required

Telecontrol (11kv, 66kv, 88kv, 132kv, substations RTU's)	19.2 kbps, IP	Latency <300ms, Loss < 0.05%	98,33%	Critical	Required
Remote Metering	64kbps, IP	Latency <300ms, Loss < 1%	98,33%	High	Required

Table 2.1 SCADA services performance's minimum requirements (Maritz & Bronkhorst, 2010)

2.6. Conclusion

This chapter discussed the various components which make up the basic SCADA system architecture. Table 2.1 summarized the SCADA requirements in an OT/IT environment and categorized security requirements for those particular services as required or optional. For this research project simulation experiment, the ones categorized as required were simulated with results given in chapter 8 and the optional ones were not done as it was deemed not necessary. The following chapter discusses Operational Technology Information Technology. This is of importance since SCADA systems and the networks which they operate in are considered Operational Technology and the convergence between Operational Technology and Information Technology must be looked at since the IP/MPLS network is a packet switched network which is considered an Information Technology network in this context.

CHAPTER 3

3. Operational Technology Information Technology (OT IT)

This section discusses and gives a background into the Operational Technology Information Technology convergence, particularly within the context of utilities where SCADA is used as the industrial control system of which encryption thereof is discussed in this research thesis.

3.1. What is OT/IT?

Operational Technology has been a field of control and monitoring systems which previously was developed and supported by engineers who were responsible for maintaining the integrity and performance critical plants and networks (Groenewald, et al., 2012).

In recent times and advancement of technology and the need for efficiency there has been a growing need to interface, collaboration and data exchange, not only between different OT systems but also IT systems. The trends in OT/IT have been such that the underlining platforms and technologies used in IT and OT are becoming similar, there is a level of standardization. There is also an increasing use of shared infrastructure. Furthermore there has been a move from proprietary, hardware-based systems particularly in OT to a more transparent and industry standardized software based technology. However this has brought in a new set of risks. This means an integrated OT/IT collaborative approach is required which is defined in the following sections.

Information Technology is generally known however for the purposes of this discussion a definition is provided. IT refers to the comprehensive technology stack which covers both hardware and software used to manipulate data for a specific purpose (Atos, 2012). IT systems, particularly in legacy systems did not directly impact with the physical world in this context.

OT/IT defines the interaction between Operational Technology and Information Technology as they converge together for better organizational performance in organizations which have implemented both. In a utility environment OT/IT convergence and integration brings together real time systems such as SCADA, EMS, DMS with corporate or enterprise applications that are IT based (Ventyx, n.d.). The following figure gives a demonstration of such integration for a utility's two networks.

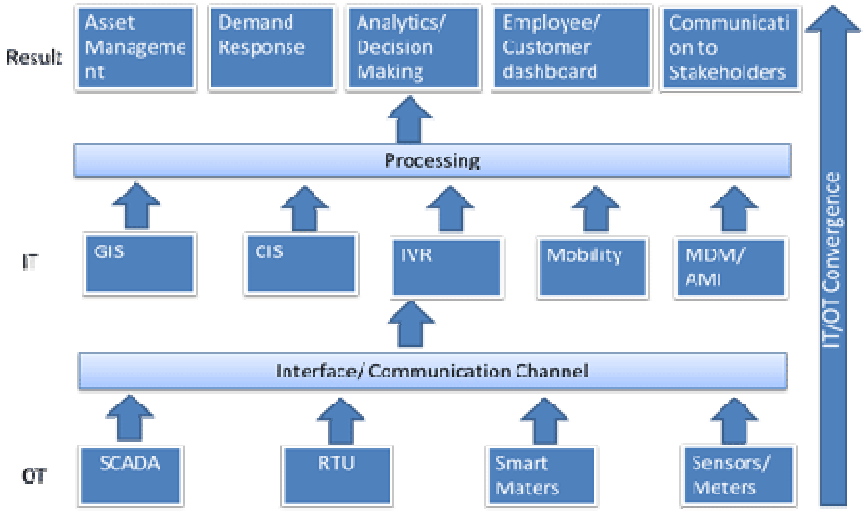


Figure 3.1 An illustration of OT/IT integration (Garg, 2013).

The vulnerabilities of OT/IT are discussed in a section looking at SCADA vulnerabilities in an OT/IT environment in chapter 6. The vulnerabilities discussed are focused on the key research question and focus area of the research project.

3.2. Governing Principles of OT/IT convergence

There are some governing principles that need to be applied when the OT/IT integration is implemented.

All Operational Technology requires following the engineering design, maintenance and operations review, quality and governance process as they would normal do and specified in their operational technology framework. This means that this would not change with the OT/IT integration. In terms of standards, both IT and OT will follow each of their specified standards with deviations more on the IT side as it has more flexibility than operational

3.3. OT/IT Convergence Characteristics

The following table summarizes convergence characteristics between Operational Technology and Information Technology. The table summarizes the individual characteristics of OT and IT and then both as a converged system. These characteristics are what need to be considered or taken into account when OT/IT integration or convergence takes place. It is important to understand IT characteristics and similarly OT characteristics before the integration and convergence. Finally it is equally important to understand the integrated or converged OT/IT characteristics.

	Operational Technology (OT)	**Information Technology (IT)**	**Converged OT/IT**
Role	Manage, Monitor, Control physical Assets and industrial network equipment	Process business transactions, provides communication, information, supports	End to end solution, convergence

		personnel	
Architecture	Embedded hardware/Software	Generic enterprise applications	Common Information Model (CIM), MPLS Core Network
Security	NERC-CIP compliant	Generic Enterprise Security – ISO27001/2	Integrated Cyber Security Approach
Interfaces	Electromechanical, coded displays, customized GUI's, computer based HMI	Windows-Based GUI's	Converged OS GUI's
Connectivity	Hardwired control networks (legacy networks), serial, non IP protocols, some may use IP as part of protocol stack.	IP-based connectivity with significantly lower performance requirements except between data centers, video and voice applications	Converged connectivity, pack switched protocols
Operating Systems	Embedded, Deterministic	Non-Deterministic, e.g. MAC or Windows	Shared of Converged OS
Environment	Typically deployed in rugged and has environments	In office and thermo regulated environments	Mixed as per need
Type of Data	Real time, time critical, near real time	Non-real time, batch orientated data for enterprise systems, real-time for IP based voice, video, etc.	For SCADA: real-time, time critical and batch dependent on service command applied
Ownership	Technology Group (Engineers)	IT group	Group owned, jointly governed as per effective business model
Usage	Always online	Some always online	Converged
Service Provisioning	Services usually cannot be outsourced due to the critical infrastructure needs	Some services can be outsourced but all under IT group	Core services should not be outsourced
Consequence of Failure	Possibly catastrophic with possible human life loss	Productivity loss, transaction delays,	Combination of both
Business Continuity	Core product mission critical systems and services	Business processes critical support systems including its facilities and support systems	End to end optimized operations
Typical Systems Examples	EMS, SCADA, Substation Automation, etc.	SAP, Outlook, Web interfaces	DMS, AMI, MDMS, MV90, etc.

Table 3.1 Characteristics of OT, IT and converged OT/IT (Groenewald, et al., 2012)

3.4. Conclusion

Chapter 3 discussed and defined OT IT, the basic governing principles of OT IT and lastly discussed the best practices and considerations for OT IT convergence in the context of a utility. The following chapter discusses the Multiprotocol Label Switching (MPLS) network. This is done because the research question looks at the performance of SCADA services over an IP/MPLS core network with encryption implemented. Thus it is important to discuss the architecture, design and security requirements for such a network.

CHAPTER 4

4. Multiprotocol Label Switching Core Network for SCADA

The purpose of this chapter is to discuss the Multiprotocol Label Switching (MPLS) Core Network for SCADA systems. The discussion gives a brief introduction to MPLS, the basic architecture requirements in SCADA and also discussing its basic requirements for cyber security of which encryption (which is the focus area of this research project) is discussed. The chapter concludes by discussing briefly several cyber security techniques which can be used to secure the SCADA MPLS core network.

4.1. Introduction

A simple definition of Multiprotocol Label Switching (MPLS) is a type of protocol which facilitates network traffic flow for faster routing. This is done by allowing packets to be moved at layer 2, which is the datalink layer instead of having to forward the frame to the layer 3, the network layer. This is done by each packet being labelled upon coming into the MPLS network by a router, this router is normally referred to by an ingress router (Rouse, 2015). Then all the following routers simply do packet forwarding to the desired destination which is entirely determined by the labels attached by the ingress router. The following figure illustrates the basic functioning of a simple MPLS core network.

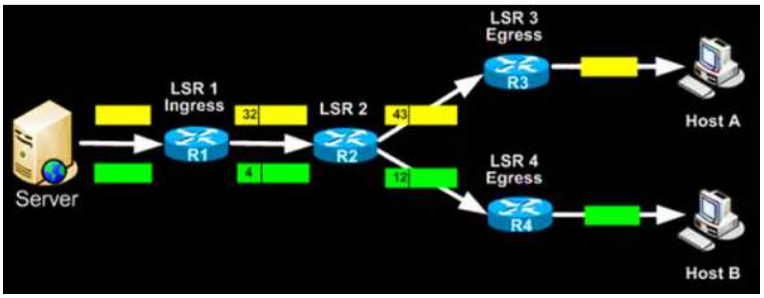


Figure 4.1 A demonstration of the basic functioning of a MPLS network (Rouse, 2015)

The destination label determines what route the packet will follow beforehand and this is referred to as the Label Switched Paths (LSP). This allows the MPLS architecture design to determine before what traffic will take which route which makes this network service aware and increases reliability and availability. For instance the routers can be configured to beforehand to determine which type of packets and from which address range must take this part. In chapter 8, for the experiment simulation part, the routers where configured to allow traffic only in a certain range of address to improve the level of security. Thus this predetermination of routes and address range not only improves security but also can improve the quality of supply such that service requirements (latency, packet loss, etc.) are met, as

defined in Table 2.1 for this research project because routers do not have to halt the process of forwarding packets so that they can go through routing tables.

4.2. Multiprotocol Label Switching core Network Architecture

The MPLS core network topology is determined by graphical mapping of the actual connections of routers and peripheral devices on the network (Alcatel Lucent, 2012). There are several topologies or architectures that can be followed when implementing an MPLS/IP core network and these include, a ring topology, a meshed or partially meshed, or a linear, hub, etc. For energy utilities it is recommend that ring architecture be used particularly if there would be access networks involved or used to access the MPLS core network as demonstrated by the proposed Eskom MPLS core network in Appendix A. The primary reason a ring topology is preferred it is because the architecture provides a level of efficiency, reliability and a level of redundancy such that traffic can be re-routed to different routes in cases of faults on the primary route. Another advantage of using ring architecture is that the feature of fast reroute in MPLS can be used such that traffic on the ring is not duplicated and as such ensure resiliency (Budka, 2014). This means that the bandwidth of the architecture can be efficiently used. The fast reroute feature in this architecture can reroute the traffic in below 50ms should there be a problem on the primary route. The following figure below gives an illustration of a whole energy utility MPLS/IP core network and various divisional services that can be provided over this core network through various access networks. As it can be viewed similarly to the proposed Eskom network in Appendix A, it is a ring core network with several access networks to the various divisional services points.

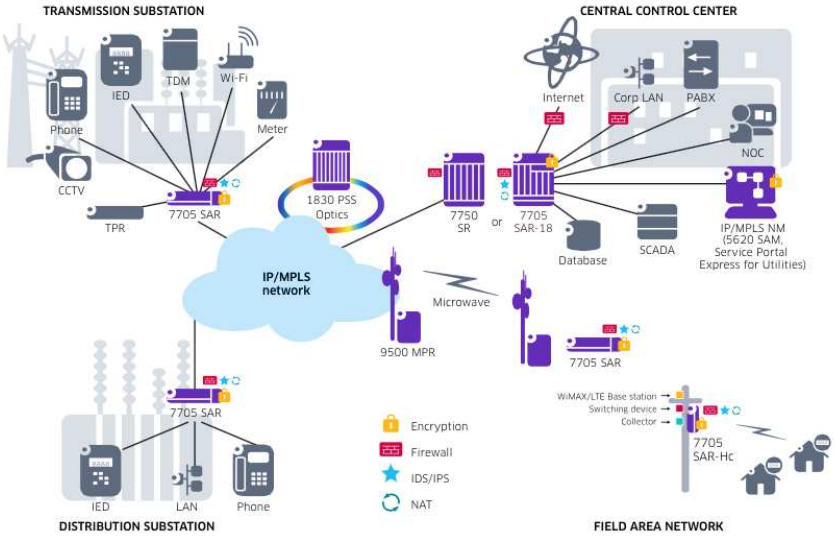


Figure 4.2 An illustration of an Energy utility IP/MPLS core ring network with several divisional services access networks (Alcatel Lucent, 2012)

For energy utilities the data packets are required to be able to propagate from the SCADA master control to the various locations within the network seamlessly. Furthermore the data packets are also required to be able to propagate amongst the various locations themselves and other networks such as the enterprise network for services such as remote substation

access through Demilitarized Zone (DMZ) network architecture. The other architecture requirement is that the IP/MPLS core must be seamlessly scalable to accommodate the growing number of sites especially in the energy distribution environment where the number of substations grows as the demand for electricity grows. The MPLS/IP core network for utilities must also be able to use high end scheduling processes to hierarchically place services as per their defined criticality as illustrated in Table 2.1. This means that services traffic will be fairly prioritized based on this criticality. This is often referred to as the virtualization of the MPLS/IP core network (Network Strategy Partners, 2009). This is achieved in the MPLS/IP core network through implementing the LSP service over certain virtual private network (VPN) process such as the circuit emulation service, virtual private LAN service (VPLS) and the Internet Protocol virtual private network (IPVPN) (Alcatel Lucent, 2012). These virtual private networks mentioned above can be used concurrently to carry different services and their specific LSP's. This means that one kind of service traffic can be transmitted over an IPVPN while another service over VPLS as an example. Circuit Emulation Service is used within the IP/MPLS core network specifically for legacy services which are using the Time Division Multiplexing protocol. This service basically transforms the TDM data or serial data into an IP or packet switched data which can propagate over the IP/MPLS core network. The following figure gives a demonstration of the circuit emulation service implementation for legacy services.



Figure 4.3 An illustration of the circuit emulation implemented in MPLS for legacy services (Alcatel-Lucent, 2011)

The virtual private LAN services basically reroutes packets based on their media access address (MAC) address. Both the source and the destination MAC address must be pre-included in the MAC address table for all the routing paths. The internet protocol virtual private network (IPVPN) is used in MPLS/IP network for IP traffic only, meaning that the router forwarding is based solely on the IP address of the destination. For this research project, the simple IP/MPLS network architecture is based on both the Circuit Emulation services due to legacy equipment and the IPVPN service. This means the data from legacy SCADA system to the legacy RTU is encapsulated via an emulation and then transmitted over a predefined IPVPN MPLS core network which is encrypted.

4.3. Basic Cyber Security Requirements for SCADA MPLS Core Network

There is an inherent requirement for cyber security for SCADA over MPLS/IP core networks as it is a packet switched network which is connected to other networks such as access networks which might be connected to other networks which are connected to the internet. These cyber security requirements must also satisfy the SCADA cyber security value chain requirements and furthermore ensure that the SCADA services performance requirements are still met. This section discusses cyber security requirements that are specific to the MPLS

core network. The focus of this research project is encryption versus performance requirements, the other cyber security requirements are discussed to indicate where in the value chain encryption fits in and also to give a holistic view of the importance of each cyber security component in the value chain. These requirements are namely; address space and routing separation, hiding the network core structure, resistance to cyber intrusions, and spoofing prevention (CISCO MPLS, 2014). The address space and routing separation means that each end system would be assigned a distinctive IP address whereby all routes to this end system is acquired by this IP address. Hiding the core network topology means the architecture of the MPLS/IP network should not be easily deduced or seen from outside networks. This means the internal addressing and routing is hidden from the outside world and this can bring a level of security. However from a reliability, resilience and continuity point of view the addressing and routing must be properly documented. Resistance to attacks means that to some extend the MPLS/IP core network must be resistive to external attacks. Preventing spoofing refers to hiding the IP address within the core network and also hiding the LSP's. This can be achieved through IPSec which was covered in chapter 8 for this research project as part of the experiment.

4.4. Cyber Security Techniques for SCADA MPLS core Network

This section discusses technique options that are available to secure the SCADA MPLS core network that gives high levels of security and ensures SCADA services performances requirements are met. Each of these are discussed to simply give an overview of where encryption is SCADA cyber security fall is within the value chain even though key focus area of the research project is only encryption. Herewith are the various techniques (Killalea, 2000):

4.4.1. Routing Authentication

Routing information must be hidden and made confidential as it is a likely point to be attacked. The hidden routing information ensures that data from one router to the next comes from a router that the next router expects. This means all peer routers during configuration, the expected address ranges must be predefined for authentication purposes. Predefined authentication works best for static routing. For dynamic routing protocols in a case where the primary link fails and there is pre-authentication configuration there are ways of achieving that and still have a level of security. One option is to update the dynamic routing protocol table such that expected ranges of IP addresses are configured on the routers. The other alternative would be to use label distribution protocol authentication, discussed in section 4.4.4. Label distribution protocol authentication uses HASH algorithms to identify network routers to provide a level of authentication in dynamic routing such that foreign routers are not introduced in the network for attacking purposes.

4.4.2. Access Control List

Access control list refers to a list of permitted objects which are given access on a network. In this context it is referred to as packet filtering meaning all other information which is not permitted will be denied access.

4.4.3. Customer Edge and Provider Edge Routers

This technique refers to when a utility's network is also connected to an external service provider's network. The customer edger router refers to the utility's router and the provider edge router refers to the service provider's router. The best configuration in this instance is to use static addresses which only allow traffic from this specific static address of the provider edge router.

4.4.4. Label Distribution Protocol Authentication

This technique refers to using hashing algorithms to authenticate routers and preventing attackers from introducing unauthentic routers. The hashing algorithms which can be used in this case are SHA-1 and MD5. MD5 however has in recent times been broken thus it is recommended that a SHA-1 algorithm is used. A detailed discussion on Hashing algorithm is given in Chapter 5 with experiment done in Chapter 8.

4.4.5. Firewalling

A firewall is essential for external connections and across different access networks and VPN's connected to the MPLS core network to prevent attacks coming in from these connections. The firewalling principle in the context of utilities' operational technology is illustrated in figure 4.2 above.

4.4.6. Trusted Devices

A list of trusted devices with their respective IP address can be added onto the routing tables such that only those addresses have access to the network.

4.4.7. IPSec in MPLS networks

More security to the IP/MPLS core network can be added by configuring IPSec on the core routers. IPSec should be configured with several of the following techniques to ensure optimal security (Shirley, 2000).

4.4.7.1. Authentication

Authentication can be implemented on peer routers so that a foreign router is not used in an attack. The key for authentication can be pre-shared between the routers during each peer router configuration.

4.4.7.2. Encryption

In a case where an intruder is able to use packet sniffing techniques for instance to sniff the traffic in the network, encryption ensures that the attacker is not able to make any sense of the traffic. This means the integrity of the information is preserved. Encryption is discussed in much more detail in Chapter 5 and the implementation thereof is discussed in Chapter 8 with the results given in Chapter 9.

4.4.7.3. Integrity of Packets

This technique allows that any changes to the packets in the network be reported.

4.4.7.4. Replay Detection

This technique is essential when an intruder has successfully been able to sniff packets and try to replay them at a later stage. This feature of IPSec will be able to detect this and prevent it.

4.4.7.5. Point-to-Point

The point-to-point feature is used between each peer router for static configuration of most of the IPSec features. It is advised that this feature be used for small core networks as it was done in this research project. It becomes difficult to use this especially in ever expanding network such as with increasing number of substations in utilities.

4.5. Conclusion

Chapter 4 discussed MPLS/IP core network, its design, architecture consideration and cyber security requirement. The security aspect as per the requirements which was implemented extensively for MPLS networks in this research experiment is the IPsec together with other techniques which focuses on encryption which is the key focus area of the research project. The following chapter discusses the various encryption/decryption techniques. It gives a background and gives a comparative analysis between the different schemes to lay the foundation in terms of the best suitable technique for implementation in the context of a utility.

CHAPTER 5

5. Encryption and Decryption Techniques

This section introduces and discusses encryption and decryption techniques.

5.1. Introduction

There are several questions that have to be asked with regards to encryption and decryption relating to the information that is to be encrypted and decrypted. These questions include how important and useful is that information, what is the cost to repair that information. Furthermore questions such as what is the cost or would it make a difference if this information is known by others to whom it is not intended or the information becomes available in the public domain. Further questions such as; is the information encrypted or decrypted valid and can the generator of the information be verified as the owner and that the information is authentic and comes from the so claimed owner. These questions are important and they normally inform what encryption techniques are suitable for use.

5.2. Cryptography History

As society developed, the need of securing information grew as result of societal aspects such as war throughout history that resulted in cryptographic schemes. This was apparent in many different cultures across the world (Khan, 1996). Cryptography can be seen in ancient Egypt with the use of Hieroglyphics as attributed to secrete communication between pharaohs and gods. Cryptography appeared also in ancient Greece with the use of a Scytale. A Scytale is made up of a cylinder with a strip parchment with a message on it (Leemburg, 1995). This parchment would have to be wind around the cylinder exactly the same way to be able to read the message. Spartans predominantly used this during times of war to execute military campaign through secure communications. The following figure below gives an example of Scytale.

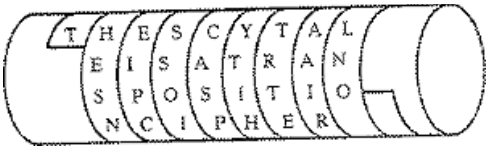


Figure 5.1 Ancient Greek's cryptographic Scytale (Leemburg, 1995)

Ancient Rome also used cryptography for secure communication by replacing Roman letters with Greek letters; this was known as the Caesar Cipher. The Caesar cipher is illustrated in the following figure.



Figure 5.2 Illustration of Caesar Cipher (Adrian, 2014)

In World War II a device called an Enigma Machine was designed by German engineer Arthur Scherbius which was basically an electron mechanical rotor cipher machine which performed multiple predetermined steps of substitution that were pre-set according to that particular's day settings. This meant each day the settings would be changed. This was used for encrypted military operations communications. The following figure 5.3 gives an illustration of the Enigma Machine. The machine worked by basically changing a letter into another letter at each stage it passed through the machine (Dade, 2006).

In today's day and age, whole government departments are created towards cryptography to still try and ensure secure communications, all with the goal for availability, integrity and authenticity of the information exchanged.

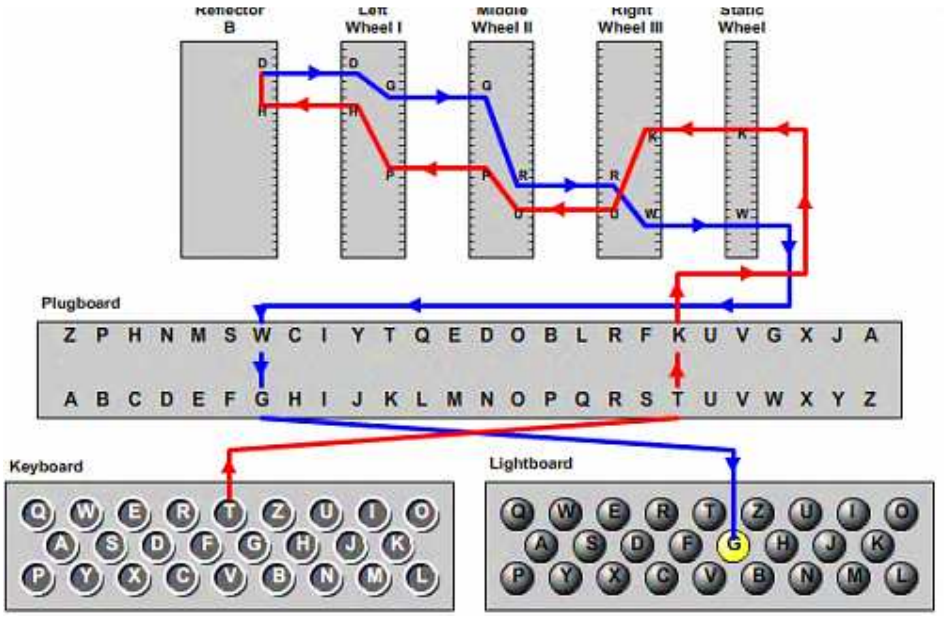


Figure 5.3 An Illustration of the internal operation of the Enigma Machine (Dade, 2006)

5.3. Cryptography Basics

The basic goal of cryptography or encryption is to protect the confidentiality of information in information systems. Cryptography is based on scrambling information such that it is completely random to anyone except those who know how to decrypt it. One of the key

challenges of cryptography is the appearance of the encrypted information as random. Another key factor in cryptography is a concept called plausible deniability which is a condition such that, a subject can “safely and believably deny any knowledge of any particular truth that may exist because no evidence of such truth’s existence can be proved” (Dr Ellefsen & Blauw, 2014). This means that a relatively good encrypted piece of information should look completely random such that it is indistinguishable from true random data and furthermore the deniability of the existence of decrypted data. This is the ultimate goal of a cryptographic system. The technical definition of cryptography is that cryptography is a way of encoding information such that an entity, being a person or machine is not able to easily or feasibly decode the encoded message without having knowledge of a key whereby this key is a large number which is not easily calculated. The reason for mentioning this technical definition is because it introduces the concept of a key which is essential in cryptographic schemes.

5.4. Cryptography Systems

A crypto system is made up of a combination of three elements; namely the algorithm used for encryption, the keying information (key) and the operational procedure. The encryption algorithm performs the actual encryption and decryption. All the other elements of a cryptography system depend on this algorithm. The Key is an algorithm that generates keys which will be used by the above mentioned encryption algorithm. The operational procedure basically states how all these parts of the encryption system functions together with one another to achieve the cryptographic goal. For example it will specify the type of input and output required by the encryption algorithm and how the key will be exchanged by parties involved in the cryptographic system.

In today’s age, cryptography systems can be found in almost any computing systems, for instance, passwords and logons secure web transactions, communication devices, etc. Cryptographic systems are very widespread and available however still prove difficult in most instances to make them common. There are several reasons for this and these include the understanding of the technology. This means that cryptographic systems are not as intuitive and are difficult to understand especially for users who are not trained. The second reason is the sharing of secret information, which is an important issue with a cryptographic system to distribute and share the key. The other reason why cryptographic systems are not common is the issue of cost. Cryptography systems require a lot of money for research and development and they require a lot of time. Thus full packaged cryptographic systems can be very expensive for this reason. Lastly another common reason why these systems are not so common is that, at times it has specialized administrative requirements whereby a person requires specialized knowledge how to fully use the system, which introduces the element of training. One aspect of encryption that has to be considered is the process of deciphering the encrypted message without a key, which is normally known as Cryptanalysis. These are techniques used by intruders to compromise the system. They use three basic categories of cryptanalysis which will be discussed later in this chapter in much more detail. They use known plaintext and the ciphertext of a message to generate the key. Pattern recognition can be used starting with the common place of a message and try and find similarities and derive

a key. The last category is the use of a brute force approach attack. This is a trial and error method such that each and every possible combination of a key is tried until the correct key is generated.

5.5. Basic Ciphers

This section discusses basic ciphers which form part of the fundamentals of encryption schemes and techniques used even in today's systems.

5.5.1. Substitution Cipher

The Substitution Cipher, otherwise known as a Caesar Cipher encodes information by substituting plaintext with other letters. The figure below illustrates an example of a basic substitution cipher.

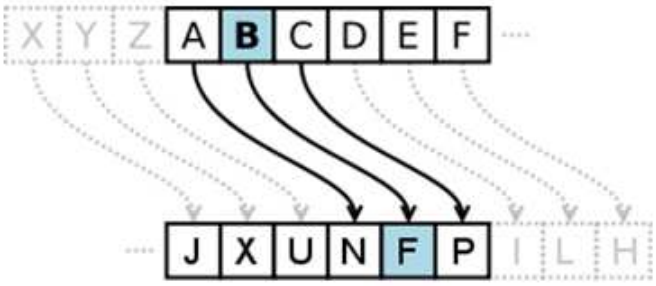


Figure 5.4 An illustration of a substitution cipher (Mat, 2009)

5.5.2. Permutation Cipher

The permutation cipher is also known as the transposition cipher whereby plaintext characters are moved according to a specified permutation formula. This is a basis of which some of the most advanced encryption schemes such as Advanced Encryption Standard (AES) are based on. The following figure gives a simple illustration of a permutation cipher.

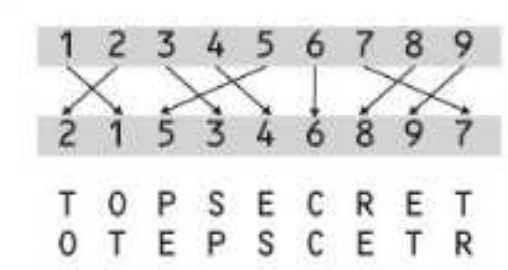


Figure 5.5 A Permutation Cipher example (Shelton, 2014)

5.5.3. Concealment Cipher

The concealment cipher is basically hiding a message in plain site within other messages. In modern day age this technique is quite advanced with the use of steganography to hide other information in images and videos which will not be so clear to the human eye. The following figure gives an illustration of a concealment cipher.

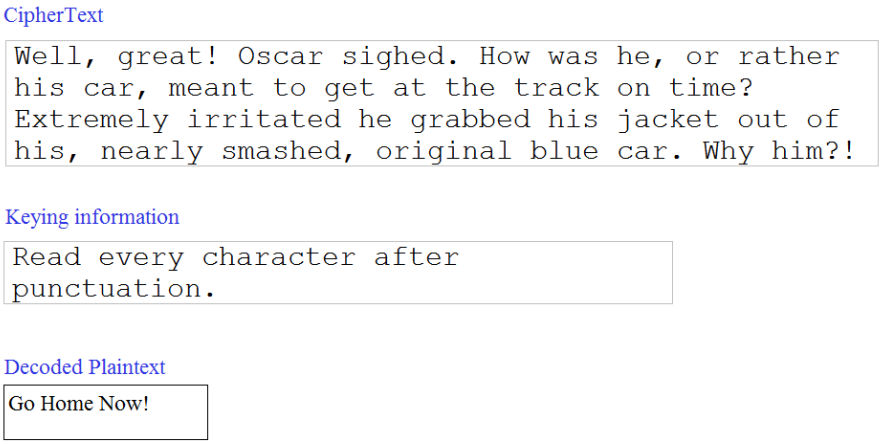


Figure 5.6 An illustration of a concealment cipher (Dr Ellefsen & Blauw, 2014)

5.6. Keying Information

Keying information forms a critical part of a cryptography system. In Section 5.3, it is mentioned as one of the three components of a cryptography system. Keying information or a key in short, is a complex sequence of bits that is produced by an algorithm of which it is protected by a passcode and this key allows encryption and decryption to take place. The strongest forms of keys in cryptography systems have to be random in nature which means they must not be known by any other parties other than the ones involved. The issue of random was raised as an important aspect of encryption in previous section and there are only few sources of pure randomness in nature such as the movement of molecules, radioactive decay and biological mutations to name a few (Dr Ellefsen & Blauw, 2014). With computing systems it nearly impossible to achieve pure randomness meaning consequently the keys generated are not purely random as computer systems are deterministic by design. True randomness in computer systems would be extensively expensive. The best way to achieve randomness in computer systems is to implement a pseudorandom process, in this context a pseudorandom generator to generate keying information or keys. A pseudorandom number generator is not necessarily a truly random sequence but generates the same sequence each time but that sequence can be altered by what is referred to as a seed. These seeds can be generated from various sources that express a form of randomness, for instance in computing systems the current millisecond clock or the past several hundred keyboard keystrokes or mouse locations. This is how relatively random keys can be generated using a pseudorandom process.

In cryptographic systems a relatively long key which is difficult to figure out is mostly desired as it makes the encryption code stronger. Thus the term keylength is important, which is a measure of the length of a key measure in amounts of bits making up that key. Keying information or keys introduces a difficult measure of security in cryptography systems which are how to keep the secret that only the vested parties know the key and still make sure the secrecy of the key is kept during exchange of the key between those two parties. There are common practices that can be implemented to ensure key secrecy such as using tokens to store the keys, encrypting the keys upon transporting them across networks for example and

keys being derived from passphrases. The other disadvantage about keys is that they are better when they are longer however rely on relatively short passcodes which human beings must be able to remember to protect these keys. This means the introduction of a human being in cryptographic system is the weakest link. The following figure gives a simple illustration of a key derivation function that summarizes the above discussion.

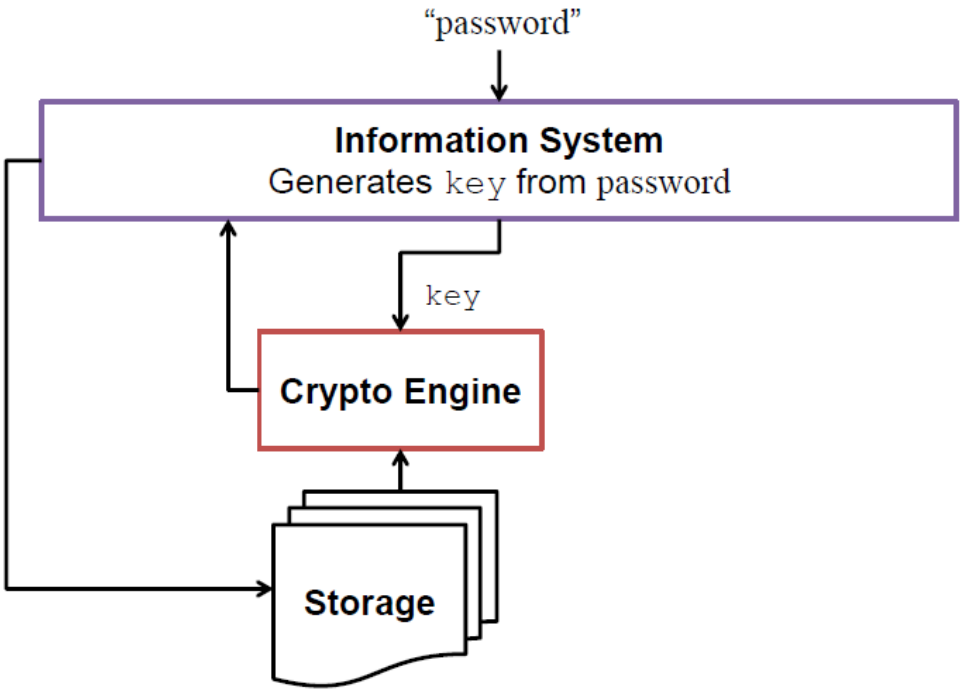


Figure 5.7 A key derivation function from a passcode illustration (Jeff, 2012)

5.7. Types of Encryption Schemes

The strength of which a cryptography system provides depends on the strength of the algorithm implemented; however there must still be a balance between algorithm complexity and computational efficiency. This means it must encrypt and decrypt information with relative ease and reasonable time frame. The balance between these two factors makes it difficult for algorithms to be unbreakable, meaning some can be breakable especially with their reliance on keying information. There are ways to make algorithms even less breakable; this is referred to as security obfuscation (Gebbie, 2002). This means that information about how the encryption of information was done must be kept secret.

There are two main types of encryption techniques namely the symmetric encryption and asymmetric encryption schemes. Symmetric encryption is when the same key is used to encrypt and decrypt data. Furthermore symmetric encryption uses fixed scrambling operations. The key for symmetric encryption is measured in bits and depending on the level of encryption the number keys can be determined from that.

If for example:

$$n = number\ of\ bits\ used\ for\ encryption \quad (5.1)$$

Then

$$number\ of\ keys, k = 2^n \quad (5.2)$$

Thus if 56 bits are used for instance for encryption then the number of keys would be

$$k = 2^{256} = 72,058 \times 10^{15} \quad (5.3)$$

Asymmetric encryption uses one key to encrypt and another to decrypt the data. The encryption key would generally be known by more than one source and it would be published and this is known as *public key* system. Asymmetric encryption uses prime numbers for encryption unlike symmetric encryption which uses complex scrambling. The following table gives a comparative analysis between the two schemes.

Scheme	Encryption	Decryption	Advantage	Disadvantage
Symmetric	Same Key	Same Key	Fast and versatile	Key distribution to both parties can compromise key.
Asymmetric	Public key	Private key	Different keys for encryption and decryption	Slower to compute

Table 5.1 A comparative analysis summary of symmetric and asymmetric encryption schemes.

Encryption schemes seek to achieve and ensure the protection of the integrity of messages, availability and confidentiality. Encryption schemes does not however solve problems introduced by access control onto systems, users who have access but are malicious, and also encrypting everything does not necessarily make information purely secure. Symmetric and Asymmetric encryption schemes are discussed in detail in the next sections respectively.

5.8. Symmetric Encryption

Symmetric encryption is categorized into two categories that have a goal to achieve confidentiality namely; Stream cipher and Block Cipher. The primary difference between these two is with regards to the amount of data or information they each encrypt at a time. Each of these schemes is discussed in the following sections respectively. Symmetric encryption schemes that achieves integrity and availability will be discussed later

5.8.1. Stream Cipher: Confidentiality

A stream cipher encrypts streams of data at a time, meaning it encrypts each bit of data at a time. The keying information requirement is that the key must be the same length or longer than the data being encrypted. This scheme is most suitable for continuous data streams of unknown length and size such as in a mobile telephony system and network connections such as SSL or TLS. Stream ciphers are relatively faster to compute but do not provide

information integrity. The following figure gives a simple illustration of a stream symmetric cipher implemented using a simple XOR gate.

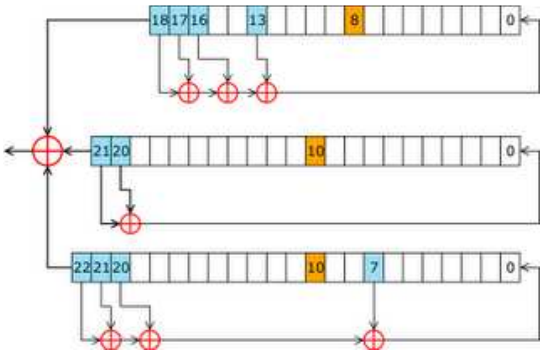


Figure 5.8 An illustration of a stream cipher (Lemke, 2014)

One of the most commonly known and used stream ciphers is the Ron's Cipher with a keylength of up to 2048 bits. It generates a pseudorandom stream of bits. It does this by generating 256 bytes in numerical order and swaps each byte with a byte dependent on each byte from the key (permutation). With each byte of data the swop happens again and the result is XORed with the plaintext. This scheme was very useful as it is fast but has a flaw that if the same key is used more than once the whole system is compromised. The RC4 cipher is shown in the following figure below.

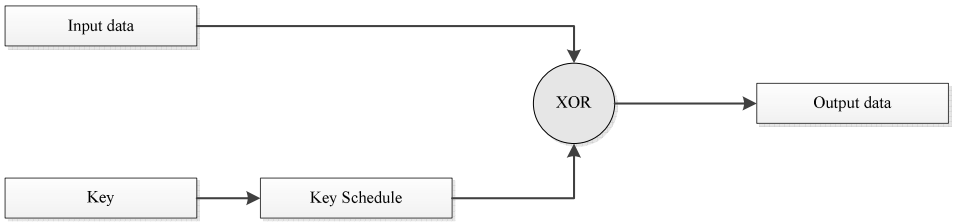


Figure 5.9 A Ron's Cipher implementation

5.8.2. Block Cipher: Confidentiality

A Block Cipher encrypts information in chunks of data blocks. All blocks must be of the same size and length, and if there is not enough data to fill blocks random data is added to fill them up. There is a specific algorithm that generates these random data and it is referred to as a padding algorithm. The Block cipher is best suitable for encrypting determined file sizes such as documents. However this means that Block ciphers require more memory. There are several examples of block ciphers and these include the Electronic Codebook (ECB) mode Encryption. Each block of data encryption in ECB mode is independent of each other but uses the same key. This is a great flaw because if one breaks one block they can get the key and be able to decrypt the information. The following figures gives an illustration of both encryption and decryption modes of Electronic Codebook mode respectively (Gattol, 2015).

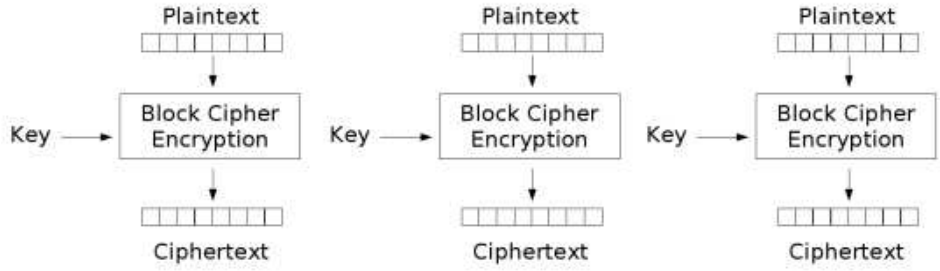


Figure 5.10 Electronic Codebook mode encryption (Brandsma, 2012)

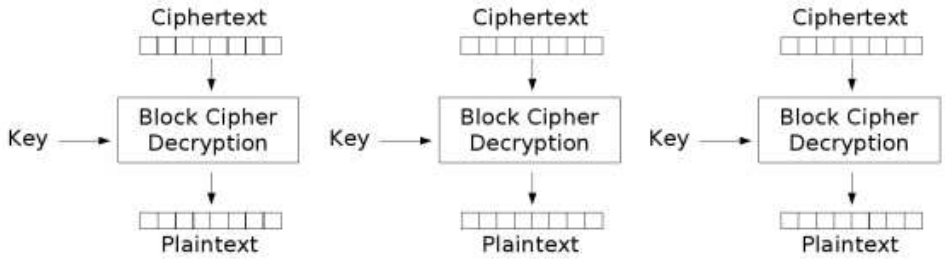


Figure 5.11 Electronic Codebook mode decryption (Brandsma, 2012)

To solve the problem of using the same key and making the blocks independent, a Cipher Block Chaining mode encryption was introduced. This uses an initial vector of the same size as the plaintext which is XORed with the plaintext and input into the cipher with a key to generate the ciphertext. This generated ciphertext becomes the initial vector of the next block in a daisy chain format. This means that each block is dependent on the previous block and breaking one block does not necessarily mean all the blocks are broken. This encryption and decryption mode is illustrated in the following figures respectively (Gattol, 2015).

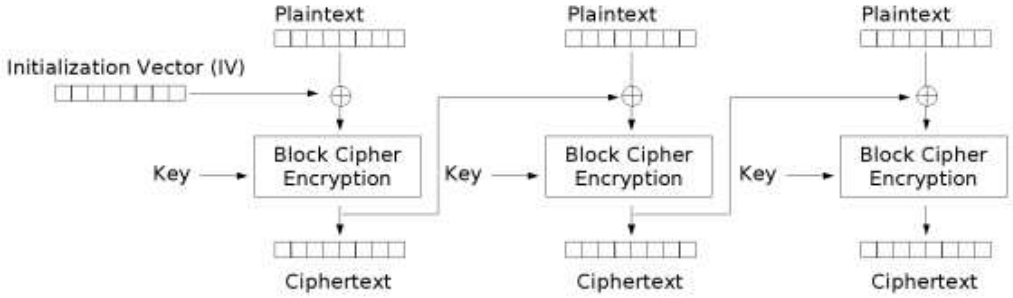


Figure 5.12 Cipher Block Chaining (CBC) mode encryption (Brandsma, 2012)

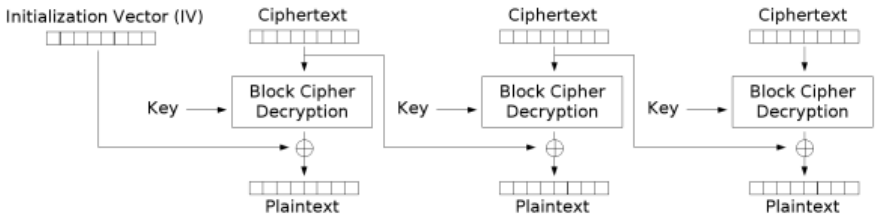


Figure 5.13 Cipher Block Chaining (CBC) mode decryption (Brandsma, 2012)

One of the most commonly known block encryption implementations was the Data Encryption Standard (DES). When it was first introduced it used a key length of 56 bits whereby it is a relatively short length and was only able to encrypt a block length of 64 bits. Later it was developed to have a key length of 128 bits and be able to encrypt 128 bit long block of data. In 1997 DES was broken and was no longer the standard for encryption. To further try to make it more secure, a triple DES scheme was introduced whereby three separate keys were used in sequence. The first key was used to encrypt the data, the second key was used to decrypt the encrypted data and the third key was used to encrypt the data again. The computational efficiency of this scheme was compromised and it was not as secure. The following figure 5.14 gives an illustration of a simple DES algorithm implementation.

DES encryption standard was replaced by the Advanced Encryption Standard (AES). AES encryption encrypts blocks of 128 bits at a time and can use 128, 192 and 256 key length bits. It uses substitution permutation network which is quick to code the algorithm in software (Secretary of Commerce, 2001). The AES-128 encryption consists of 10 rounds of processing (Kak, 2014). The first step to be taken before processing can take place is that the input state array is XORed with first four words of the key which happens in the same manner during the decryption stage, whereby the last four words of the key is used instead. The encryption process steps include a single-byte based substitution step; it also includes a row permutation and a column mixing with the mixing of a round key. The 128 bit block of data that AES encrypts in a single round consists of a matrix of bytes, typically a *4 X 4* matrix. Each of the subsequent bytes of the 128 block occupies each of the columns of the matrix one after the other. This matrix is referred to as a *state array* (Niyaz, n.d.). This means each round of processing processes the input state array to produce an output state array. The output state array of the final round of processing is mixed into a 128 bit output block. In terms of the decryption each round of processing is made up of the following stages, inverse and shifts the rows, then inverse and substitute bytes, then adds a round key and finally inverse and mixes the columns. This means that AES follows a very simple design that is easily implementable in modern hardware and still provides efficiency in performance. The following figure 5.15 illustrates the AES structure for 10 rounds of processing for an encryption key that is 128 bits long. Both the encryption and the decryption side are illustrated in the figure below. AES-256 encryption scheme which is implemented in this research project is an extension of the AES-128 encryption scheme.

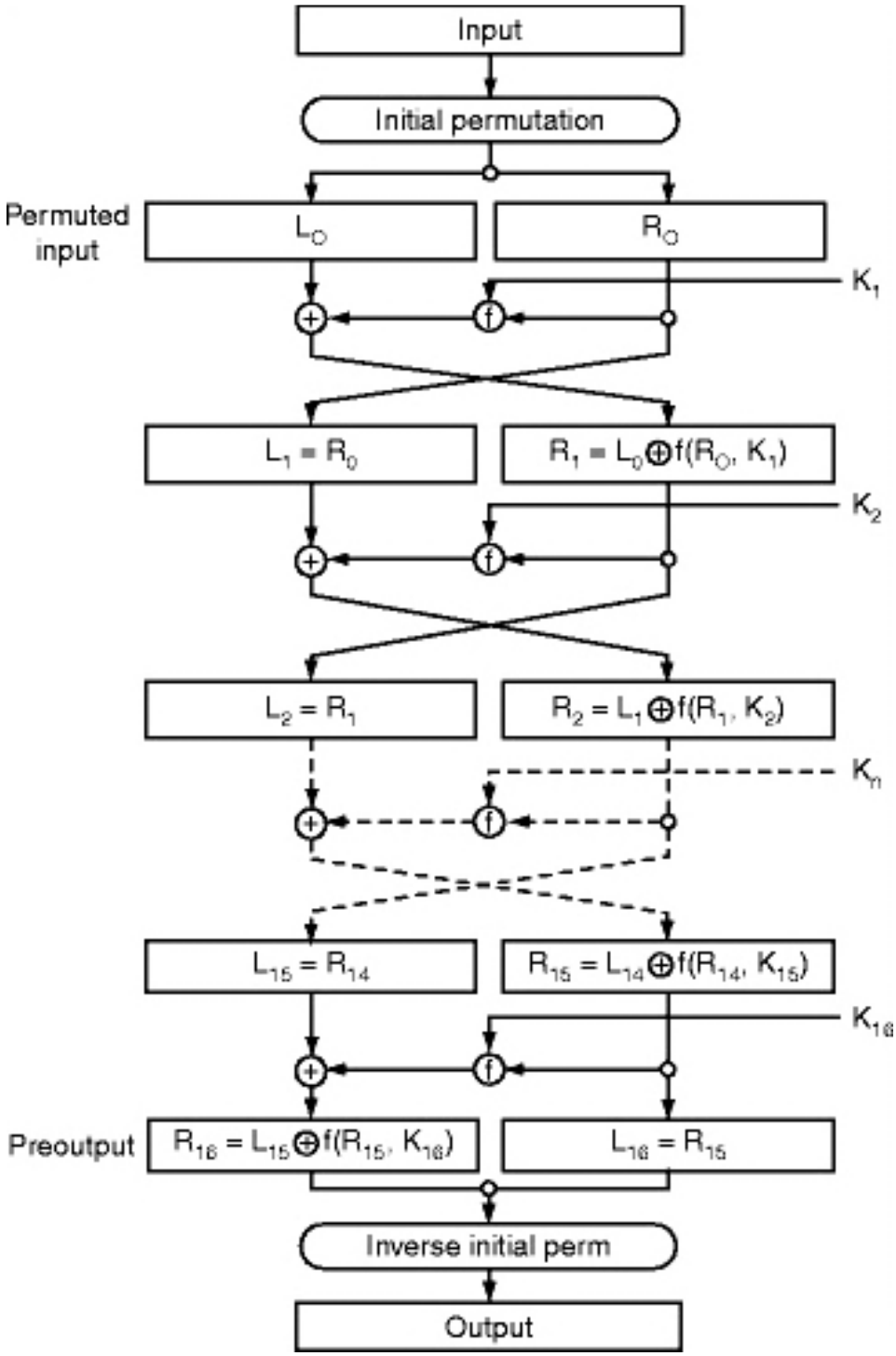


Figure 5.14 DES encryption algorithm illustration (Van Tilborg & Jajodia, 2011)

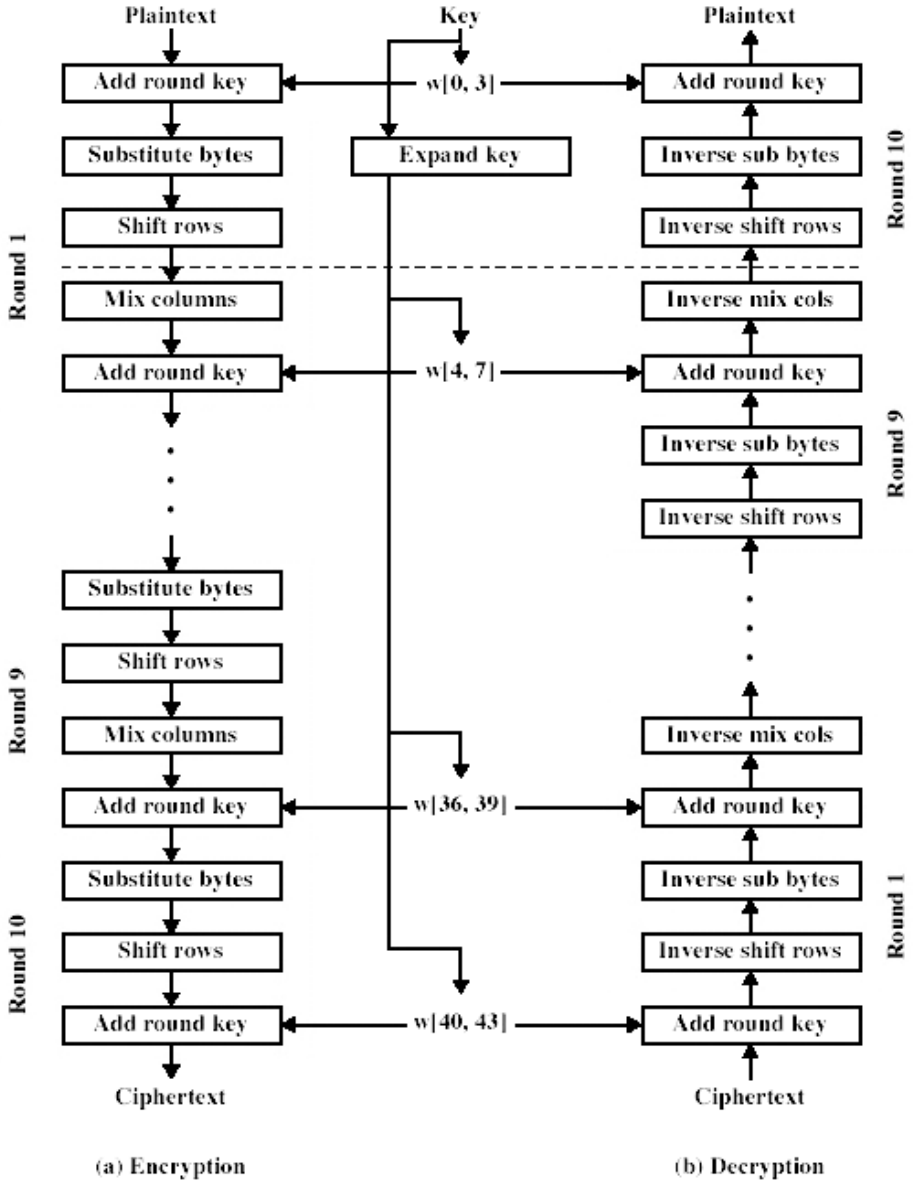


Figure 5.15 AES overall structure (Secretary of Commerce, 2001)

5.8.3. Information Integrity

Data Integrity refers to the safe keeping of information from alteration and manipulation by unauthorized parties. This means that it ensures that information has not been compromised and thus requires information to be verified. However verifying the integrity of information is no enough by itself because the information still has to be coming from an authentic party. Thus authentication and integrity goes hand in hand.

Information integrity can be verified by implementing two techniques namely; Message Integrity Code (MIC) and Message Authentication Code (MAC). The Message Integrity Code is a string of bits which forms part of a message or information which verifies the integrity of a message. The Message Integrity Code is illustrated in the following figure below.

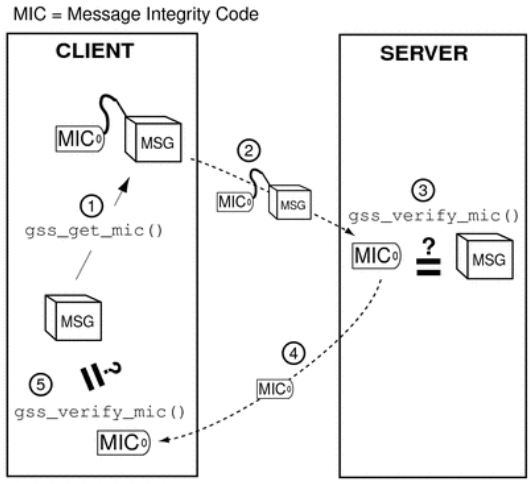


Figure 5.16 Message Integrity Code illustration (Oracle, 2010)

The Message Authentication Code is also a string of data but it authenticates and verifies the integrity of a message using a key. The key can be generated using a symmetric block cipher or using a cryptographic hash function which will be discussed in the next section. The following figure shows a function of a Message authentication Code.

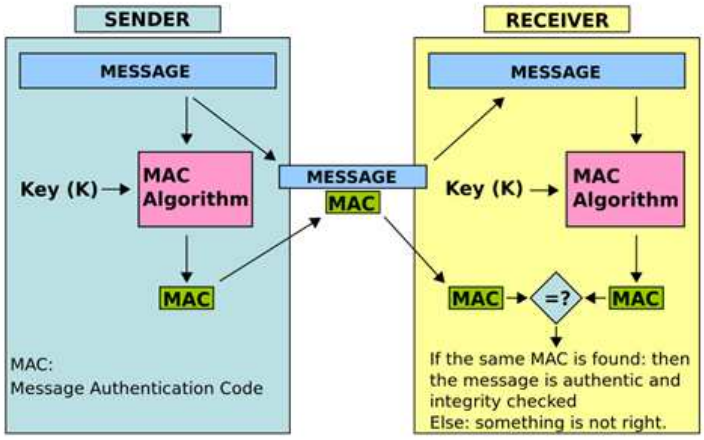


Figure 5.17 Message Authentication Code illustration (Bernstein, n.d.)

5.8.4. Cryptographic Hashing

Cryptographic Hashing refers to a process of taking an amount of data, despite the size and returning a block of data that is of a fixed size. This output is referred to as a cryptographic hash value and the data that is being hashed is called the message or input. Cryptographic hash function must have the following attributes:

- It must compute a hash digest for any input
- It must be virtually impossible to determine the original message from the digest

- Any minute alteration on the message should completely change the digest or hash value
- Finally it should be impossible to find two messages with the same digest.

Cryptographic hashing can be used for password verification, pseudorandom number generation, file or data identification and file or data integrity by comparing the original hash value with the one generated by the transmitted or saved data. There are several examples of cryptographic hash algorithms such as the Message Digest 5 (MD5) which is a 128 bit message digest. MD5 was broken in 2007. Another example is the Secure Hash Algorithm 1 (SHA1) which is a 160 bit message digest and the most recent the SHA-2 and SHA-3.

5.9. Asymmetric Encryption

The following section discusses Asymmetric encryption.

5.9.1. Introduction

Asymmetric encryption uses one key to encrypt and another to decrypt the data. The two keys are the private key which must be kept private and secrete at all times and the other key is a public key which can be distributed to anyone. These public keys and private keys go in pairs to create what is referred to as a public-private key pair. Public-Private key pairs are generated using very large prime numbers multiplied together. In asymmetric encryption, data that is encrypted using a public key can only be decrypted using a corresponding private key; the converse is true as well. Thus the public-private key in asymmetric encryption can be used to ensure authentication of where the information comes from as it would be encrypted by a public-private key pair combination. The primary reason why prime numbers are used to generate the public-private key pairs is that it very difficult to factorize very large prime numbers into the very same prime numbers that were used to create it to consequently get each of the public key and the private key. Thus the use of prime numbers in this context in cyber security is considered a Mathematical sinkhole whereby it creates a security system that is relatively robust and resistant to some attacks. One of the most commonly used asymmetric encryption would be the Rivest, Shamir, and Adleman (RSA) encryption (Anonymous, 1999). It was created in the late 1970’s and became one of the most used asymmetric encryption scheme because of its ease of use and the use of very large prime numbers for public-private key pairs. RSA encrypts data using a symmetric encryption algorithm such as RC4 or AES. The symmetric key is encrypted with the recipient’s public key meaning only the recipient with their private key can be able to decrypt and get the symmetric key to decrypt the data. This is often referred to as end-to-end encryption. The following figure gives an illustration of RC4 encryption scheme.

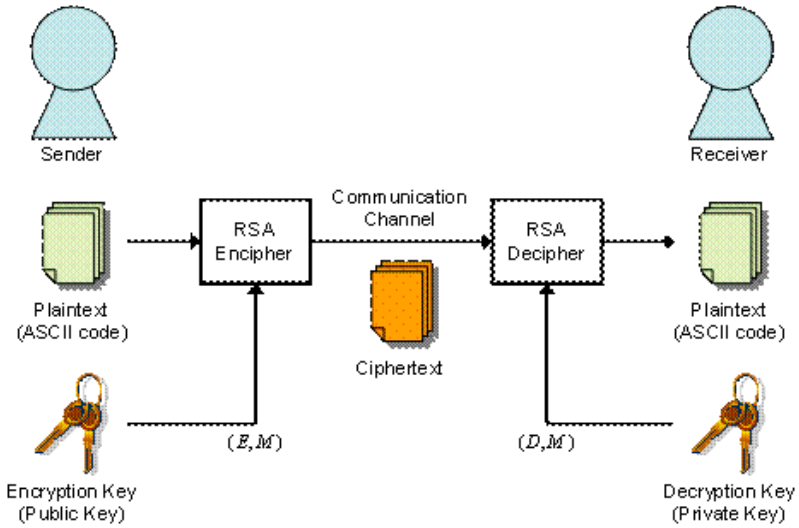


Figure 5.18 RSA encryption scheme illustration (Lib4U, 2013)

With reference to Table 5.1 above, one needs to know when to use asymmetric encryption and when to use symmetric encryption considering both advantages and disadvantages of both techniques. Thus one can use asymmetric encryption when encrypting small amounts of information as the amount of time will impact the performance. Secondly as aforementioned asymmetric encryption can be used when one wants to verify authenticity.

5.9.2. Public Key Infrastructure

Public key infrastructure (PKI) is an implementation created to give a holistic solution to digital identities, particularly to ensure authenticity in asymmetric encryption schemes. The public key infrastructure is based on the trust between a party and a certification authority, between a vendor and a certification authority and lastly between various certification authorities. Public key Infrastructure use digital certificates with encrypted public keys to authenticate users, furthermore to ensure integrity and ensure integrity and ensure non-repudiation of users as discussed in Section 5.4. A public key infrastructure consist of hardware, software, policies and entities in a form of certificate authorities, digital certificates, computers and servers, certificate directories, and registration authorities. The public key infrastructure was intended to be a global system however the problem with that ideal is who would take responsibility for, funding and access to it, thus several authorities interacting with each other. The public key infrastructure can be fully implemented within a utility.

5.10. Authentication Systems

The following section discusses authentication systems.

5.10.1. Secure Socket Layer

Secure Socket Layer (SSL) is used to encrypt network communication and forms part of using certificates and digital signatures for authenticity and integrity verification. Secure

Socket Layer uses keys exchange system and digital certificates to enable secure communication. IP secure and HTTPS uses secure socket layer. The following is a simple example of secure socket layer session implementation.

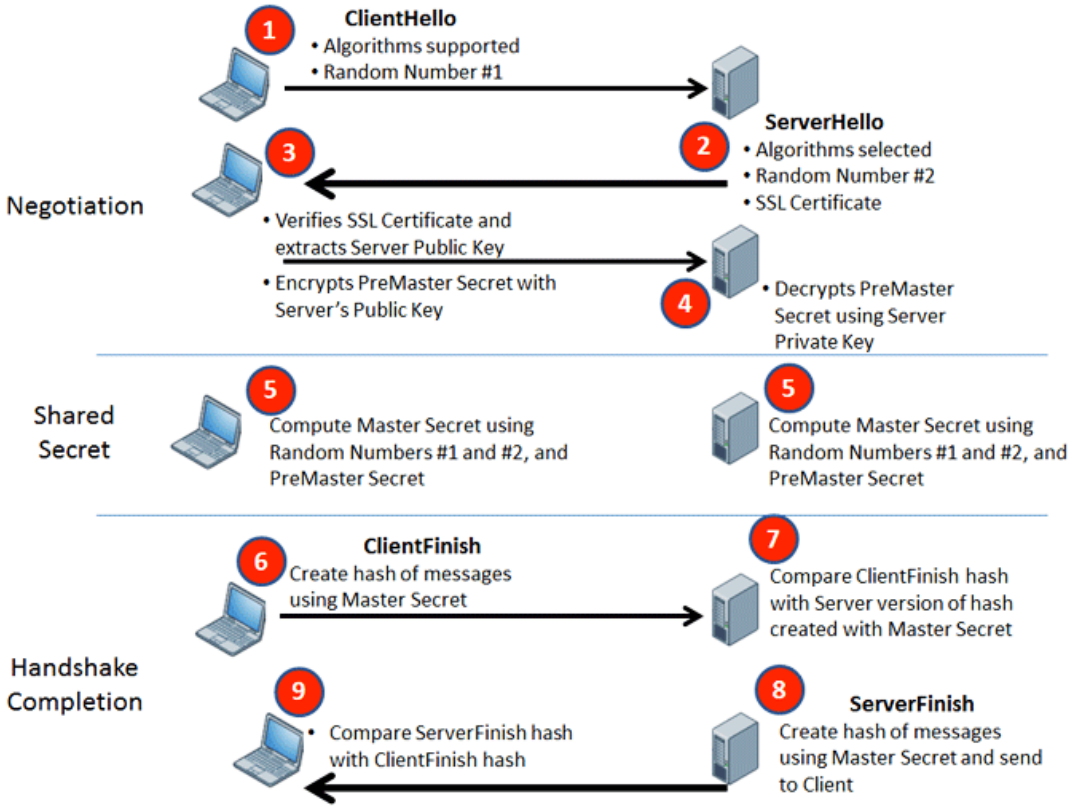


Figure 5.19 A Secure Socket Layer session example (IdenTrust SSL, 2015)

5.10.2. Secure Shell (SSH)

Secure Shell is a means of communication between two computer systems that is secure. It utilizes passwords, usernames and digital certificates to create an encrypted path or tunnel between two computer systems. The advantage of SSH is that it can verify a party and who that part claims to be. SSH can be combined with many other authentication technologies to create a robust authentication system. This can be used in this context with remote terminal units. The weakest link in this system is still the human element in a form of very weak passwords.

5.11. Virtual Private Networks (VPN)

The following section discusses Virtual Private Networks which are essential in creating an MPLS core network for a SCADA system.

5.11.1. Introduction

Virtual Private Network (VPN) is virtual network that enables a party to securely connect to a particular network remotely. Virtual networks are designed or created on a combination of encryption techniques as discussed in the previous sections such as digital certificates which

are used to negotiate for session keys, symmetric encryption to encrypt the actual data or information and also an authentication mechanism to authenticate the user on accessing the network. The performance and connection of virtual private networks must be monitored particularly for time critical functions because some connections might be slow with the latency increase due to the multi-factor encryption techniques mentioned above. There are several virtual private network protocols and a few are discussed in the following sections.

5.11.2. Point to Point Tunneling Protocol (PPTP)

Point to Point Tunneling Protocol is a tunneling protocol and does not necessarily use encryption whereby encryption can be added as a separate layer. This was used in legacy Windows based operating systems equipment. This protocol does not provide much security.

5.11.3. Layer 2 Tunneling Protocol (L2TP)

Layer 2 Tunneling Protocol can be considered as an evolution from Point to Point Tunneling. Layer 2 Tunneling is capable of authenticating users and also has the ability to encrypt data and also compress it over a secure tunnel. What needs to be noted is that the data is one that is encrypted in tunnel and not the tunnel itself.

5.11.4. Internet Protocol Security (IPSec)

Internet Protocol Security it is a layer above the conventional protocols used for Internet. When the Internet and its protocol were first designed it was not done with security in mind, hence why IPSec was introduced later as a layer. IPSec can be used together with other various VPN solutions to provide robust security. IPSec can be used with either keys or digital certificates. For this research project IPSec was used as part of the MPLS network security with the encryption implemented within this design. More details are provided in chapter 4, 8 and 9.

5.11.5. Open Virtual Private Network (OpenVPN)

Open Virtual Private Network is an open source virtual private network solution which uses SSL/ TLS to create an encrypted tunnel.

5.12. Conclusion

Chapter 5 covered various encryption techniques and their basic functions in the context of the algorithm, the keying information and the governing operating principles for that particular scheme. It also covered the difference between symmetric and asymmetric encryption and how the two can be used in a system including advantages and disadvantages of each technique under these schemes. This literature led to the best implementation of encryption in the simulation experiment considering what was discussed in this chapter and also in Chapter 4. The following chapter discusses SCADA vulnerabilities by categorizing them into different sections. It also looks at how these are determined through basic ethical hacking process.

CHAPTER 6

6. SCADA Systems Ethical Hacking and Vulnerability Assessment

The following chapter discusses SCADA systems ethical hacking and vulnerability assessment techniques. The importance of ethical hacking and vulnerability assessment in this context is to ensure that vulnerabilities are identified before an attack occurs and are addressed accordingly, thus vulnerability assessment and ethical hacking is a preventative measure and a preparation measure before something happens. The actual lab experiment of SCADA ethical hacking and vulnerability assessment on an actual SCADA system is beyond the scope of this research project and hence was not conducted; however it is still important in this context to discuss the various vulnerabilities, vulnerability and ethical hacking process as given in this chapter.

6.1. Ethical Hacking Overview

Ethical Hacking is a process of performing penetration and security tests to determine vulnerabilities in computer systems (Van Der Haar & Leung, 2014). A hacker was initially defined as a "a person who enjoys learning the details of computer systems and how to stretch their capabilities as opposed to most users of computers who prefer to learn the bare minimum" (Palmer, 2001). There is a difference between security testing and penetration testing. Penetration is a process of getting into a computer system's network to try and determine weaknesses whereas security testing goes further than breaking into computer systems' networks but also includes the analysis of computer systems' network security and policy as well. However in this discussion the two will be referred to in a generic form as penetration testers. The difference between normal hackers and ethical hackers is that even though they basically perform the same functions, they do so with different intentions and outcomes. Ethical hackers hack systems with the permission of system owners with the intention of identifying weaknesses but hackers do so with an intention of harm.

Penetration Testers should have a form of certification from an accredited organization such as the SANS institute, Certified Information Systems Security Professional (CISSP), etc. Beyond the accreditation, penetration testers and security testers must have technical skills and higher levels of insight into networks. Penetration testers use programming languages such as C, scripting and other collection of tools to perform penetration and security testing.

There are basic models of penetration testing that are industry followed; namely the white box model, the black box model and the gray box model (Leung, et al., 2014). The white box model refers to a model whereby the penetration tester is given information about the computer systems, the network topology and technology on which he will be performing the penetration test. Also in this model the staff and personnel at that organization where the penetration and security test is performed, are informed about the penetration test.

The Black box model refers to a model whereby the penetration tester is not given information about the computer systems, the network topology and technology and also the staff and personnel of that organization are not informed about the test and the penetration tester is not permitted to consult or interview those staff. This means the penetration tester must figure out the network topology, technology and computer system. The last model is the Gray model whereby some information about the network topology and computer system is given to the penetration tester. This is also referred to as the Hybrid model (Leung, et al., 2014). In any of these model a penetration tester must ensure that they perform the security test as per the agreement signed with the systems owner and also that everything done is legal using legal tools and that the process is documented.

6.2. SCADA Vulnerability Assessment Process

This section discusses in detail the process to be undertaken when performing the vulnerability assessment on a SCADA system.

6.2.1. Assessment Planning

In order to perform a security assessment and penetration testing a detailed plan must be formulated and be followed. The plan must include a time frame, costing, and objectives of the assessment, resources and tools requirements. The plan must also include the agreement contract the organization and tester including limitations of the test and also any non-disclosure agreements (Idaho National Laboratory, 2011).

6.2.2. Actual Testing and Assessment Process

The assessment process focuses primarily on weaknesses that are common in various SCADA systems. SCADA assessments or penetration testing must be done on disconnected backup or development systems and not on live systems (Idaho National Laboratory, 2011). The vulnerability assessment must include the SCADA administrators, network administrators, penetration testers and security specialist, Information Technology Personnel and other Stakeholders, meaning that the SCADA penetration testing and vulnerability assessment is more likely to follow a white box or a grey box model as opposed to a black box model (Mell, et al., 2007). Hereto is the process:

1. The first thing would be to compile the scope of work. This outlines the working contract which includes the goals of the assessment, the systems to be assessed, the equipment to be used, the team, cost, non-disclosure agreements, etc.
2. The final scope of work is finalized as a collaborative input from both the assessment team and the system owner.
3. The team can start with a full look at and understanding of the SCADA system deployed with the use of documentation and practical demonstration of the system. This will serve as learning and system understanding process for the team.
4. The assessment equipment, hardware and software can be configured and calibrated and interfaced with system to be assessed.
5. The actual penetration testing and vulnerability assessment can be conducted, with continuous documentation and categorization of the vulnerability using a scoring process

6. A draft final assessment report can be produced for the system owner (Permann & Rohde, 2005).
7. Compile the final report and report key learning areas which do not contravene the non-disclosure agreement can be documented for public release. This is highly important not only for a national but also and international collaborative effort for cyber security for critical infrastructure. This is also discussed in more detail in chapter 10.

6.3. SCADA Vulnerabilities in a OT/IT Environment

The following section presents the high level common SCADA vulnerabilities which can be identified using the processes and plan outlined in the previous sections.

6.3.1. Network Attacks

The follow discusses Network attacks

6.3.1.1. Denial of Service Attack (DoS)

A Denial of Service (DoS) attack prohibits authentic users from using and gaining access to network infrastructure.

6.3.1.2. Distributed Denial of Service Attack (DDoS)

A Distributed Denial of Service Attack (DDoS) is an attack similar to the DoS attack, however the attack uses a number of attack computers or servers such that the attack is not from one source but distributed. This means that in a packet switched network, the network could be flooded with a great number of packets which results in loss of bandwidth.

6.3.1.3. Buffer Overflow Attack

This type of attack exploits code that is written in poor fashion, for example a code that doesn't check or verify the amount of memory utilized by code. In this type of attack the attacker would write a program that overflows the memory buffer in operating system consequently would result in the attacker gaining access as an administrator.

6.3.1.4. Ping of Death Attack

The Ping of Death Attack is when an attacker creates a large ICMP packet in the IP addressing header more than the allowable 65 kilobytes which results in a crash of the destination pointer which cannot process the large ICMP packet.

6.3.1.5. Session Hijacking

This type of attack is when an attacker joins a TCP/IP session and creates an impression to both authentic parties that the other party is communicating with the other authentic party meanwhile it is the attacker.

6.3.2. Attacks on Encryption Systems

This section discussed general and common attacks on encryption systems. There are several reasons why encryption systems would be attacked and these include; accessing the information, to get access to that network, perhaps to do an investigation or as an information recovery process to mention a few. As mentioned in Section 5.4 and 5.6 the strength of an encryption mostly lies in the strength or complexity of the keying information however this

does not guarantee that attacks on a system will not be effective, however it can make difficult or even impossible. Hereto discussed are several encryption systems attacks.

6.3.2.1. Man-in-the-Middle Attack (MITM)

The Man-in-the-Middle (MITM) attack is when an attacker eavesdrops or listens to an encrypted communication channel and try to decrypt that information. It is a very popular type of an attack.

6.3.2.2. Timing Attacks

The timing attack considers and checks how long or the amount of time it takes to encrypt and decrypt information which is then used to figure how long is the key and what the key is. It is a difficult type of attack to perform which is why it is not as popular.

6.3.2.3. Brute-force Attack

The brute force attack tries each and every combination possible for passwords until the correct one is found. This is very time consuming though, although there are advanced brute-force attack systems which start with most common possible password for a key.

6.3.2.4. Rainbow Tables

This is an effective type of attack whereby the hash function is attacked by using extremely large pre-computed tables to quickly figure out hashes of passwords. These pre-computed tables can even be bought online.

6.3.2.5. Birthday Attack

Attackers use this attack to Hash functions (section 5.8.4) to force the hash function to generate the same output for different inputs.

6.3.2.6. Chosen Ciphertext and Plaintext Attacks

This is a reverse engineering attack whereby the attacker would use the ciphertext and plaintext to try and reverse engineer the algorithm or the encryption key.

6.3.2.7. Port Scanning Attacks

Port Scanning attempts to discover which services a host computer is rendering and identifies that as vulnerabilities. Port scanning would report on all closed ports, open ports, filtered ports and also determine the services running on those ports and also the operating system. There are different types of port scans namely;

- SYN scan which is a subtle port scan,
- ACK scan which normally used to get through firewalls,
- XMAS scan whereby different flags are altered,
- Connect scan which completes the required three way handshake.

6.3.3. Physical Access Attacks

Physical Access Attacks refers to physical security vulnerabilities in the physical access control system which may result to cyber security attack. This is extremely important because not only is it a possible flaw in modern industrial networks but also in legacy industrial networks which were designed to be isolated as a security measure. Now if an attacker is

somehow able to gain access onto the SCADA system's infrastructure using physical means this can result in a cyber-attack. The other reason why physical security is important is that an attack is more likely to happen inside than externally (Leung, et al., 2014). Physical access attacks include the following:

6.3.3.1. Physical Access Control

This is the actual physical access to the infrastructure. Most infrastructure premises use deadbolt locks which an average person can pick seamlessly. The other flaw in access control is that most premises do not keep proper records of individuals accessing the infrastructure premises. Some premises utilize magnetic strip access cards which can easily be lost, stolen, or even duplicated with ease. This will be demonstrated even further under social engineering attacks.

6.3.3.2. Key loggers

Physical key loggers can be inserted onto unsuspecting user's computers to capture every key strokes of a keyboard and can later be retrieved to gain critical information such as passwords.

6.3.3.3. Footprinting

Footprinting is a method of acquiring information about a network, in this case an industrial network in order to be able to gain access onto that network. Weaknesses in the network are identified which can be exploited. Several methods of Footprinting are available such as email, using HTTP basics and other web tools.

6.3.4. Social Engineering Attacks

The human element is always the weakest link in cyber security and that is why one of the most effective methods of initiating cyber attackers begins with a social engineering attack. A social engineering attack extracts important information through persuasion, intimidation, coercion, urgency, kindness, position of authority, etc. (Leung, et al., 2014). This attack is harder to protect against as it involves human behavior. It is hypothesized that if the human element is completely removed there would be a great improvement in cyber security. Several techniques are used to instigate a social engineering attack such as;

6.3.4.1. Shoulder Surfing

This attack refers to when a potential attackers reads what a user enters on their keyboard.

6.3.4.2. Dumpster Diving

An attacker goes through trash in order to obtain useful information which can be used with other useful information obtained in other means. The information in the trash can include, passwords on sticky notes, calendars and schedules, resumes, utility bills, financial reports, directories and telephone numbers, etc.

6.3.4.3. Piggybacking or tailgating

This attack is attempting to gain access onto a restricted area by using the human element of kindness, by tailgating authorized personnel entering that restricted area.

6.3.4.4. Phishing

Phishing can be done through various means including emails. For instance a user can get an email about a sick baby and through compassion they can open attachments which may result in an intrusion.

6.3.5. Operating Systems Vulnerabilities

There are several vulnerabilities which can be exploited within an operating system. Since most SCADA systems run on Windows based machines, this section will discuss briefly Windows based operating system vulnerabilities. These vulnerabilities are as follows:

6.3.5.1. New Technology File System (NTFS)

A File system can be vulnerable, particularly the New Technology File System (NTFS) because of its inherent feature of being able to hide information behind existing files without affecting the function and size of the other information (Buttner, 2008).

6.3.5.2. File Allocation Table

The biggest problem about a Windows based file allocation table is that it does not allow for an access control list (ACL) at the individual file level. The ACL is important for administering permissions on files particularly in a multi-user environment (Leung, et al., 2014).

6.3.5.3. Remote Procedure Cell

This is another great vulnerability as this remote procedure cell allows software running on one host to be run as code on another remote host.

6.3.5.4. Structure Query Language (SQL) Server

The most common vulnerabilities in SQL servers environment of which some SCADA systems uses as form of its databases is the human element where by system administrator accounts have no passwords or the username and password are left default. This would give administrative rights to an intruder if access is gained.

6.3.5.5. Passwords and Authentication

The issue about passwords as mentioned in encryption discussion in the section above is the human factor element whereby a user must select a password they can remember. This results in very weak passwords, which can be cracked using social engineering techniques and also computing techniques.

6.3.6. Legislative constraints

The legislative constraints are another big issue in SCADA and can be considered as vulnerability on a generic basis that most of our laws are not able to keep up with the rate at which technology is changing. The second legislative constraint not only in the context of SCADA systems cyber security is the issue of jurisdiction, whereby it is difficult to arrest and persecute a criminal should an attack be coming from another country. Another issues is that with regards to digital evidence to be obtained, a warrant is required especially if the property of the suspected perpetrator is to be analysed. The bureaucratic process of this can compromise the time required to obtain digital evidence without the perpetrator getting the

opportunity to alter the state of that evidence. The other challenge with obtaining a warrant is the difficulty of demonstrating probable cause. Thus legislative constraints can be deemed as vulnerabilities.

6.4. Conclusion

Chapter 6 discussed the vulnerabilities in SCADA systems. It categorized them into network, encryption, physical, social engineering, operating system and legislative vulnerabilities each with details of specific attacks. The chapter also discussed the process which can be followed under ethical hacking to determine these vulnerabilities. Chapter 6 concludes the non-technical component of this research project. Chapter 7 onwards discusses the actual simulation experiment. It gives the simulation experiment setup and configuration.

CHAPTER 7

7. Lab experiment and simulation setup and configuration

This chapter gives the overall description of the experiment and simulation performed. In Section 7.1 it introduces the experiment's objective, scope and area of focus. In Section 7.2 it briefly gives the experiment equipment and tools required and discuss in Section 7.3 the SCADA services that we simulated during the experiment. In Section 7.4 the lab experiment and simulation setup overview is given. Section 7.5 discusses the MPLS/IP network design and configuration including the encryption. The last section discusses how the SCADA and RTU was setup and calibrated.

7.1. Experiment Objective

The key research question for this project is *"whether using encryption in SCADA systems, the services performance requirements are still met in OT IT environment over an MPLS core network"?* The core focus area of the research project was to determine if Encryption is used, will SCADA performance requirements still be met. This means that the research project specifically focuses on the Encryption, which is a single component of the whole security value chain, versus SCADA services performance requirements over packet switched network. This means that the experiment will solely focus on the encryption portion and not any other part of the cyber security value chain pertaining to SCADA cyber security. Other aspects of the SCADA cyber security value chain are outside the scope of this research project and can be considered as future work or other areas of research. The experiment was to determine the three main aspects of encryption which are integrity, availability and confidentiality if they are met as discussed in Chapter 1 and ensures that SCADA service requirements are also met. The experiment design and setup will ensure that encryption meets the requirements of integrity, confidentiality and availability over a packet switched network (MPLS/IP network). Furthermore, SCADA services will be tested over this encrypted MPLS/IP network to determine if the service performance requirements as per Table 2.1 are met. The key performance measure and indicator of testing the hypothesis will be the overall latency of each SCADA service experienced over the encrypted MPLS/IP network compared to the basic required one as per Table 2.1. This means that if the latencies experiences by these services are lower than the ones specified as basic performance requirements then the experiments would have been successful and the converse would be mean they were not successful. Another key performance measure will be availability requirements of SCADA services as per Table 2.1. If the availability is higher than stipulated in Table 2.1 then the experiment was a successful and the converse would be true. The integrity and confidentiality aspects of the encryption was achieved during the design and configuration given in section 7.5 with the results discussed in Chapter 9.

7.2. Lab experiment equipment and tools used

The following equipment and tools were required for the successful completion of the lab and simulation experiment.

- 1x HP laptop: Running the SCADA Simulator and Serial/TCP conversion platform

- 1x Toshiba laptop: Running the GNS3(IP/MPLS core network), Loop-back network, TCP/IP to Serial conversion
- 1x RS232 RTU 120ohm's cable
- GNS3: IP/MPLS core network simulator
- Enterprise IOS installed in GNS3 for CISCO routers (incl. Security Modules)
- Com By TCP (TCP/IP-Serial Converter)
- Serial/IP Control Panel (Serial-TCP/IP Converter)
- IST Talus ERTU (Enhanced Remote Terminal Unit)
- ERT1-13800 Dual Isolated RS-232 Card
- SCADA Simulator: FieldComm
- 1x MacBook Pro Laptop
- NETGEAR Local Area Network Switch

7.3. Description of simulated SCADA services

The following SCADA services were simulated during the experiment to measure if their performance requirements are met when their signals are transmitted over an encrypted MPLS/IP core network. This meant each of the SCADA service signals had to be received and executed by the RTU within the acceptable time period as per that particular service's performance requirement. The time period include all latencies such as the propagation time, encryption and decryption time and other latencies which were supposed to be within a certain acceptable period of time. This means that latency was a key measure of performance as discussed in the introduction and as given in table 2.1. Hereto are the main SCADA services;

7.3.1. Telecontrol

Telecontrol in FieldComm (SCADA simulator) was represented by two modes of messages which are Direct Operate and Operate. For Direct Operate mode the instructions that were sent to the RTU via the encrypted MPLS/IP core network are, *close, trip and null* which all three operate a substation relay that is normally connected to the RTU. Similarly, for Operate mode, the instructions that were sent through the encrypted MPLS/IP network are *close and trip.*

7.3.2. Substation Condition

The substation condition for this experiment was represented by using sending an integrity poll message, sending a cold restart message and sending a warm restart message as the RTU used was relatively old and does not offer other substation condition messages.

7.3.3. Remote Monitoring/Metering

At the time of the experiment there was no metering equipment connected to the RTU thus the monitoring/metering function was simulated using the "Read and Write" functions. This suffices as it a function of getting information from the IED through the RTU. The function used to represent the monitoring is "*read date and time*". A time stamp was read from the RTU to represent the acquisition of information such as metering. The other function used was the "*write date and time*". If the performance requirements for these functions were

satisfactory then it can be extrapolated that they are satisfactory for reading metering and monitoring information.

7.3.4. RTU Error Statistics

RTU error statistics is list of errors encountered when the SCADA simulator (FieldComm) communicates with the RTU, in this case over the encrypted MPLS/IP network. This error report will be generated and used as a measure to determine the level of availability of the encrypted MPLS/IP core network and the ability of the SCADA services message to propagate through this network without many issues. Availability is an integral part of encryption, which are confidentiality, integrity and availability. The following error types are measured:

- *Bad CRC:* counted when a bad CRC is received from the RTU.
- *User Data Overflow*: SCADA (FieldComm) allows the application layer fragments to be 2048 bytes and if this is exceeded by a slave (RTU) then it is counted as an error.
- *Sequence Errors*: this error is counted when a Transport layer sequence number arrives not in the correct sequence from the RTU
- *Confirm Retries*: this error is when SCADA performs a Link Layer Retry after an initial *Timeout*
- *Confirm Failure*: This error is incremented after the pre-determined number of retries is exceeded without success and SCADA fails the link transmission.
- *AL No Responses*: This refers to errors whenever no response is received to a request within the configured Application Layer Response Timeout.

7.4. Lab experiment and simulation setup overview

The overview of the Lab experiment simulation setup is as follows.

- A HP laptop was connected to a NETGEAR network Switch.
- A Toshiba laptop was connected to the HP laptop also via the NETGEAR network switch.
- The Switch was also connected to the Eskom Enterprise network.
- The Toshiba laptop was connected to the RTU via the RS232 cable.
- IP addresses assignment. The IP addresses assignment for each interface, Ethernet card, emulator and any other devices in this simulation experiment had to ensure they were within the same subnet to ensure correct routing.

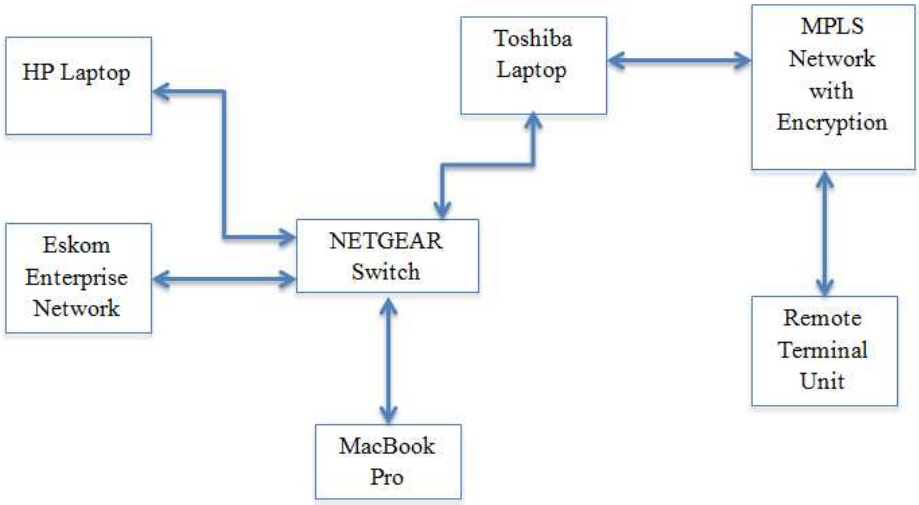


Figure 7.1 Experiment and Simulation block diagram

The following various images are illustrating the actual experimental setup.



Figure 7.2 Illustration of a D20 RTU and the IST ERTU used for the simulation and experiment



Figure 7.3 The Toshiba laptop running the MPLS/IP core network on GNS3 simulator, TCP/IP-RS232 conversion and it is connected to the RTU via RS232



Figure 7.4 The HP laptop running the SCADA simulator and RS232-TCP/IP conversion which transmits the data to the Toshiba laptop's GNS3 network via a network switch



Figure 7.5 An illustration of the switch and the MacBook Laptop connected to generate traffic and to try and send messages to the network



Figure 7.6 An overview of the overall simulation and experiment setup

The functional process for the experimental setup and simulation was as follows.

- The HP Laptop was running the SCADA Simulator (FieldComm) terminating the SCADA services messages out of a COM port.
- The SCADA services messages were then encapsulated into a TCP/IP data stream using a RS232- TCP conversion emulator. The IP Address for the HP laptop was 192.168.0.2
- The Emulator was configured as a client accepting data from the SCADA COM port and encapsulating it and transmitting it through the Ethernet card via a LAN network through the network Ethernet switch/router to Toshiba Laptop with IP Address192.168.0.3.
- The LAN network of the Toshiba was connected to the GNS3's IP/MLPS core network as per GNS3 external LAN configuration.
- The new transmitted encapsulated TCP/IP SCADA services messages were then encrypted using the AES 256 encryption standard and transmitted over the GNS3's IP/MPLS core network.
- The data was then decrypted after propagation through the virtual IP/MPLS core network in GNS3, on the exiting interface of the router connected to a loopback facility.
- The loopback network was used to re-route the data internally to the laptop to be able to convert it back to RS-232 before sending it to the RTU.
- The data from the loopback was send to the TCP/IP-RS232 conversion emulator and it was converted to RS232 format removing the TCP/IP encapsulation.
- The data was then sent through COM1 port of the Toshiba laptop to the RTU for SCADA services messages execution by the RTU.
- The RTU would execute the service messages and gave responses or feedback. The responses from the RTU back to the SCADA simulator would go through the same process in reverse.

The following figure 8.7 gives an illustration of the experiment test set-up bench.

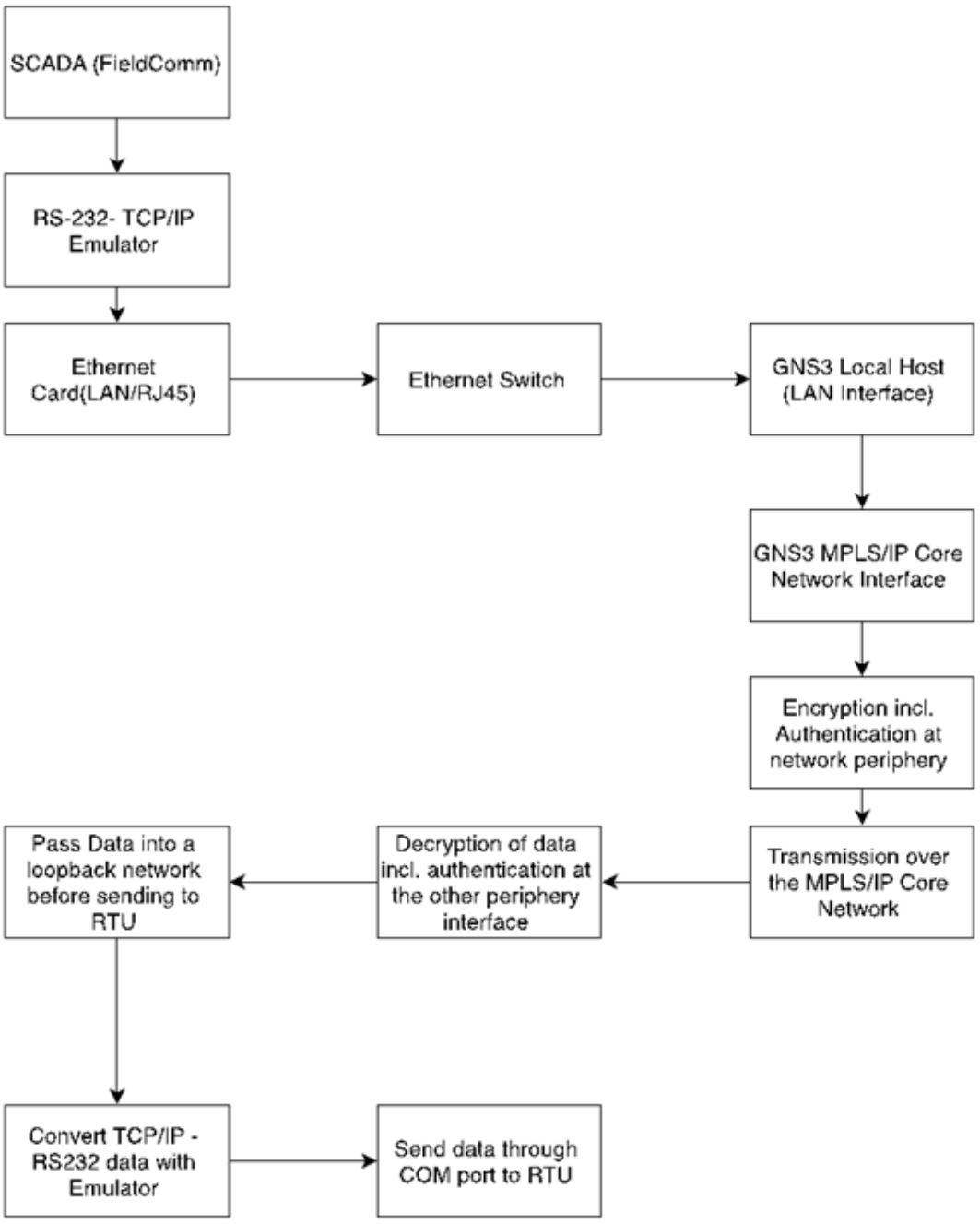


Figure 7.7 Functional block diagram for the lab experiment and simulation

7.5. MPLS/IP core network design and configuration

This section discusses the MPLS/IP core network setup and routers configuration including the AES-256 encryption and decryption.

7.5.1. Basic Network design and Router Configuration

The basic router configuration in GNS3 after the successful installation and initialization of GNS3 involved for each router configuring the various terminals to be used. Since the routers were connected to an external network, they needed to be configured as such. To be able to convert the information from TCP/IP back to RS-232 for the RTU to be able to process, a loopback configuration was used to re-route the data from the GNS3 IP/MPLS core network back into the Toshiba laptop (host) such that the TCP/IP-RS232 emulator can be able to facilitate the conversion. Herewith is the configuration of the routers.

- Firstly the various interfaces of the routers were configured, with Router 1 connected to Router 2. The other interface of Router 1 was connected to the Local Area Network interface card of the Toshiba Laptop which was connected to a switch connecting other devices and other networks such as the Eskom Network in this case. The other interface of Router 2 was connected to the loopback network to be able to convert the information back to RS-232 for processing by the RTU as discussed above. The other interfaces of Router 1 and 2 were connected to other routers to complete the IP/MPLS core network.
- The interfaces were configured as Fast Ethernet and were assigned as discussed above.
- The next phase involved navigating to the global configuration mode of the routers.
- Then the IP CEF mode was configured
- The IP domain look-up was configured
- The Router Rip version was assigned, version 2 in this case
- The network was assigned. This means only devices in a particular range of IP addresses can have access or be part of this network. This is an additional security feature/ layer as recommended for an IP/MPLS core network in Chapter 4.
- The next step involves configuring the IP Addresses of all the interfaces, each interface at a time.
- Assign the duplex mode
- Assigned the speed
- Then activate the fastEthernet interfaces
- Then verify the configuration (shown in the results section in Chapter 8)

The loopback allowed GNS3 to connect to the host computer and it was configured as follows.

- Firstly the loopback adapter had to be created by adding legacy hardware in Microsoft Device Manager.
- The next step involved configuring the loopback adapter, assigning the correct IP Address and Subnet in the range according to the network the MPLS/IP core network fell into.
- The next stage was connecting the loopback network onto the GNS3 MPLS/IP core network which was a simple matter of configuring the interface of the correct router and selecting this loopback adapter as it appears under a list of all networks available.

The following figure gives an illustration of the basic simulated IP/MPLS core network connected to a LAN on periphery and to a loopback (consequently to the emulator and a RTU) access networks on the other end.

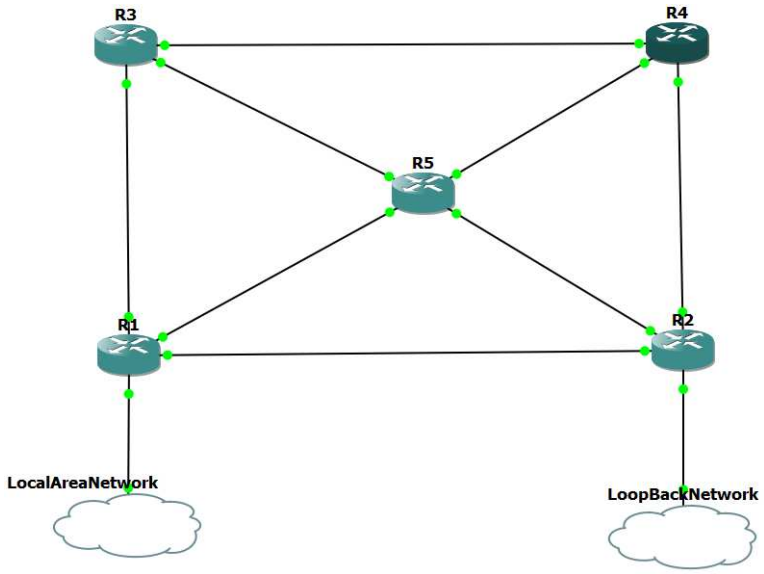


Figure 7.8 An illustration of an IP/MPLS core network connected to two access networks (LAN and Loopback)

The following figure gives an illustration of the complete connection, particularly what is happening within the clouds as illustrated in the figure above.

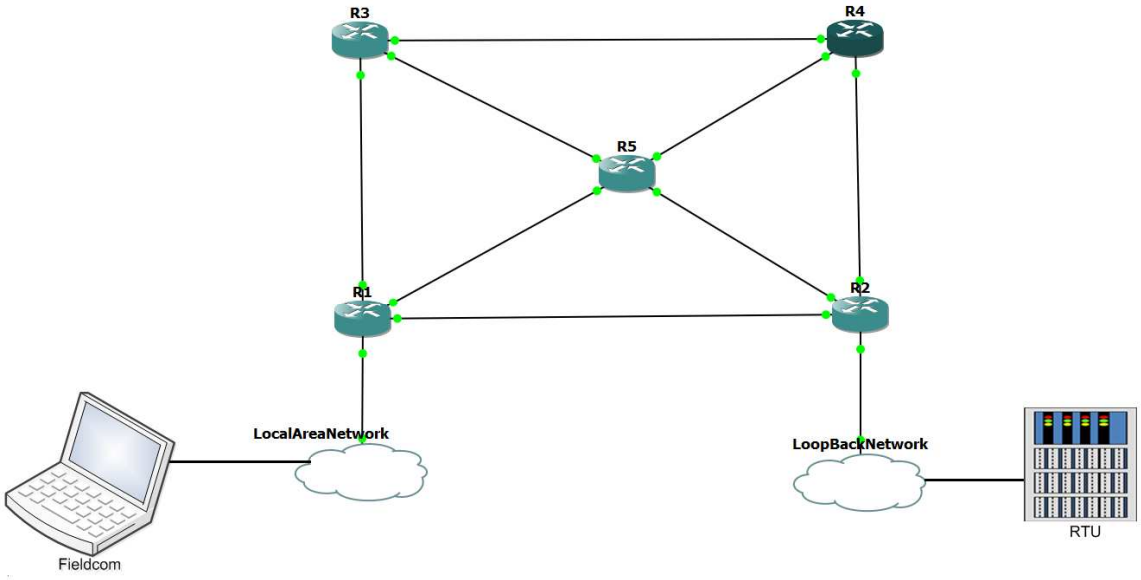


Figure 7.9 The GNS3 network connected to SCADA and the RTU through the access networks (LAN and Loopback) illustration

The full configuration of the routers is given in Appendix B

7.5.2. AES 256 Encryption in the MPLS/IP network

This section discussed how the encryption was configured onto the MPLS/IP core network configured above. The method of encryption chosen to be used for this research project was the symmetric Advanced Encryption Standard (hereto, AES) encryption standard, particularly the AES-256 using the IPSEC tunnelling mode. The 256 in AES simply refers to the number of bits of the key used as discussed in Chapter 5. This section discusses the logical steps undergone to configure the routers on the periphery of the network for encryption and decryption. This configuration process was based on the basic router configuration process from the above Section 7.3.1. The process is as discussed in detail in the following steps.

1. The first thing that needed to be done after entering the *config* mode was to enable the *isakmp.* Then after the enablement, choose a policy standard that would be used, in this case *policy 10* was chosen. The policy is needed because the simulated router initially attempts a connection in a particular mode.
2. The next thing was to configure the mode used for authentication. For a substation environment and based on literature survey as discussed in Chapter 5 the best option was to use a *pre-shared key (PSK).*
3. The next step involved enabling the Hash method, which in this case *SHA* was used. The other mode is MD5 but as discussed in chapter 5, *SHA* is much better than *MD5*.
4. The following step involved the configuration of the encryption algorithm. There are various different algorithms which can be configured depending on strength and use. In this environment it is important to strike a balance between performance and strength of the encryption algorithm. As discussed above the *Advanced Encryption Standard-256 (AES-256)* was configured.
5. The following step involved the enablement of the Group which configure the modulus size of the Diffe-Hellman key Exchange (Moonie, 2011). The group used for this configuration is *Group 5* which is the 1536-bit Diffie-Hellman group. Then subsequent to this was to configure the *lifetime* which is the time measured in seconds for security association, for this configuration the maximum which is 86400 was configured which is the default.
6. The next step was to configure the *IPSEC tunnel* mode using the *256 bit AES* and *SHA.*
7. Then the next step involved the defining the traffic that was to be secured via the encrypted channel.
8. The next step was to configure the range of IP on the other side or periphery also matching the above defined traffic to the other router in the periphery and configuring the transform set to be used.
9. The next step was to assign the configured crypto map to the outgoing interface.
10. The last step involved the testing and verification of this configuration.

The full code for configuration is given as part of Appendix B.

7.6. SCADA and RTU simulation set-up calibration

This section discusses the RTU and SCADA set-up.

7.6.1. Routers configuration calibration

This initial set-up involved connecting the routers back to back with a direct link between the two routers. A ping command was used to ping the X21 and the interfaces to determine if there was communication between the two routers and that the configuration was done properly. If the ping was successful then it would have implied that the configuration was successful. This calibration set-up is illustrated in the figure below.

7.6.2. SCADA (FieldComm) Configuration

Herewith is the configuration of the SCADA. The SCADA in this case uses Serial communication, which will be converted to packet switched. The channel used for the Master is COM1 and Channel B uses COM3, with the baud rate of 9600 kbps, data length 8 bits, 1 stop bit and no parity. The above configuration is given in the screen shot in the following figure.

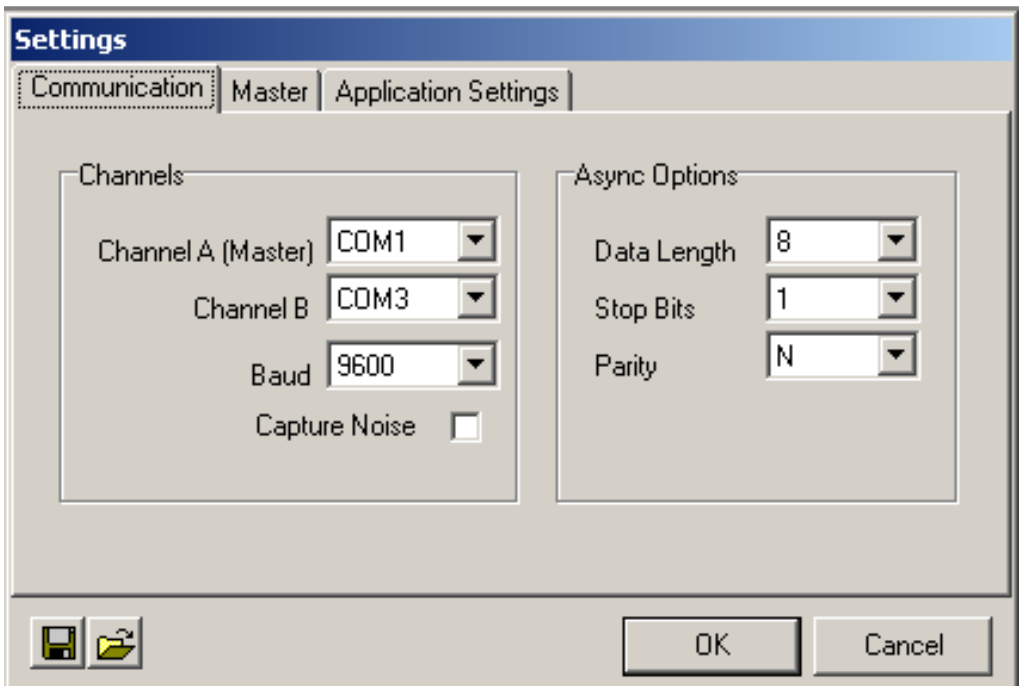


Figure 7.10 An illustration of SCADA (FieldComm) configuration for serial communication

The address of this SCADA station is assigned to 3, meaning RTU configured to recognize Master station 3 will only communicate with Master or SCADA with address 3. An intermesssage delay of 20ms is assigned. The maximum number of RTU's which can be can be connected to this master station of SCADA is 100. The other settings are left as default. The following figures give an illustration of the further SCADA configuration.

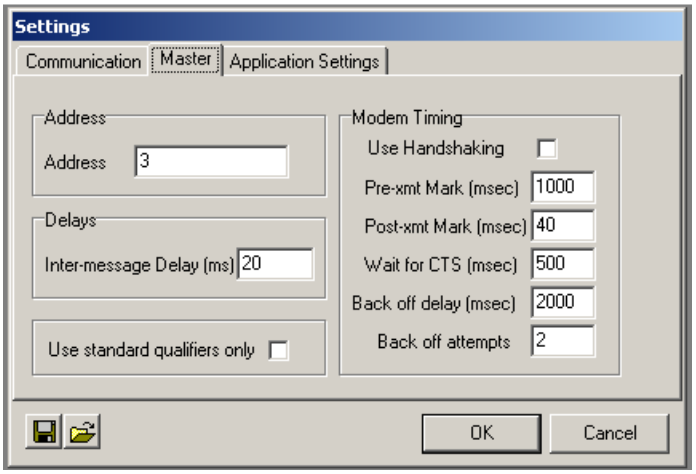


Figure 7.11 An illustration of SCADA settings with the Address of the SCADA

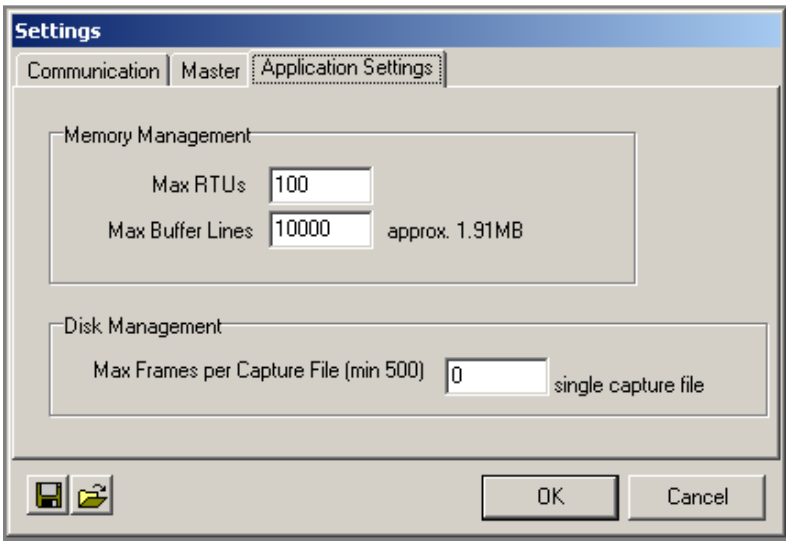


Figure 7.12 SCADA settings configuration continued

7.6.3. RTU Configuration

The following section discusses the configuration for the Remote Terminal Unit (RTU) within the SCADA module or application. The address assigned to this RTU is 1234 and an E-RTU for the purposes of this lab experiment was used. The response Timeout is assigned to be 200ms, meaning if messages are sent and delay for more than 200ms the request will time out. Polling the RTU was also enabled to allow the SCADA to poll the RTU with the poll interval assigned as 1s. This constant polling of the RTU after 1s was essential for constantly generating traffic to the RTU.

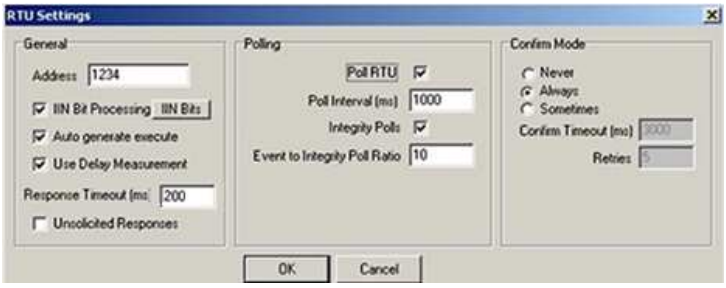


Figure 7.13 An Illustration of RTU settings configuration

The commands sent from the SCADA to the RTU are referred to as messages. These messages basically perform various functions to achieve various goals such as Telecontrol, substation condition and post analysis, remote metering and monitoring, Teleprotection, etc. The following figure shows the command prompt for selecting a particular message or command to be sent to the RTU via the network.

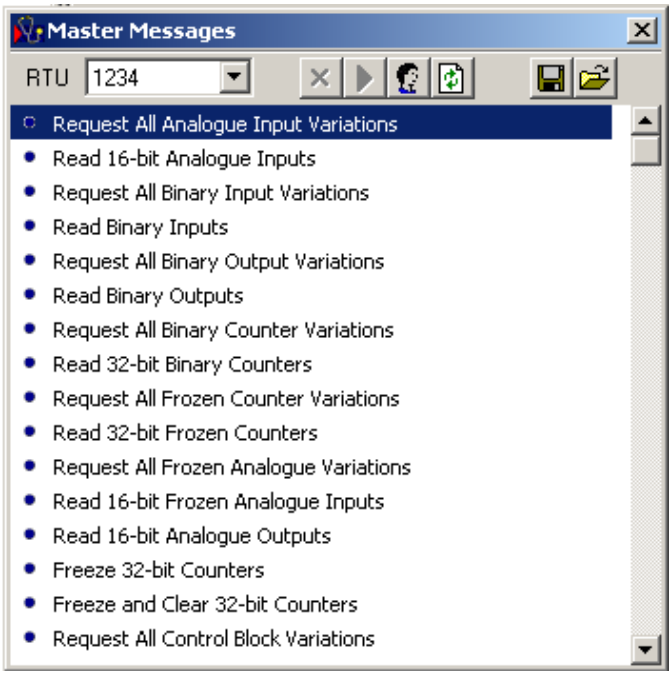


Figure 7.14 An illustration of various messages which can be sent to the RTU for execution

7.7. Conclusion

Chapter 7 covered the simulation experiment setup and configuration. It discussed the tools and equipment required for the simulation experiment and gave the overall system overview. The setup and configuration of the IP/MPLS core network and the encryption thereof was discussed including the SCADA simulator (FieldComm) and the RTU configuration. The following chapter gives the results obtained having conducting the simulation experiment.

CHAPTER 8

8. Experimental and Simulation Results

8.1. Introduction

Chapter 8 follows from the discussions of Chapter 7 and discusses the results of the experiment and simulation conducted for this research project. The results in this chapter are based on the experimental design, setup configuration, etc. discussed in Chapter 7. These results in this chapter are only focused on the key research question, which is *"whether using encryption in SCADA systems, the services performance requirements are still met in OT IT environment over an MPLS core network"?* The key focus area of the research question for this MSc 50/50 mini-thesis is encryption versus SCADA services performance requirements, and no other areas in the whole cyber security value chain. Thus the key outcome measure for this research project results would be based on Table 2.1 in chapter 2, which is latency and availability for each of the services over an encrypted MPLS/IP network versus the basic service requirements. Various sections in this chapter will present results conducted in a systematic manner in order to build on some of the results as stages. These stages are discussed in Chapter 1, Section 1.4 through to Section 1.8.

Section 8.2 shows the MPLS network's successful function on its own. Section 8.3 gives results of the encryption/decryption designed and applied at the network periphery. Section 8.4 gives results of connecting the SCADA to the RTU directly without any encryption, without the IP network (8.4.1). Then Section 8.4.3 gives various successful connections through the encrypted MPLS core network with encryption and decryption functional as tested in Section 8.3. Section 8.4.3 is the final integration stage just prior the various services commands are sent over this encrypted MPLS/IP core network. The various sub-sections of Section 8.5 give the results of the various SCADA services over this encrypted MPLS/IP network. Section 8.6 gives the results of the RTU Error Statistics. From Chapter 7, Section 7.3.4 it was discussed that RTU Error Statistics will be used as measure to determine the level of availability of the encrypted MPLS/IP core network and the ability of the SCADA services message to propagate through this network without many issues and the losses associated. This chapter then concludes with Section 8.7 whereby the results of the experiment and simulation are analysed and summarized.

8.2. IP/MPLS Core network Routers Configuration Results

The following figure gives an illustration of the successful configuration of the Router 1 and Router 2 without the encryption. It shows that all interfaces have been configured correctly and that they are working fine as per the status "OK?" given as yes for each interface. The figures also show the IP addresses for each interface where it was assigned. Where the IP address is not assigned it is because the type of interface chosen for this particular interface during the routers configuration is a switch and thus no requirement for assigning an IP

address. This was done for interfaces connected to other routers that are not essential for this experiment, which is router 3 and 4. Router 1 and 2 are the critical ones and where the encryption and decryption was applied to preserve processing power as the computer that was used for the experiment is not powerful. However the configuration design can be extended to an innumerable number of routers if the processing power is available. Thus for this simulation experiment, router 1 and router 2 suffices.

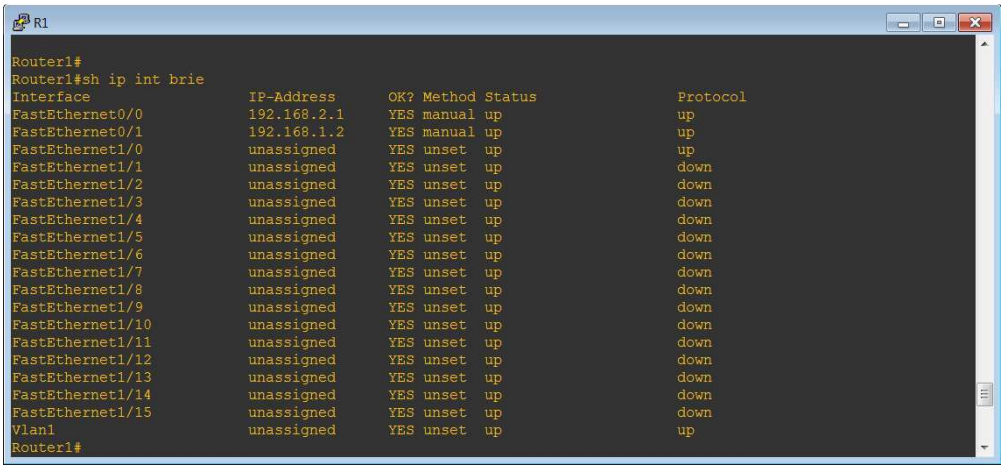


Figure 8.1 An illustration of Router 1 interfaces configuration

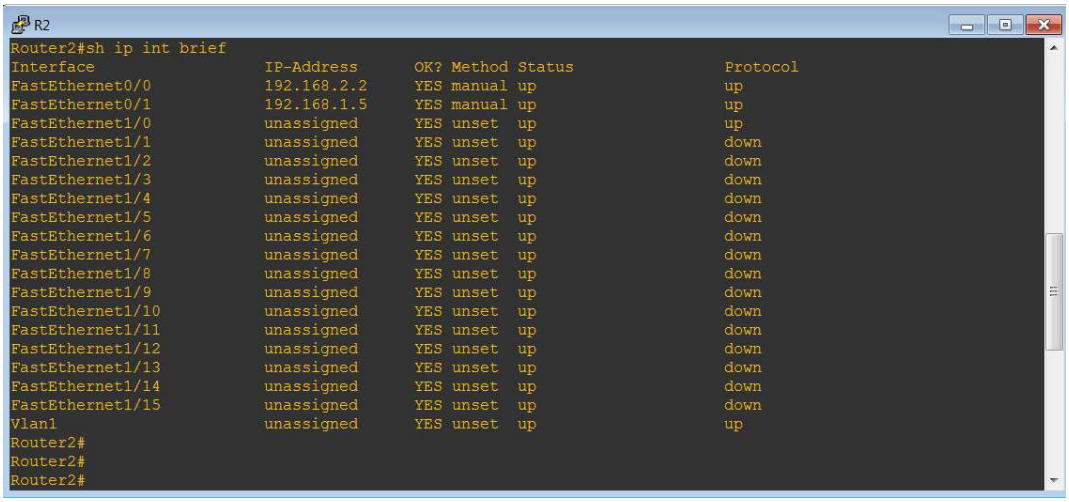


Figure 8.2 An illustration of Router 2 interfaces configuration

Router 1 pings

The following figures give an illustration of successful communication between the various interfaces through successful pings to the various interfaces.

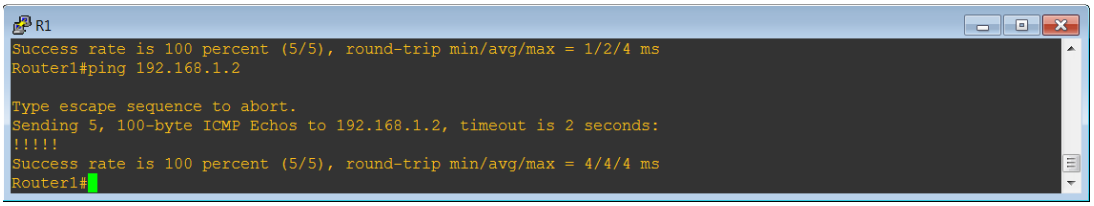


Figure 8.3 Router 1 to 192.168.1.2 (interface facing the LAN network) interface successful ping

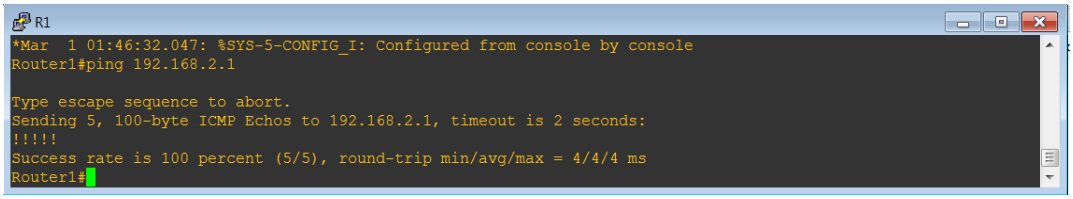


Figure 8.4 Router 1 to 192.168.2.1 interface successful ping

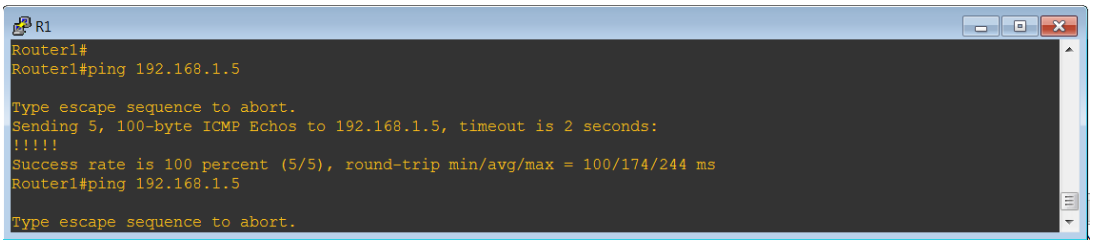


Figure 8.5 Router 1 to 192.168.1.5 interface successful ping

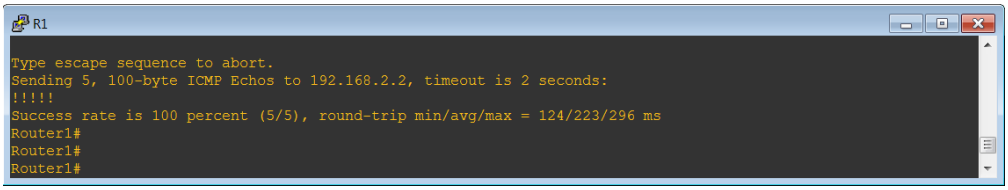


Figure 8.6 Router 1 to 192.168.2.2 interface successful ping

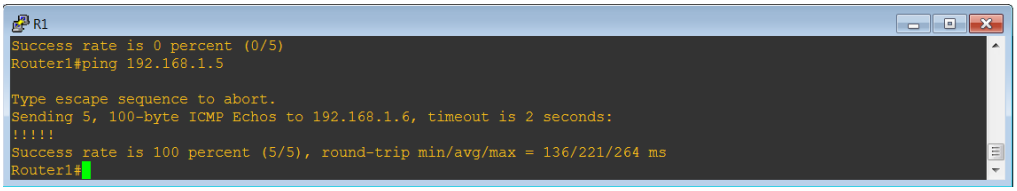


Figure 8.7 Router 1 to 192.168.1.6 (Loopback network) interface successful ping

Router 2 pings

The following figures give an illustration of successful communication between the various interfaces through successful pings to the various interfaces by Router 2.

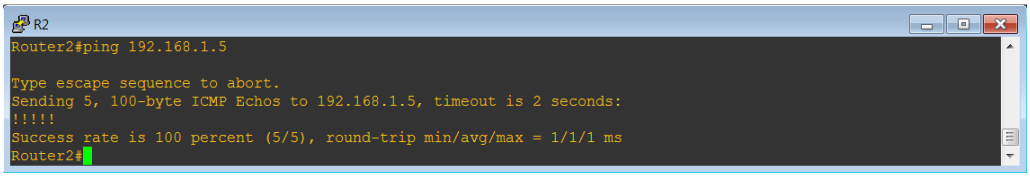


Figure 8.8 Router 2 to 192.168.1.5 interface successful ping

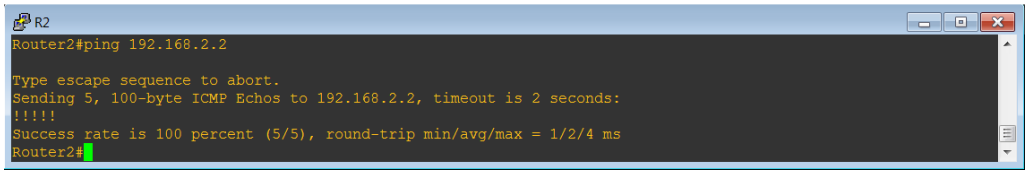


Figure 8.9 Router 2 to 192.168.2.2 interface successful ping

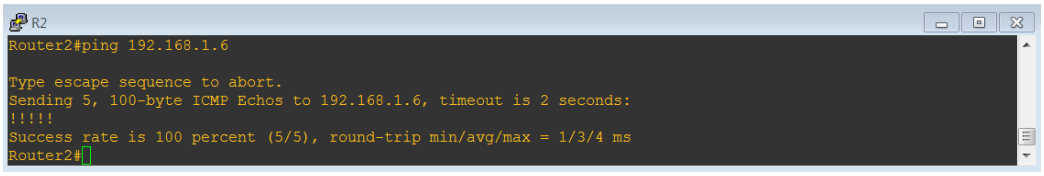


Figure 8.10 Router 2 to 192.168.1.6 interface successful pings

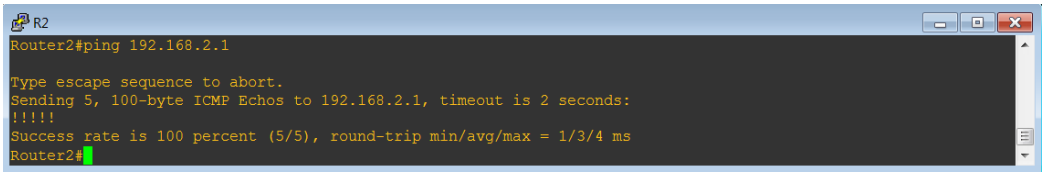


Figure 8.11 Router 2 to 192.168.2.1 interface successful pings

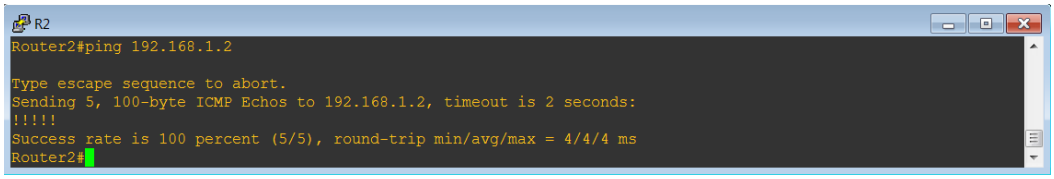


Figure 8.12 Router 2 to 192.168.1.2 (interface facing LAN) interface successful ping

8.3. IP/MPLS Core network Encryption configuration Results

The following is a simple screen dump which shows the crypto engine correctly applied to the respective routers within the simulation.

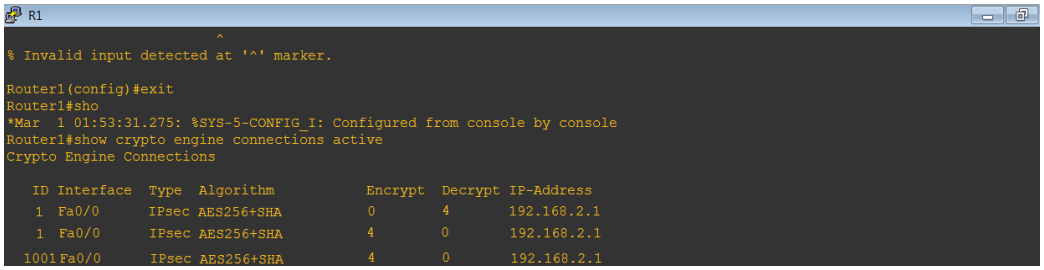


Figure 8.13 An illustration of the correct application of crypto engine to Router 1

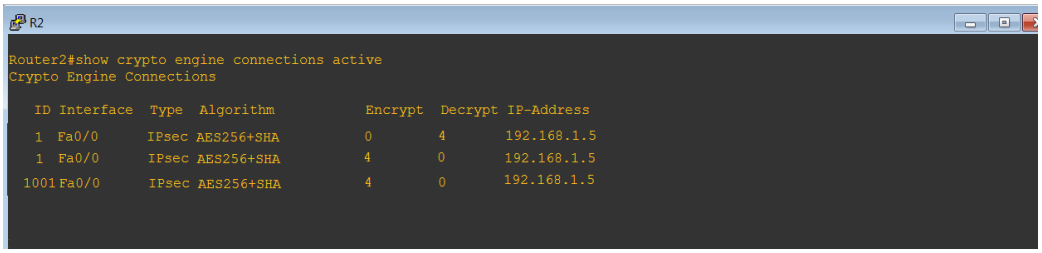


Figure 8.14 An illustration of the correct application of crypto engine to Router 2

8.4. SCADA and RTU Connection Results

The following set of results confirm, firstly the successful connection of SCADA to the RTU as a direct connection to the RTU without going through the IP/MPLS core network, basically a direct link from SCADA to the RTU. Then the results are followed by a SCADA connection to the RTU through the encrypted MPLS/IP core network through the confirmation of the conversion from RS232 to TCP, then the confirmation of the other emulator converting back from TCP/IP to RS232 and lastly the confirmation of the connection to the RTU with a response.

8.4.1. SCADA (FieldComm) and RTU connection confirmation without going through the IP/MPLS Core network

The following section gives results confirming the connection of the SCADA to the RTU. Firstly the results show the human machine interface of SCADA when the connection has not been established yet. From the following Figure 8.15 below, it can be seen that the RTU 1234 in the bottom red bubble it is stated as offline in the window named Point Database. In the main window the SCADA (Src) with address 3 is sending data to RTU with address 1234 with no response as shown by the red bubble in the window titled “Summary”. This is done to show the difference between when a SCADA is not connected to the RTU and also when the connection is successful as it is illustrated below.

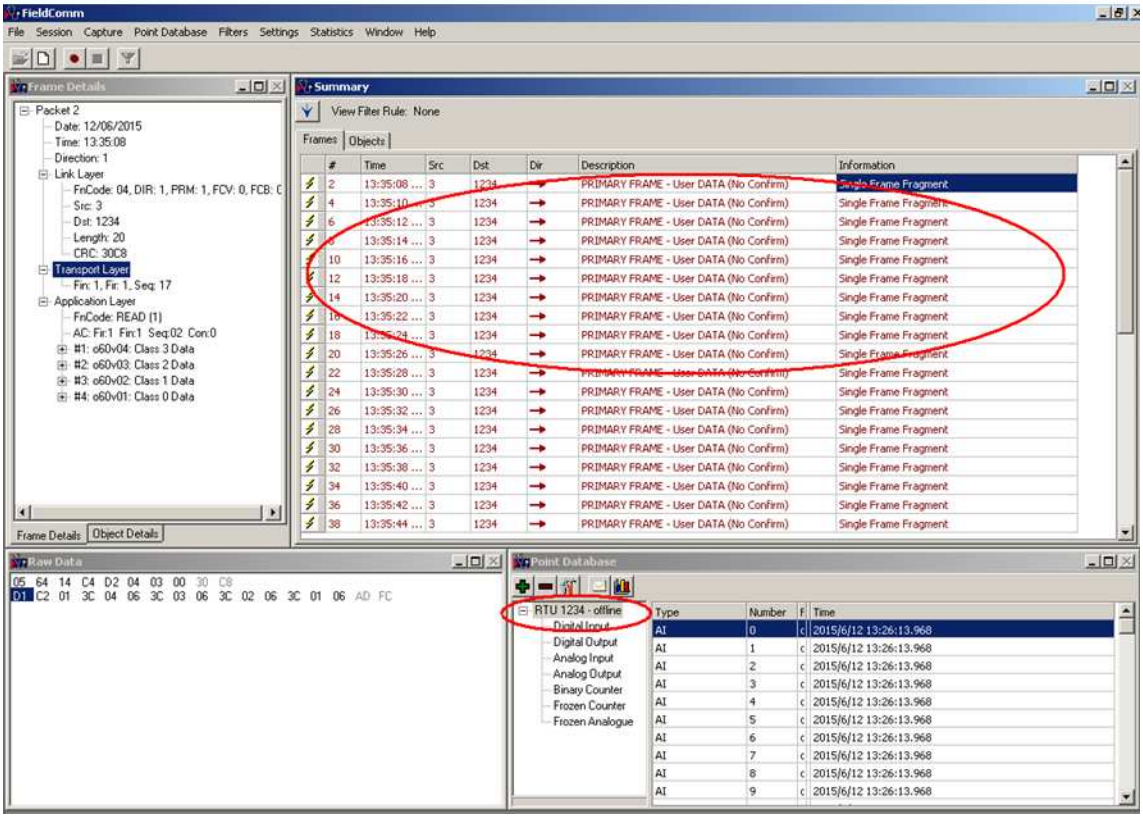


Figure 8.15. The figure shows no connection between the SCADA and RTU.

The following screen shot in Figure 8.16 illustrates a successful connection between SCADA and the RTU. In the window stated as summary, it can be noted that the SCADA with address 3 is sending data to RTU with address 1234 and also there is a response from RTU with address 1234 to SCADA with address 3. It can further be noted that in the window named point database the RTU is no longer offline as in the previous figure above. This confirms the connection of the SCADA to the RTU.

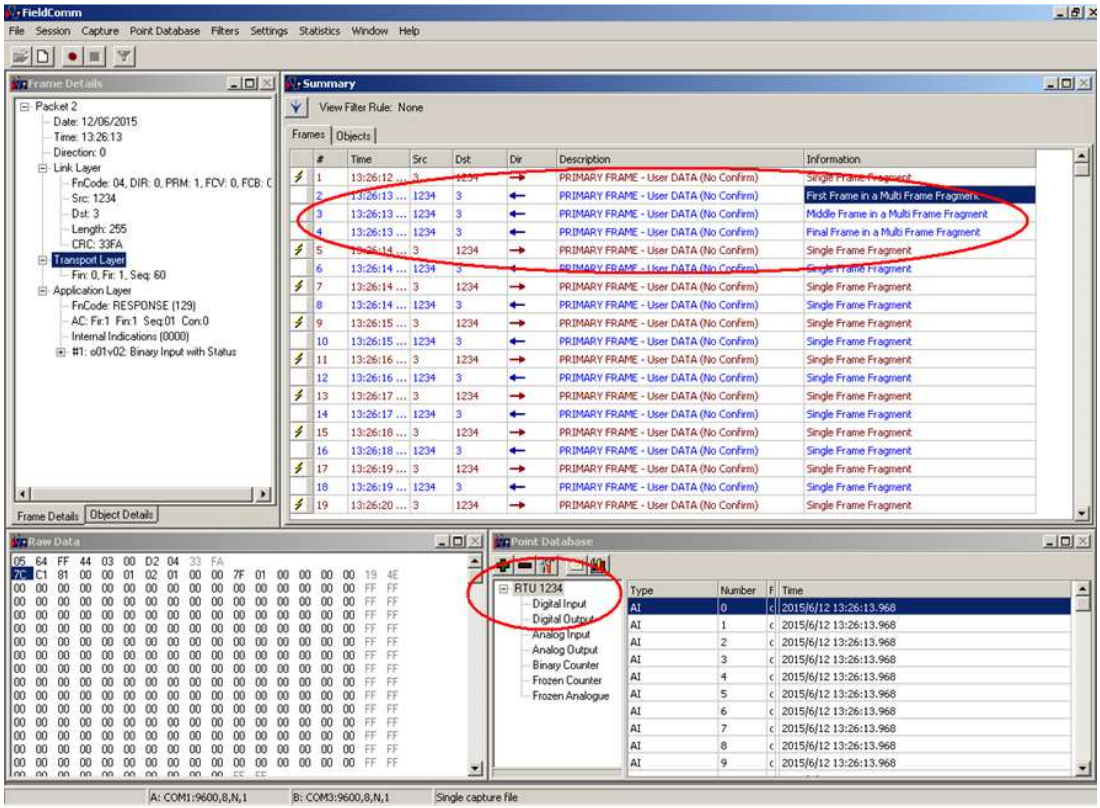


Figure 8.16. The figure shows the confirmation of SCADA connection to the RTU.

8.4.2. Emulator TCP/IP Serial Confirmation

The results in the previous section are from when the SCADA is directly connected to the RTU via a RS232 link and not through the IP/MPLS network. In order for the communication to be possible between the current SCADA and RTU, which both use RS232 for communication through the TCP/IP network, the RS232 information needs to be encapsulated into TCP/IP data packets as discussed in the previous chapter. For this experiment a TCP/IP-RS232 emulator was used.

The first converter or emulator used was to convert the RS232 or serial information generated by the SCADA into TCP/IP data packets so that it can be transmitted over an IP/MPLS core network. In this particular case the SCADA sends the serial information to COM port 2 and the emulator takes the information from COM port 2 and encapsulates it into a TCP/IP packet with the destination IP address 192.168.1.6 that is the intended IP address of the connection point connected to the periphery just before the intended RTU, in this case the loopback IP address. The following figure illustrates the configuration, the encapsulation of COM2 data to IP address 192.168.1.6.

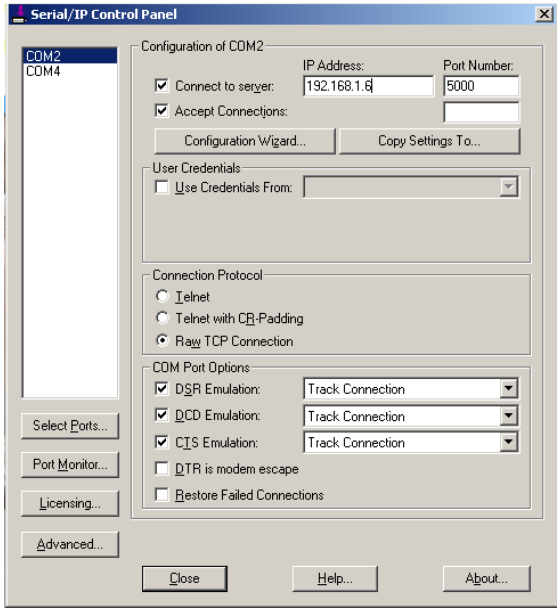


Figure 8.17. Emulator used for Serial to TCP/IP conversion of SCADA information sent over the IP/MPLS core network.

The following figure illustrates the results from the above Serial to TCP/IP convertor giving confirmation of the connection to the desired device on the periphery adjacent to the RTU with IP address 192.168.1.3

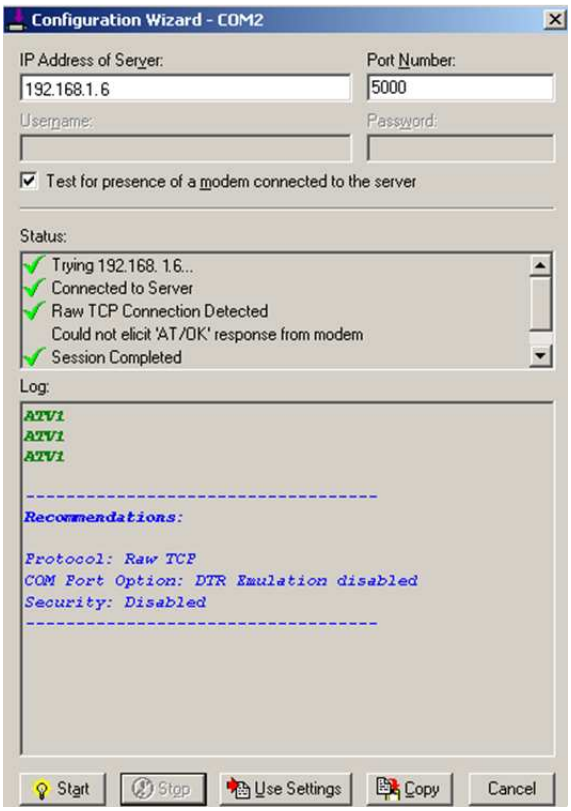


Figure 8.18 Confirmation of emulator connection to destination 192.168.1.6 (Loopback)

The connection of the above emulator had to also be confirmed on the other emulator as it is illustrated in the next figure confirming the connection between 192.168.0.2 (SCADA) and 192.168.1.6 (Loopback) before the conversion from TCP/IP-RS232 to the RTU. After the newly encapsulated information exits the IP/MPLS core network another TCP/IP to serial convertor is used to convert the TCP/IP data send by SCADA back to RS232 for processing by the RTU. The following figure confirms the connection from the device running the SCADA with IP address device 192.168.0.2. It confirms it as the source. The data will be converted back from TCP/IP to Serial communication and sending to Serial port COM 1 to the RTU.

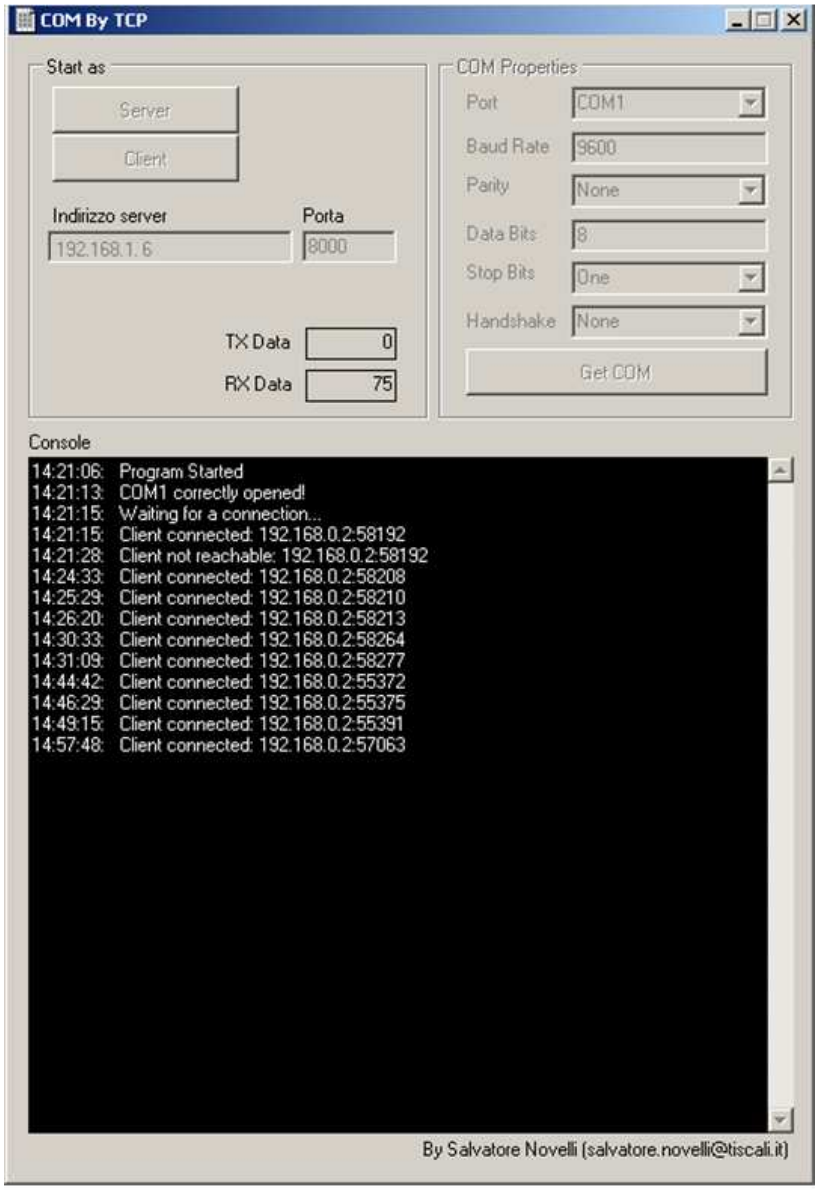


Figure 8.19 Emulator used for TCP/IP to Serial conversion of SCADA information to be executed by the RTU after propagation through encrypted IP/MPLS core network and also confirmation of connectivity between the two emulators through the network

8.4.3. SCADA (FieldComm) and RTU connection confirmation through the Encrypted IP/MPLS Core network

After the connection between the Serial to TCP/IP converter and the TCP/IP to Serial convector though the encrypted IP/MPLS core network is confirmed, the connection between SCADA and RTU had to be confirmed. This was to confirm the following in this order:

- the conversion from Serial to TCP was successful
- the encryption of the encapsulated data
- the propagation through the encrypted IP/MPLS core network was successful
- the decryption of the encapsulated data
- the conversion of TCP/IP back to Serial was successful
- Lastly the communication between SCADA and the RTU was successful through messages and responses exchanged between SCADA and the RTU.

The following figure gives the confirmation of the connection between SCADA and RTU after the conversion to TCP and propagation through the IP/MPLS core network with encryption and conversion back to Serial communication. The figure shows the RTU no longer offline with the data sent by SCADA and replies from the RTU.

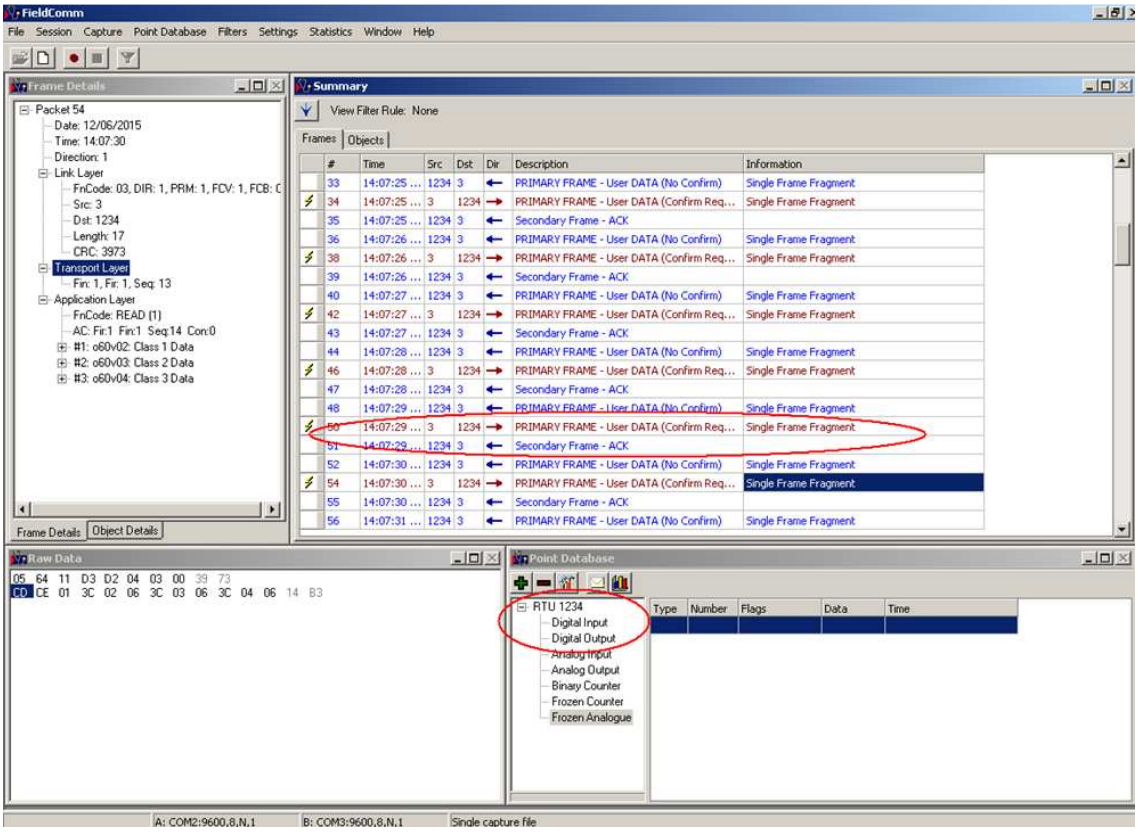


Figure 8.20 The figure confirms the communication between SCADA and RTU through the IP/MPLS core network with data conversion from Serial to TCP and back from TCP to Serial

8.5. Results of SCADA Services Tested over an encrypted IP/MPLS core network

The following section gives results of SCADA services tested over an IP/MPLS core network which was created and configured with encryption.

8.5.1. Telecontrol

For this simulation, Telecontrol in FieldComm was represented by two modes of messages namely; Direct Operate and Operate (slightly different from direct operate). Each of the above message modes for Telecontrol have different modes of functions, generally *"null, trip and close"* for relay operation of which each of these messages was sent for execution by the RTU and the results recorded hereto as printscreens of the SCADA dashboard of messages sent and responses.

8.5.1.1. Direct Operate

The direct operate mode was operated as a binary output with the function selected as *"trip, close or null"*. The following figure illustrates the SCADA services message dashboard for *"Direct Operate"*.

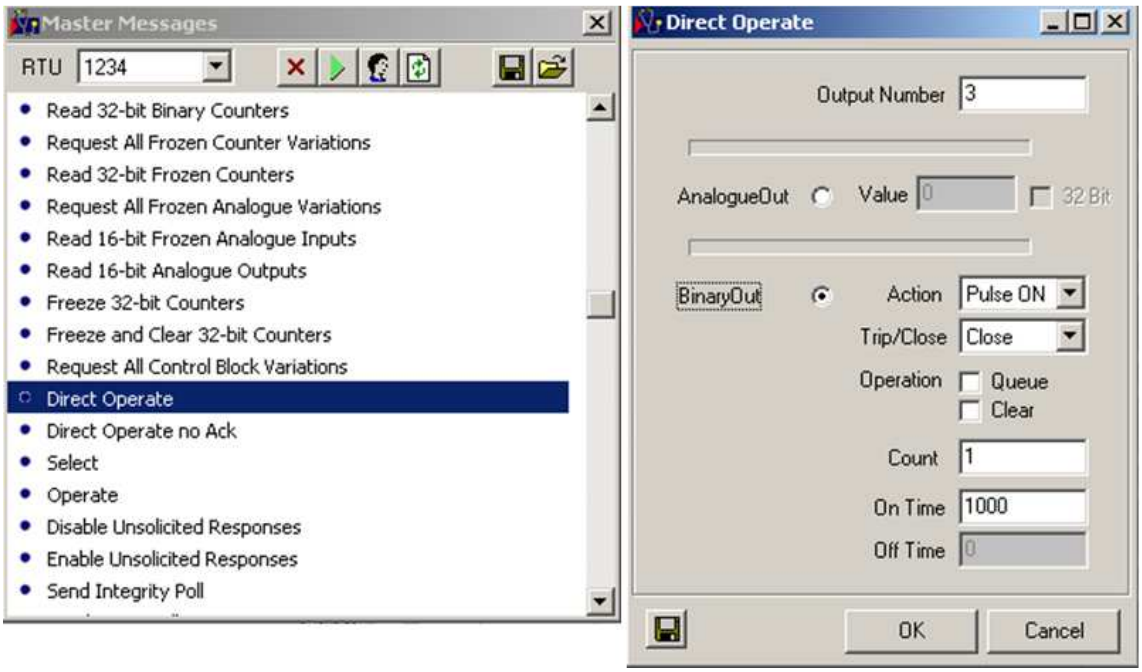


Figure 8.21 An illustration of SCADA service message; "Direct Operate" dashboard

- Null

The first message sent under the "Direct Operate" service function is a *"null"* message to the RTU. The frame details as per the SCADA dashboard is illustrated in the following figure with the response from the RTU given the subsequent figure.

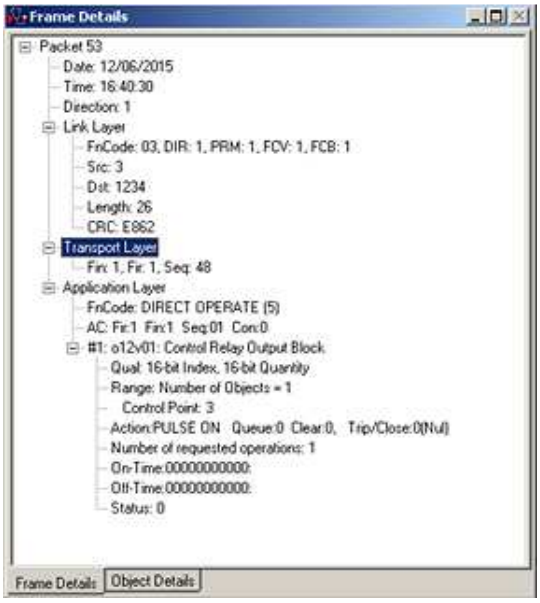


Figure 8.22 Frame details of null function by SCADA to the RTU for Direct Operate service function

The following figures gives the reply from the RTU which means the message successfully reached the RTU and RTU provided a response.

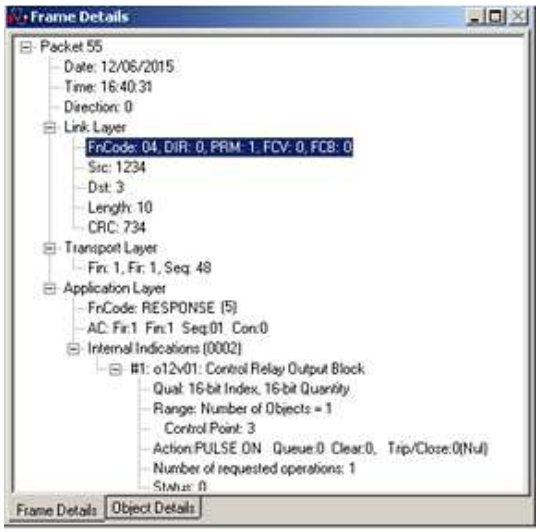


Figure 8.23 Frame details of a reply by the RTU for the null function sent by SCADA to the RTU for Direct Operate service function

- Trip

The other message sent under the "Direct Operate" service function is a *"Trip"* message to the RTU which is normally used to trip a breaker through a relay. The frame details as per the SCADA dashboard is illustrated in the following figure with the response from the RTU given the subsequent figure.

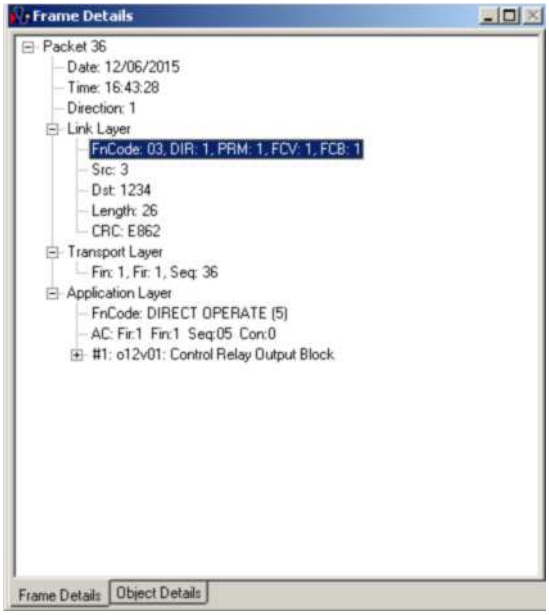


Figure 8.24 Frame details of trip function by SCADA to the RTU for Direct Operate service function

The following figure gives a response frame detail from the RTU to the SCADA regarding the execution of the *"Trip"* SCADA function for the Direct Operate service function.

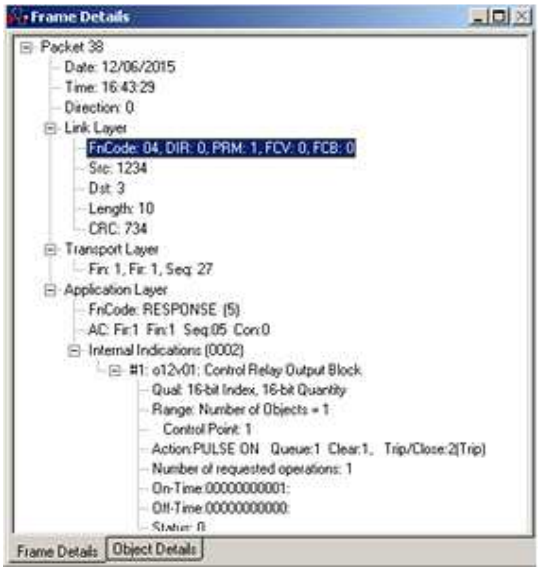


Figure 8.25 Frame details of a reply by the RTU for the trip function sent by SCADA to the RTU for Direct Operate service function

- *Close*

The last message sent by SCADA under the "Direct Operate" service function is a *"Close"* message to the RTU which is normally used to close a breaker through a relay. The frame details as per the SCADA dashboard is illustrated in the following figure with the response from the RTU given the subsequent figure.



Figure 8.26 Frame details of trip function by SCADA to the RTU for Direct Operate service function

The following figure gives a response frame detail from the RTU to the SCADA regarding the execution of the *"Close"* SCADA function for the Direct Operate service function.

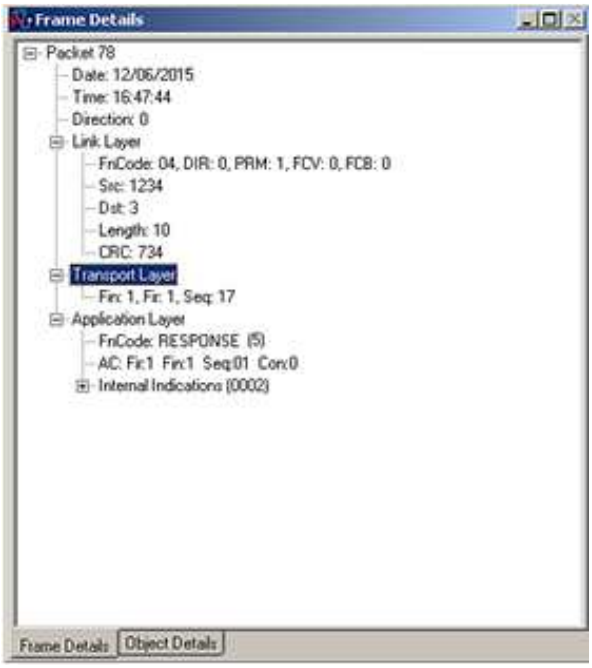


Figure 8.27 Frame details of a reply by the RTU for the close function sent by SCADA to the RTU for Direct Operate service function

8.5.1.2. Operate

The operate mode was operated as a binary output with the function selected as *trip or close*. The following figure illustrates the SCADA services message dashboard for *"Operate"*.

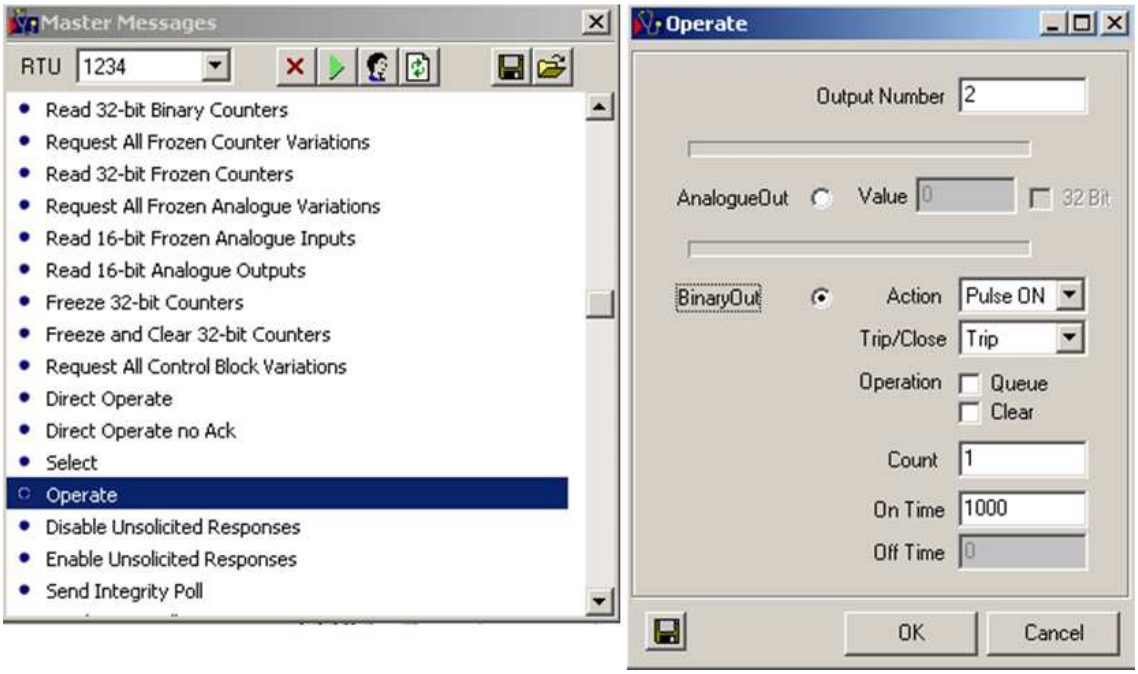


Figure 8.28 The Operate message dashboard illustration

- *Trip*

The message sent under the "Operate" service function is a "*Trip*" message to the RTU which is normally used to trip a breaker through a relay. The frame details as per the SCADA dashboard is illustrated in the following figure with the response from the RTU given the subsequent figure.



Figure 8.29 Frame details of trip function by SCADA to the RTU for Operate service function

The following figure gives a response frame detail from the RTU to the SCADA regarding the execution of the "*Trip*" SCADA function for the "*Operate*" service function.

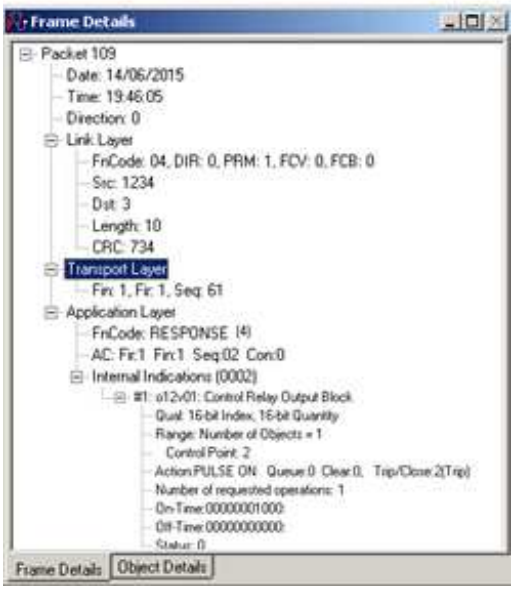


Figure 8.30 Frame details of a reply by the RTU for the trip function sent by SCADA to the RTU for Operate service function

- *Close*

The last message sent by SCADA under the "*Operate*" service function is a "*Close"* message to the RTU which is normally used to close a breaker through a relay. The frame details as per the SCADA dashboard is illustrated in the following figure with the response from the RTU given the subsequent figure.

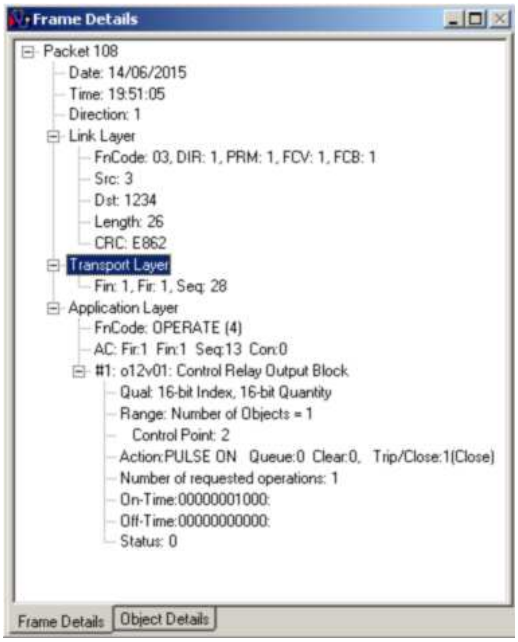


Figure 8.31 Frame details of close function by SCADA to the RTU for Operate service function

The following figure gives a response frame detail from the RTU to the SCADA regarding the execution of the *"Close"* SCADA function for the *Operate* service function.

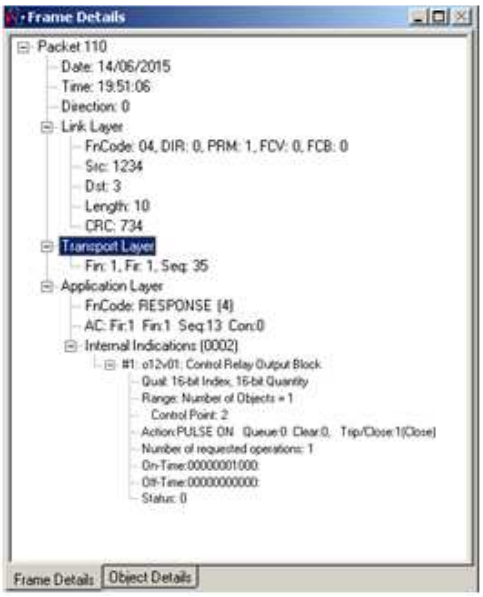


Figure 8.32 Frame details of a reply by the RTU for the close function sent by SCADA to the RTU for Operate service function

8.5.2. Substation Condition

The substation condition was simulated using cold restart, warm restart and integrity poll message.

8.5.2.1. Integrity poll

As aforementioned one of the messages sent by SCADA to simulate the substation condition was an *"Integrity poll"* message to the RTU. The frame details as per the SCADA dashboard is illustrated in the following figure with the response from the RTU given the subsequent figure.

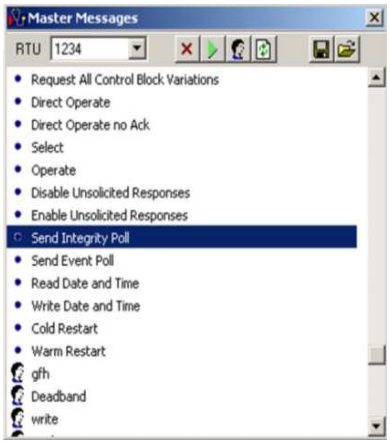


Figure 8.33 An Illustration of an integrity poll message

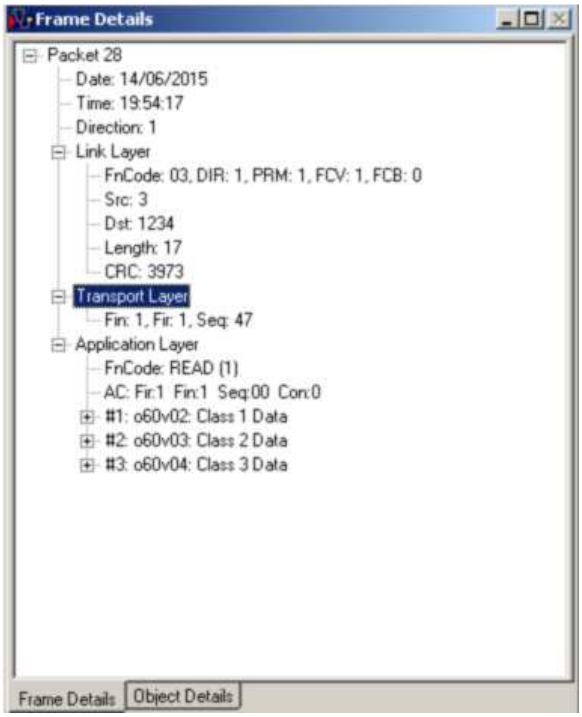


Figure 8.34 Frame details of Integrity Poll function by SCADA to the RTU

The following figure gives a response frame detail from the RTU to the SCADA regarding the execution of the *"Integrity Poll"* SCADA function.

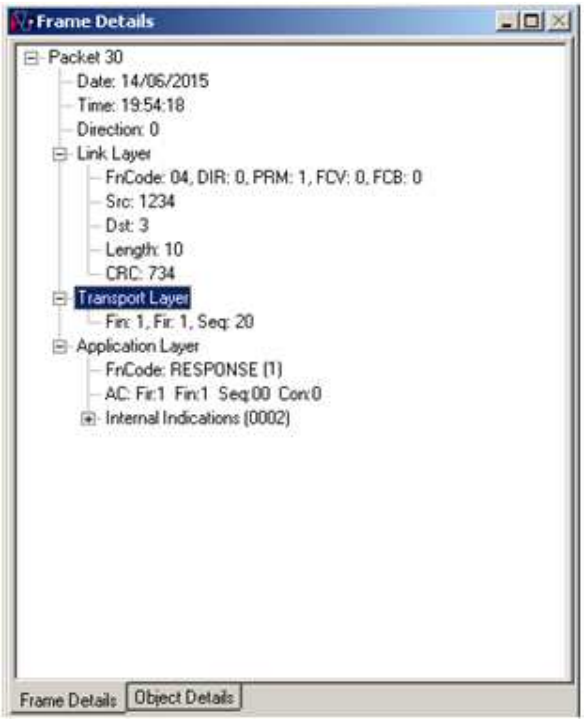


Figure 8.35 Frame details of a reply by the RTU for the Integrity Poll function sent by SCADA to the RTU

8.5.2.2. Cold Restart

As aforementioned one of the messages sent by SCADA to simulate the substation condition was a *"Cold restart"* message to the RTU. The frame details as per the SCADA dashboard is illustrated in the following figure with the response from the RTU given the subsequent figure.

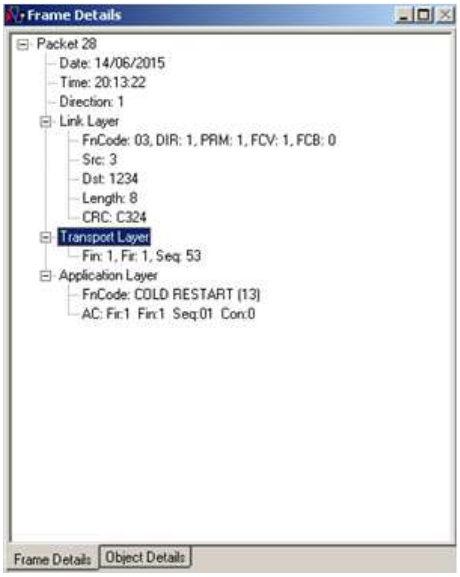


Figure 8.36 Frame details of Cold Restart function by SCADA to the RTU

The following figure gives a response frame detail from the RTU to the SCADA regarding the execution of the *"Cold Restart"* SCADA function.

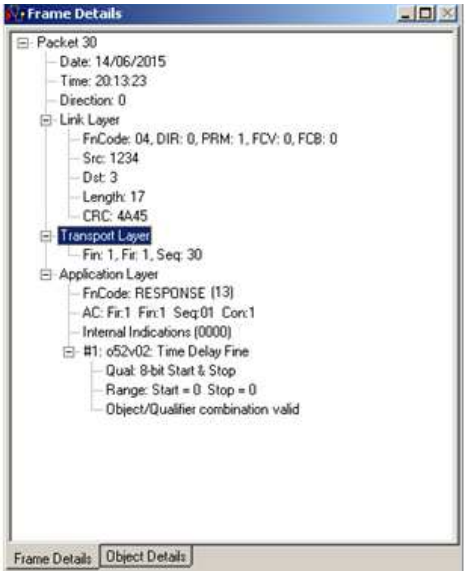


Figure 8.37 Frame details of a reply by the RTU for the Event Poll function sent by SCADA to the RTU

8.5.2.3. Warm Restart

As aforementioned one of the messages sent by SCADA to simulate the substation condition was a *"Warm Restart"* message to the RTU. The frame details as per the SCADA dashboard is illustrated in the following figure with the response from the RTU given the subsequent figure.

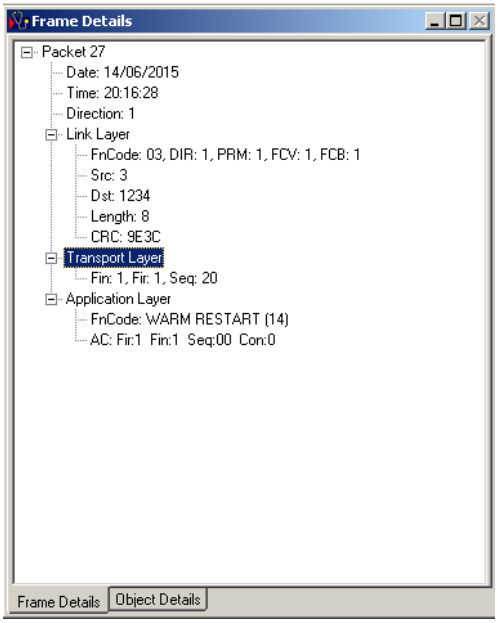


Figure 8.38 Frame details of Cold Restart function by SCADA to the RTU

The following figure gives a response frame detail from the RTU to the SCADA regarding the execution of the *"Warm Restart"* SCADA function. The *"Warm Restart"* was implemented successfully as shown by the response frame detail response and also the with the time delay attributed as fine.

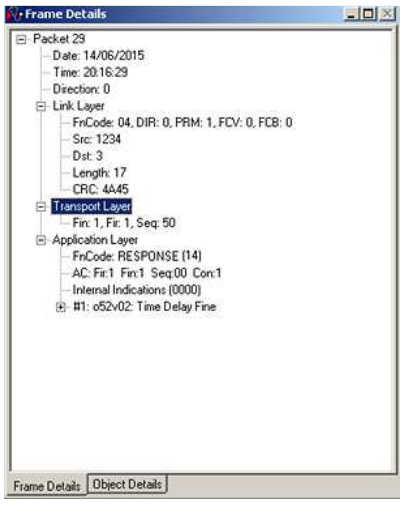


Figure 8.39 Frame details of a reply by the RTU for the Warm Restart function sent by SCADA to the RTU

8.5.3. Remote Monitoring/Metering

The monitoring and metering function was simulated by the "*read date and time"* function and also the "*write date and time"* function.

8.5.3.1. Read date and time

The first message sent by SCADA to simulate the remote monitoring and metering is the *"Read date and time"* message to the RTU. The frame details as per the SCADA dashboard is illustrated in the following figure with the response from the RTU given the subsequent figure.

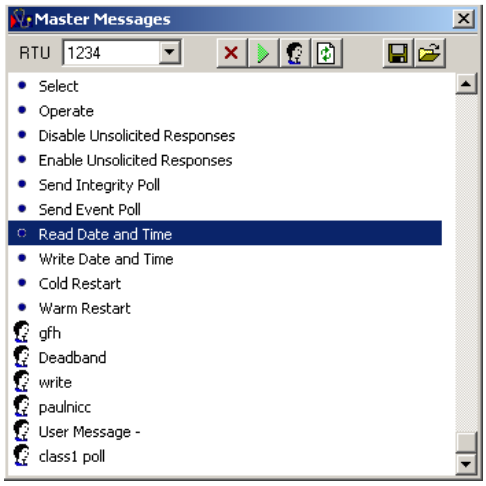


Figure 8.40 An Illustration of a read date and time message

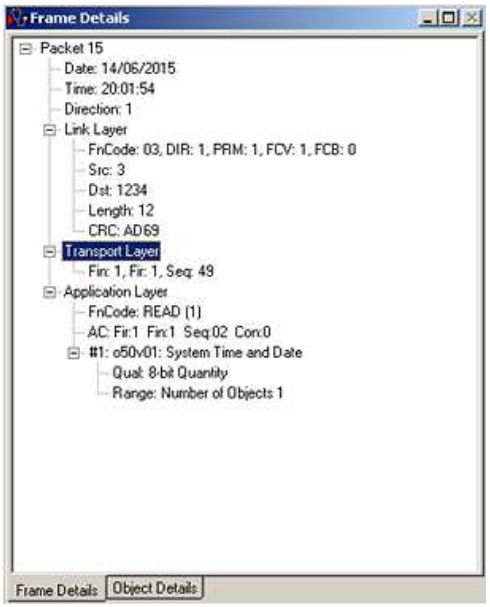


Figure 8.41 Frame details of Read Date and Time function by SCADA to the RTU

The following figure gives a response frame detail from the RTU to the SCADA regarding the execution of the *"Read Date and Time"* SCADA function. It shows the time and date which were read from the RTU. The propagation time and the processing time in this instance

was slight longer due to reasons which could not be determined, however since this part of the simulation illustrates a non-critical, non-time sensitive service, the propagation delay and latency introduced is acceptable. This part of the simulation represents remote device monitoring which monitors the health status of the device which is not time critical. Furthermore this part of the simulation can represent the reading of metering information which is also not that much of a time critical event during the actual reading.

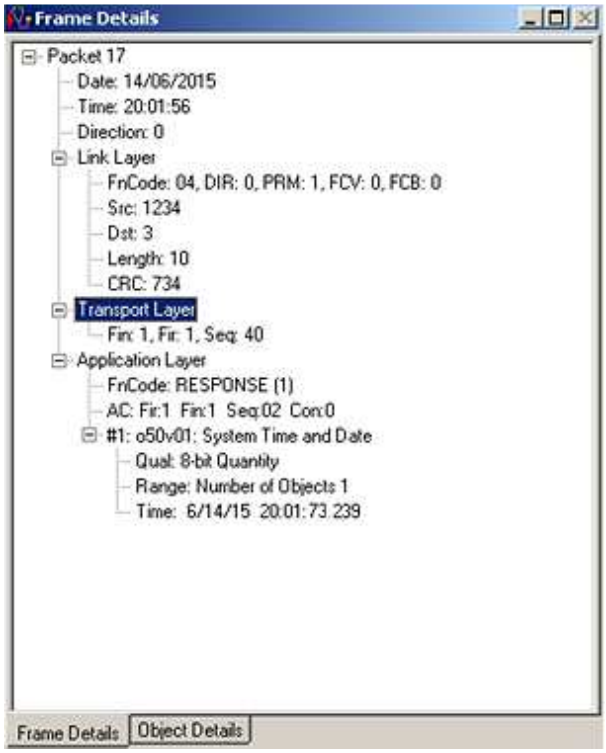


Figure 8.42 Frame details of a reply by the RTU for the Read Date and Time function sent by SCADA to the RTU

8.5.3.2. Write Date and Time

The other message which was sent by SCADA to simulate the remote monitoring service is the *"Write date and time"* message to the RTU. This message represent an instance whereby data was initially read from RTU and other IED devices in the substation and simple changes needs to be made, hence the representation by *"Write date and time"*. The frame details as per the SCADA dashboard is illustrated in the following figure with the response from the RTU given the subsequent figure.

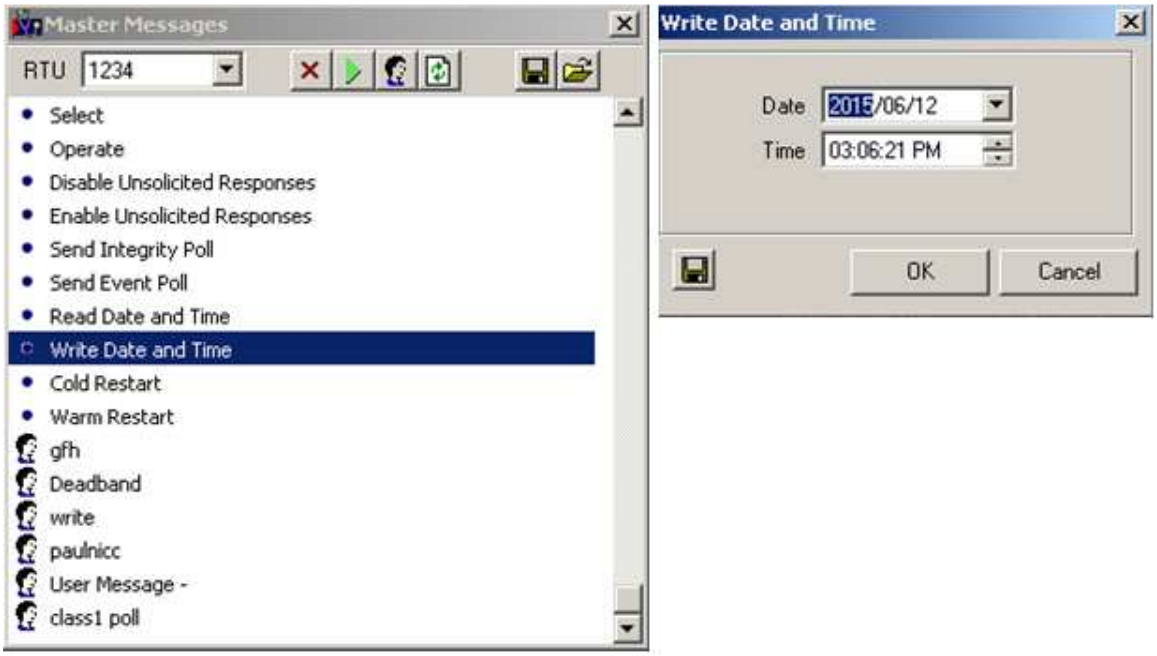


Figure 8.43 An Illustration of a Write date and time message

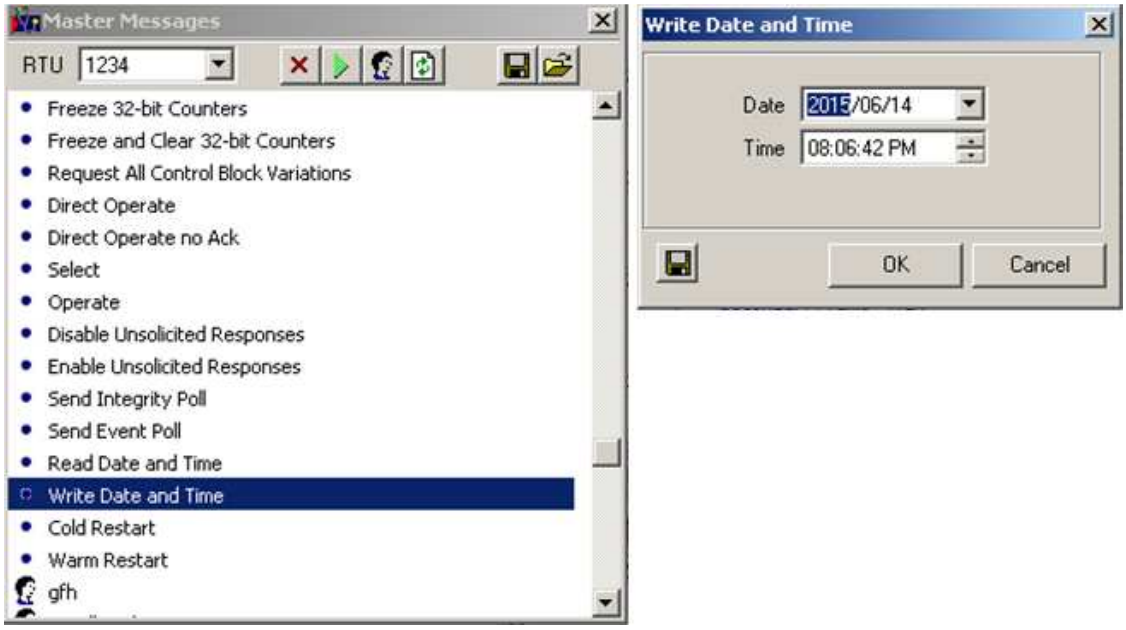
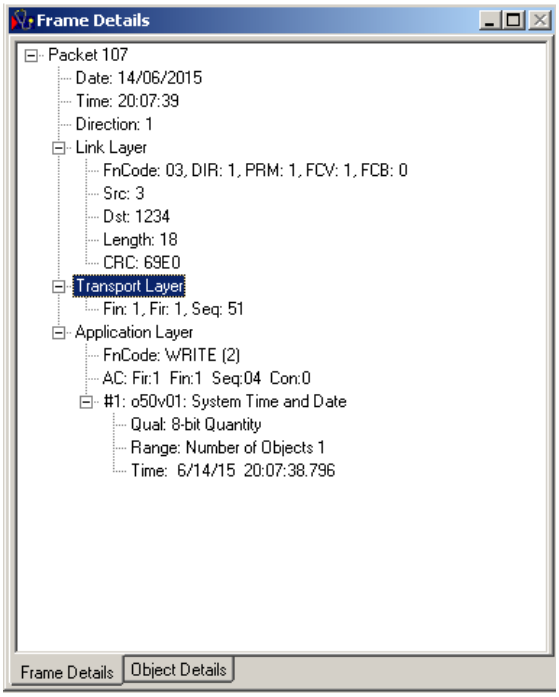


Figure 8.44 An Illustration of a Write date and time message with a date change



Frame details of Write Date and Time function by SCADA to the RTU

The following figure gives a response frame detail from the RTU to the SCADA regarding the execution of the *"Write Date and Time"* SCADA function. As it can be seen the time recorded on the response frame is similar to the one sent with a just a variation of milliseconds introduced due to propagation time and processing time.

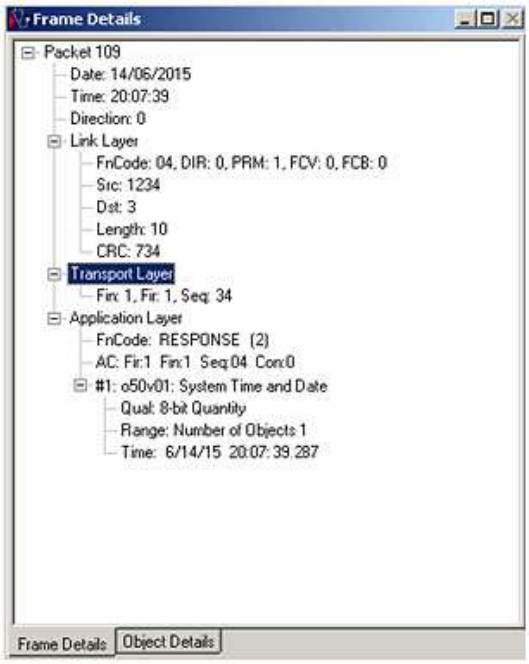


Figure 8.45 Frame details of a reply by the RTU for the Write Date and Time function sent by SCADA to the RTU

8.6. RTU Error Statistics

This sections gives overall results of the SCADA (FieldComm)'s RTU Error Statistics. It gives a graph of Errors encountered when the SCADA simulator communicates with the RTU. The error types as discussed in Chapter 7 are as follows:

- *Bad CRC:* counted when a bad CRC is received from the RTU.
- *User Data Overflow*: SCADA (FieldComm) allows the application layer fragments to be 2048 bytes and if this is exceeded by a slave (RTU) then it is counted as an error.
- *Sequence Errors*: this error is counted when a Transport layer sequence number arrives not in the correct sequence from the RTU
- *Confirm Retries*: this error is when SCADA performs a Link Layer Retry after an initial *Timeout*
- *Confirm Failure*: This error is incremented after the pre-determined number of retries is exceeded without success and SCADA fails the link transmission.
- *AL No Responses*: This refers to errors whenever no response is received to a request within the configured Application Layer Response Timeout.

The following figure gives the results of a RTU error Statistics for the first iteration of SCADA communication with a RTU over an encrypted IP/MPLS core network in this simulation experiment. Overall as it can be seen there are no errors according to the SCADA with the exception of the AL No Responses which experienced overall 2 errors. These errors could have resulted due to initialization of the simulation experiment. The overall error statistics is reasonably good.

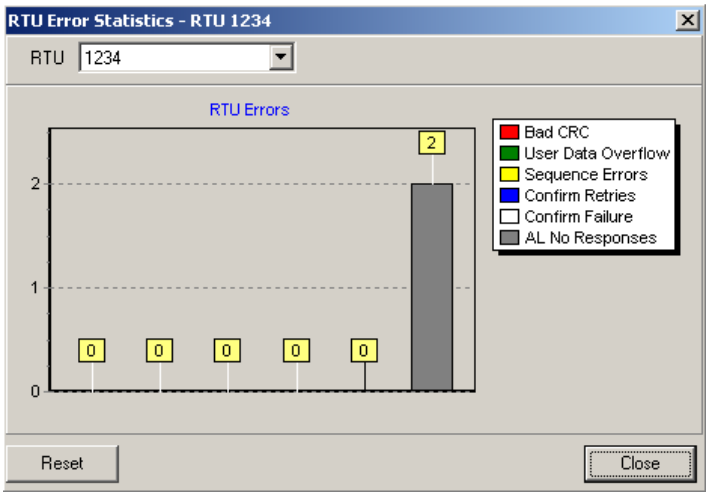


Figure 8.46 The RTU Errors Statistics for the first iteration of SCADA communication with a RTU over an encrypted IP/MPLS core network

The following figure gives the results of a RTU error Statistics for the second iteration of SCADA communication with a RTU over an encrypted IP/MPLS core network simulation experiment. The second iteration is even much better than the first one, there were no errors encountered as illustrated in the RTU graph below. This implies that the communication

between the SCADA and the RTU over an encrypted IP/MPLS core link with encryption is successful with no errors due to packet loss or delays.

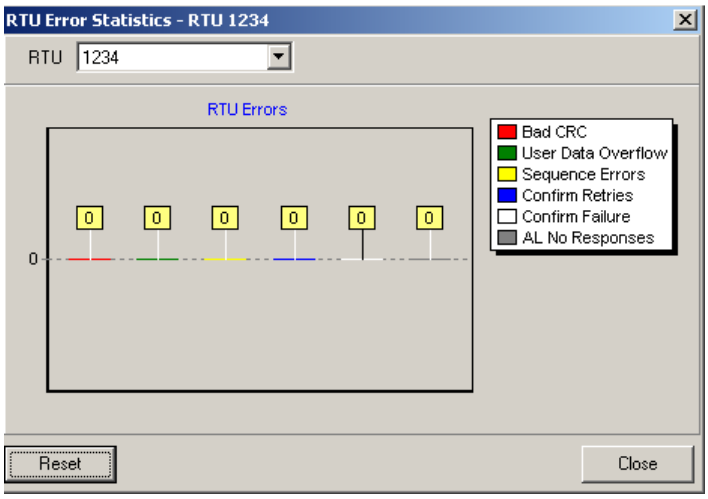


Figure 8.47 The RTU Errors Statistics for the second iterations of SCADA communication with a RTU over an encrypted IP/MPLS core network

8.7. Summary of Results

This section discusses the summary of the results obtained in this chapter and analyzes them.

Section 8.2 results demonstrate that the design and configuration of the MPLS/IP network was successful. The various interfaces worked perfectly and were able to send information across each interface of the routers. The propagation round-trip in some of the pings of various interfaces were recorded with a maximum of 4ms, an average of 4ms and a minimum also 4ms. Other propagation round-trips were recorded, some were extremely lower and others relatively higher. Some of the higher ones were due to mostly initialization issues. This implies that the average round-trip for the MPLS/IP core network is less than 4ms on average. The protocol indicated as up meaning and IP addresses assign and the status, *"OK?"* of the routers as YES. This meant that the MPLS/IP design and configuration as discussed in Chapter 7, Section 7.3 was done correctly and successfully.

Section 8.3 results demonstrate that the crypto engine (encryption and decryption) was designed and applied correctly to the respective routers' interfaces for the simulated MPLS/IP network. The screen dumps of Figure 8.13 and Figure 8.14 shows the crypto engine connections on the routers as active. These figures further describe the type which IPSec and the algorithm used, which is AES256 with SHA. The analysis is that the encryption was successful designed and implemented. The code thereof is given in Appendix B.

Section 8.4 gives results of SCADA and RTU connections results. In Section 8.4.1 a direct connection between the SCADA and RTU was implemented and the results given. At this stage the connection between the SCADA and RTU is without going through the encrypted MPLS/IP core network, but simply a direct link. This was done as aforementioned for calibration purposes such that it is known what a successful direct connection looks like.

Figure 8.15 gives results of an "*Unsuccessful*" connection between an RTU and SCADA. The figure shows that there is no response in the SCADA (FieldComm) Summary window. The point database displays the RTU as offline. Both of these are given by red bubbles in Figure 8.15. Figure 8.16 gives results of a successful connection between the SCADA and the RTU as a direct connection without going through the encrypted MPLS/IP core network. It this figure is demonstrated by SCADA (FieldComm) summary window stating and showing successful communication through initial communication to the RTU and a response. Furthermore the window named RTU database shows the RTU as online and no longer says offline as it did in Figure 18.15. Section 8.4.2 gives the results of the emulation that is the encapsulation of data from RS232 to TCP/IP. Figure 8.17 to Figure 8.19 shows that the emulation was successful. Section 8.4.3 can be considered as the final integration stage, where the RTU and SCADA communicate via the previously tested designed MPLS/IP core network with encryption implemented and the emulation of RS232 to TCP/IP information. This meant that information send by the SCADA (FieldComm) was encapsulated into TCP/IP, then encrypted by the crypto engine, then propagate through the TCP/IP network, decrypted, converted back to RS232 for the RTU. Figure 8.20 shows that this communication between SCADA and RTU through the encrypted MPLS/IP network is successful. This is similar to the results obtained by a direct link between the SCADA and RTU, given by Figure 8.16 which was discussed briefly above. There is communication between SCADA and a response from the RTU. The RTU is not shown as offline but it is online.

After the above successful results, the next results were results obtained from sending various SCADA services, as given in Table 2.1, over this encrypted MPLS/IP core network. Section 8.5.1 gives the results of messages that were sent representing the Telecontrol service. The first message sent is *"Direct Operate"* which has three commands, *null, trip* and *close*. Figure 8.22 shows the successfully sent *"Null"* command by SCADA and Figure 8.23 shows the response from the RTU. The response message shows the time the response was received. The time in these messages are only measure up until only seconds, nothing beyond that. Similarly the *"Trip"* and *"Close"* commands' results (RTU responses) are given in Figure 8.25 and Figure 8.27 respectively. All the commands for Telecontrol, Direct Operate messages were successful. The next set of messages for Telecontrol was Operate. With the Operate, trip and close commands were sent through the encrypted MPLS/IP network through to the RTU. Similarly to the Direct Operate the Telecontrol, Operate message commands were successful because responses were received from the RTU. These commands messages that were sent and with their responses received from the RTU are given in Figures 8.28 to Figure 8.32 without any issues. Substation condition was simulated using cold restart, warm restart and integrity poll messages as discussed Section 8.5.2. The experiment showed successful communication for the various messages between SCADA and RTU over the encrypted MPLS/IP core network without and issues. These successful communications are demonstrated in Figures 8.33 to Figure 8.39. Remote Monitoring/Metering as previously discussed was simulated by the "*read date and time"* function and also the "*write date and time"* function. The messages were successfully sent from SCADA to RTU through the encrypted MPLS/IP core network with successful responses from the RTU. These results are given successfully by Figure 8.40 to Figure 8.45.

In addition to having successfully sent various SCADA services messages to the RTU over an encrypted IP/MPLS network, the important thing was to determine whether they reached the RTU in the appropriate time as prescribed by the minimum SCADA service requirements as described in Table 2.1. This meant determining the overall maximum latency (delay) that could be experienced by packets propagating from the SCADA to the RTU. In terms of the latency (delay) calculation, for this research project, the total latency value can be determined and estimated by the following equation below.

$$\boldsymbol{Latency}_{\text{total}} = \boldsymbol{Latency}_{\boldsymbol{MPLS}} + \boldsymbol{Latency}_{\boldsymbol{Encryption}} + \boldsymbol{Latency}_{\boldsymbol{Decryption}} \quad (8.1)$$

During the simulation experiment it was difficult to measure latency for each encrypted and decrypted packet. The latency for propagating through the virtual IP/MPLS core network was relatively easier to determine although it was subjective to the processing capability of the computer used, however for the purpose of this experiment that determined latency suffices. The latency for the MPLS core network can be determined by one end of the network interface *ping* the other end of the network interface. This is shown as the *"round-trip"* time in the various figures in Chapter 8, Section 8.2. As discussed above the average propagation round-trip was on average 4ms meaning a single trip of 2ms, however there were other propagation round-trip recorded. The maximum measured for this network was a *round-trip* of *26ms* which means a single trip of *13ms.* Thus in order to fully analyze the results the maximum possible measured propagation round-trip was used. In terms of the encryption and decryption latency, it can be assumed that the latency for each of them is the same as it takes roughly the same amount of time to encrypt and decrypt the same amount of data. This means that equation (8.1) is simplified to be;

$$Latency_{\text{total}} = Latency_{MPLS} + 2 \times Latency_{Encryption} \quad (8.2)$$

For this experiment the encryption processing time was measured as the latency between the interface at which a packet to be encrypted enters router 1 and also the interface in which it leaves after encryption. The challenging part was tracing packets to measure their latencies, however this was achieved by sampling a packet and measure its *"Time since previous frame in this TCP stream"* which is a function under *Time Stamps*. Thus a small number of packets were selected and their values noted down. Most of the packets picked had *0.00000 seconds.* One or two of the selected packets, had the value of 0.000065000 seconds. This is illustrated in the following figure.

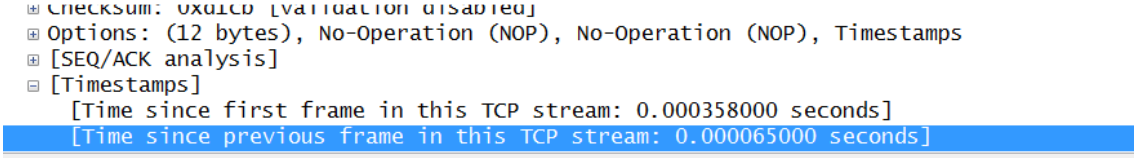


Figure 8.48 A demonstration of an encryption introduced latency

This meant that the latency introduced by encryption and decryption was negligible compared to the propagation latency introduced by the IP/MPLS core network. This is due to the type of encryption chosen for this research project. Using equation (8.2)

$$Latency_{\text{total}} = Latency_{MPLS} + 2 \times Latency_{Encryption} \quad (8.3)$$

$$Latency_{\text{total}} = 13ms + 65\mu s$$

$$Latency_{\text{total}} \approx 13ms$$

$$Latency_{\text{total}} < 14ms \ll 200ms$$

This means that the latency introduced is around 13ms, which could be even less as it was also caused to some extent by the processing capabilities of the computer used for this simulation. This latency is still way less than the average 200ms latency requirements of the various SCADA services. Furthermore it must be recalled that in Figure 7.11, Section 7.6.2 during the RTU and SCADA (FieldComm) configuration, the *"Inter-message Delay"* was set to a limit of 20ms. This meant that messages which took longer than 20ms would be discarded. Since all the messages send to the RTU returned the expected response successfully, it can be concluded that the latency for all these messages were acceptable and lower than 20ms. Furthermore during the RTU configuration, the delay was limited to less than 200ms, as illustrated in Figure 7.13, Section 7.6.3. This meant that if the propagation for a single command is longer than 200ms the request will time out and a time out response will be issued to the SCADA simulator. This was to further check and verify that the acceptable and successful communication between SCADA and RTU were indeed successful. This was so done because the lowest latency requirement of SCADA services is less 200ms as given in Table 2.1. This further ascertain the latency calculation above, that the latency was lower than 14ms which is acceptable for all the SCADA services in an OT/IT environment as given in Table. 2.1.

In terms of the losses, Section 8.6 recorded the loss statistics as recorded by the RTU. These are loss statistics which gives an indication of the different types and number of losses in terms of the messages by the RTU. In Figure 8.46 it can be seen that there two losses which could have been due to initialization and in Figure 8.47 there no losses recorded meaning that all messages were received successfully by the RTU which answers the question on loss and availability in terms of the requirements. This means that the losses were significantly less than 1% with availability more than satisfactory.

The summary of the experimental and simulation results as compared to the requirements in Table 2.1 are given in Table 8.1 below. The conclusion of the results is that they were more than satisfactory for each of the SCADA services requirements.

				SCADA Service Performance Requirements		Simulation Experiment Results		
SCADA Service	**Channel**	**Priority**	**Security**	**Latency, Loss**	**Availability**	**Latency, Loss**	**Availability**	**Comments**
Substation Condition Monitoring and post analysis	IP	High	Required	Latency <200ms, Loss < 1%	98%	Latency <15ms, Loss << 1%	100%	Service Requirements Met
Telecontrol (11kv, 22kv, 33kv, substations RTU's)	IP, Ethernet	Critical	Required	Latency <300ms, Loss < 0.05%	98,33%	Latency <15ms, Loss << 0.05%	100%	Service Requirements Met
Telecontrol (11kv, 66kv, 88kv, 132kv, substations RTU's)	IP, Ethernet	Critical	Required	Latency <300ms, Loss < 0.05%	98,33%	Latency <15ms, Loss << 0.05%	100%	Service Requirements Met
Remote Metering	IP,	High	Required	Latency <300ms, Loss < 1%	98,33%	Latency <15ms, Loss < <1%	100%	Service Requirements Met
Control Centre Voice services	IP	Critical	Optional	Latency <150ms, Loss <1%	98%	N/A	N/A	This part of the simulation was not conducted as security is optional and due to lack of equipment. It forms part of future work

Table 8.1 Summary of results of SCADA services over an encrypted IP/MPLS Core Network

CHAPTER 9

9. Conclusions and Recommendations for Future Work

This chapter discusses the conclusion of the research report and also the recommendations.

9.1. Research Project Conclusion

The goal of this research project was to determine whether using encryption in SCADA systems over IP/MPLS core network in an OT IT environment, the service performance requirements are still met as given chapter 1. This means the key focus area of the research project was encryption against SCADA service requirements over a packet switched networks only. This meant that the research project only focused on encryption only in the whole SCADA cyber security value chain. Other aspects of this value chain could be considered further and future work. With regards to encryption, as stated in Chapter 1, the research project needed to demonstrate confidentiality, availability and integrity which are key characteristics of encryption. When these are met, which are fundamental to encryption the service performance requirements of SCADA needed to be met over an MPLS/IP network.

Confidentiality meant ensuring that the information source is really from the desired source in this case SCADA. This was demonstrated through the encryption configuration of the routers applied with identical and pre-authorization requirement to the two critical routers. This meant that no any other connecting router without the identical crypto engine applied to it would have been able to successfully interface with the configured routers. Furthermore, the IP address range allowable for communication was preconfigured and added as part of the crypto engine.

Integrity refers to the information not altered and this was ascertained by the implementation of the AES-256 encryption with a SHA implementation. As discussed in chapter 5 a SHA value of any stream of data gives the same SHA value, however a slight alteration in the data stream changes the whole SHA value thus if the resulting two SHA values are not the same it would demonstrate that the information has been altered. Thus in this research project the integrity of the information was preserved through the implementation of SHA.

Availability meant that the SCADA system is available all the time and the IP/MPLS network is reliable. This was achieved by how the MPLS configuration was done, it was done by implementing a tunneling strategy with encryption and other security features implemented. These ensured a reliable connection between the two critical routers. This achieves the average SCADA service requirement of around 99% in terms of availability and 1% in terms of losses.

In terms of the latency, for this research project as given in Section 8.7 it was determined to be approximately 13ms. This was further ascertained by introducing the *"Inter-Message*

Delay" of *20ms* during SCADA and RTU configuration such that services messages which exceeded *20ms* in propagation after another message would not be received by the RTU. Another assertion was made through the configuration of the RTU such that the maximum allowable message delay be limited to 200ms. This meant that messages that took longer than 200ms would be time out and a time response would be given as a response to the SCADA simulator. Since all expected responses from the RTU were received for each message, this meant that all messages met this criteria, thus it can be concluded that the latency requirement was met. The loss and availability requirement was ascertained by the report given by the RTU error statistic.

The conclusion to the key research question, *"Whether using encryption in SCADA systems, the services performance requirements are still met in OT IT environment over an encrypted MPLS core network?",* is that the requirements are met as demonstrated in this research project.

9.2. Recommendations for Future work

The following section discusses the recommendation for future work which can be undertaken to further expand on this research project.

9.2.1. Proposed experiment for VHF (Voice) Services over IP/MPLS with encryption

The simulation experimental work can be further extended to other services such as voice for substations operations. Herewith in this section is a brief summary of the future work that can be implemented. These proposed tests can in future be used to verify if the replacement of BME (Bandwidth Management Equipment)'s with Cisco routers (IP/MPLS core network) will meet the latency and quality of service requirements for SCADA voice service requirements as it did in other services. The future work could cover the implementation of the UHF and VHF services over an IP based network. These tests would be vital towards the migration away from the obsolete BME equipment in utilities by proving the services compatible with an underlying IP network platform as opposed to the incumbent BME platform as aforementioned. This lab experiment would involve the test of VHF voice transmission over an IP network and also the encryption and decryption thereof from a VHF Repeater at the far site to the Customer's Control Centre equipment, as voice is part of the SCADA services even though security is characterized as optional. The encryption and decryption scheme would be implemented within the routers during configuration as it was done in the simulation in this research project. The lab experiment setup is as show in the following figure below.

9.2.2. Proposed experiment for Teleprotection over IP/MPLS with encryption

The experimental work can further be extended to other services such as Teleprotection over an encrypted MPLS/IP network. This type of experimentation would require real time network testing as the service requirements for this services are extremely strict and violation of this service could result in serious consequences.

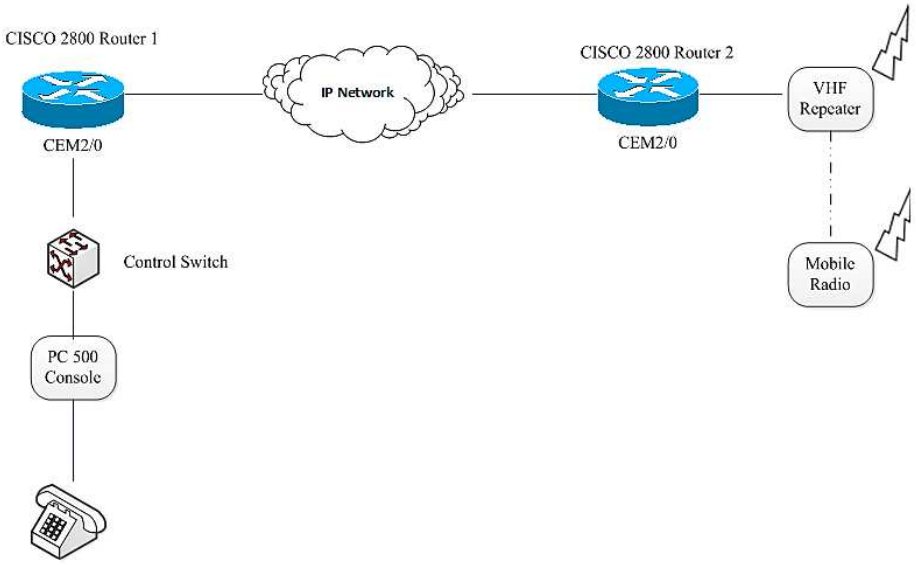


Figure 9.1 Testing of voice services over IP network

9.2.3. The Role for Digital Forensics in SCADA systems

To further extend this research topic as future work and not only focusing on encryption but on the whole cyber security value chain in SCADA systems for value chain, the subject of digital forensics in SCADA system can be researched. This is of utmost importance as SCADA digital forensics deals with scenarios when something has happened in a SCADA system. Computer and Digital forensics is the process of identifying, preserving, analyzing and presenting digital and computer evidence in way that is legally acceptable (Van Der Haar & Leung, 2014). Similar principles of computer and digital forensics can be utilized in SCADA forensics. There is a great need for computer and digital forensics in SCADA systems not only for the abovementioned reason for finding out what has happened, but also as a legal requirement. It is required that organizations show that measures are put in place to try and defend its infrastructures against intrusion. If this cannot be demonstrated, the organization's senior management can be held personally liable for damages. This intrusion can be proved by using digital forensics tools and principles making it easier and transparent to find out what has happened. The main objective of digital forensics in SCADA systems is to determine what happened in a case of an event through an investigative process. The investigation process follows a trail that an intruder leaves during the commissioning of the intrusion to SCADA systems. This process then ties the system intruder to the intrusion and the system that was compromised. This investigative process is based on the commonly known Locard's Exchange principle used in normal criminal investigations. Locard's Exchange principle states that a contact between two items always results in an exchange (Petherick, et al., 2010). For example in a crime scene, there is an exchange of evidence between the offender and the victim, between the victim and the weapon, between the offender and the weapon and between the victim, the offender and the scene itself. The same principle applies in digital forensics for SCADA whereby there can be an exchange between the intruder and the SCADA system. Thus the Locard's Exchange principle has been extended to the digital realm as illustrated in the Figure 9.3 below.

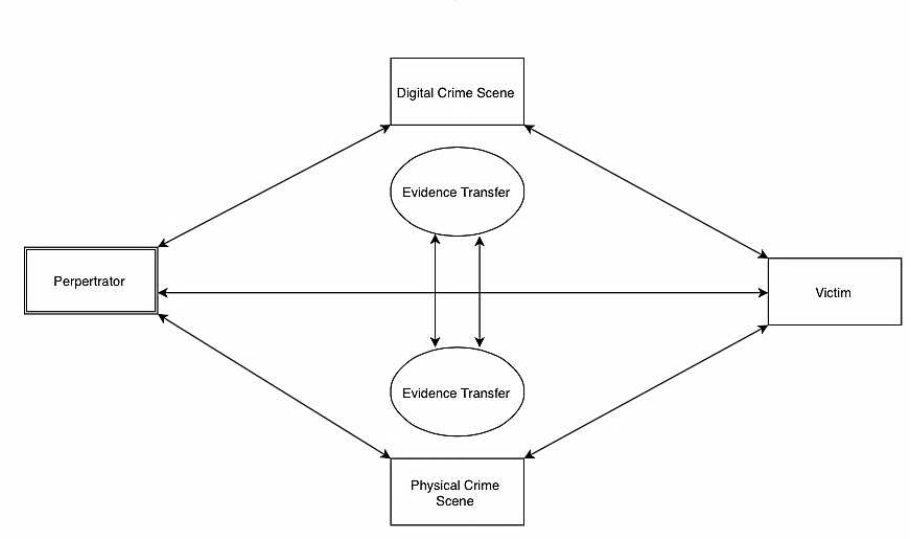


Figure 9.2 An illustration of the Locard's Exchange Principle (Casey, 2004)

It is important to note that the evidence obtained during SCADA digital forensics be done in a proper manner as failure to do so may deem the evidence unusable in court for litigation and other legal or corporate proceedings. This means the evidence obtained through this process must be authentic, meaning it must be in the same state that it was found. The evidence must remain in the state it was collected and this must be demonstrated through Chain of Custody (Ryder, 2002). Lastly the evidence must be obtained legally. There is a shortage of tools and skills with regards to SCADA digital forensics and hence why it is recommended and as future work towards the cyber security value chain.

Bibliography

Adrian, 2014. *Math Cryptography*. [Online] Available at: http://cryptography-adrian-lam.blogspot.co.za/2014/02/caesar-cipher.html [Accessed 2014].

Alcatel Lucent, 2012. Network Tranformation to Reliable, Secure End-to-End Packet Based Infrastructures. *Deploying IP/MPLS Communications Networks for Smart Grids Practice Note.*

Alcatel-Lucent, 2011. *Deploying IP/MPLS Communicaitons for Smart Grids, application note.* [Online] Available at: http://resources.alcatel-lucent.com/asset/162351 [Accessed 2015].

Anon., 2011. *Supervisory control and data acquisition system.* [Online] Available at: http://itlaw.wikia.com/wiki/Supervisory_control_and_data_acquisition_system [Accessed July 2014].

Anon., 2012. *Deploying IP/MPLS Communications for Smart Grid, application note,* s.l.: Alcatel-Lucent.

Anon., 2014. *Operational IP Network.* [Art] (Eskom Telecommunications).

Anonymous, 1999. How Encryption and Digital Signatures Work. *Bionic Buffalo Tech Note #35:*, 12 May, pp. 4-10.

Atos, 2012. The Convergence of IT and Operational Technology. *Ascent Thought Leadership from Atos*, November.

Benoit, J., 2013. *An Introduction to Cryptography as Applied to Smart Grid.* s.l.:s.n.

Bernstein, D., n.d. *The Poly 1305-AES Message Authentication Code.* Chicago: University of Illinois.

Brandsma, H., 2012. *Cyptography.* [Online] Available at: http://crypto.stackexchange.com/questions/2476/cipher-feedback-mode

Budka, K., 2014. *The Future of Smart Grid Communications,* s.l.: Alcatel Lucent.

Buttner, A., 2008. *Common Platform Enumeration (CPE) - Specification.* 2.1 ed. Neal Ziring: The MITRE Corporation.

Casey, E., 2004. The Role of Digital Evidence. In: *Digital Evidence and Computer Crime, Second Edition.* s.l.:ACADEMIC PRESS.

Casey, E., 2011. *Digital Evidence and Computer Crime - Forensic Science, Computers and the Internet.* 3rd Edition ed. s.l.:Academic Press.

Chego, L. N., 2014. *Whether using encryption in SCADA systems, the services performance requirements are still met in OT IT environment over an MPLS core network? Research Project Proposal.* Mini-Thesis Proposal ed. Johannesburg(Gauteng): University of the Witwatersrand.

CISCO MPLS, 2014. *Security of the MPLS Architecture,* s.l.: s.n.

CISCO, 2004. *Cisco 2800 Series Software Configuration Guide,* s.l.: CISCO.

CISCO, 2006. *Configuring a Router IPsec Tunnel Private-to-Private Network with NAT and a Static,* s.l.: CISCO.

CISCO, 2006. *Configuring IOS-to-IOS IPSec Using AES Encryption,* s.l.: Cisco.

CISCO, 2008. *An Introduction to IP Security (IPSec) Encryption,* s.l.: s.n.

Corregedor, M. R., 2014. *Introduction to Malware and Anti-Malware Techniques.* Johannesburg: University of Johannesburg.

Dade, L., 2006. *How Enigma Machines Work.* [Online] Available at: http://enigma.louisedade.co.uk/howitworks.html [Accessed 20 June 2014].

Dr Ellefsen, I. D. & Blauw, F. F., 2014. *Module 2: Introduction to Encryption Techniques (INTENT2).* Johannesburg: University of Johannesburg.

E&A Engineering Solutions, 2014. *E&A Engineering Solutions.* [Online] Available at: http://www.eaengineering.co.in/scada-system.html [Accessed 2014].

Garg, A., 2013. *Fusion of IT and OT in Utilities,* s.l.: Intelligent Utility.

Gattol, M., 2015. *Block Layer Encryption.* [Online] Available at: http://www.markus-gattol.name/ws/dm-crypt_luks.html

Gebbie, S., 2002. *A Survey of The Mathematics of Cryptology,* Johannesburg: University of the Witwatersrand.

General Electric, 2009. *D400 Substation Data Manager: Software Configuration Guide,* s.l.: General Electric.

Groenewald, P. R., Gydien, Z. & Gutschow, D., 2012. *Definition of Operational Technology (OT) and OT/IT Collaboration Accountabilities.* Germiston: Eskom.

Hale, G., 2011. *More SCADA Vulnerabilities Found,* s.l.: s.n.

Harrell, C., 2010. *Overall DF Investigation Process.* [Online] Available at: http://journeyintoir.blogspot.co.za/2010/10/overall-df-investigation-process.html
[Accessed 2014].

Huawei, 2016. *New IP/MPLS Network Overview*, Johannesburg: Eskom.

Idaho National Laboratory, 2011. *Vulnerability Analysis of Energy Delivery Control Systems*, Idaho: Office of Electricity Delivery and Energy Reliability.

IdenTrust SSL, 2015. *What is SSL?*. [Online] Available at: https://www.identrustssl.com/learn.html [Accessed 2014].

Jeff, 2012. *Password Based Key Derivation Function (PBKDF2) diagram*, s.l.: Agile Bits.

Kak, A., 2014. Lecture 8: AES: The Advanced Encryption Standard. *Lectures Notes on Computer and Network Security*, 6 March, p. 3.

Kalapatapu, R., 2004. *SCADA Protocols and Communication Trends.* Houston, Texas, The Instrumentation, Systems and Automation Society, pp. 1-2.

Khan, D., 1996. *The Code Breakers*, s.l.: Scribner.

Killalea, T., 2000. *Recommneded Internet Service Provider Security Services and Procedures*, s.l.: s.n.

Klipatrick, T. et al., 2008. An Architecture For SCADA Network Forensics. In: *Advances in Digital Forensics II.* s.l.:s.n., pp. Chapter 2: 275-284.

Leemburg, S., 1995. *Classical Cryptography*, s.l.: s.n.

Lemke, A., 2014. *How does the E0 cipher work?*, s.l.: s.n.

Leung, W. S., Blauw, F. F. & Ellefsen, I. D., 2014. *Introduction to Malicious Software and Ethical Hacking.* October(Gauteng): University of Johannesburg.

Lib4U, 2013. *RSA Public Key Encryption System.* s.l.:s.n.

Mabotja, E., 2011. *Proof of Concept Scope of Work: Testing of Dx SCADA over the CISCO Network*, Germiston: Eskom Distribution.

Mabotja, E., Mbebe, Z. & Naidoo, C., 2014. *VHF/UHF Services of IP Test Procedure*, Johannesburg: Eskom .

Mackiewitz, J., 2014. *Overview of IEC61850 and Benefits.* [Online] Available at: http://electrical-engineering-portal.com/three-generations-of-scada-system-architectures
[Accessed February 2014].

Maritz, S. & Bronkhorst, G., 2010. *User Requirement Specification for Communication Services Requirements*, Simmerpan, Germiston: Eskom.

Mat, 2009. *Python: Cryptography Substitution Cipher improving on the Caesar cipher.* [Online]

Available at: http://www.stealthcopter.com/blog/2009/12/python-cryptography-substitution-cipher-improving-on-the-caesar-cipher/

Mell, P., Scarfone, K. & Romanosky , S., 2007. *A Complete Guide to the Common Vulnerability Scoring System Version 2.0.* Version 2 ed. s.l.:National Institute of Standards and Carnegie Mellon University .

Moonie, A., 2011. *Steps to configure an IPSEC site to site VPN on CISCO IOS Device (GNS3 Lab).* [Online]
Available at: https://www.m00nie.com/2011/03/steps-to-configure-an-ipsec-site-to-site-vpn-on-a-cisco-ios-device-gns3-lab/
[Accessed April 20155].

National Archives UK, 1586. *The Babington Plot,* s.l.: s.n.

Network Strategy Partners, 2009. *Business Drivers and Design Cuidelines for Network Convergence and Virtualization of IP/MPLS Core Networks.* s.l., Management Consultants to the Networking Industry.

Niyaz, P. K., n.d. *Advanced Encryption Standard Implementation in C.* [Online]
Available at: comp.ist.utl.pt/ec-csc/Code/Ciphers/AES_Encrypt.cpp
[Accessed 1 July 2014].

Oracle, 2010. *Oracle Solaris Security for Developers Guide,* s.l.: s.n.

Palmer, C., 2001. Ethical Hacking. *IBM Systems Journal,* 40(3), pp. 769-780.

Patzlaff, H., 2013. D7.1 Forensics. *SEVENTH FRAMEWORK PROGRAMME: Theme SEC-2011.2.5-1 (Cyber attacks against critical infrastructures).*

Permann, M. R. & Rohde, K., 2005. *Cyber Assessment Methods for SCADA Security,* Idaho Falls: Idaho National Laboratory.

Petherick, W. A., Turvey, B. E. & Fergurson, C. E., 2010. *Forensic Criminology. London: Elsevier Academic Press..* [Online]
Available at: http://aboutforensics.co.uk/edmond-locard/
[Accessed 2015].

Robles, R. J. & Choi, M.-k., 2009. Assessment of the Vulnerabilities of SCADA, Control Systems and Critical Infrastructure Systems. *International Journal of of Grid and Distributed Computing,* June, 2(2), pp. 27-34.

Rouse, M., 2015. *Multiprotocol Label Switching (MPLS) definition.* [Online]
Available at: http://searchenterprisewan.techtarget.com/definition/Multiprotocol-Label-Switching
[Accessed 31 July 2015].

Ryder, K., 2002. Computer Forensics – We've had an incident, who do we get to investigate?. *GSEC Certification Assignment Version 1.3*.

Secretary of Commerce, 2001. Announcing The Advanced Encryption Standard. *Federal Information Processing Standards Publication 197,* 26 November.Issue 197.

Shelton, B., 2014. *Introduction to Cryptography.* [Online] Available at: http://www.infosectoday.com/Articles/Intro_to_Cryptography/Introduction_Encryption_Algorithms.htm

Shirley, R., 2000. *Internet Security Glossary,* s.l.: s.n.

Soloman, M. G. et al., 2011. *Computer Forensics Jumpstart.* 2nd ed. s.l.:Wiley.

Stirland, J., Jones, K., Janicke, H. & Wu, T., 2014. *Developing Cyber Forensics for SCADA Industrial Control Systems.* Kuala Terengganu, Malaysia, Airbus Group and 2 De Montfort University.

Systems & Network Analysis Center, n.d. *Securing Supervisory Control and Data, page 1.* [Online]
Available at: https://www.nsa.gov/ia/_files/factsheets/scada_factsheet.pdf [Accessed 12 March 2016].

Taveras, P., 2013. *SCADA LIVE FORENSICS: REAL TIME DATA ACQUISITION PROCESS TO DETECT, PREVENT OR EVALUATE CRITICAL SITUATIONS.* Azores, Portugal, 1st Annual International Interdisciplinary Conference.

Ten, C.-W., Lui, C. C. & Govindarasu, M., n.d. Vulnerability Assessment of Cybersecuirty for SCADA Systems Unsing Attack Trees.

Terezinho, F., n.d. *SCADA Systems Automate Electrical Distribution: PC-based supervisory control and data acquisition systems increase uptime, cut costs and improve utilization.,* s.l.: InduSoft.

Thales, 2013. *Cyber Security for SCADA Systems,* s.l.: s.n.

TruData Consulting, 2008. *FieldComm DNP3.0 Test Sest User Guide,* s.l.: s.n.

Van Der Haar, D. T. & Leung, W. S., 2014. *Introduction to Digital Forensics.* Johannesburg(Gauteng): University of Johannesburg.

Van Der Knijff, R. M., 2014. *Control Systems/SCADA Forensics, what's the difference in Digital Investigation.* s.l., s.n., pp. 112-243.

Van Tilborg, H. C. A. & Jajodia, S., 2011. Data Encryption Standard (DES). *Encyclopedia of Cryptography and Security,* Issue 2.

Vanucci, D., 2013. *Seecrypt Cryptography.* [Online] Available at: https://www.seecrypt.com/en/seecrypt/technology/security-encryption-sc [Accessed Aug 2014].

Ventyx, n.d. *Convergence of Information and Operation Technologies (IT & OT) to build a successful Smart Grid,* s.l.: ABB.

Wander, A. S., Eberle, H., Gupta, V. & Shantz, S. C., n.d. Energy Analysis of Public-Key Cryptography for Wireless Sensor Networks.

Wilhoit, K., n.d. *ICS, SCADA, and Non-Traditional Incident Response.* s.l., TREND Micro.

Wu, T., Disso, J. F., Jones, K. & Campos, A., n.d. *Towards a SCADA Forensics Architecture,* Newport: EADS Innovation Works Quadrant House Celtic Springs .

Appendix A: A Typical Utility IP/MPLS Core Network

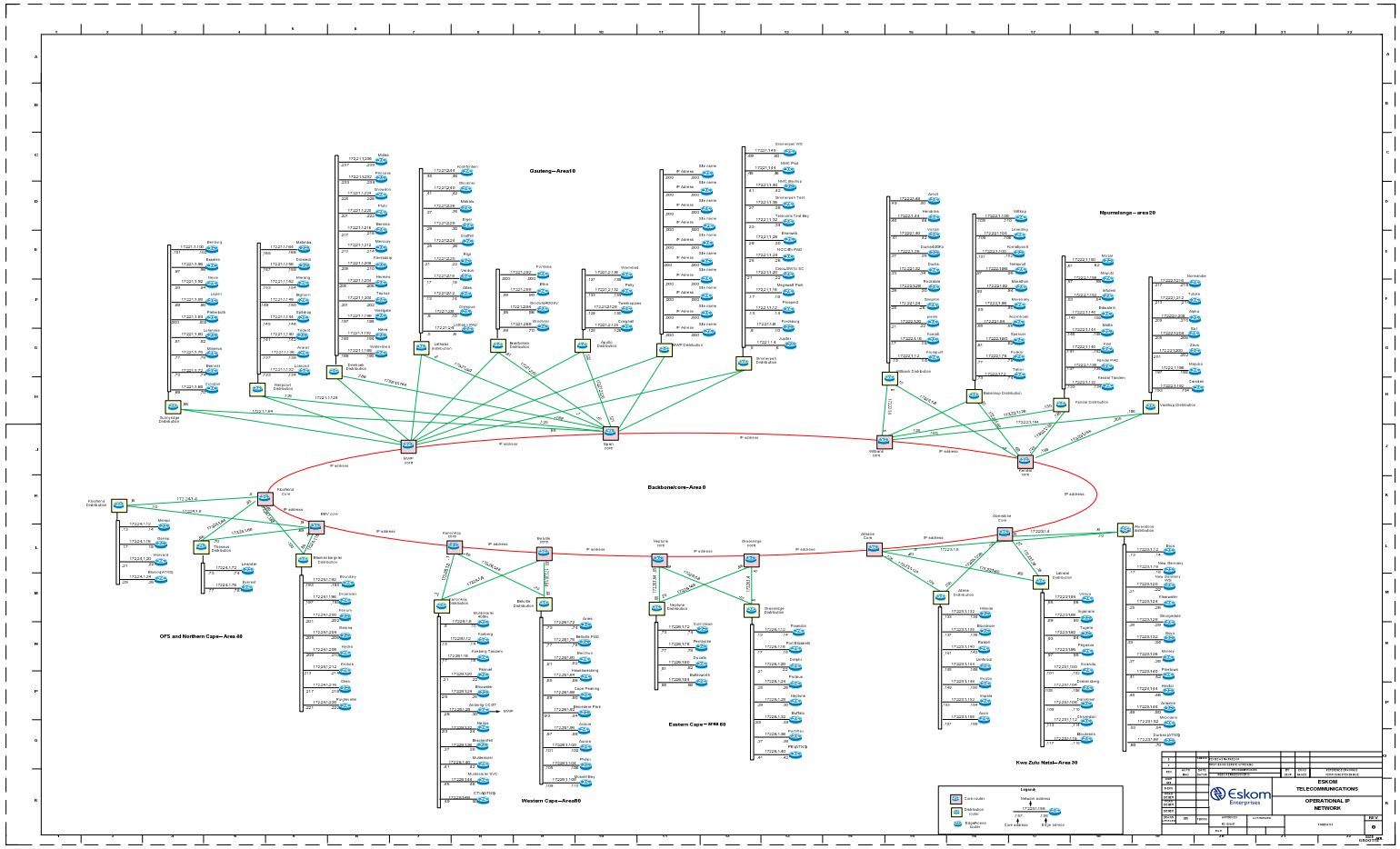


Figure 0.1: An illustration of a typical utility IP/MPLS core network (Anon., 2014)

Appendix B: Code for Configuration

Connected to Dynamips VM "R1" (ID 0, type c3725) - Console port

Press ENTER to get the prompt.

Router1#

*Mar 1 00:37:06.475: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router1#

Router1#configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Router1(config)#hostname Router3

Router3(config)#no aaa new-model

Router3(config)#ip cef

Router3(config)#no ip domain lookup

Router3(config)#multilink bundle-name

*Mar 1 00:38:06.471: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router3(config)#multilink bundle-name authenticated

Router3(config)#router rip version 2

                  ^

% Invalid input detected at '^' marker.

Router3(config)#router rip

Router3(config-router)#version2

                      ^

% Invalid input detected at '^' marker.

Router3(config-router)#version 2

Router3(config-router)#network 192.168

*Mar 1 00:39:06.467: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router3(config-router)#network 192.168.2.0

Router3(config-router)#network 192.168.0.0

Router3(config-router)#interface fastethernet0/0

Router3(config-if)#description To_Router2

Router3(config-if)#ip

*Mar 1 00:40:06.459: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router3(config-if)#ip address 192.169.2.1 255.255.255.0

Router3(config-if)#duplex auto

Router3(config-if)#speed auto

Router3(config-if)#

*Mar 1 00:41:06.455: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router3(config-if)#interface fastethernet0/1

Router3(config-if)#ip address 192.168.0.4 255.255.255.0

Router3(config-if)#duplex auto

Router3(config-if)#speed auto

Router3(config-if)#no shutdown

Router3(config-if)#

*Mar 1 00:42:06.451: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router3(config-if)#exit

Router3(config)#interface fastethernet0/0

Router3(config-if)#no shutdown

Router3(config-if)#exit

Router3(config)#exit

Router3#

*Mar 1 00:42:32.279: %SYS-5-CONFIG_I: Configured from console by console

Router3#sh ip int brie

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	192.169.2.1	YES	manual	up	up
FastEthernet0/1	192.168.0.4	YES	manual	up	up
NVI0	192.169.2.1	YES	unset	up	up

Router3#

Building configuration...

[OK]

Router3#

*Mar 1 00:42:53.495: %SYS-2-MALLOCFAIL: Memory allocation of 32768 bytes failed from 0x600286B4, alignment 0

Pool: Processor Free: 26732 Cause: Not enough free memory

Alternate Pool: None Free: 0 Cause: No Alternate pool

-Process= "Exec", ipl= 0, pid= 244, -Traceback= 0x61467A3C 0x600165C4 0x6001C400 0x6001CBAC 0x600286BC 0x600 29620 0x614AF920 0x614AF994 0x614AFA2C 0x60594608 0x614B7090 0x6059E8C0 0x614AF694 0x61476F68 0x61477830 0x614 D3978

Router3#config

*Mar 1 00:43:06.447: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router3#configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Router3(config)#hostname Router1

Router1(config)#exit

Router1#wr

Building configuration...

*Mar 1 00:43:19.203: %SYS-5-CONFIG_I: Configured from console by console[OK]

Router1#

*Mar 1 00:43:24.123: %SYS-2-MALLOCFAIL: Memory allocation of 32768 bytes failed from 0x600286B4, alignment 0

Pool: Processor Free: 26620 Cause: Not enough free memory

Alternate Pool: None Free: 0 Cause: No Alternate pool

-Process= "Exec", ipl= 0, pid= 244, -Traceback= 0x61467A3C 0x600165C4 0x6001C400 0x6001CBAC 0x600286BC 0x600 29620 0x614AF920 0x614AF994 0x614AFA2C 0x60594608 0x614B7090 0x6059E8C0 0x614AF694 0x61476F68 0x61477830 0x614 D3978

Router1#wr

Building configuration...

[OK]

Router1#wr

Building configuration...

[OK]

Router1#

*Mar 1 00:44:06.443: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router1#

*Mar 1 00:45:06.451: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router1#

*Mar 1 00:46:06.475: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router1#

*Mar 1 00:47:06.495: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router1#

*Mar 1 00:48:06.523: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router1#interface fastethernet0/1

^

% Invalid input detected at '^' marker.

Router1#configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Router1(config)#interface fastethernet0/1

Router1(config-if)#ip address 192

*Mar 1 00:49:06.547: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router1(config-if)#ip address 192.168.1.2 255.255.255.0

Router1(config-if)#exit

Router1(config)#exit

Router1#

*Mar 1 00:49:58.667: %SYS-5-CONFIG_I: Configured from console by console

Router1#sh ip int brie

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	192.169.2.1	YES	manual	up	up
FastEthernet0/1	192.168.1.2	YES	manual	up	up
NVI0	192.169.2.1	YES	unset	up	up

Router1#

*Mar 1 00:50:06.563: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router1#configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Router1(config)#interface fastethernet0/1

```
Router1(config-if)#ip address 192.168.1.2 255.255.255.0
Router1(config-if)#no shutdown
Router1(config-if)#exit
Router1(config)#network 192.168.1.0
                ^
% Invalid input detected at '^' marker.
Router1(config)#configure terminal
                ^
% Invalid input detected at '^' marker.
Router1(config)#exit
Router1#confgir
*Mar  1 00:51:55.787: %SYS-5-CONFIG_I: Configured from console by console
Router1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#router rip
Router1(config-router)#version 2
Router1(config-router)#network 192.168.1.0
Router1(config-router)#network 192.168.2.0
Router1(config-router)#exit
Router1(config)#end
Router1#
*Mar  1 00:52:50.963: %SYS-5-CONFIG_I: Configured from console by console
Router1#sh ip int brie
Interface              IP-Address      OK? Method Status                Protocol
FastEthernet0/0        192.169.2.1     YES manual up                    up
FastEthernet0/1        192.168.1.2     YES manual up                    up
NVI0                   192.169.2.1     YES unset  up                    up
Router1#ping 192.168.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:
.....
Success rate is 100 percent (5/5), round-trip min/avg/max = 22/24/46 ms
```

```
Router1#configure terminala
                    ^
% Invalid input detected at '^' marker.
Router1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#router rip
Router1(config-router)#version 2
Router1(config-router)#network 192.168.2.0
Router1(config-router)#network 192.168.1.0
Router1(config-router)#exit
Router1(config)#interface fastethernet0/0
Router1(config-if)#ip address 192.168.2.1 255.255.255.252
Router1(config-if)#no shutdown
Router1(config-if)#duplex auto
Router1(config-if)#speed auto
Router1(config-if)#exit
Router1(config)#end
Router1#
*Mar  1 01:00:08.643: %SYS-5-CONFIG_I: Configured from console by console
Router1#sh ip int brie
Interface              IP-Address      OK? Method Status                Protocol
FastEthernet0/0        192.168.2.1     YES manual up                    up
FastEthernet0/1        192.168.1.2     YES manual up                    up
NVI0                   192.168.2.1     YES unset  up                    up
Router1#ping 192.168.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/26/44 ms
Router1#ping 192.168.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
```

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/8 ms

Router1#ping 192.168.1.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.5, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

Router1#ping 192.168.2.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms

Router1#ping 192.168.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/7/8 ms

Router1#ping 192.168.2.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 12/36/72 ms

Router1#ping 192.168.1.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.5, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

Router1#ping 192.168.0.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.0.3, timeout is 2 seconds:

.....

Router1#configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Router1(config)#interface fastethernet0/1

Router1(config-if)#ip address 192.168.1.2 255.255.255.252

Router1(config-if)#no shutdown

Router1(config-if)#duplex auto

Router1(config-if)#speed auto

Router1(config-if)#exit

Router1(config)#router rip

Router1(config-router)#version 2

Router1(config-router)#network 192.168.1.0

Router1(config-router)#network 192.168.2.0

Router1(config-router)#exit

Router1(config)#end

Router1#

*Mar 1 01:10:09.527: %SYS-5-CONFIG_I: Configured from console by console

Router1#sh ip int brie

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	192.168.2.1	YES	manual	up	up
FastEthernet0/1	192.168.1.2	YES	manual	up	up
NVI0	192.168.2.1	YES	unset	up	up

Router1#ping 192.168.2.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/8 ms

Router1#ping 192.168.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

Router1#ping 192.168.2.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 16/28/44 ms

Router1#ping 192.168.1.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.5, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 20/25/36 ms

Router1#wr

Building configuration...

[OK]

Router1#

*Mar 1 01:11:16.891: %SYS-2-MALLOCFAIL: Memory allocation of 32768 bytes failed from 0x600286B4, alignment 0

Pool: Processor Free: 25176 Cause: Not enough free memory

Alternate Pool: None Free: 0 Cause: No Alternate pool

-Process= "Exec", ipl= 0, pid= 244, -Traceback= 0x61467A3C 0x600165C4 0x6001C400 0x6001CBAC 0x600286BC 0x60029620 0x614AF920 0x614AF994 0x614AFA2C 0x60594608 0x614B7090 0x6059E8C0 0x614AF694 0x61476F68 0x61477830 0x614D3978

Router1#ping 192.168.2.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/8 ms

Router1#ping 192.168.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/8 ms

Router1#ping 192.168.2.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 20/32/56 ms

Router1#ping 192.168.1.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.5, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 20/28/48 ms

Router1#wr

Building configuration...

[OK]

Router1#

*Mar 1 01:15:12.887: %SYS-2-MALLOCFAIL: Memory allocation of 32768 bytes failed from 0x600286B4, alignment 0

Pool: Processor Free: 25152 Cause: Not enough free memory

Alternate Pool: None Free: 0 Cause: No Alternate pool

-Process= "Exec", ipl= 0, pid= 244, -Traceback= 0x61467A3C 0x600165C4 0x6001C400 0x6001CBAC 0x600286BC 0x60029620 0x614AF920 0x614AF994 0x614AFA2C 0x60594608 0x614B7090 0x6059E8C0 0x614AF694 0x61476F68 0x61477830 0x614D3978

Router1#

*Mar 1 01:19:59.647: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

*Mar 1 01:20:00.643: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router1#

*Mar 1 01:20:01.647: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router1#

*Mar 1 01:21:01.647: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router1#

*Mar 1 01:22:34.651: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router1#

*Mar 1 01:22:35.655: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

*Mar 1 01:22:36.647: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router1#

*Mar 1 01:23:36.651: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router1#

*Mar 1 01:24:36.647: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router1#

*Mar 1 01:25:36.639: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router1#

*Mar 1 01:26:36.631: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router1#

*Mar 1 01:27:36.915: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router1#

*Mar 1 01:28:36.907: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router1#

*Mar 1 01:29:37.447: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router1#

*Mar 1 01:30:37.443: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router1#

*Mar 1 01:31:37.439: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router1#

*Mar 1 01:32:37.431: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPF0F75584939D Port 2 (full duplex).

Router1#sh ip int brief

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	192.168.2.1	YES	manual	up	up
FastEthernet0/1	192.168.1.2	YES	manual	up	up
NVI0	192.168.2.1	YES	unset	up	up

Router1#ping 192.168.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 8/16/24 ms

Router1#wr

Building configuration...

[OK]

Router1#

*Mar 1 01:45:28.927: %SYS-2-MALLOCFAIL: Memory allocation of 32768 bytes failed from 0x600286B4, alignment 0

Pool: Processor Free: 25152 Cause: Not enough free memory

Alternate Pool: None Free: 0 Cause: No Alternate pool

-Process= "Exec", ipl= 0, pid= 244, -Traceback= 0x61467A3C 0x600165C4 0x6001C400 0x6001CBAC 0x600286BC 0x60029620 0x614AF920 0x614AF994 0x614AFA2C 0x60594608 0x614B7090 0x6059E8C0 0x614AF694 0x61476F68 0x61477830 0x614D3978

Router1#ping 192.168.1.6

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.6, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

Router1#configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Router1(config)#ip route 0.0.0.0 0.0.0.0 192.168.1.6

Router1(config)#exit

Router1#end

*Mar 1 01:47:54.499: %SYS-5-CONFIG_I: Configured from console by console

```
Router1#end
Translating "end"
Translating "end"
% Unknown command or computer name, or unable to find computer address
Router1#exit
Router1 con0 is now available
Press RETURN to get started.
Router1#sh ip int brief
Interface              IP-Address      OK? Method Status                Protocol
FastEthernet0/0        192.168.2.1     YES manual up                    up
FastEthernet0/1        192.168.1.2     YES manual up                    up
NVI0                   192.168.2.1     YES unset  up                    up
Router1#ping 192.168.2.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Router1#ping 192.1.2
% Unrecognized host or address, or protocol not running.
Router1#ping 192.168.2.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/7/8 ms
Router1#ping 192.168.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/8 ms
Router1#ping 192.168.2.2


Type escape sequence to abort.
```

Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 40/49/60 ms

Router1#ping 192.168.1.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.5, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 16/31/52 ms

Router1#ping 192.168.1.6

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.6, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

Router1#ping 192.168.1.6

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.6, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

Router1#

Connected to Dynamips VM "R2" (ID 1, type c3725) - Console port

Press ENTER to get the prompt.

R2#

R2#configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

R2(config)#hostname Router2

Router2(config)#boot-start-marker

Router2(config)#boot-end-marker

```
Router2(config)#no aaa new-model
Router2(config)#ip cef
Router2(config)#no ip domain lookup
Router2(config)#multilink bundle-name authenticated
Router2(config)#interface fastethernet0/0
Router2(config-if)#ip address 192.168.2.2 255.255.255.0
Router2(config-if)#no shutdown
Router2(config-if)#duplex auto
Router2(config-if)#speed auto
Router2(config-if)#description Frm_Router1
Router2(config-if)#interface fastethernet0/1
Router2(config-if)#ip address 192.168.1.5 255.255.255.0
Router2(config-if)#duplex auto
Router2(config-if)#speed auto
Router2(config-if)#no shutdown
Router2(config-if)#exit
Router2(config)#router rip
Router2(config-router)#version 2
Router2(config-router)#network 192.168.1.0
Router2(config-router)#network 192.168.2.0
Router2(config-router)#exit
Router2(config)#end
Router2#
*Mar  1 00:48:20.503: %SYS-5-CONFIG_I: Configured from console by console
Router2#sh ip int brie
Interface              IP-Address      OK? Method Status                Protocol
FastEthernet0/0        192.168.2.2     YES manual up                    up
FastEthernet0/1        192.168.1.5     YES manual up                    up
NVI0                   192.168.2.2     YES unset  up                    up
Router2#ping 192.168.2.2

Type escape sequence to abort.
```

Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/8 ms

Router2#ping 192.168.2.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

Router2#interface fastethernet 0/0

^

% Invalid input detected at '^' marker.

Router2#configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Router2(config)#interface fastethernet0/0

Router2(config-if)#ip address 192.168.2.2 255.255.255.252

Router2(config-if)#duplex auto

Router2(config-if)#speed auto

Router2(config-if)#no shutdown

Router2(config-if)#exit

Router2(config)#interface fastethernet0/1

Router2(config-if)#ip address 192.168.1.5 255.255.255.252

Router2(config-if)#no shutdown

Router2(config-if)#duplex auto

Router2(config-if)#speed auto

Router2(config-if)#exit

Router2(config)#exit

Router2#confi

*Mar 1 00:56:28.887: %SYS-5-CONFIG_I: Configured from console by console

Router2#configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

```
Router2(config)#router rip

Router2(config-router)#version e

 Wrong Rip version, only version 1 or version 2 is valid

Router2(config-router)#network 192.168.1.0

Router2(config-router)#network 192.168.2.0

Router2(config-router)#exit

Router2(config)#end

Router2#

*Mar  1 00:57:18.515: %SYS-5-CONFIG_I: Configured from console by console

Router2#sh ip int brief

Interface              IP-Address      OK? Method Status              Protocol

FastEthernet0/0        192.168.2.2     YES manual up                  up

FastEthernet0/1        192.168.1.5     YES manual up                  up

NVI0                   192.168.2.2     YES unset  up                  up

Router2#wr

Building configuration...

[OK]

Router2#

*Mar  1 00:57:50.083: %SYS-2-MALLOCFAIL: Memory allocation of 32768 bytes failed from 0x600286B4, alignment 0

Pool: Processor  Free: 20288  Cause: Not enough free memory

Alternate Pool: None  Free: 0  Cause: No Alternate pool

 -Process= "Exec", ipl= 0, pid= 92,  -Traceback= 0x61467A3C 0x600165C4 0x6001C400 0x6001CBAC 0x600286BC
0x60029620 0x614AF920 0x614AF994 0x614AFA2C 0x60594608 0x614B7090 0x6059E8C0 0x614AF694 0x61476F68
0x61477830 0x614D3978

Router2#ping 192.168.2.1


Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 16/24/44 ms

Router2#ping 192.168.1.5


Type escape sequence to abort.
```

Sending 5, 100-byte ICMP Echos to 192.168.1.5, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/8 ms

Router2#ping 192.168.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 20/28/52 ms

Router2#ping 192.168.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 20/40/72 ms

Router2#wr

Building configuration...

[OK]

Router2#

*Mar 1 01:03:30.983: %SYS-2-MALLOCFAIL: Memory allocation of 32768 bytes failed from 0x600286B4, alignment 0

Pool: Processor Free: 19636 Cause: Not enough free memory

Alternate Pool: None Free: 0 Cause: No Alternate pool

-Process= "Exec", ipl= 0, pid= 92, -Traceback= 0x61467A3C 0x600165C4 0x6001C400 0x6001CBAC 0x600286BC 0x60029620 0x614AF920 0x614AF994 0x614AFA2C 0x60594608 0x614B7090 0x6059E8C0 0x614AF694 0x61476F68 0x61477830 0x614D3978

Router2#ping 192.168.2.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 20/36/60 ms

Router2#ping 192.168.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 16/26/40 ms

Router2#sh ip int brief

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	192.168.2.2	YES	manual	up	up
FastEthernet0/1	192.168.1.5	YES	manual	up	up
NVI0	192.168.2.2	YES	unset	up	up

Router2#ping 192.168.2.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/8 ms

Router2#ping 192.168.1.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.5, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/7/8 ms

Router2#ping 192.168.2.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 20/26/36 ms

Router2#ping 192.168.2.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 20/28/36 ms

Router2#ping 192.168.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 16/26/40 ms

Router2#ping 172.0.0.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.0.0.1, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

Router2#ping 192.168.0.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.0.3, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

Router2#sh ip int brie

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	192.168.2.2	YES	manual	up	up
FastEthernet0/1	192.168.1.5	YES	manual	up	up
NVI0	192.168.2.2	YES	unset	up	up

Router2#ping 192.168.2.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 20/24/36 ms

Router2#ping 192.168.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 20/28/52 ms

Router2#wr

Building configuration...

[OK]

Router2#

*Mar 1 01:11:59.195: %SYS-2-MALLOCFAIL: Memory allocation of 32768 bytes failed from 0x600286B4, alignment 0

Pool: Processor Free: 19636 Cause: Not enough free memory

Alternate Pool: None Free: 0 Cause: No Alternate pool

-Process= "Exec", ipl= 0, pid= 92, -Traceback= 0x61467A3C 0x600165C4 0x6001C400 0x6001CBAC 0x600286BC 0x60029620 0x614AF920 0x614AF994 0x614AFA2C 0x60594608 0x614B7090 0x6059E8C0 0x614AF694 0x61476F68 0x61477830 0x614D3978

Router2#wr

Building configuration...

[OK]

Router2#

*Mar 1 01:15:17.339: %SYS-2-MALLOCFAIL: Memory allocation of 32768 bytes failed from 0x600286B4, alignment 0

Pool: Processor Free: 19636 Cause: Not enough free memory

Alternate Pool: None Free: 0 Cause: No Alternate pool

-Process= "Exec", ipl= 0, pid= 92, -Traceback= 0x61467A3C 0x600165C4 0x6001C400 0x6001CBAC 0x600286BC 0x60029620 0x614AF920 0x614AF994 0x614AFA2C 0x60594608 0x614B7090 0x6059E8C0 0x614AF694 0x61476F68 0x61477830 0x614D3978

Router2#sh ip int brief

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	192.168.2.2	YES	manual	up	up
FastEthernet0/1	192.168.1.5	YES	manual	up	up
NVI0	192.168.2.2	YES	unset	up	up

Router2#ping 192.168.2.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/6/8 ms

Router2#ping 192.168.1.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.5, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/6/8 ms

Router2#ping 192.168.1.6

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.6, timeout is 2 seconds:

.!!!!

Success rate is 80 percent (4/5), round-trip min/avg/max = 8/9/12 ms

Router2#ping 192.168.1.6

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.6, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/16/32 ms

Router2#ping 192.168.2.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/31/36 ms

Router2#ping 192.168.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 16/24/44 ms

Router2#ping 192.168.1.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.3, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 20/42/84 ms

Router2#ping 192.168.1.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.3, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 20/36/84 ms

Router2#ping 192.168.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

Router2#ping 192.168.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

Router2#ping 192.168.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

Router2#wr

Building configuration...

[OK]

Router2#

*Mar 1 01:45:15.583: %SYS-2-MALLOCFAIL: Memory allocation of 32768 bytes failed from 0x600286B4, alignment 0

Pool: Processor Free: 19636 Cause: Not enough free memory

Alternate Pool: None Free: 0 Cause: No Alternate pool

-Process= "Exec", ipl= 0, pid= 92, -Traceback= 0x61467A3C 0x600165C4 0x6001C400 0x6001CBAC 0x600286BC 0x60029620 0x614AF920 0x614AF994 0x614AFA2C 0x60594608 0x614B7090 0x6059E8C0 0x614AF694 0x61476F68 0x61477830 0x614D3978

Router2#wr

Building configuration...

[OK]

Router2#wr

Building configuration...

[OK]

Router2#

*Mar 1 01:52:39.643: %SYS-2-MALLOCFAIL: Memory allocation of 32768 bytes failed from 0x600286B4, alignment 0

Pool: Processor Free: 19636 Cause: Not enough free memory

Alternate Pool: None Free: 0 Cause: No Alternate pool

-Process= "Exec", ipl= 0, pid= 92, -Traceback= 0x61467A3C 0x600165C4 0x6001C400 0x6001CBAC 0x600286BC 0x60029620 0x614AF920 0x614AF994 0x614AFA2C 0x60594608 0x614B7090 0x6059E8C0 0x614AF694 0x61476F68 0x61477830 0x614D3978

Router2#wr

Building configuration...

[OK]

Router2#

//Code for Routers encryption configuration.

Connected to Dynamips VM "R1" (ID 11, type c3725) - Console port

Press ENTER to get the prompt.

[OK]

Smart Init is disabled. IOMEM set to: 5

Using iomem percentage: 5

Restricted Rights Legend

Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706

Cisco IOS Software, 3700 Software (C3725-ADVENTERPRISEK9-M), Version 12.4(15)T5, RELEASE SOFTWARE (fc4)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2008 by Cisco Systems, Inc.
Compiled Wed 30-Apr-08 18:27 by prod_rel_team
Image text-base: 0x60008930, data-base: 0x63647200

BIST FAILED...
Unknown file system detected.
Use format command to format the card as DOS File System.
Or use erase command to format the card as Low End File System.

This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for

compliance with U.S. and local country laws. By using this product you

agree to comply with applicable laws and regulations. If you are unable

to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:

http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to

export@cisco.com.

Cisco 3725 (R7000) processor (revision 0.1) with 124928K/6144K bytes of memory.

Processor board ID FTX0945W0MY

R7000 CPU at 240MHz, Implementation 39, Rev 2.1, 256KB L2, 512KB L3 Cache

2 FastEthernet interfaces

DRAM configuration is 64 bits wide with parity enabled.

55K bytes of NVRAM.

16384K bytes of ATA System CompactFlash (Read/Write)

Installed image archive

SETUP: new interface FastEthernet0/0 placed in "shutdown" state

SETUP: new interface FastEthernet0/1 placed in "shutdown" state

Press RETURN to get started!

*Mar 1 00:00:15.159: %LINEPROTO-5-UPDOWN: Line protocol on Interface VoIP-Null0, changed state to up

*Mar 1 00:00:15.163: %LINEPROTO-5-UPDOWN: Line protocol on Interface IPv6-mpls, changed state to up

*Mar 1 00:00:15.667: %SYS-5-CONFIG_I: Configured from memory by console

*Mar 1 00:00:15.759: %LINK-5-CHANGED: Interface FastEthernet0/1, changed state to administratively down

*Mar 1 00:00:15.759: %LINK-5-CHANGED: Interface FastEthernet0/0, changed state to administratively down

*Mar 1 00:00:16.779: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to down

*Mar 1 00:00:16.779: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to down

*Mar 1 00:00:17.123: %SYS-5-RESTART: System restarted --

Cisco IOS Software, 3700 Software (C3725-ADVENTERPRISEK9-M), Version 12.4(15)T5, RELEASE SOFTWARE (fc4)

Technical Support: http://www.cisco.com/techsupport

Copyright (c) 1986-2008 by Cisco Systems, Inc.

Compiled Wed 30-Apr-0

R1#

R1#

8 18:27 by prod_rel_team

*Mar 1 00:00:17.147: %SNMP-5-COLDSTART: SNMP agent on host R1 is undergoing a cold start

*Mar 1 00:00:17.359: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is OFF

*Mar 1 00:00:17.363: %CRYPTO-6-GDOI_ON_OFF: GDOI is OFF

R1#

R1#

R1#

R1#

R1#

R1#conf t

Enter configuration commands, one per line. End with CNTL/Z.

R1(config)#crypto isakmp enable

R1(config)#crypto isakmp policy 10

R1(config-isakmp)#authentication pre-share

R1(config-isakmp)#hash sha

R1(config-isakmp)#encryption aes 256

R1(config-isakmp)#group 5

R1(config-isakmp)#lifetime 86400

R1(config-isakmp)#exit

R1(config)#crypto isakmp key SCADASECURE address 192.168.2.1

R1(config)#crypto isakmp keepalive 10 2 periodic

R1(config)#exit

R1#

```
*Mar  1 00:02:54.243: %SYS-5-CONFIG_I: Configured from console by console
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#crypto ipsec transform-set MYSETNAME esp-aes 256 esp-sha-hmac
R1(cfg-crypto-trans)#mode tunnel
R1(cfg-crypto-trans)#crypto ipsec transform-MYSETNAME
                    ^
% Invalid input detected at '^' marker.

R1(cfg-crypto-trans)#ah-md5-hmac AH-HMAC-MD5 transform
                    ^
% Invalid input detected at '^' marker.

R1(cfg-crypto-trans)#exit
R1(config)#conf t
                ^
% Invalid input detected at '^' marker.

R1(config)#$ 101 permit ip 192.168.0.0 0.0.0.255 192.168.2.0 0.0.0.255
R1(config)#crypto map Router1_to_Router2 10 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
        and a valid access list have been configured.
R1(config-crypto-map)#set peer 192.168.2.2
R1(config-crypto-map)#match address 101
R1(config-crypto-map)#set transform-set MYSETNAME
R1(config-crypto-map)#interface fastethernet
% Incomplete command.

R1(config)#interface fastethernet0/0
R1(config-if)#crypto map Router1_to_Router2
R1(config-if)#ip
*Mar  1 00:09:32.883: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
```

R1(config-if)#ip route 192.168.1.6 255.255.255.252 192.168.2.1

%Inconsistent address and mask

R1(config)#ip route 192.168.1.6 255.255.255.255 192.168.2.1

R1(config)#exit

R1#

*Mar 1 00:10:58.519: %SYS-5-CONFIG_I: Configured from console by console

R1#conf t

Enter configuration commands, one per line. End with CNTL/Z.

R1(config)#interface fastethernet 0/0

R1(config-if)#ip address

% Incomplete command.

R1(config-if)#ip address 192.168.2.1 255.255.255.252

R1(config-if)#boot-start-marker

R1(config)#boot-end-marker

R1(config)#no aaa new-model

R1(config)#ip cef

R1(config)#no shutdown

% Incomplete command.

R1(config)#no ip domain lookup

R1(config)#multilink bundle-name authenticated

R1(config)#no shutdown

% Incomplete command.

R1(config)#exit

R1#no shu

*Mar 1 00:13:28.499: %SYS-5-CONFIG_I: Configured from console by console

R1#no shutdown

     ^

% Invalid input detected at '^' marker.

R1#conf t

Enter configuration commands, one per line. End with CNTL/Z.

R1(config)#interface fastethernet 0/1

R1(config-if)#ip address 192.168.1.2 255.255.255.252

R1(config-if)#no shutdown

R1(config-if)#

*Mar 1 00:15:56.591: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up

*Mar 1 00:15:57.591: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up

R1(config-if)#exit

R1(config)#router rip

R1(config-router)#version 2

R1(config-router)#

*Mar 1 00:16:17.195: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPA0CF5B815B5B Port 2 (full duplex).

R1(config-router)#network 192.168.0.0

R1(config-router)#network 192.168.1.0

R1(config-router)#network 192.168.2.0

R1(config-router)#exit

R1(config)#end

R1#

*Mar 1 00:16:43.291: %SYS-5-CONFIG_I: Configured from console by console

R1#

ROUTER 2 CONFIGURATION

R2#

R2#conf t

Enter configuration commands, one per line. End with CNTL/Z.

R2(config)#hostname Router2

Router2(config)#boot-start-marker

Router2(config)#boot-end-marker

Router2(config)#no aaa new-model

Router2(config)#ip cef

```
Router2(config)#no ip domain lookup

Router2(config)#multilink bundle-name authenticated

Router2(config)#interface fastethernet 0/0

Router2(config-if)#ip address 192.168.2.2 255.255.255.252

Router2(config-if)#no shutdown

Router2(config-if)#

*Mar  1 00:20:26.207: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up

*Mar  1 00:20:27.207: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

Router2(config-if)#interface fastethernet 0/1

Router2(config-if)#ip address 192.168.1.5 255.255.255.252

Router2(config-if)#no shutdown

Router2(config-if)#

*Mar  1 00:21:31.351: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up

*Mar  1 00:21:32.351: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up

Router2(config-if)#exit

Router2(config)#conf t
               ^

% Invalid input detected at '^' marker.


Router2(config)#crypto isa

*Mar  1 00:22:17.415: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex),
with SEPA0CF5B815B5B Port 2 (full duplex).

Router2(config)#crypto isakmp enable

Router2(config)#crypto isakmp policy 10

Router2(config-isakmp)#authentication pre-share

Router2(config-isakmp)#hash sha

Router2(config-isakmp)#encryption aes 256

Router2(config-isakmp)#group 5

Router2(config-isakmp)#lifetime

*Mar  1 00:23:17.415: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex),
with SEPA0CF5B815B5B Port 2 (full duplex).

Router2(config-isakmp)#lifetime 86400

Router2(config-isakmp)#exit
```

Router2(config)#crypto isakmp key SCADASECURE address 192.168.2.2

Router2(config)#crypto isakmp keepalive 10 2 periodic

Router2(config)#

*Mar 1 00:24:17.419: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPA0CF5B815B5B Port 2 (full duplex).

Router2(config)#exit

Router2#c

*Mar 1 00:24:21.619: %SYS-5-CONFIG_I: Configured from console by consoleo

Router2#conf t

Enter configuration commands, one per line. End with CNTL/Z.

Router2(config)#crypto ipsec transform-set MYSETNAME esp-aes 256 esp-sha-hmac

Router2(cfg-crypto-trans)#mode tunnel

Router2(cfg-crypto-trans)#crypto ipsec transform-set MYSETNA

*Mar 1 00:25:17.423: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPA0CF5B815B5B Port 2 (full duplex).

Router2(cfg-crypto-trans)#crypto ipsec transform-set MYSETNAME?

% Unrecognized command

Router2(cfg-crypto-trans)#crypto ipsec transform-set MYSETNAME

                ^

% Invalid input detected at '^' marker.

Router2(cfg-crypto-trans)#crypto ipsec transform-set MYSETNAME?

% Unrecognized command

Router2(cfg-crypto-trans)#crypto ipsec transform-set MYSETNAME

                ^

% Invalid input detected at '^' marker.

Router2(cfg-crypto-trans)#exit

Router2(config)#access-list permit ip 192.1

*Mar 1 00:26:17.423: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPA0CF5B815B5B Port 2 (full duplex).

Router2(config)#$ permit ip 192.168.1.0 0.0.0.255 192.168.0.0 0.0.0.255

access-list permit ip 192.168.1.0 0.0.0.255 192.168.0.0 0.0.0.255

```
          ^
% Invalid input detected at '^' marker.


Router2(config)#$ permit ip 192.168.1.0 0.0.0.255 192.168.0.0 0.0.0.255
access-list permit ip 192.168.1.0 0.0.0.255 192.168.0.0 0.0.0.255
          ^
% Invalid input detected at '^' marker.


Router2(config)#exit
Router2#
*Mar  1 00:27:05.131: %SYS-5-CONFIG_I: Configured from console by console
Router2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router2(config)#access-list 101 per
*Mar  1 00:27:17.419: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex),
with SEPA0CF5B815B5B Port 2 (full duplex).
Router2(config)#access-list 101 permit ip 192.168.0.0 0.0.0.255 192.168.2
*Mar  1 00:28:17.419: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex),
with SEPA0CF5B815B5B Port 2 (full duplex).
Router2(config)#$ 101 permit ip 192.168.0.0 0.0.0.255 192.168.2.0 0.0.0.255
Router2(config)#interface fastethernet 0/1
Router2(config-if)#duplex full
Router2(config-if)#no shut
*Mar  1 00:28:54.195: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
Router2(config-if)#no shutdown
Router2(config-if)#exit
Router2(config)#crypto
*Mar  1 00:29:17.459: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex),
with SEPA0CF5B815B5B Port 2 (full duplex).
Router2(config)#crypto map Router2_to_Router1 10 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
      and a valid access list have been configured.
Router2(config-crypto-map)#set peer 192.168.2.1
```

Router2(config-crypto-map)#match address 101

Router2(config-crypto-map)#

*Mar 1 00:30:17.467: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPA0CF5B815B5B Port 2 (full duplex).

Router2(config-crypto-map)#set transform-set MYSETNAME

Router2(config-crypto-map)#interface fastethernet 0/0

Router2(config-if)#cryptomap Router2_to_Router1

*Mar 1 00:31:17.475: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPA0CF5B815B5B Port 2 (full duplex).

Router2(config-if)#cryptomap Router2_to_Router1

                  ^

% Invalid input detected at '^' marker.

Router2(config-if)#crypto map Router2_to_Router1

Router2(config-if)#

*Mar 1 00:31:37.835: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON

Router2(config-if)#ip route 192.168.0.0

*Mar 1 00:32:17.475: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPA0CF5B815B5B Port 2 (full duplex).

Router2(config-if)#ip route 192.168.0.0 0.0.0.255 192.168.2.0

%Inconsistent address and mask

Router2(config)#ip route 192.168.0.0 0.0.0.255 192.168.1.0

%Inconsistent address and mask

Router2(config)#ip route 192.168.0.0 255.255.255.255 192.168.1.0

Router2(config)#

*Mar 1 00:33:17.483: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with SEPA0CF5B815B5B Port 2 (full duplex).

Router2(config)#exit

Router2#

*Mar 1 00:33:26.291: %SYS-5-CONFIG_I: Configured from console by console

Router2#protocol

Translating "protocol"

Translating "protocol"

% Unknown command or computer name, or unable to find computer address

Router2#conf t

Enter configuration commands, one per line. End with CNTL/Z.

Router2(config)#protocol

^

% Invalid input detected at '^' marker.

Router2(config)#p

% Ambiguous command: "p"

Router2(config)#exit

Router2#p

Protocol [ip]:

*Mar 1 00:34:00.131: %SYS-5-CONFIG_I: Configured from console by console

Target IP address:

% Bad IP address

Router2#

Router2#