Utilization of different Encryption Schemes for Securing SCADA Component Communication

Minkyu Choi

Abstract: This dissertation is a study on the comparison of different Encryption Schemes for Securing SCADA Component Communication. SCADA Communication is a core component of a SCADA Monitoring System. SCADA (Supervisory Control and Data Acquisition) communication can take place in a number of ways. Early SCADA communication took place over radio, modem, or dedicated serial lines. Today, it is much more common for SCADA communications to travel over LAN or WLAN. The process of communication over a SCADA system involves several different SCADA system components. These include the sensors and control relays, Remote Terminal Units (RTUs), SCADA master units, and the overall communication network. Each of these parts is necessary for effective SCADA communication. A system can effectively monitor alarms and status updates within the network only when all of these system components function properly. For more complete monitoring of SCADA communications, operators must deploy advanced RTUs. The RTU is where most SCADA communication is gathered within the system. Values from inputs and outputs, referred to as SCADA points, are sent from individual sensors to the RTU. The RTU is responsible for forwarding these SCADA communications to the master station, or Human-Machine Interface (HMI). Common misconception regarding SCADA security was SCADA networks were isolated from all other networks and so attackers could not access the system. As the industry grows, the demand for more connectivity also increased. From a small range network, SCADA systems are sometimes connected to other networks like the internet. The open standards also make it very easy for attackers to gain in-depth knowledge about the working of these SCADA networks. The use of COTS hardware and software to develop devices for operating in the SCADA network also contribute to its lack of security. Devices that are designed to operate in safety critical environments are usually designed to failsafe, but security vulnerabilities could be exploited by an attacker to disable the fail-safe mechanisms. This makes these devices must not only be designed for safety but also for security. Because of so many vulnerabilities encryption Schemes are applied to secure the communication between the components. This work compares different Encryption Schemes for Securing SCADA Component Communication.

Keywords: SCADA Security, Encryption, SCADA Component Communication, SCADA Networks

1. Introduction

As SCADA (Supervisory Control and Data Acquisition) communications can contain sensitive data, it is important to utilize secure LAN networks when transmitting data to RTUs (Remote Terminal Units) and the master station. However, LAN can be expensive to install at remote sites, and it may not be feasible for an organization to immediately incorporate all of their remote sites into the LAN network. SCADA Communication is a core component of a SCADA Monitoring System. SCADA communication can take place in a number of ways. Early SCADA communication took place over radio, modem, or dedicated serial lines. They have traditionally used combinations of radio and direct serial or modem connections to meet the communication requirements. Today, it is much more common for SCADA communications to travel over LAN or WLAN. The process of communication over a SCADA system involves several different SCADA system components. These include the sensors and control relays, RTUs, SCADA master units, and the overall communication network. Each of these parts is necessary for effective SCADA communication. A system can effectively monitor alarms and status updates within the network only when all of these system components function properly. For more complete monitoring of SCADA communications, operators must deploy advanced RTUs. The RTU is where most SCADA communication is gathered within the system. Values from inputs and outputs, referred to as SCADA points, are sent from individual sensors to the RTU. The RTU is responsible for forwarding these SCADA communications to the master station, or Human-Machine Interface (HMI).

For the past several years several researches have been done on the SCADA security issues. Along with the works in the research community, the international standard bodies also have worked to derive the standard documents for the SCADA security. The purpose of this study is not only to define the challenges for a known isolated SCADA system, but also to organize the results that these isolated cases is now vulnerable to cyber attack threats. Several solutions such as Application of Asymmetric-key Encryption to SCADA Security, Symmetric-Key Encryption for Wireless Internet SCADA, Communication Security for SCADA using a Cross Crypto Scheme was discussed to answer the challenges and SCADA communication security issues that are raised. The current results on these challenges will be summarized from the efforts of the international organization as well as research communities.

1.2 Statement of the Problem

As more components of control systems become interconnected with the outside world using IP-based standards, the probability and impact of a cyber attack heighten. The complexity of modern SCADA systems leaves many vulnerabilities as well as vectors for attack. Attacks can come from many places, including indirectly through the corporate network, virtual private networks (VPN), wireless networks, and dial-up modems. Possible attack vectors on an SCADA system include:

- Backdoors and holes in network perimeter;
- Vulnerabilities in common protocols;
- Database attacks;
- Communications hijacking and 'man-in-the-middle' attacks.

All this listed attacks are threat to SCADA's Confidentiality, Authentication, Integrity and Non-repudiation aspects.

1.3 Vulnerabilities in SCADA

Control Systems such as SCADA can be described as a system that is controlling and monitoring a process or processes. Examples of SCADA controlled processes are the opening and closing of water valves, control of power relays, and switching of train tracks. SCADA is composed of four basic components: the HMI (Human Machine Interface), Master Station, sensors, and a communication network. SCADA systems are what were responsible for controlling the above-mentioned facilities, and in each case these were either failed or compromised.

Traditionally, a lot of these control systems operated in isolated environments with proprietary technologies. Most of them operate in its own private network. Consequently, they faced little to no cyber-security risk from external attackers. However, today's modernization and the adoption of available commercial technologies have resulted in these systems becoming increasingly connected and interdependent. SCADA systems nowadays utilize the public networks such as the internet. In fact, almost every major operating system is being used across the range of vendor products. Normally, machines with operating systems such as Windows XP, Windows and Linux in this space are installed on rugged machines that can handle industrial conditions and utilize redundancy in the design of the hardware. It is known that the said operating systems are not mainly designed for this purpose [1].

Common misconception regarding SCADA security was SCADA networks were isolated from all other networks and so attackers could not access the system [2]. As the industry grows, the demand for more connectivity also increased. From a small range network, SCADA systems are sometimes connected to other networks like the internet. The open standards also make it very easy for attackers to gain in-depth knowledge about the working of these SCADA networks. The use of COTS (Commercial Off-the-Shelf) hardware and software to develop devices for operating in the SCADA network also contribute to its lack of security. Devices that are designed to operate in safety-critical environments are usually designed to failsafe, but security vulnerabilities could be exploited by an attacker to disable the fail-safe mechanisms. This makes these devices must not only be designed for safety but also for security [3].

Supervisory Control and Data Acquisition Systems and its communication protocols designed decades ago, that time security was not the primary concern because of the closed nature of the communications networks and the general model of trusting the data on them since it was installed in a private network. Outsiders will find it difficult to attack the system because of its close nature. As technology modernized, they have become interconnected and have started running more modern services such as Web interfaces and have implemented remote configuration protocols [1]. Because of this upgrades, security has become a bigger issue.

The transition from proprietary technologies to more standardized and open solutions together with the increased number of connections between SCADA systems and office networks and the Internet has made them more vulnerable to attacks. Consequently, the security of SCADA-based systems has come into question as they are increasingly seen as extremely vulnerable to cyberwarfare/cyberterrorism attacks [4][5][23][35].

SCADA is vulnerable because of the following:

- Insufficient concern about security and authentication in the design, deployment and operation of existing SCADA networks;
- The notion that SCADA systems have the benefit of security through obscurity with the use of specialized protocols and proprietary interfaces;
- The notion that SCADA networks are secure because they are disconnected from the Internet;
- The notion that SCADA networks are secure because they are physically secured.

There are two distinguishable threats to a modern SCADA system. The first one is the threat of unauthorized access to the control software, whether it is human access or changes induced intentionally or accidentally by virus infections and other software threats residing on the control host machine. Another one is the threat of packet access to the network segments hosting SCADA devices.

1.4 Organization

In this thesis, approaches to secure the communication between remote components of SCADA over the internet are discussed. The following sections provide information about the Common Threats and Vulnerabilities of Critical Infrastructures, Vulnerabilities in SCADA and Critical Infrastructure Systems, Related Literature such as the background and information about Control Systems, SCADA Systems and Web SCADA. Assessment of the Vulnerabilities of SCADA, Control Systems and Critical Infrastructure Systems as well as The Categorizations of SCADA Communication Protocols is also discussed.

In Section 3, solutions to the cited issues are discussed. Solutions such as Application of Asymmetric-key Encryption to SCADA Security, Symmetric-Key Encryption for Wireless Internet SCADA, Communication Security for SCADA using a Cross Crypto Scheme.

In Section 4, discussion about the application and implementation of the encryption Schemes to current SCADA Systems and the Future of SCADA is explained.

Section 5 contains the conclusion and the summary as well as possible future works related to this subject.

2. Related Literature

2.1 Control Systems

A control system is a device or set of devices to manage, command, direct or regulate the behavior of other devices or systems. There are two common classes of control systems, with many variations and combinations: logic or sequential controls, and feedback or linear controls. Some devices or systems are inherently not controllable. An automatic sequential control system may trigger a series of mechanical actuators in the correct sequence to perform a task. For example various electric and pneumatic transducers may fold and glue a cardboard box, fill it with product and then seal it in an automatic packaging machine. In the case of linear feedback systems, a control loop, including sensors, control

algorithms and actuators, is arranged in such a fashion as to try to regulate a variable at a set point or reference value. An example of this may increase the fuel supply to a furnace when a measured temperature drops. PID (Proportional Integral Derivative) controllers are common and effective in cases such as this. Control systems that include some sensing of the results they are trying to achieve are making use of feedback and so can, to some extent, adapt to varying circumstances. Open-loop control systems do not make use of feedback, and run only in pre-arranged ways [6][7].

2.2 SCADA Systems

Supervisory Control and Data Acquisition (SCADA) existed long time ago when control systems were introduced. SCADA systems that time use data acquisition by using strip chart recorders, panels of meters, and lights. Not similar to modern SCADA systems, there is an operator which manually operates various control knobs exercised supervisory control. These devices are still used to do supervisory control and data acquisition on power generating facilities, plants and factories [3][8]. Telemetry is automatic transmission and measurement of data from remote sources by wire or radio or other means. It is also used to send commands, programs and receives monitoring information from these remote locations. SCADA is the combination of telemetry and data acquisition. Supervisory Control and Data Acquisition system is compose of collecting of the information, transferring it to the central site, carrying out any necessary analysis and control and then displaying that information on the operator screens. The required control actions are then passed back to the process [9][10].

The measurement and control system of SCADA has one master terminal unit (MTU) which could be called the brain of the system and one or more remote terminal units (RTU). The RTUs gather the data locally and send them to the MTU which then issues suitable commands to be executed on site. A system of either standard or customized software is used to collate, interpret and manage the data. Supervisory Control and Data Acquisition (SCADA) is conventionally set upped in a private network not connected to the internet. This is done for the purpose of isolating the confidential information as well as the control to the system itself [8].

Because of the distance, processing of reports and the emerging technologies, SCADA can now be connected to the internet. This can bring a lot of advantages and disadvantages which will be discussed in the sections. Conventionally, relay logic was used to control production and plant systems. With the discovery of the CPU (Central Processing Unit) and other electronic devices, manufacturers incorporated digital electronics into relay logic equipment. Programmable logic controllers or PLC's are still the most widely used control systems in industry. As need to monitor and control more devices in the plant grew, the PLCs were distributed and the systems became more intelligent and smaller in size. PLCs (Programmable logic controllers) and DCS (distributed control systems) are used as shown in the next Figure.

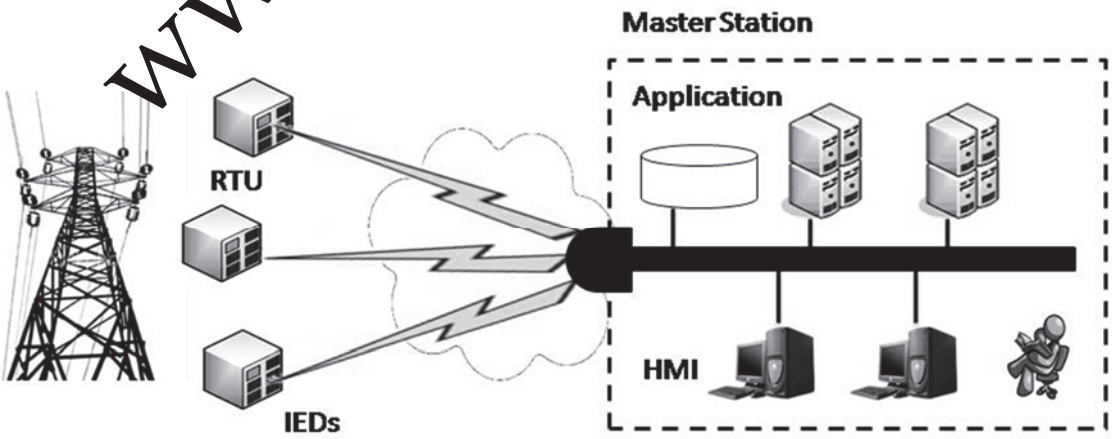

Figure 1. Conventional SCADA Architecture

Data acquisition begins at the RTU, IED (Intelligent Electronic Device) or PLC level and includes meter readings and equipment status reports that are communicated to SCADA as required. Data is then compiled and formatted in such a way that a control room operator using the HMI can make supervisory decisions to adjust or override normal RTU (PLC) controls. Data may also be fed to a Historian, often built on a commodity Database Management System, to allow trending and other analytical auditing [8].

2.2.2 SCADA Hardware

A SCADA system consists of a number of remote terminal units (RTUs) collecting field data and sending that data back to a master station, via a communication system [4]. The master station displays the acquired data and allows the operator to perform remote control tasks. The accurate and timely data allows for optimization of the plant operation and process. Other benefits include more efficient, reliable and most importantly, safer operations. These results in a lower cost of operation compared to earlier non-automated systems [8].

Supervisory Control and Data Acquisition Systems usually have Distributed Control System components. PLCs or RTUs are also commonly used; they are capable of autonomously executing simple logic processes without a master computer controlling it. A functional block programming language, IEC 61131-3, is frequently used to create programs which run on these PLCs and RTUs. This allows SCADA system engineers to perform both the design and implementation of a program to be executed on an RTU or PLC. From 1998, major PLC manufacturers have offered integrated HMI/SCADA systems, many use open and non-proprietary communications protocols. Many third-party HMI/SCADA packages, offering built-in compatibility with most major PLCs, have also entered the market, allowing mechanical engineers, electrical engineers and technicians to configure HMIs themselves. Much other hardware are also basing its functionality to those of PLC's [11].

The communications system provides the pathway for communication between the master station and the remote sites. This communication system can be wire, fiber optic, radio, telephone line, microwave and possibly even satellite. Specific protocols and error detection philosophies are used for efficient and optimum transfer of data. The master station (or sub-masters) gather data from the various RTUs and generally provide an operator interface for display of information and control of the remote sites. In large telemetry systems, sub-master sites gather information from remote sites and act as a relay back to the control master station [8].

2.2.3 SCADA Software

Supervisory Control and Data Acquisition software can be divided into proprietary type or open type. Proprietary software are developed and designed for the specific hardware and are usually sold together. The main problem with these systems is the overwhelming reliance on the supplier of the system. Open software systems are designed to communicate and control different types of hardware. It is popular because of the interoperability they bring to the system [9]. WonderWare and Citect are just two of the open software packages available in the market for SCADA systems. Some packages are now including asset management integrated within the SCADA system [8].

2.2.4 SCADA Human Machine Interface

In SCADA and in the industrial design field of human-machine interaction, the user interface is (a place) where interaction between humans and machines occurs. The goal of interaction between a human and a machine at the user interface is effective operation and control of the machine, and feedback from the machine which aids the operator in making operational decisions. Examples of this broad concept of user interfaces include the interactive aspects of computer operating systems, hand tools, heavy machinery operator controls and process controls [8].

The goal of human-machine interaction engineering is to produce a user interface which makes it easy, efficient, and enjoyable to operate a machine in the way which produces the desired result. This generally means that the operator needs to provide minimal input to achieve the desired output, and also that the machine minimizes undesired outputs to the human. [8] Ever since the increased use of personal

computers and the relative decline in societal awareness of heavy machinery, the term user interface has taken on overtones of the (graphical) user interface, while industrial control panel and machinery control design discussions more commonly refer to human-machine interfaces [3].

2.2.5 SCADA Communication

SCADA systems have traditionally used combinations of radio and direct serial or modem connections to meet communication requirements, although Ethernet and IP over SONET / SDH is also frequently used at large sites such as railways and power stations. The remote management or monitoring function of a SCADA system is often referred to as telemetry. This has also come under threat with some customers wanting SCADA data to travel over their pre-established corporate networks or to share the network with other applications [8].

The legacy of the early low-bandwidth protocols remains, though. SCADA protocols are designed to be very compact and many are designed to send information to the master station only when the master station polls the RTU. Typical legacy SCADA protocols include Modbus RTU, RP-570, Profibus and Conitel [4]. These communication protocols are all SCADA-vendor specific but are widely adopted and used. Standard protocols are IEC 60870-5-101 or 104, IEC 61850 and DNP3. These communication protocols are standardized and recognized by all major SCADA vendors. Many of these protocols now contain extensions to operate over TCP/IP. It is good security engineering practice to avoid connecting SCADA systems to the Internet so the attack surface is reduced [8].

Recently, OLE for Process Control (OPC) has become a widely accepted solution for intercommunicating different hardware and software, allowing communication even between devices originally not intended to be part of an industrial network. Central computer of the data acquisition system, located in the hydro power plant, provides measurements performance according to a preset program, the instrumentation existing at this time and remote communications by RS485 bus, using Master-Slave architecture and IEC1107, Modbus RTU, ASCII protocols [12].

2.3 The Categorization of SCADA Communication Protocols

Communication is very important in SCADA systems. In communication, protocols are needed to be implemented to avoid miscommunications, signaling and authentication errors, and other problems [13]. In order for SCADA systems to obtain its functionality, it needs a protocol for transmitting data. Some of the SCADA protocols include Modbus RTU, RP-570, Profibus and Conitel [4]. These communication protocols are all SCADA-vendor specific but are widely adopted and used. Standard protocols are IEC 61850 (in which T101 branched out), IEC 60870-5-101 or 104, and DNP3. These communication protocols are standardized and recognized by all major SCADA vendors. Many of these protocols is now improved and contain extensions to operate over TCP/IP [13].

Three of the most important part of a SCADA system are Master Station, Remote Terminal (RTU, PLC, IED) and the communication between them. In order to have good communication betweens them, there must be a communication protocol. DNP3 and T101 are two of the most common protocols today. These two open communication protocols that provide for interoperability between systems for telecontrol applications. DNP is widely used in North America, South America, South Africa, Asia and Australia, while IEC 60870-5-101 or T101 is strongly supported in the Europe [13].

2.3.1 IEC 60870-5 Standards

IEC 60870-5 is the collection of standards produced by the IEC(International Electrotechnical Commission). It was created to provide an open standard for the transmission of SCADA telemetry control and information [13]. It provides a detailed functional description for telecontrol equipment and systems for controlling geographically widespread processes specifically for SCADA systems. The standard is intended for application in the electrical industries, and has data objects that are specifically intended for such applications. It is also applicable to general SCADA applications in any industry. But IEC 60870-5 protocol is primarily used in the electrical industries of European countries [14].

2.3.2 DNP3 Protocol

The DNP3 or Distributed Network Protocol is a set of communications protocols used between components in process automation systems [15]. It is usually used is in utilities such as water and electric companies. It is also technically possible to use it in other utilities. It was specifically developed to facilitate communications between various types of data acquisition and control systems. It plays a crucial role in SCADA systems. It is used by SCADA Master Stations or Control Centers, Remote Terminal Units, and Intelligent Electronic Devices. It is primarily used for communications between a master station and IEDs or RTU's. DNP3 supports multiple-slave, peer-to-peer and multiple-master communications. It supports the operational modes of polled and quiescent operation. The latter is also referred to as reporting by exception [13][16][17][18].

2.3.2 Comparison of T101 and DNP3 Protocols

IEC 60870-5-101/104 and DNP3 have basically the same functionality. They both provide solutions to first level of Data Acquisition Interoperability. Many factors are needed when selecting a protocol to be used like the kind of utility where SCADA will be implemented before choosing the proper protocol. The location should also be considered. Like for example if your system is located in America, it is better to use DNP3 since it is better to get technical assistance in case something is wrong.

As discussed, DNP3 is popular in America. Since DNP3 and T101 are open standards, SCADA operators should monitor the development, and make contributions when appropriate, to T101 and DNP3. They should also pursue the developers to include security features on the protocols. This could help develop or improve the protocols in SCADA communication. [13] [13][16][17][18]

Table 1 shows the comparison of both protocols, the DNP3 and the T101.

Table 1. Comparison of T101 and DNP3

	DNP3	**T101**
Organization	DNP user's group	IEC TC 57 WG 03
Standard	Open Industry Specification	IEC Standard
Dominant Market	North America	Europe
Architecture	4-Layer architecture Also supports 7 layer TCP/IP or UDP/IP	3-layer EPA architecture
Main Coverage	Application Layer (Services and Protocol)	Application layer (Services and Protocol)
Device Addressing	Link Contains both source and destination address both always 16 bits Application layer does not	Link address could be 0, 1, 2 bytes Unbalanced link contains slave address

	contain address 32 b point addresses of each data type per device	Balanced link is point to point so link address is optional (may be included for security)
Parameter setting control	Change some communication related parameters; more under development	Few
Cyclic transmission	Available but interval cannot be remotely adjusted	Eliminates static data poll message from master Interrupt by event triggered communication request
Open for other encoding solutions	Yes open for other encoding solutions like XML	No

2.4 Internet based SCADA

Conventional SCADA only have 4 components, the master station, plc/rtu, fieldbus and sensors. Internet SCADA replaces or extends the fieldbus to the internet. This means that the Master Station can be on a different network or location.

In Figure 2, you can see the architecture of SCADA which is connected through the internet. Like a normal SCADA, it has RTUs/PLCs/IEDs, The SCADA Service Provider or the Master Station. This also includes the user-access to SCADA website. This is for the smaller SCADA operators that can avail the services provided by the SCADA service provider. It can either be a company that uses SCADA exclusively. Another component of the internet SCADA is the Customer Application which allows report generation or billing. Along with the fieldbus, the internet is an extension. This is setup like a private network so that only the master station can have access to the remote assets. The master also has an extension that acts as a web server so that the SCADA users and customers can access the data through the SCADA provider website [19][34].

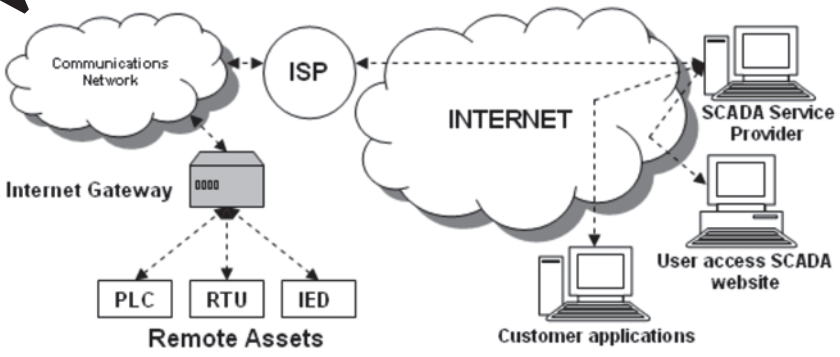

Figure 2. Internet SCADA Architecture [20]

AS the system evolves, SCADA systems are coming in line with standard networking technologies. Ethernet and TCP/IP based protocols are replacing the older proprietary standards. Although certain characteristics of frame-based network communication technology (determinism, synchronization, protocol selection, environment suitability) have restricted the adoption of Ethernet in a few specialized applications, the vast majority of markets have accepted Ethernet networks for HMI/SCADA.

A few vendors have begun offering application specific SCADA systems hosted on remote platforms over the Internet. This removes the need to install and commission systems at the end-user's facility and takes advantage of security features already available in Internet technology, VPNs and SSL. Some concerns include security [20], Internet connection reliability, and latency.

3. Encryption Schemes

In this Section, solutions to the issues and vulnerabilities in SCADA and Web SCADA are discussed. Solutions such as Application of Asymmetric-key Encryption to SCADA Security, Symmetric-Key Encryption for Wireless Internet SCADA and the Proposed Communication Security for SCADA using a Cross Crypto Scheme.

3.1 Asymmetric-key Encryption for SCADA Security

The internet SCADA facility has brought a lot of advantages in terms of control, data generation and viewing. With these advantages, comes the security issues regarding web SCADA. In this section, web SCADA and its connectivity along with the issues regarding security will be discussed. A web SCADA security solution using asymmetric-key encryption will be explained.

Asymmetric key encryption uses different keys for decryption/encryption. These two keys are mathematically related and they form a key pair. One key is kept private, and is called private-key, and the other can be made public, called public-key. Hence this is also called Public Key Encryption. Public key can be sent by mail. A private key is typically used for encrypting the message-digest; in such an application private-key algorithm is called message-digest encryption algorithm. A public key is typically used for encrypting the secret-key; in such a application private-key algorithm is called key encryption algorithm [22][29]. The asymmetric key encryption algorithm is shown in Figure 3.

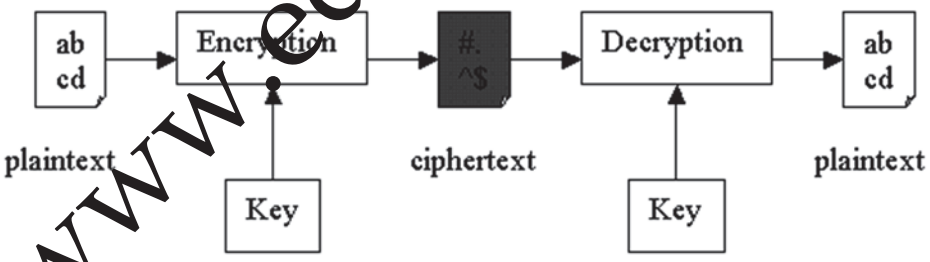

Figure 3. Asymmetric key encryption uses different keys for decryption and encryption

3.2 Symmetric-Key Encryption for Wireless Internet SCADA

As stated in the previous sections, SCADA was connected only in a limited private network when it was introduced. With new technology and facilities, there are also demands of connecting SCADA though the internet. The internet SCADA facility has brought a lot of advantages in terms of control, data viewing and generation. Aside from connecting SCADA to the internet, there are also operators who want to connect their system wirelessly. This can save budget for communication lines [24]. Along with the advantages it brings, are security issues regarding wireless internet SCADA. In this section, we discuss internet SCADA, its connection through wireless communication and the security issues surrounding it. To answer the security issues, a symmetric-key encryption for wireless internet SCADA was proposed under the scope of this work [24].

Symmetric-key algorithms are a class of algorithms for cryptography that use trivially related, often identical, cryptographic keys for both decryption and encryption. The encryption key is trivially related to the decryption key, in that they may be identical or there is a simple transform to go between the two keys. The keys, in practice, represent a shared secret between two or more parties that can be used to maintain a private information link [25][29].

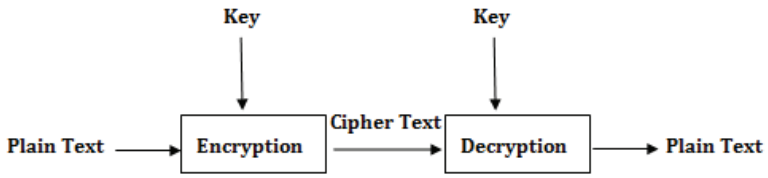

Figure 4. Symmetric Key utilizing same key to encrypt and decrypt the data

Figure 4 shows how symmetric key algorithm works. Symmetric-key algorithms can be divided into stream ciphers and block ciphers. Stream ciphers encrypt the bytes of the message one at a time, and block ciphers take a number of bytes and encrypt them as a single unit. Blocks of 64 bits have been commonly used; the Advanced Encryption Standard algorithm approved by NIST in December 2001 uses 128-bit blocks [25].

3.3 Proposed Communication Security for SCADA using Cross Crypto Schemic Cipher

Symmetric and asymmetric ciphers each have their own advantages and disadvantages. Symmetric ciphers are significantly faster than asymmetric ciphers, but require all parties to somehow share a secret (the key). The asymmetric algorithms allow public key infrastructures and key exchange systems, but at the cost of speed. So, in this study a combination of the best features of both symmetric and asymmetric encryption techniques is presented in the form of a crossed cipher for SCADA system.

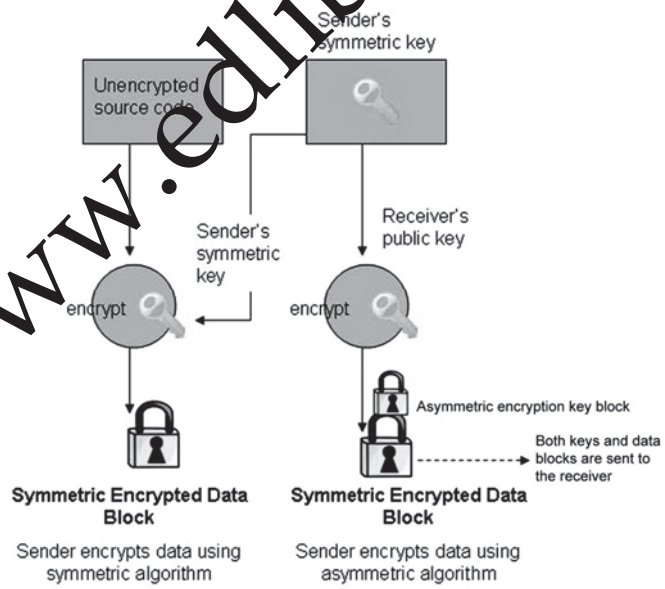

Figure 5. The crossed-cipher scheme

This crossed-cipher is capable of providing implicit authentication for the sender's identity. From the two major types of encryptions, asymmetric encryption provides more functionality than symmetric encryption, at the expense of speed and hardware cost. On the other hand symmetric encryption provides cost-effective and efficient methods of securing data without compromising security and should be considered as the correct and most appropriate security solution for many applications. In some instances, the best possible

solution may be the complementary use of both symmetric and asymmetric encryption. The algorithm presented here combines the best features of both the symmetric and asymmetric encryption techniques. The plain text data is to be transmitted in encrypted using the AES algorithm. Where in it will generate a random secret key for a symmetric cipher (AES)[26][31], and then encrypt this key via an asymmetric cipher (ECC) using the recipient's public key. The message itself is then encrypted using the symmetric cipher and the secret key. Both the encrypted secret key and the encrypted message are then sent to the recipient. This algorithm is shown in Figure 5.

4. Discussion

In this section, the application and implementation of the proposed solutions will be discussed. Each of them will be analyzed and the outcome is evaluated.

It contain the implementation of solutions like the Integration of Asymmetric-key Encryption to Internet SCADA; Symmetric Key Encryption in Wireless SCADA Environment; and the Implementation of the Cross Crypto Scheme Cipher to Secure communications for SCADA.

4.1 Integration of Asymmetric-key Encryption to Internet SCADA

Authentication will be required to access the data and reports so that only users who have enough permission can access the information. Quality system administration techniques can make all the difference in security prevention [21]. SCADA web server must always be secure since the data in it are very critical. Web server security software can also be added.

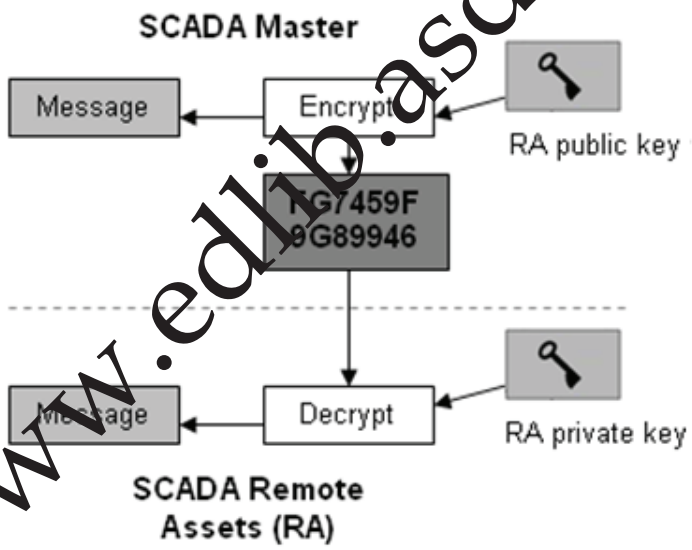

Figure 6. Asymmetric-key encryption applied to internet SCADA

Communication from the customer or client will start with an http request to the master server. The client will be authenticated before the request will be completed. The SCADA master will then send back the requested information to the client. The information will also be encrypted using the same encryption that is proposed to be used between the SCADA master and the remote assets [21]. This scenario is described in Figure 6 utilizing the asymmetric-key encryption algorithm. In this cryptography scheme, one of the keys is designated the public key and may be advertised as widely as the owner wants. The other key is designated the private key and is never revealed to another party. This method could be also used to prove who sent a message and can address non-repudiation vulnerability of a system.

To test the usability of this scheme, it was tested using the web base Asymmetric-key Encryption simulator. Since there are many kinds of Asymmetric-key Encryption, in this simulator, RSA Cipher is used. A demonstration of this algorithm is shown in Figure 7.

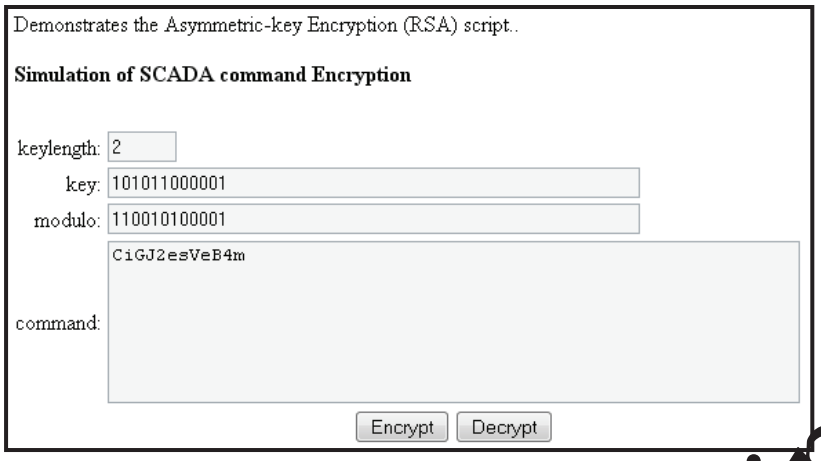

Figure 7. Browser based RSA Cipher Simulator

Table 2 shows the results of encrypted commands. The first column shows the command; the second column shows the key length; the third column shows the Modulo, the fourth column shows the key which is used for encrypting the command, the fifth column shows the encrypted data; the sixth column shows the key which is used to decrypt the data and the last column shows the actual command.

Table 2. Asymmetric-key Encryption of SCADA commands

Command	Keylength	Modulo	Key 1	Encrypted data	Key 2	Decrypted data
command 1	2 bytes	110010100001	10001	KAq[illegible]0dXb6h6	101011000001	turn on
command 2	2 bytes	110010100001	10001	9R[illegible]"7TEXWLsc	101011000001	turn off
command 3	2 bytes	110010100001	10001	qS0fd_L"ti	101011000001	connect
command 4	2 bytes	110010100001	10001	[illegible]Wx5P_4o6JuC5B4	101011000001	disconnect
command 5	2 bytes	110010100001	10001	LaO2p5HZXTHLS_7	101011000001	open valve
command 6	2 bytes	110010100001	10001	0XGvoFO4i7mIP3_M	101011000001	close valve
command 7	2 bytes	110010100001	10001	MNG1pMdWdR3nG6g	101011000001	half open
command 8	2 bytes	110010100001	10001	kRWkd7"nudFndww2	101011000001	half close

SCADA systems connected through the internet can provide access to real-time data display, alarming, trending, and reporting from remote equipment. But it also presents some vulnerabilities and security issues. In this section, the security issues in internet SCADA were pointed out. The utilization of asymmetric key encryption is suggested. It can provide security to the data that is transmitted from the SCADA master and the remote assets. Once a system is connected to the internet, it is not impossible for other internet users to have access to the system that is why encryption is very important [21][30].

4.2 Symmetric Key Encryption in Wireless SCADA Environment

Symmetric cryptography uses the same key for both encryption and decryption. Using symmetric cryptography, it is safe to send encrypted messages without fear of interception. This means only the SCADA master and the remote assets can communicate with each other because of the said key.

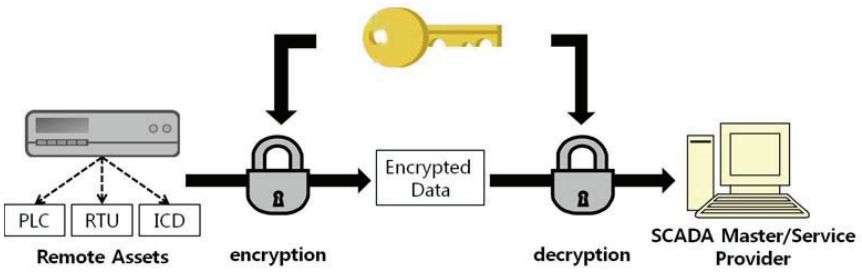

Figure 8. Symmetric cryptography between SCADA Master Station and Remote Components

Figure 8 describes the algorithm for the symmetric cryptography between the SCADA Master Station and the remote components. The SCADA Master Station and the remote components shares the same key for encrypting and decrypting messages. With this form of cryptography, it is obvious that the key must be known to both the Master Station and the remote components. This cryptography schemes are generally categorized as being either stream ciphers or block ciphers. Stream ciphers operate on a single bit (byte or computer word) at a time and implement some form of feedback mechanism so that the key is constantly changing. A block cipher scheme encrypts one block of data at a time using the same key on each block. In general, the same plaintext block will always encrypt to the same ciphertext when using the same key in a block cipher whereas the same plaintext will encrypt to different ciphertext in a stream cipher.

WEP was included as the privacy of the original IEEE 802.11 standard. WEP uses the stream cipher RC4 [27] for confidentiality, and the CRC-32 checksum for integrity. It can be implemented to wireless SCADA as it is implemented to other wireless systems. Messages between remote RTU's can be converted to ciphertext by utilizing this mechanism [24].

The use of symmetric key encryption specifically the RC4 cipher was also is applicable in a wireless Web-SCADA. It can provide security to the data that is transmitted from the SCADA master and the remote assets and also communication between remote RTU's. Once a system is connected to the internet specially wirelessly, it is not impossible for other internet users to have access to the system that is why encryption should be implemented. Data and report generation is also in demand so the internet SCADA is designed to have a web based report generation system through http. And to cut off the budget for communication lines, SCADA operators utilize the wireless based SCADA [24].

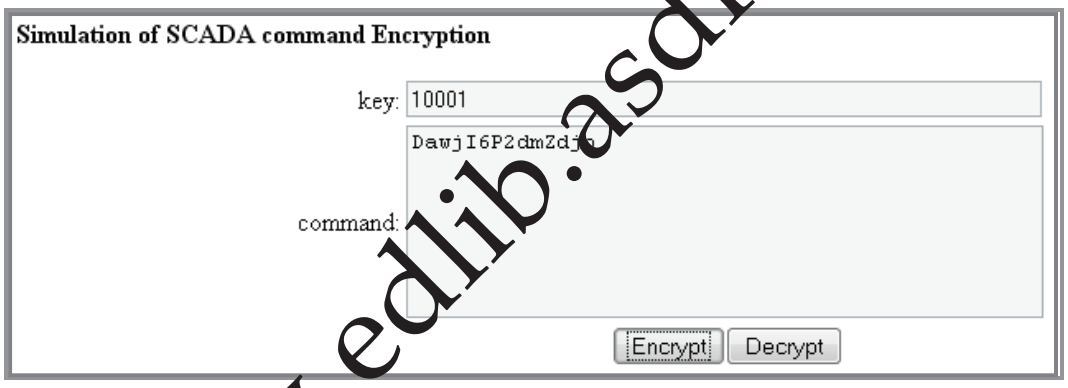

Figure 9. Browser based RC4 Simulator

The following table shows the results of encrypted commands. The first column shows the command; the second column shows the key which is used for encryption; the third column shows the encrypted data and the last column shows the actual command.

Table 3. Symmetric-key Encryption of SCADA commands

Command	Key 1	Encrypted data	Decrypted data
command 1	10001	JqMgRYo7ca	turn on
command 2	10001	JqMgRYo7kig	turn off
command 3	10001	04NbRMk4ya	connect
command 4	10001	ZG3gMoA7ce2dCb	disconnect
command 5	10001	4ewdRYE9nGMgnb	open valve
command 6	10001	003b2M6OAugaEXa	close valve
command 7	10001	"ahbJYo7CeMa	half open
command 8	10001	"ahbJYo4aS2hnb	half close

4.3 Implementation of the Cross Crypto Scheme

The implementation was done using out in J2SE (Java 2, Standard Edition) v 1.4.0. J2SE has the built-in classes for AES, and MD5 Hashing. The code uses these packages and the header files have the following header.

```
import java.security.*;

import javax.crypto.*;

import javax.crypto.spec.*;

import java.io.*;
```

Using Java a method has been developed for elliptic curve generation, base point generation, keys (both public and private) generation and encryption and decryption. The class of BigInteger in Java has been used to handle large integers and the method of IsProbablePrime to determine whether the large integer is prime or not. The software was running on Intel® Corel™i5 CPU @ 3.20 GHz and 2.99GB RAM [28].

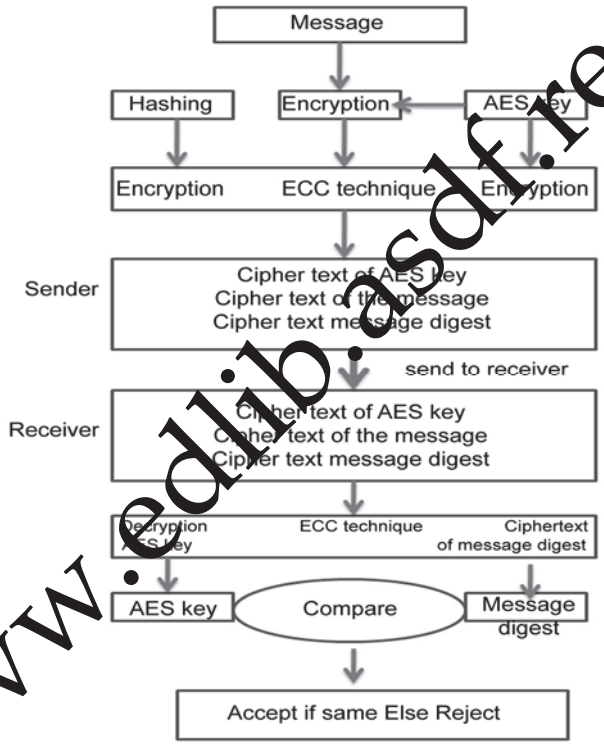

Figure 10. Chain of operation

Figure 10 depicts the chain of operation in the proposed cipher scheme. The AES key which is used to encrypt the data is encrypted using ECC. The cipher text of the message and the cipher text of the key are then sent to the receiver. To ensure integrity of the data that is transmitted, the data is subjected to MD5 hash algorithm [19]. The message digest obtained by this process is also encrypted using ECC technique. Thus the sender sends (1) Cipher text of the message, (2) Ciphertext of the AES key, and (3) Ciphertext of the message digest. The receiver upon receiving the Cipher text of the message, Ciphertext of the AES key, and Ciphertext of the message digest, first decrypts the Ciphertext of the AES key to yield the AES key. This is then used to decrypt the cipher text of the message to yield the plain text. The plaintext is again subjected to MD5 hash algorithm. This process yields a message digest. The ciphertext of the message digest is decrypted using ECC technique to obtain the message digest sent by the sender. This value is compared with the computed message digest. If both of them are equal, the message is accepted else rejected. Figure 11 shows the simple application developed based on the chain of operation.

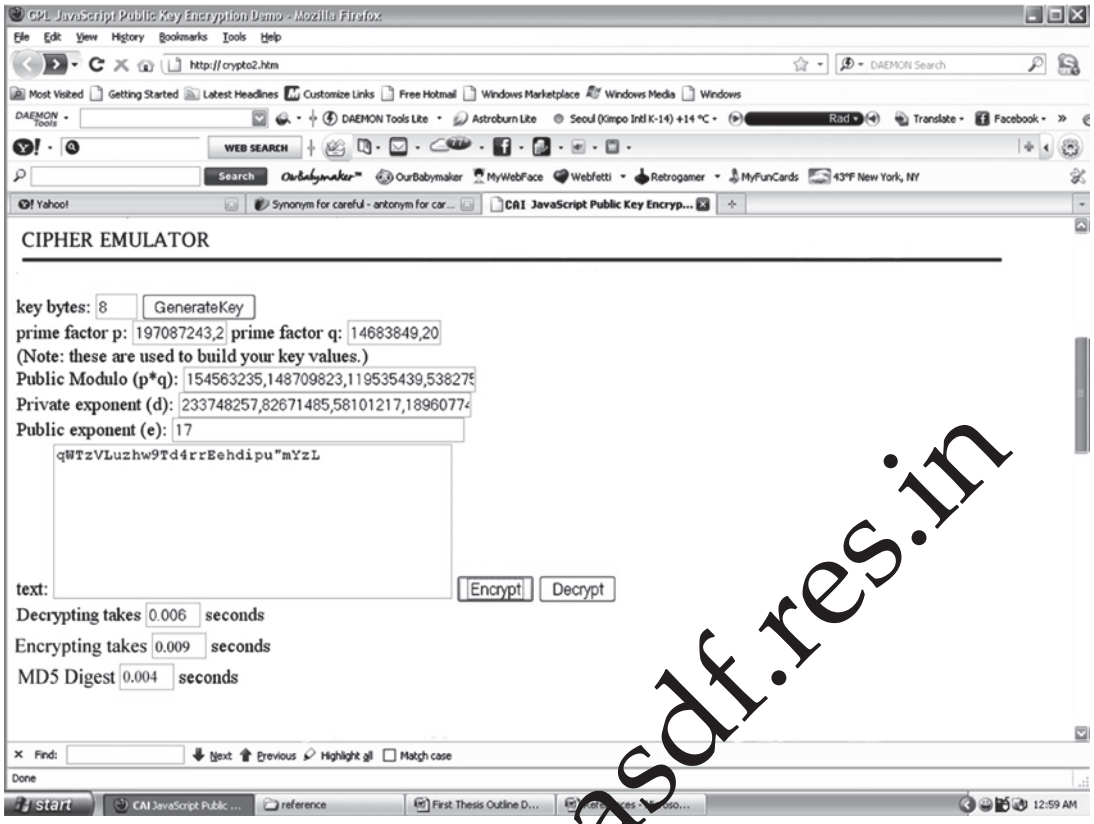

Figure 11. Cipher emulator

4.3.1 Steps in AES

The algorithm consists of four stages that make up a round, which is iterated 10 times for a 128-bit length key, 12 times for a 192-bit length key and 14 times for a 256- bit length key.

Stage 1: "Sub Bytes" transformation is a non-linear for each byte of the block.

Stage 2: "Shift Rows" transformation cyclically shifts (permutes) the bytes within the block.

Stage 3: "Mix Columns" transformation groups 4-bytes together forming 4-term polynomials and multiplies the polynomials with a fixed polynomial mod(x^4+1).

Stage 4: "Add Round Key" transformation adds the round key with the block of data.

4.3.2 Steps in Finding Base

Step 1: Take the Elliptic curve y2 mod p= x3 + ax +b mod p where p is a prime number.

Step 2 : For values from 0 to p-1, compute LHS and RHS

Step 3 : Locate points P where LHS = RHS

Step 4 : Count the number of points 'n'. The total number of points is always n + 1 (one point at infinity)

Step 5 : Find the prime factors of (n+1) and choose the largest among them.

Step 6: Find the negative point for every point computed in step 3.

Step 7 : Now perform addition operation of the each of the points obtained in step 3. Addition refers to finding 2P, 3P, 4P, 5P…. and tabulate them

Step 8 : Repeat step 7 until one gets the point at infinity.

Step 9 : Identify the largest prime factor from step 5. From the table created in step 7, locate for what points of P, the value is O (point at infinity)

Step 10: From the list of points, one can choose any point which will be the base point.

4.3.3 Steps in Key Generation

Step 1: Sender and receiver agree on the elliptic curve E and the base point G with order n. The order of n must be large. Hence E,G and n are known to everyone.

Step 2: Sender chooses a random number ds which is 1 < ds < n-1. He then computes ds * G. For him, ds is the private key and ds * G is the public key.

Step 3 : Receiver chooses a random number dr which is 1 < dr < n-1. He then computes dr*G. For him, dr is the private key and dr * G is

4.3.4 Steps in Encryption

Step 1 : Both sender and receiver agree on the elliptic curve E and the base point G with order n. Hence E,G and n are known to everyone.

Step 2 : Sender then encodes the message M as a point.

Step 3: Sender then generates a random number k. He then computes the value of kG (again a point).

Step 4: Sender takes the public key (drG) of the receiver, multiples the same with k (result is a point), and adds that with M. The result is again a point. Sender sends { kG, M + kdrG } to the receiver.

4.3.5 Steps in Decryption

Receiver gets { kG, M + kdrG } sent by sender

Step1: Extracts kG portion.

Step 2 : Multiples the same with his private key dr. He obtains kGdr.

Step 3: He then extracts M + kdrG portion. Subtracts the output of Step 2. i.e. M + kdrG - kGdr. which results in M, the plain text.

4.3.6 Steps in MD5

Input: b-bit message

Step 1: Appending padding bits: Padding is performed as follows: a single "1" bit is appended to the message, and then "0" bits are appended so that the length in bits of the padded message becomes congruent to 448, modulo 512. In all, at least one bit and at most 512 bits are appended.

Step 2: Append Length: A 64-bit representation of b (the length of the message before the padding bits were added) is appended to the result of the previous step. At this point, the resulting message has a length that is an exact multiple of 512-bits.

Step 3: Initialize the MD Buffer: Four word buffers to compute the Message Digest. Each one is a 32-bit register.

Step 4: Process Message in 16-word block

Step 5: Output: The message digest produced as output in each of the 4 MD Buffer. Begin with low order byte of the Buffer and end with the high order byte of the Buffer.

4.3.6 Testing the Scheme

Testing the cipher scheme on a test data of various sizes, Table 4 provides details on the time taken for encryption; decryption and calculation of MD5 message digest process. The following table depicts information on Encryption & Decryption of 128 bit AES key and MD5 message digest using ECC.

Table 4. AES encryption and decryption and calculation of MD5 message digest

Size (Kb)	128 bit AES Encryption	128 bit AES Decryption	MD5 message Digest
25	0.951 sec	0.948 sec	0.683 sec
50	1.126 sec	1.221 sec	0.752 sec
100	2.38 sec	2.217 sec	1.081 sec

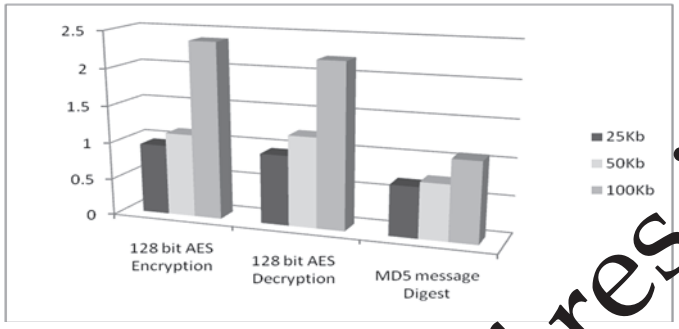

Figure 12. Graphical analysis of AES and MD5

Table 4 depicts the total time taken for performing encryption, decryption, MD5 message digest process calculation. Figure 12 shows the comparison of the time taken for performing the 128-bit encryption, 128-bit decryption and MD5 message digest using AES algorithm applied to different data sizes of 25, 50 and 100 Kb.

Table 5. Encryption & Decryption of AES key and MD5 message digest using ECC

	Encryption	Decryption
128 bit AES key	32	36
MD5 Hashing of Ciphertext	33	38

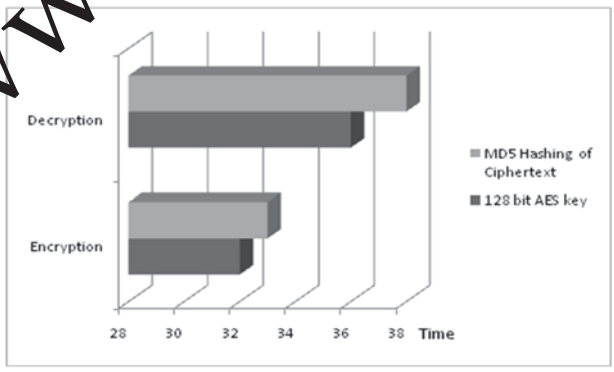

Figure 13. Graphical analysis of Encryption & Decryption of AES key and MD5 message digest using ECC

Table 5 depicts the encryption and decryption of AES key and MD5 message digest using the Elliptic Curve Cryptography or ECC. The graphical analysis from these table is shown in Figure 13.

Since the ECC key sizes are so much shorter than comparable RSA keys, the length of the public key and private key is much shorter in elliptic curve cryptosystems. This results into faster processing times, and

lower demands on memory and bandwidth. With any cryptographic system dealing with 128 bit key, the total number of combination is 2128. The time required to check all possible combinations at the rate of 50 billion keys/second is approximately 5 x 1021 years. Computational complexity for breaking the elliptic-curve cryptosystem for an elliptic curve key size of 150 bits is 3.8 x 1010 MIPS (Million Instructions Per Second years) [29]. While ECC may be relatively difficult to understand for the layman, it is nevertheless an important technology that has great potential to prosper in the future. The challenging and somewhat complicated nature of elliptic curve groups makes it harder to crack the ECC discrete logarithm problem. With less bits required to give the same security, ECC has fared favorably compared to RSA.

5. Conclusion

5.1 Summary

SCADA (Supervisory Control and Data Acquisition) communication can take place in a number of ways. SCADA Communication is a core component of a SCADA Monitoring System. Early SCADA communication took place over radio, modem, or dedicated serial lines. The process of communication over a SCADA system involves several different SCADA system components. These include the sensors and control relays, Remote Terminal Units (RTUs), SCADA master units, and the overall communication network. Encryption is also an important part of communication. Solutions such as Application of Asymmetric-key Encryption to SCADA Security, Symmetric-Key Encryption for Wireless Internet SCADA and Communication Security for SCADA using a Cross Crypto Scheme are discussed in this thesis.

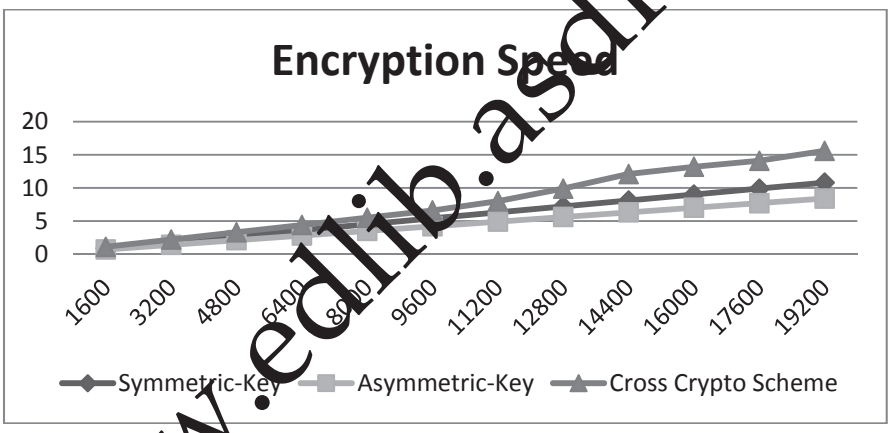

Figure 14. Encryption Speed Comparison

An important thing to be considered is the Encryption Speed. Compared to Asymmetric Key Encryption, Symmetric Key Encryption appears to be slower. However, because of many processes, the Cross crypto Scheme seemed to be the slowest as shown in Figure 14.

It's important to note right from the beginning that beyond some ridiculous point, it's not worth sacrificing speed for security. However, the measurements will still help us make certain decisions.

The security of SCADA systems is important because compromise or destruction of these systems would impact multiple areas of society far removed from the original compromise. The data that is gathered by the system is very important. The system reacts to the data it gets. Imagine what will happen if the data is not accurate. It can damage the society. To improve the security is SCADA communication, this work analysis and compare Encryption Schemes which are used in SCADA communication.

Devising a crossed-cipher scheme as presented in this study is one way to retrofit on to the system and be able to address the Confidentiality, Integrity, and Non-repudiation issues in SCADA.

The design and implementation of the crossed-cipher scheme was done in Java combining the best of both symmetric (AES) and asymmetric (ECC) cryptograophy and to ensure integrity of the data, the MD5 hash algorithm was adopted. A test for teh scheme for various sizes of files was done. By combining AES, the

algorithm which can use a variable block length and allowed any combination of key lengths of 128, 192 or 256 bits and blocks of length 128, 192 or 256 bits proven to be effected against attacks. The design and strength of all key lengths of the AES algorithm (i.e., 128, 192 and 256) are sufficient to protect classified information up to the SECRET level. TOP SECRET information will require use of either the 192 or 256 key lengths. This symmetric cryptography AES was used along with ECC asymmetric cryptography. ECC are mathematical objects that have been subject to many scrutiny by top mathematicians over the past 150 years. They have many important, elegant, and delightful properties and many research papers have been written solely exploring the various characteristics of these objects.

An important feature of these curves is that their points can be interpreted as part of a mathematical group and the challenging and somewhat complicated nature of elliptic curve groups makes it harder to crack the ECC discrete logarithm problem. With less bits required by ECC to give the same security compared to other existing asymmetric cryptography, ECC is indeed a reliable cryptographic scheme that will be important in the near future.

5.2 Future Work

The Internet based SCADA system designed provides a flexible and scalable solution to the problem of transferring data from an industrial control system using open Internet protocols. It has the ability to be modified quite easily to solve some of the more difficult issues that still require resolution. These are issues such as security and Quality of Service of the real-time data.

One important area that requires further work is the area of security. This is a major concern for industry and business in general. An Internet based SCADA system will require a relatively impenetrable security system before it is widely accepted. The implementation of a Quality of Service system to improve the flow of real-time traffic across a network that is shared with non-real-time traffic is also an area that should be researched further. Some systems were briefly examined for this project but their implementation was beyond the scope of the project. These systems will improve the functionality of Web-based SCADA immensely and help increase its performance.

Technology matures rapidly that SCADA systems are becoming increasingly ubiquitous. Some operators extend the HMI to mobile phones which is where SCADA systems leads to. Securing this mobile SCADA environment is an interesting riddle to answer in the future.

6. References

1) CNN Interactive, "Teen Hacker Faces Federal Charges", March 18, 1988, http://www.cnn.com/TECH/computing/9803/18/ juvenile.hacker/index.html [May 2011]
2) Quinn-Judge, P. Cracks in the system. TIME Magazine (9th Jan 2002).
3) Reed, T. At the Abyss: An Insider's History of the Cold War. Presidio Press, March 2004.
4) http://www.iscm.com/outsideplant.html
5) Washington Post. Dissertation could be security threat. http://www.washingtonpost.com/ac2/wp-dyn/ A23689-2003Jul7
6) Rosslin John Robles and Tai-hoon Kim, "Security Encryption Scheme for Communication of Web Based Control Systems", Communications in Computer and Information Science, Volume 123, Signal Processing and Multimedia, Pages 317-325, ISSN: 1865-0929
7) Mihaela Ciortea, (2004), "ASPECTS REGARDING THE TYPES OF PROCESS CONTROL SYSTEMS", Proceedings of the International Conference on Theory and Applications of Mathematics and Informatics - *ICTAMI 2004*, Thessaloniki, Greece
8) Tai-hoon Kim, (2010), "Weather Condition Double Checking in Internet SCADA Environment", *WSEAS TRANSACTIONS on SYSTEMS and CONTROL*, Issue 8, Volume 5, August 2010, ISSN: 1991-8763, pp. 623
9) D. Bailey and E. Wright, Practical SCADA for Industry, 2003
10) Andrew Hildick-Smith, Security for Critical Infrastructure *SCADA Systems,* 2005

11) Ramón Martínez-Rodríguez-Osorio, Miguel Calvo-Ramón, Miguel Á. Fernández-Otero, Luis Cuellar Navarrete, "Smart control system for LEDs traffic-lights based on PLC", Proceedings of the 6th WSEAS International Conference on Power Systems, Lisbon, Portugal, September 22-24, 2006, pp. 256-260
12) Costin Cepisca, Horia Andrei, Emil Petrescu, Cristian Pirvu, Camelia Petrescu, "Remote Data Acquisition System for Hydro Power Plants", Proceedings of the 6th WSEAS International Conference on Power Systems, Lisbon, Portugal, September 22-24, 2006, pp. 59-64
13) Rosslin John Robles, Min-kyu Choi, Tai-hoon Kim, "The Taxonomy of SCADA Communication Protocols", Proceedings of the 8th KIIT IT based Convergence Service workshop & Summer Conference, Mokpo Maritime University (Mokpo, Korea), pp. 23, ISSN 2005-7334
14) C. Clarke, D. Reynders, E. Wright, Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems, 2004
15) Station Automation COM600 3.4 IEC 60870-5-101 Master (OPC) User's Manual
16) DNP Users Group, "Overview of the DNP3 Protocol" http://www.dnp.org/About/Default.aspx [March 2011]
17) DPS Telecom, "DNP3 Protocol" http://www.dpstele.com/dpsnews/techinfo/dnp3_knowledge_base/dnp3_protocol.php [March 2011]
18) A DNP3 Protocol Primer. Revision A [March 2005]
19) Rosslin John Robles, Kum-Taek Seo, Tai-hoon Kim, "Communication Security solution for internet SCADA", Korean Institute of Information Technology 2010 IT Convergence Technology - Summer workshops and Conference Proceedings, 2010.5, pp. 461 - 462
20) D. Wallace, (2003), "Control Engineering. How to put SCADA on the Internet", Available: http://www.controleng.com/article/CA321065.html [January 2010]
21) McClanahan, R,H 2003, 'SCADA AND IP: Is Network Convergence Really Here?', IEEE Industry Applications Magazine, March/April.
22) Minkyu Choi, Rosslin John Robles, Taihoon Kim, "Application Possibility of Asymmetric-key Encryption to SCADA Security", The Journal of Korean Institute of Information Technology, Vol.7 No.4, August 2009, pp. 208-217, ISSN: 1958-8619
23) Dan Kaplan (2008) SC Magazine "Rare SCADA vulnerability discovered" Available: http://www.scmagazineus.com/Rare-SCADA-vulnerability-discovered/article/109956/ [October 2009]
24) Rosslin John Robles and Min-Kyu Choi, "Symmetric-Key Encryption for Wireless Internet SCADA", Communications in Computer and Information Science, Volume 58, Security Technology, Pages 289-297, ISSN: 1865-0929
25) RSA LAboratories "What is RC4?", Available: http://www.rsa.com/rsalabs/node.asp?id=2250 [June 2009]
26) [26] P. Prasithsangaree and P. Krishnamurthy, "Analysis of Energy Consumption of RC4 and AES Algorithms in Wireless LANs", 2003
27) "RC4", Available: http://www.wisdom.weizmann.ac.il/~itsik/RC4/rc4.html [June 2009]
28) Surge in criminal-driven cyber attacks anticipated in 2006, IBM Global Business Security Index Report, Dec 2005
29) Stallings W. Cryptography and Network Security. Prentice Hall, Upper Saddle River, New Jersey, USA, second edition, 1999
30) Tenable Network Security, "Protecting Critical Infrastructure: SCADA Network Security Monitoring", whitepaper, August 1, 2008.
31) "AES", http://en.wikipedia.org/wiki/Advanced_Encryption_Standard, 2011
32) "PLC", http://en.wikipedia.org/wiki/Programmable_logic_controller, 2011
33) "DCS", http://en.wikipedia.org/wiki/Distributed_control_system, 2011
34) Vale, Z., Morais, H., Silva, M., Ramos, C., "Towards a future SCADA", Power & Energy Society General Meeting, 2009, PES'09, pp.1-7, 26-30 July 2009.
35) Rosslin John Robles and Min-kyu Choi (2009), "Assessment of the Vulnerabilities of SCADA, Control Systems and Critical Infrastructure Systems", International Journal of Grid and Distributed Computing, Vol. 2, No. 2, June 2009, pp. 27-34