12-1-2013

Application of NTRU Cryptographic Algorithm for securing SCADA communication

Amritha Puliadi Premnath
University of Nevada, Las Vegas, ppamri@gmail.com

APPLICATION OF NTRU CRYPTOGRAPHIC ALGORITHM FOR SECURING SCADA COMMUNICATION

By

Amritha Puliadi Premnath

Bachelor of Engineering, Computer Science
Thiagarajar College of Engineering, India
2010

A thesis submitted in partial fulfillment
of the requirements for the

Master of Science - Computer Science

School of Computer Science
Howard R. Hughes College of Engineering
The Graduate College

University of Nevada, Las Vegas
December 2013

UNLV
UNIVERSITY OF NEVADA LAS VEGAS

THE GRADUATE COLLEGE

We recommend the thesis prepared under our supervision by

Amritha Puliadi Premnath

entitled

Application of NTRU Cryptographic Algorithm for Securing SCADA Communication

is approved in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science

Department of Computer Science

Ju-Yeon Jo, Ph.D., Committee Chair

Yoohwan Kim, Ph.D., Committee Member

Laxmi Gewali, Ph.D., Committee Member

Venkatesan Muthukumar, Ph.D., Graduate College Representative

Kathryn Hausbeck Korgan, Ph.D., Interim Dean of the Graduate College

December 2013

ABSTRACT

Application of NTRU Cryptographic Algorithm for securing SCADA Communication

by

Amritha Puliadi Premnath

Dr. Ju-Yeon Jo, Examination Committee Chair

Associate Professor of Computer Science

University of Nevada, Las Vegas

Supervisory Control and Data Acquisition (SCADA) system is a control system which is widely used in Critical Infrastructure System to monitor and control industrial processes autonomously. Most of the SCADA communication protocols are vulnerable to various types of cyber-related attacks. The currently used security standards for SCADA communication specify the use of asymmetric cryptographic algorithms like RSA or ECC for securing SCADA communications. There are certain performance issues with cryptographic solutions of these specifications when applied to SCADA system with real-time constraints and hardware limitations. To overcome this issue, in this thesis we propose the use of a faster and light-weighted NTRU cryptographic algorithm for authentication and data integrity in securing SCADA communication. Experimental research conducted on ARMv6 based Raspberry Pi and Intel Core machine shows that cryptographic operations of NTRU is two to thirty five times faster than the corresponding RSA or ECC. Usage of NTRU algorithm reduces computation and memory overhead significantly making it suitable for SCADA systems with real-time constraints and hardware limitations.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my committee chair, Dr. Ju-Yeon Jo for her strong support and guidance throughout my thesis. It gives me great pleasure in acknowledging her valuable assistance.

I am extremely grateful to Dr.Yoohwan Kim for his indispensable advice, information, and valuable guidance on different aspects of my research work.

I would like to thank my husband, Karthikkumar for his continuous support and motivation. His valuable thoughts and immense cooperation was imperative to my completion of this degree. I owe thanks to my family and friends for their immense love and encouragement.

I would also like to thank Dr.Laxmi Gewali and Dr. Venkatesan Muthukumar for being a part of my committee. I convey my special thanks to the graduate coordinator, Dr. Ajoy K. Datta for all his support and advocacy. I extend my gratitude to the Student Affairs Maintenance department for funding my Master's degree.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Critical Infrastructure represents the basic facilities, services and installations necessary for functioning of a community, such as water, power lines, transportation, communication systems, and so on. Any act or practice that causes a real-time Critical Infrastructure System to impair its normal function and performance will have debilitating impact on security and economy, with direct implication on the society. Critical infrastructure system operation involves the exchange of real-time data from various distributed control systems along the local and wide area communication networks to support a variety of vital mechanisms. To enable such mechanisms messages have to be delivered in a secure and timely manner using a cost-efficient and compatible communication protocol.

SCADA (Supervisory Control and Data Acquisition) system [19] is a control system which is predominantly used in Critical Infrastructure System to monitor and control industrial processes autonomously. SCADA can be seen as a combination of hardware, software, controllers, networks and computers that assist in the remote monitoring and co-ordination of control systems of an adverse infrastructure like smart grids, chemical plants, transportation systems etc. These systems have been in use from 1960's. Ever since then SCADA has been gradually evolving along with new upcoming technologies making it more flexible, yet more vulnerable.

SCADA systems previously designed were connected to limited private network. There was no need of protecting such closed architecture against any cyber-attack. They were designed to be tolerant towards few human errors that were very low in

severity. The SCADA system used today is easily affected by cyber-attacks due to the arrival of IP technologies and standards into the design of such systems. This integration supports new IT capabilities, but it provides significantly less isolation for SCADA systems from the outside world than predecessor systems, creating a greater need to secure these systems. SCADA provides automation solutions using several standards such as the IEC-61850 [11], DNP3 [12], IEC 60870-5 [13] and Modbus [14]. Most of these protocols run over unsecure TCP/IP networks using high speed switched Ethernet to obtain necessary response times. It is therefore imperative that system security and risk mitigation be at the forefront of the minds of all SCADA system users.

1.1 Need for Current Work

Encryption and authentication are highly effective methods to reduce some of these cyber threats to SCADA communications. Recently, there have been several efforts to secure the SCADA systems. Security communities have been trying to make security policies, operational, quality and system recommendations to provide security systems for SCADA infrastructure. There are two open standards for SCADA communications available on the market today that were developed to provide security through encryption and authentication: IEC 62351suite [6 and 11] and IEEE6189 suite (also known as AGA-12 incorporated in IEEE 1711), these standards secure SCADA equipment communication [3]. However there have been few noted performance issues in meeting the timing requirements of utilities such as smart grid and water companies, while implementing IEC 62351 and AGA-12 [1 and 15].

AGA-12 and IEC 62351 standards approve the use of asymmetric algorithms such as RSA [16] and ECC [17] for digital signing which is used for authentication purposes. Unfortunately in practical, some SCADA applications involving delay constraints limit their security to just authentication. They don't adopt any encryption technique to secure the integrity of the message as the digital signing process using RSA is time consuming and process intensive [15]. Also there are number of insecure connections in the SCADA network unprotected with the absence of authentication and encryption [4] (due to the expensive asymmetric cryptographic operations), e.g. ports used for maintenance of SCADA system, examination of the SCADA system, obtaining remote access to the system etc. Such devices or applications and the communication channel it uses is highly susceptible to attacks and hence results in compromise of the integrity of data transmitted. Although ECC based authentication mechanisms can provide better performance results when compared to RSA, in practical it is necessary to consider algorithms faster than ECC for real-time applications. The objective of this thesis is to provide a better solution for SCADA device/channel authentication and data integrity by introducing the use of faster and light-weighted NTRU cryptographic, rather than the currently used slow RSA or comparatively slow ECC.

Our experiments were performed on ARMv6 based Raspberry Pi and Intel machine running Windows 7 for evaluating the performance of different asymmetric cryptosystems. Our results show that usage of a light-weight asymmetric key protocol like NTRU is necessary for supporting a secure and faster real-time critical application.

1.2 Outline

Chapter 2 gives a general background of the SCADA architecture, advantages of using internet-based SCADA systems and an overview of SCADA Communication protocols. Chapter 3 discusses the need for SCADA security and the common attacks it encounters. Chapter 4 gives a brief introduction on the role of cryptography in SCADA security and practical difficulties involved in implementing them. Chapter 5 discusses about the NTRU cryptographic algorithm and the proposed approach of applying it to improve the security standards of SCADA system. Chapter 6 subsequently describes about our experimentation on Raspberry Pi and Intel Core Machines to evaluate and compare the performance of NTRU with RSA & ECC cryptographic operations and discuss our findings. Chapter 7 gives the conclusion and the improvements that can be made in the future.

CHAPTER 2

BACKGROUND

This chapter gives an overview of SCADA architecture, advantage of using Internet based SCADA system and overview of the SCADA communication protocols

2.1 SCADA Architecture

SCADA is an acronym for Supervisory Control and Data Acquisition, which is a computer-based control system that is used for collecting and analyzing real-time data. SCADA systems are designed to collect field information, transfer it to a central computer facility, and display the information to the operator graphically or textually, thereby allowing the operator to monitor or control an entire system from a central location in real time. Based on the sophistication and setup of the individual system, control of any individual system, operation, or task can be automatic, or it can be performed by operator commands.

The basic SCADA architecture consists of following fundamental components:

Remote Terminal Unit (RTU)

RTU's are microprocessor based devices deployed in the field at specific sites and locations to support SCADA remote stations. They serve as local collection points gathering information for the control center, from field control devices remotely and issue commands to the field control system. Control devices are components like sensors, actuators, electric motors, console lights, switches, and valves etc. that are deployed in the field to perform and control local operation. Local operation can involve data collection from sensor systems, opening and closing of valves, turning on and off of switches and so on.

Programmable Logic Controllers (PLC)

The PLC is a small industrial computer originally designed to perform the logic functions executed by electrical hardware (relays, drum switches, and mechanical timer/counters). PLCs have evolved into controllers with the capability of controlling complex processes, and they are used substantially in SCADA systems. They provide the same control as RTU except that RTUs are designed for specific control applications.

Intelligent Electronic Devices (IED)

An IED is a "smart" sensor/actuator containing the intelligence required to acquire data, communicate to other devices, and perform local processing and control. An IED could combine an analog input sensor, analog output, low-level control capabilities, a communication system, and program memory in one device. The use of IEDs in SCADA systems allows for automatic control at the local level.

Master Terminal Unit (MTU)

The Master units (MTUs) serve as the central processor of the SCADA system. They monitor and control large number of RTUs/PLCs. They acquire information from RTU's/PLC's, carryout necessary analysis and control; provide the reading and equipment details to the human operators through HMI. In an internet-based SCADA network, the MTU's can be on a different network or location.

Human Machine Interface (HMI)

HMI is a computer system that runs powerful graphics for displaying status and historical information to operators. Human operators use HMIs to configure set points, control algorithms, and adjust and establish parameters in the field control

devices remotely by observing the readings and sending instructions to specific RTU's.

Communication Network

One of the most important elements of the SCADA system is the communication network which acts as a bridge between the control systems. SCADA communication is conducted over leased lines/switched telephone, wide area network/internet, radio/microwave and satellite.

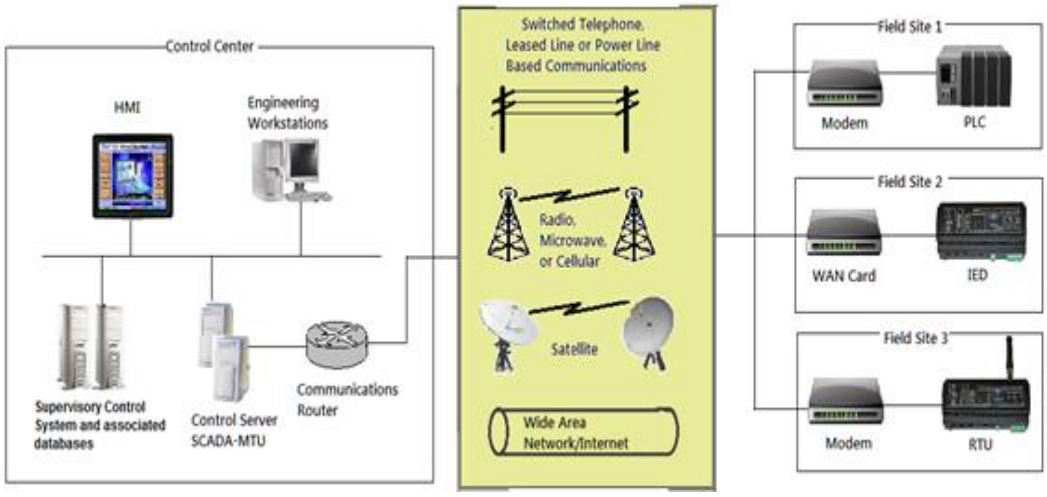

Figure 2.1 Basic SCADA Architecture

Figure 2.1 shows a basic SCADA architecture which is interconnected with the internet. The traditional proprietary and closed SCADA architecture cannot meet the ever-changing requirements of Critical Infrastructure Industry. Internet based architecture is highly essential to provide an ideal and flexible platform for this constantly changing business.

2.1.1 Advantages of using Internet-based SCADA Architecture

We come across many companies that have leveraged internet for their SCADA systems, either by building new applications from scratch or by enabling internet accessibility to the existing SCADA systems. The major reasons that motivates companies to adopt IP technologies into their SCADA design include,

- It reduces the infrastructure cost, as they have the benefit of using public Internet instead of using the expensive dedicated lines.
- It allows them to access information in an easier way from remote sites and assists in improving system efficiency and performance.
- It provides immediate access to real-time data.
- It reduces the cost involved in repairing and other labor costs required for troubleshooting or service when a dedicated line fails.
- It facilitates compliance with regulatory agencies through automated report generating from remote equipment.
- It is more flexible in terms of choosing equipment and systems based on price/performance rather than compatibility with installed base.
- It supports scalability quickly from few sites to thousands.

2.2 SCADA Communication Standards and Trends

The information/control signals exchanged between SCADA devices and other control systems through a network, or other media is governed by rules and conventions that can be set out in technical specifications called Communication protocols standards.

2.2.1 SCADA Communication Protocols

Protocol designs in SCADA are compact and are so designed as to send information to MTU only in case the RTU is polled for information by the MTU. Typical legacy SCADA protocols include Modbus RTU, ASCII, RP-570, Profibus and Conitel. These communication protocols are all SCADA-vendor specific. Standard protocols are IEC 60870-5-101 or 104, IEC 61850 and DNP3 [13]. These communication protocols are standardized and recognized by all major SCADA vendors. Communication protocols with extensions can operate in internet protocol TCP/IP. For e.g. Modbus TCP/IP has now become standard for lot of hardware manufacturers and is widely accepted communication protocol. Although it is advisable not to connect it to internet and expose it to risk, Ethernet TCP/IP has found its way into industrial automation breaking the barriers in majority of SCADA market.

The following protocols are emerging as virtual standards in modern SCADA systems.

2.2.1.1 Distributed Network Protocol (DNP3)

DNP3 is a protocol that defines communications between master stations, remote terminal units (RTUs), and IEDs in SCADA. IEEE has opted DNP3 as a standard for Electric power system communications [12]. It is also widely used in water infrastructure, oil, gas, security and other industries. Initially, DNP3 was designed without any security features. DNP3 is extended to DNP3 Secure Authentication (SA) [26], which was designed to meet requirements of IEC 62351-5. DNP3-SA employs techniques including symmetric cryptography and hashed message

authentication codes (HMACs).Implementation presumes that both master station and outstation share a common secret key, called an update key, which is used to generate a session key. The recently released DNP3-SA5 reinforces overall security for data information gathering, exchange, and use in SCADA systems.

Network Architecture: DNP3 was initially designed with four layers: physical, data link, transport and application layer. Originally physical layer involved serial communication protocols such as RS-232, RS-422 or RS-485. Today's DNP3 has been ported over TCP/IP layer to support recent communication technologies, and thus can be considered as three-layer network protocol operating upon the TCP/IP layer[27] to support end-to-end communications. Figure 2.2 shows DNP3 protocol wrapped inside TCP/IP data packet.

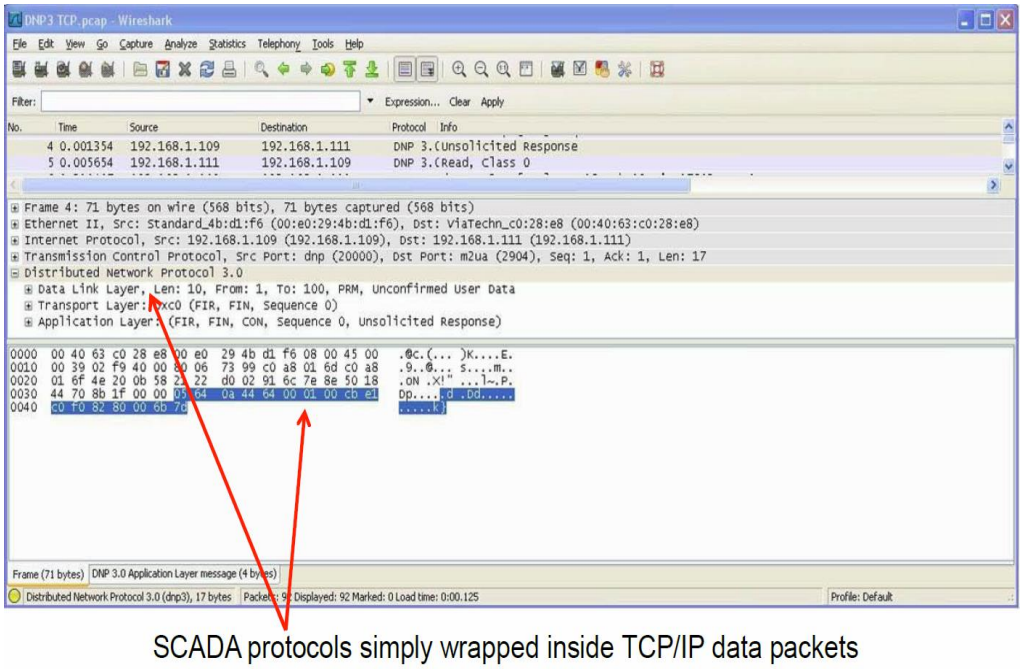

Figure 2.2 DNP3 with TCP

2.2.1.2 IEC 61850

IEC 61850 was published as a standard by IEC (International Electrochemical Commission) for Substation Automation system. It was created to be an internationally standardized method of communication and integration to support systems built from IEDs and RTUs independent of the device manufacturer. It also defines certain performance classes for different communication methods. Table 2.1 shows a list of delay requirements for IEC 61850 messages, which reveals that power substation communication contains a number of time-critical messages with application layer delay constraints varying from 3ms to 500ms.

Message Type	**Delay Constraints(ms)**
Type 1A/P1	3
Type 1A/P2	10
Type 1B/P1	100
Type 1B/P2	20
Type 2	100
Type 3	500

Table 2.1 Timing Requirements for messages in power substations

Network Architecture: Differing from DNP3 that is based on TCP/IP, IEC 61850 specifies a series of protocol stacks for variety of services including TCP/IP, UDP/IP, and an application-to-MAC stack for time-critical messages.

Any security standard that attempts to secure IEC 61850 based traffic must take into consideration of these performance requirements.

2.2.1.3 IEC 60870-5

IEC 60870-5 provides a communication profile for sending basic tele-control messages between two systems, which uses permanent directly connected data circuits between the systems. It is one of the widely accepted standards in Electric power systems that enable interoperability among compatible tele-control equipment.

2.3 Overview of IEC 62351

For some years now, Critical infrastructure systems using SCADA architecture have been attempting to secure the different protocols it uses. This push for security is mainly due to the movement from “point to point” communication between devices to large TCP/IP networks. This resulted in the emergence of IEC 62351 series. Its primary objective was to undertake the development of standards for security of the communication protocols defined by IEC TC 57, specifically IEC 61850, IEC 60870-5 series and its derivatives (i.e., DNP3).However, the current scope of IEC 62351 is aimed at defining numerous mechanisms to protect exchange of information in automation applications [6]. The major goal of this standardization is to provide end-to-end security in power automation systems. Table 2.2 IEC 62351 specification covering different OSI layers. Some IEC 62351-3 standards are as follows:

- **IEC 62351-3** identifies how to ensure secure TCP/IP-based protocols using transport layer security (TLS).

• **IEC 62351-5** defines security for IEC 60870-5 and its derivatives, providing different solutions for serial and networked versions. It uses TLS for TCP/IP profiles and encryption for serial profiles. It specifies how to incorporate user and device authentication, and data integrity. Existing protocols like DNP3 has been extended to meet the authentication requirements of IEC 62351-5.

• **IEC 62351-6** provides security for IEC 61850 profiles.

Part	**Scope**	**OSI Layers**				
		1	**2**	**3**	**4**	**5-7**
3	Profiles Including TCP				X	
4	Profiles Including MMS					X
5	Security for IEC 60870-5 and Derivatives		X	X	X	X
6	Security for IEC 61850 Profiles		X	X		X
9	Cyber security key management for power system equipment		X	X	X	X
11	Security for XML Files(Pending)					X

Table 2.2 IEC 62351 covering different layers of OSI model

Besides power systems, other SCADA systems, and other critical infrastructure systems can deploy the specified security measures in IEC 62351 because they have several common requirements.

CHAPTER 3

SECURITY CONSIDERATIONS FOR SCADA COMMUNICATION

This chapter focuses on the need for securing SCADA communication and common threats associated with SCADA network.

3.1 Need for securing SCADA Communication

Traditional SCADA systems were designed to be closed networks; they were separated from other enterprise or public networks. They also used proprietary hardware, software and network protocols which increased the difficulty of understanding SCADA systems. So security was not considered as a big issue.

However due to advent of internet-based communication, today more organizations connect SCADA networks with other potentially unsecure networks to leverage the benefits listed before. Although it looks beneficial, several attacks have been reported in this modern internet-based SCADA system. According to the latest ICS-CERT (Industrial Control Systems Cyber Emergency Response team) report in the first-half of fiscal year 2013, over 200 attempted intrusions were detected. From Figure 3.1 highest percentage of incidents were reported in the energy sector at 53%. Due to the internet, technical information needed to attack these systems is widely discussed making the SCADA system even more vulnerable. Critical security flaws have become well known to potential hackers. It is feared that SCADA systems can be victimized by hackers, criminals or terrorists.

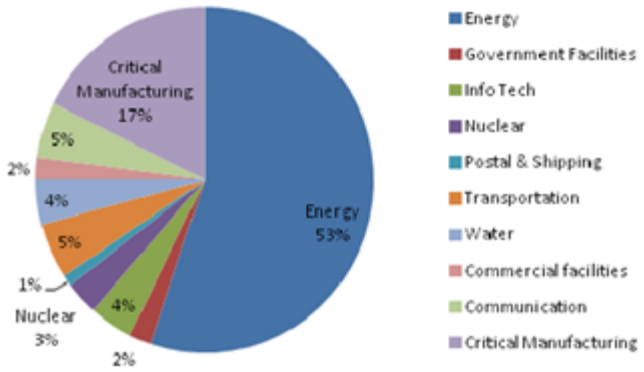

Figure 3.1 Percentage of incidents reported across all Critical Infrastructure sectors (Oct, 2012 – May, 2013)

3.1.1 Risk Factors associated with SCADA Architecture

During the analysis for providing a secure SCADA communication, few factors were reported to have contributed to the escalation of risk to SCADA system. Those include:

- Usage of standardized technologies whose vulnerabilities are well known to attackers. For e.g. nine out of ten SCADA systems use Windows, others use Unix-like operating system. Attackers are knowledgeable in these technologies, so it becomes easier for them to wage attacks on these systems.
- Insecure connections: The communication link that most SCADA enterprise uses (e.g. leased line, internet, wide area network etc.) to transmit data between control systems and remote locations, could be easily compromised.
- Readily available technical information about control systems: Internet is flooded with information on infrastructures and control systems. Hackers and attackers use this information to understand about the system and find ways to attack them.

- Sometimes control systems which are installed incorrectly might also act as a threat as they can bridge networks together unintentionally.
- In par with an external threat, there is also the risk of internal threat where an attack is caused by an employee who has greater access to the SCADA control systems.

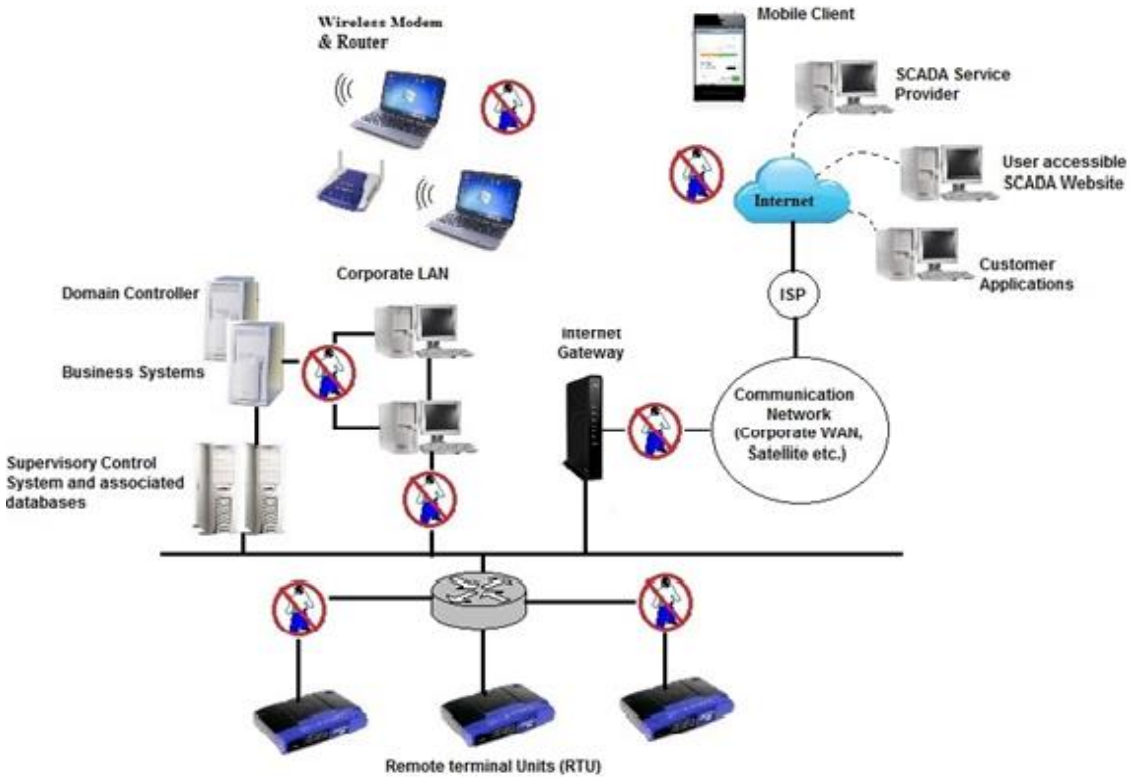

Figure 3.2 Points of vulnerability in a SCADA network- Example

As a consequence of all these issues and as attackers are becoming more sophisticated in performing cyber-related criminal and terrorist activities, there is an urgent need for providing high-quality cost-efficient security for SCADA

communication. Figure 3.2 shows a security compromised SCADA network and the potential points of vulnerability.

3.2 Attacks in SCADA Communication - Classification

Current SCADA devices are effective in detecting and preventing well-known Internet attacks, but until recently they have not addressed SCADA communication protocol attacks completely. SCADA vendors are beginning to develop and incorporate attack signatures [18] for various SCADA protocols such as Modbus, IEC-61850, and DNP3. Attacks that possibly affect the SCADA communication are listed below:

Data Integrity Attacks

In SCADA system, attacks that result in modification or destruction of control and sensing signals/messages is referred to as Data Integrity attacks and any prolonged loss of data results in Denial of Service (DoS) attack. These attacks could cause the system to behave in an unstable manner by hijacking its normal operation.

- Example – An attacker can make unauthorized changes to programmed instructions/status values in RTUs, resulting in damage to equipment, premature shutdown of processes, or even disabling control equipment.

Authentication Attacks

Authentication is the process of verifying the identity of an entity. SCADA system is vulnerable to an unauthorised party who can send fake messages which may damage the industrial control process controlled by SCADA. Hence devices have to provide their identity details for communication. Whenever a SCADA device receives a command to perform some control, it challenges the sending device for its identity.

Only when the receiving device is satisfied with the identity response, it acts upon the original command.

- Example – An attacker can send false information to control system operators to disguise unauthorized changes or to initiate inappropriate actions by system operators.

Confidentiality Attacks

Confidentiality attacks in SCADA system are caused by gaining access to sensitive data, either by eavesdropping on the network (non-secure communication line) or accessing the repository. Disclosure of sensitive data results in loss or damage to the entire SCADA system. To protect sensitive data from unauthorized users, data is encrypted before it is transmitted through an unprotected communication channel like a public network. Encrypted data becomes meaningless or unintelligible to an eavesdropper. Only the intended recipient can decrypt the message with the secret key.

- Example 1 – An attacker can over hear a communication between control systems and can en-route a data exchange by assuming exchanger's identity.
- Example 2 – They can eavesdrop and acquire desired information, such as customer's private details.

Non-repudiation Attacks

Non-repudiation is a service that provides proof of integrity and origin of data thereby assuring an authentication to be genuine. Origin of data is more important in SCADA communication involving multicast communication.

- Example – An attacker can guess the private key corresponding to the signing certificate and change the message origin.

The key requirement of a secure SCADA system is to provide solutions to defend these attacks.

3.3 Attack simulated in Power System

Many vendors use TCP/IP to transport SCADA messages. Link layer frames are embedded into TCP/IP packets for transmission. This approach has enabled SCADA architecture to take advantage of Internet technology and allow collecting data economically and controlling geographically separated devices. This has made the system more vulnerable to cyber-attacks.

An attack was simulated with a DNP3 simulator to show how an attacker can modify the data (in this case, increase the current load above 400 amps) without control center/MTU being notified of the changes. Triangle MicroWorks Protocol Test Harness Tool was used to perform a data-integrity attack in an electric power system.

3.3.1 Terminology in Power system

Polling: Polling refers to actively sampling the status of an external device by a client program as a synchronous activity.

Relay: A relay is an electrically operated switch. Many relays use an electromagnet to operate a switching mechanism mechanically, but other operating principles are also used. Relays with calibrated operating characteristics are used to protect electrical circuits from overload or faults: in modern electric power systems these functions are performed by digital instruments called “protective relays”.

3.3.2 Simulation

- Figure 3.3 shows RTU simulator that displays the normal current flow with a closed circuit state. The load current is 200 Amps.

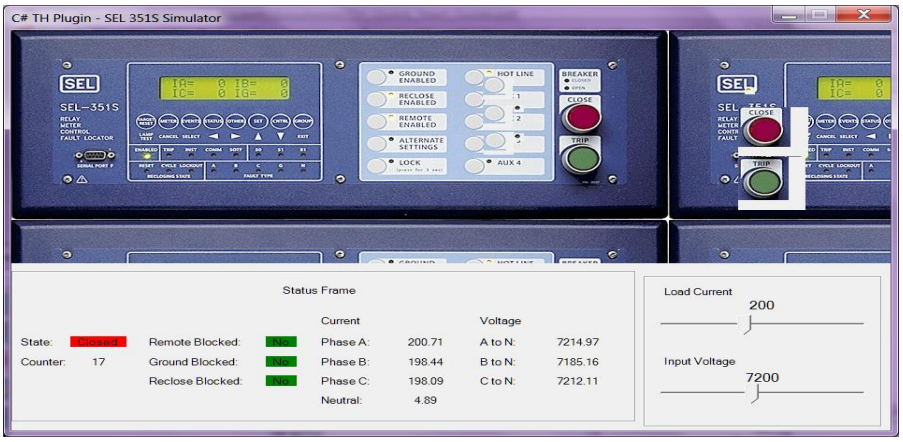

Figure 3.3 RTU displaying normal current flow

- MTU keeps monitoring the status of its RTUs and controls it. Figure 3.4 shows the status view as seen from a MTU. It indicates a normal current flow.

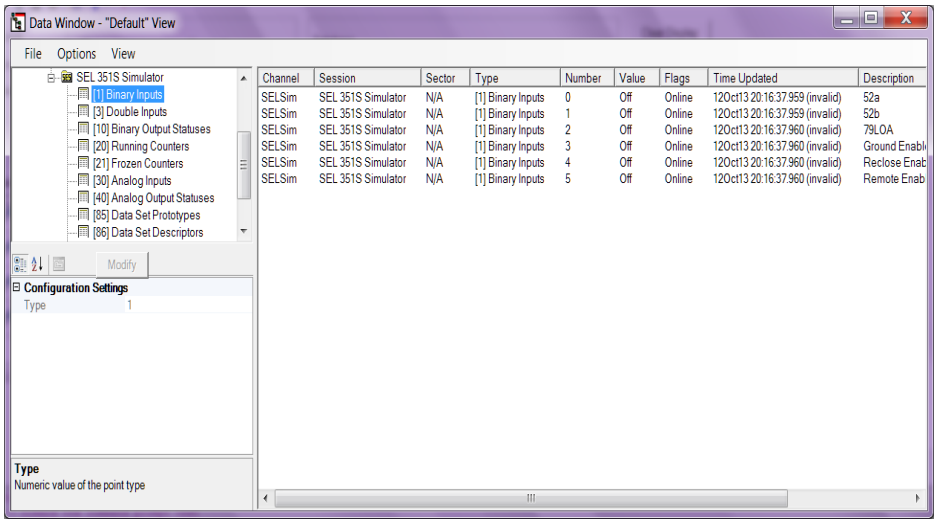

Figure 3.4 Status view from the MTU

- To perform an attack, DNP3 polling was shut down (as seen in Figure 3.5) and the current load was increased to 400 Amps. As the polling was disabled, the master station will not receive any status message from the RTU.

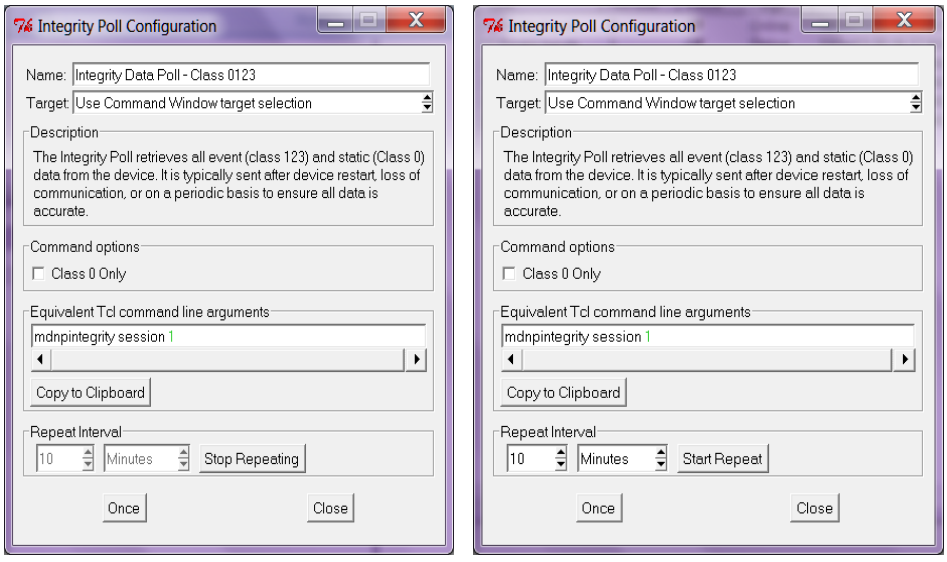

Figure 3.5 Polling is Shutdown

- This caused the relay to trip and stopped the current flow which can be seen in Figure 3.6.

Figure 3.6 RTU showing zero current flow

- However, there was no status change in the master view because polling was not enabled. The status was the same as seen in Figure 3.4.
- If the polling had been enabled, the master view would have looked like the one shown in Figure 3.7.

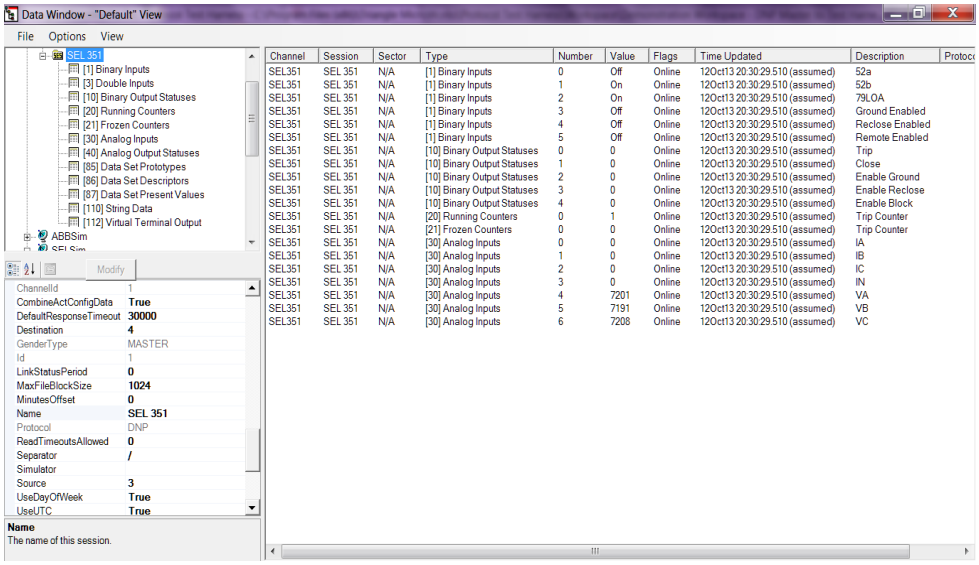

Figure 3.7 Master view if polling was not shutdown

This simulation shows what could have happened if an attacker/hacker had turned off polling. Though it looks like a simple attack, its impact is huge. This has increased the need for securing a SCADA system. Securing SCADA communication protocols has been one of the goals of technical specification IEC 62351.

CHAPTER 4

CRYPTOGRAPHIC SOLUTIONS IN SCADA SECURITY

This chapter provides an overview of cryptographic solutions in SCADA security and discusses on the practical difficulties while implementing those.

4.1 Role of Cryptography in SCADA security

For securing the overall SCADA communication completely the existing SCADA protocols must ensure to provide end-to-end authentication, data integrity, non-repudiation and confidentiality [8]. Cryptography is a hidden component of all these security measures or cyber security policies. NIST [20], IEEE, AGA (American Gas Association) and many other organisations have been sincerely engaged in developing cryptographic standards to secure SCADA communication [2, 3, and 4].

Cryptographic primitive approaches are needed in SCADA system to deal with attacks targeting integrity and confidentiality that cause negligible effect on the network performance.

4.1.1 Terminology

Encryption and Decryption

Encryption and decryptions are cryptographic methods used to achieve secure communication and information. Encryption is the conversion of data into a form, called a cipher text, which cannot be easily understood by unauthorized people. Decryption is the process of converting encrypted data back into its original form, so it can be understood by the intended recipient. The design and choice of encryption scheme is the essential mechanism to protect data confidentiality and integrity in any SCADA system.

Cryptosystem

A cryptosystem can be considered as a suite of three algorithms: for key generation, encryption and decryption. The two primary cryptosystems used are Symmetric and Asymmetric depending upon the nature of keys used to encrypt a message.

Comparison Feature	**Symmetric Cryptosystem**	**Asymmetric Cryptosystem**
Key Management	Symmetric encryption uses only one key that all parties to the message exchange must know, so the key is the same on either side of the transmission. Complicates key management as it requires secure exchange and update of secret keys among the communication SCADA systems/devices.	Asymmetric encryption uses a public and a private key. The owner holds the private key and never shares it with anyone. The public key is available to anyone to decrypt a message from the owner, and only the owner's private key can encrypt the message.
Speed	Faster due to simplicity of algorithm.	Relatively slower than symmetric algorithms.
Key Length Requirement	Uses shorter key length generally.	Requires longer key lengths to achieve a given level of security.
Security Risk	Risk is the disclosure of shared key to an unauthorized entity.	Less risky since private key is not shared to anyone.
Resource Utilization	Requires approximately constant computational resources regardless of key size.	Requires more computational resources for long key size.
Approved algorithms for SCADA system	Advanced Encryption Standard(AES), Data Encryption Standard (DES), Triple DES	RSA, Elliptic Curve Cryptography(ECC)

Table 4.1 Comparison of Symmetric and Asymmetric Cryptosystem

Table 4.1 shows a detailed comparison of the two cryptosystems. It is necessary to determine the appropriate choice of cryptosystem in SCADA communication to provide a secure end-to-end communication. Both symmetric and asymmetric cryptographic solutions have their own advantages and disadvantages. Application of such solutions in SCADA communication may present design and operational challenges. So it is necessary to identify appropriate solution for specific SCADA control system.

IEC 62351 is one of the recommended standards by NIST for securing the communication between control systems. It is widely adopted in substation automation system. Table 4.2 shows the different attacks in SCADA system with the cryptographic solutions proposed by IEC 62351 to handle them.

Type of Attack	**Cryptographic Solutions by IEC 62351**
Data-Integrity Attack	Symmetric (AES, DES, TDES) and Asymmetric Encryption algorithms (RSA, ECC), Hashing algorithms (SHA-1 , SHA-2, SHA-256)
Authentication Attack	HMAC, Asymmetric Digital Signature schemes
Confidentiality Attack	Symmetric and Asymmetric Encryption algorithms, Asymmetric Digital Signature schemes
Non-repudiation	Asymmetric Digital Signature schemes

Table 4.2 Cryptographic solutions for different attacks in SCADA communication

4.2 Role of Asymmetric Cryptography in SCADA Communication Protocols

Both symmetric and asymmetric based approaches become major counter measures against such attacks [7]. The following describes the need for asymmetric cryptography in SCADA communication on two major topics:

4.2.1 In Encryption and Decryption

Encryption & Decryption are elementary cryptographic methods to achieve secure communication and information protection from unauthorized users. In SCADA systems, most devices are expected to have at least basic cryptographic capabilities, including the support for symmetric and asymmetric cryptography. Although symmetric key encryption is faster and uses less computational resources than asymmetric counterpart, sharing a common secret key increases the risk for attacks. For SCADA system involving millions of devices, adopting symmetric counter measure leads to generation of several keys, one each for a communication with every different party. Key management becomes difficult and moreover authenticity of the message cannot be verified. Hence there is a need to use a faster and a light weight asymmetric encryption scheme in SCADA systems.

4.2.2 In Authentication

It is a system for certifying the origin of a communication or the process for verifying that an entity or object is who or what it claims to be. Authentication is a crucial identification process to eliminate attacks targeting data integrity. Authentication protocols used in SCADA system should be highly efficient, tolerant to attacks and faults, and also support multicast traffic. Keyed-Hash Message

Authentication Code (HMAC) [21] and digital signature schemes are the common authentication mechanisms in SCADA.

Multicast has wide applications in SCADA systems, including monitoring, protection, and information dissemination e.g. in substation communication systems [11]. The most straightforward multicast authentication scheme is to use asymmetric digital signatures, which is also recommended by a recent security standard for substation communication, IEC 62351.This is mainly because HMAC does not provide data-origin authentication in multicast traffic. In group traffic, all grouped members share the same single HMAC key (symmetric key) and hence the identity of the sender is not uniquely established. Although HMAC provides group-level security, data-origin authentication is not achieved. Since asymmetric digital signature uses 2 keys, one which is never shared, it can provide true data-origin authentication making it a valuable choice.

4.3 Challenges in implementing Cryptographic solutions for SCADA security

It is always necessary to understand the performance impacts for any Critical infrastructure using SCADA system before introducing any cryptographic solution for its security. The typical characteristics of SCADA network make it challenging to adapt cryptographic protocols such as asymmetric cryptosystems into SCADA systems with limited resources and SCADA systems involving real-time traffic. Existing cryptographic technologies for authorization, authentication, encryption and decryption require more bandwidth, processing power and memory than what the current SCADA device generally has. Hence, the application of cryptographic

solution to SCADA systems poses significant challenges mainly due to following constraints.

- **Limited computational capacity**: The remote equipment such as RTUs is an embedded system having low computational and space capacity.
- **Low rate data transmission**: Since the SCADA system has been used for a long time, the communication line of the SCADA network has low bandwidth.
- **Real-time processing**: The SCADA system should behave accurately. Delay of data processing could cause serious problem.

The difficulty of applying security technology to the system makes the constraints to be a basic consideration for applying security mechanism.

4.3.1 Performance Issues while implementing IEC 62351

The scope of IEC 62351 lies in the development of standards for security of communication protocols defined by IEC TC57, IEC 60870-5& 6 series, IEC 61850 series and IEC 61968 series for Substation Automation control systems. They try to provide the standards for authenticating and encrypting SCADA communication link in Power system. IEC 62351-3 to IEC 62351-6 provide various levels of protocol security, depending upon the protocol (e.g. MMS, GOOSE, SMV, DNP3 etc.). IEC 62351 approves the use of the following asymmetric algorithms for providing cryptographic solutions:

- RSA with 2048 bits until 2029
- RSA with 3072 bits for CA’s after 2030

- Elliptic Curve Cryptography (ECC) with curves P-224, K-233 or B-233 until 2029
- ECC with curves P-256, P-384, P-521, K-283, K-409, K-571, B-283, B-409 and B-571 after 2030.

RSA and ECC are the widely recommended algorithms for digital signing. Embedded SCADA devices/RTU has little computational power and only a small portion can be made available for protection and control. Some SCADA applications using protocols like GOOSE and SV have strict real-time constraints.

Currently, IEC 62351 explicitly specifies the use of RSA as a solution to protect and authenticate time-critical messages. The following issues were found while implementing IEC 62351 in Substation Automation system.

- Software implementation of RSA digital signature scheme did not meet the real-time requirements with today's existing RTU's/IED's hardware.
- RSA requires longer keys in order to be secured compared to other cryptosystems like ECC. Though a longer key length in itself is not so much disadvantage, it contributes to slower encryption and decryption which makes it unsuitable for SCADA applications with real-time constraints. RSA theory says that for an n-bit key, computational effort for encryption is proportional to n^2, while effort for decryption is proportional to n^3.
- The delay involved in implementing RSA digital signature schemes, leaves few devices and communication channel unsecured without adopting any cryptographic technique for encryption.

This indicates that although RSA is highly recommended by IEC 62351, its low computational efficiency indeed affects the communication performance of time-critical applications, which demands the need to consider an alternative new and faster asymmetric cryptosystem. ECC has attracted increased attention in SCADA networks over RSA, with few researchers in terms of required key lengths and processing times. Although ECC provides a better performance over RSA, it is necessary to consider approaches or techniques faster and secure than ECC for SCADA real-time applications.

Our research is aimed at overcoming some of the limitations and shortcomings of the presently specified cryptographic security measures for SCADA real-time systems and improving their performance by proposing the application of faster and light-weighted NTRU cryptographic algorithm for SCADA security, over RSA or ECC.

CHAPTER 5

INTRODUCTION TO NTRU CRYPTOGRAPHIC ALGORITHM AND PROPOSED WORK

This chapter provides a brief description of the NTRU encryption and signature algorithm, and the proposed work of using NTRU for SCADA security.

Critical Infrastructure using SCADA system incorporates millions of electronic devices and users. To meet the cyber security requirements, every node/device in the SCADA system must have at least basic cryptographic functions, such as symmetric and asymmetric cryptographic primitives, to perform data encryption and authentication. This thesis is intended to provide a secure and faster cryptographic solution for SCADA system security using NTRU lattice based asymmetric cryptographic algorithm.

5.1 Introduction to NTRU

NTRU is a public key cryptosystem (PKCS) and an IEEE 1363.1 and X9.98 standard [APPENDIX I]. It was first published in 1996 by J.Hoffstein, J.Pipher and Silverman. That same year, the developers of NTRU joined with D. Lieman and founded the NTRU Cryptosystems, Inc., and were given a patent on the cryptosystem. In 2009, the company was acquired by Security Innovation, a software security company. It uses lattice based cryptography to encrypt and decrypt data. NTRU is based on algebraic structures of certain polynomial rings. The hard problem on which NTRU is based is the Short Vector Problem (finding a short vector in lattice). It consists of two algorithms: NTRUEncrypt, which is used for encryption, and NTRUSign, which is used for digital signatures. NTRU encryption

is proposed as a public-key encryption enabling high-speed processing. The NTRU encryption performs encryption and decryption by polynomial operations that can be implemented at higher speeds, as compared to RSA encryption that carries out modulo exponentiation under a certain rule and ECC that performs scalar multiplication for points on an elliptic curve. Another added advantage is unlike RSA and ECC, NTRU is not known to be vulnerable to quantum computer based attacks.

5.2 NTRU Public Key Cryptosystem

NTRU stands for n^{th} degree Truncated Ring polynomial Unit. NTRU is a relatively new Public Key Cryptosystem (PKCS) that uses lattice-based cryptography to encrypt and decrypt data. The algorithm is based on embedding messages in a polynomial ring, R. The ring R consists of truncated polynomials of degree N-1 having integer coefficients that are reduced modulo certain parameters, after every math operation. The notation for the Ring is given as:

$$\mathbf{R= Z[X] / (X^{N-1})}$$

Where Z represents the set of integers and N is 1 more than the degree of the polynomial. A full mathematical explanation is beyond the scope of this thesis and the reader is referred to the literature for an in-depth analysis of NTRU cryptography.

A brief explanation of the algorithm is as follows:

5.2.1 NTRU Parameters

NTRU PKCS is specified by a number of parameters and keys as shown in table 5.1

NTRU Parameter	Explanation
N	The polynomials in the truncated polynomial ring have degree N-1 (Non-secret)
q	Large modulus: The coefficients of the truncated polynomials will be reduced mod q. (Non-secret)
p	Small modulus: The coefficients of the message are reduced to mod p (Non secret)
f	A polynomial that is the private key (Secret)
g	A polynomial that is used to generate the public key h from f (Secret but discarded after initial use)
h	A polynomial that is the public key
r	The random "blinding polynomial. (Secret but discarded after initial use)
k	A security parameter which controls resistance to certain types of attacks, including plaintext awareness.
$\mathbf{d_f}$	The polynomial f has d_f coefficients equal to 1, (d_f-1) coefficients equal to -1, and the rest equal to 0.
$\mathbf{d_g}$	The polynomial g has d_g coefficients equal to 1, d_g coefficients equal to -1, and the rest equal to 0.
$\mathbf{d_r}$	The polynomial r has d_r coefficients equal to 1, d_r coefficients equal to -1, and the rest equal to 0.

Table 5.1 NTRU Parameters and Keys

5.2.2 Key Generation

Bob wants to create a public/private key pair for the NTRU public key cryptosystem.

- Bob chooses 2 random "small" polynomials **f** and **g** in the defined ring R A "small" polynomial is relative to a random polynomial **mod q**, i.e., the coefficients are much smaller than **q**.

- Bob then computes the inverse of **f modulo q** and the inverse of **f modulo p**.

  The inverses are denoted as $\mathbf{f_q}$ and $\mathbf{f_p}$ respectively.

  $$\mathbf{f*f_q = 1 \text{ (modulo q) and } f*f_p = 1 \text{(modulo p)}}$$

  Bob should select f such that its inverses $\mathbf{f_q}$ and $\mathbf{f_p}$ exists.

- Bob computes the product, h= $\mathbf{pf_q * g}$ **(modulo q).**
- Bob's private key is the pair of polynomials **f** and $\mathbf{f_p}$. Bob's public key is the polynomial **h**.

5.2.3 Encryption

Alice wants to send a message to Bob using Bob's public key **h**.

- Alice converts her message in the form of a polynomial **m** whose coefficients are chosen modulo **p**, between **–p/2** and **p/2** (m is a small polynomial modulo q)
- Alice randomly chooses a random polynomial **r**. This is the "blinding value", which is used to obscure the message.
- Alice computes the polynomial **e = pr * h + m (modulo q)**.
- The polynomial **e** is the encrypted message which Alice sends to Bob.

5.2.4 Decryption

Bob on receiving Alice's encrypted message **e**, wants to decrypt it.

- Bob uses his private polynomial **f** to **compute a = f * e (modulo q)**. Since Bob is computing **a modulo q**, he chooses the coefficients of **a** to lie between **–q/2** and **q/2**.
- Bob next computes the polynomial b = a (modulo p) reducing each of the coefficients of **a modulo p.**
- Bob uses his other private polynomial $\mathbf{f_p}$ to compute $\mathbf{c = f_p * b}$ **(modulo p).**

- Polynomial **c** will be Alice's original message **m.**

5.2.5 Example

Let **N** = 11, **q** = 32, **p** = 3, $\mathbf{d_f}$ = 4, $\mathbf{d_g}$ = 3.

Bob needs to choose a polynomial **f** of degree 10 with four 1's and three -1's, and he needs to choose a polynomial **g** of degree 10 with three 1's and three -1's. Suppose he chooses:

$$\mathbf{f = -1 + X + X^2 - X^4 + X^6 + X^9 - X^{10}}$$

$$\mathbf{g = -1 + X^2 + X^3 + X^5 - X^8 - X^{10}}$$

Next Bob computes the inverse $\mathbf{f_p}$ of **f modulo p** and the inverse $\mathbf{f_q}$ of **f modulo q** He finds that:

$$\mathbf{f_p = 1 + 2X + 2X^3 + 2X^4 + X^5 + 2X^7 + X^8 + 2X^9}$$

$$\mathbf{f_q = 5 + 9X + 6X^2 + 16X^3 + 4X^4 + 15X^5 + 16X^6 + 22X^7 + 20X^8 + 18X^9 + 30X^{10}}$$

The final step in key creation is to compute the product

$\mathbf{h = pf_q * g = 8 + 25X + 22X^2 + 20X^3 + 12X^4 + 24X^5 + 15X^6 + 19X^7 + 12X^8 + 19X^9 + 16X^{10}}$ **(modulo 32)**

Bob's private key is the pair of polynomials **f** and $\mathbf{f_p}$ and his public key is the polynomial **h.**

For the purposes of this tutorial, let $\mathbf{d_r}$ **= 3.** Now, suppose Alice wants to send the message,

$$\mathbf{m = -1 + X^3 - X^4 - X^8 + X^9 + X^{10}}$$

to Bob using Bob's public key,

$\mathbf{h = 8 + 25X + 22X^2 + 20X^3 + 12X^4 + 24X^5 + 15X^6 + 19X^7 + 12X^8 + 19X^9 + 16X^{10}}$ **(modulo 32)**

She first chooses a random polynomial **r** of degree 10 with three 1's and three -1's. Say she chooses,

$$\mathbf{r = -1 + X^2 + X^3 + X^4 – X^5 – X^7}$$

Then her encrypted message **e** is,

$$\mathbf{e = r*h+m = 14 + 11X + 26X^2 + 24X^3 + 14X^4 + 16X^5 + 30X^6 +7X^7 + 25X^8 + 6X^9 + 19X^{10} \text{ (modulo 32)}}$$

Alice sends the encrypted message **e** to Bob.

Upon decryption, he uses his private key **f** to compute,

$$\mathbf{a = f*e = 3 -7X -10X^2 -11X^3 + 10X^4 + 7X^5 + 6X^6 + 7X^7 + 5X^8 – 3X^9 -7X^{10} \text{ (modulo 32)}}$$

Note that when Bob reduces the coefficients of **f*e** modulo 32, he chooses values lying between -15 and 16, not between 0 and 31. It is very important that he chooses the co-efficient in this way. Next Bob reduces the coefficients of **a** modulo 3 to get,

$$\mathbf{b = a = -X –X^2 + X^3 + X^4 + X^5 + X^7 –X^8 –X^{10} \text{ (modulo 3)}}$$

Finally Bob uses $\mathbf{f_p}$, the other part of his private key, to compute

$$\mathbf{c = f_p * b = -1 + X^3 –X^4 –X^8 + X^9 + X^{10} \text{ (modulo 3)}}$$

The polynomial **c** is Alice's message **m**, so Bob has successfully decrypted Alice's message.

5.2.6 Theoretical Operating Specifications

This section gives an overview of the theoretical operating characteristics of the NTRU PKCS. There are four integer parameters (N, P, Q, and K) as described before. The following table 5.2 summarizes the NTRU PKCS characteristics in terms of these parameters.

Plain Text Block	(N-K) $\log_2 P$ bits
Encrypted Text Block	N $\log_2 Q$ bits
Encryption Speed	$O(N^2)$ operations
Decryption Speed	$O(N^2)$ operations
Private Key Length	$2N\log_2 P$ bits
Public Key Length	$N\log_2 Q$ bits

Table 5.2 NTRU PKCS operating characteristics

5.3 NTRU Signature Scheme

Digital signature schemes are a type of public-key encryption that is used for identifying a sender and preventing data falsification when data is sent from a receiving machine/client to a machine/client. The transmitting client creates signature data for data desired to be transmitted using a private key of the transmitting client, and then transmits the signature data to the receiving client together with the desired data. The receiving client performs a verification of the signature data using a public key corresponding to the private key of the transmitting apparatus to judge whether the desired data has been falsified .It is difficult to calculate a value of the private key from the public key.

In the key generation under the NTRUSign signature scheme, the private key and the public key are generated by using multiple elements in a polynomial ring R with integer coefficients and an ideal of the ring R modulo polynomial X^{N-1}. For generating a signature under the NTRUSign signature scheme for a message, the generated private key and 2N- dimensional vector, which is a hash value of the message, are used. For the signature verification of the NTRUSign signature

scheme, the public key, the signature of the message, and the 2N- dimensional vector are used. A full mathematical explanation is beyond the scope of this paper and the reader is referred to the literature [10] for an in-depth analysis of NTRU Signature scheme.

5.4 Advantages of NTRU over other PKCS

The benefits of using NTRU has been listed below which makes it a right choice for application in SCADA environment.

- NTRU has been observed to be multiple times faster than RSA and ECC.
- It consumes minimal resources including CPU and battery.
- Significantly reduces server resource utilization for large-scale deployments.
- Improves the data throughput(over RSA) when integrated with SSL
- Ideal for low power or hard to access environments, for embedded devices where code size is a major limitation.
- Resistant to Quantum computing attacks.

5.5 Proposed Work

This section provides a brief description about the issues with cryptographic solutions in SCADA with previous approaches to handle it and the proposed work.

5.5.1 Issue & Previous Approaches

Security standards for SCADA protocols such as IEC 62351 and AGA-12 explicitly specify the use of RSA & ECC based digital signature schemes for providing authentication, data-integrity, confidentiality and non-repudiation. Due to the practical difficulties in implementing RSA digital signature scheme in limited environment, the security community is making significant efforts to design an

alternative solution. HMAC [21] and HORS [22] are two such alternatives designed. However they have their own limitations. HMAC does not provide true data-origin authentication and HORS implementation requires a large public key size on the order of 10 KB, resulting in non-negligible overhead for both communication and storage. Although ECC based authentication mechanisms can provide better performance results when compared to RSA, in practical it is better to consider algorithms faster than ECC for real-time applications. Considering these issues, we propose the use of light-weighted and faster NTRU asymmetric cryptosystem [9 and 10] into SCADA systems for use in encryption and digital signature. No previous research work suggests the use of NTRU as asymmetric algorithm in SCADA communication.

5.5.2 Proposed Approach

In our proposed approaches for encryption and authentication, NTRU is considered as the asymmetric cryptography. NTRU is chosen over RSA or ECC because it not only necessitates less power consumption and computation, but also reduced amounts of data transmitted and stored. Experiments conducted previously reveal that NTRU delivers substantial performance and size advantages over its competitors running multiple times faster while consuming minimal resources including CPU and battery [23 and 24]. Unlike RSA and ECC, NTRU is not known to be vulnerable to quantum computer based attacks. All these factors make it an ideal choice for its use in SCADA systems/devices working under a limited environment.

The proposed Encryption and Authentication mechanisms in SCADA system include Certificate Creation I as its first phase where the key-generation operation and certificate issuance takes place.

5.5.2.1 Key Generation and Certificate Creation

To implement asymmetric/public-key cryptography in SCADA systems for providing a secure communication on an in-secure public network, it requires the use of digital certificates to verify the identity of the SCADA device/client machine. A public-key infrastructure (PKI) [25] is a system for the creation, storage, and distribution of digital certificates which are used to verify that a particular public key belongs to a certain entity. Every device/client machine in SCADA system that involves in transmitting messages has to create a NTRU private and public key pair using the NTRU key generation algorithm and stores it in the local key store. Administrators of these devices/machines direct a Certificate Signing Request (CSR) to the organization's physically protected Certificate Authority (CA). CA which is a part of PKI system signs the CSR after analyzing the requester and then issues digital certificates that contain a public key and the identity of the owner. The public key is publicly made available to all parties with whom the requester communicates or it is exchanged during communication. This certificate is then used for encryption and authentication purposes. Figure 5.1 shows the steps involved in creating a NTRU digital certificate.

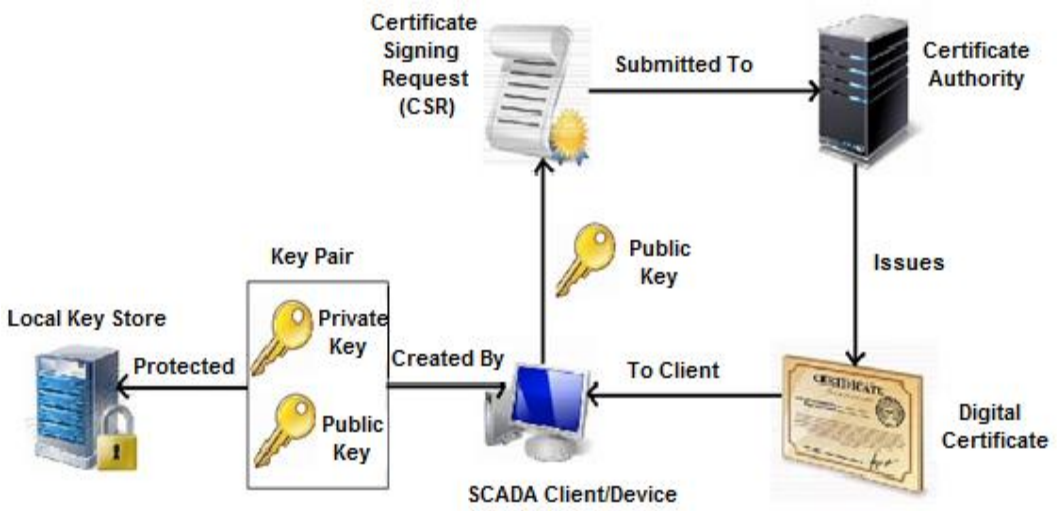

Figure 5.1 Key Generation and Certificate Creation

For SCADA applications involving real-time traffic, certificate exchange is not done as part of the messages; the digital certificates must be pre-installed on the receiving nodes.

5.5.2.2 NTRU Encryption mechanism in SCADA communication

The NTRU algorithm performs encryption and decryption by polynomial operations that can be implemented at higher speeds, as compared to RSA encryption that carries out modulo exponentiation under a certain rule and an elliptic curve cryptosystem that performs scalar multiplication for points on an elliptic curve. Hence, the NTRU encryption in SCADA achieves higher-speed processing than conventional public-key encryption, and is also capable of performing, when used in software processing, the processing in a practical period of time.

In order to ensure data-integrity, SCADA messages have to be encrypted when they are transmitted. In this proposed mechanism, the messages are encoded into a truncated polynomial ring R. When a SCADA device/client “A” wishes to transmit

message/control signals to another device/client “B”, “A” uses the public key of “B” published by the Certificate Authority to encrypt the encoded message to create a cipher message. On the receiving end, “B” decrypts the cipher message using its own private key to obtain the original message. Figure 5.2 shows how encryption and decryption of SCADA message is done using NTRU asymmetric keys.

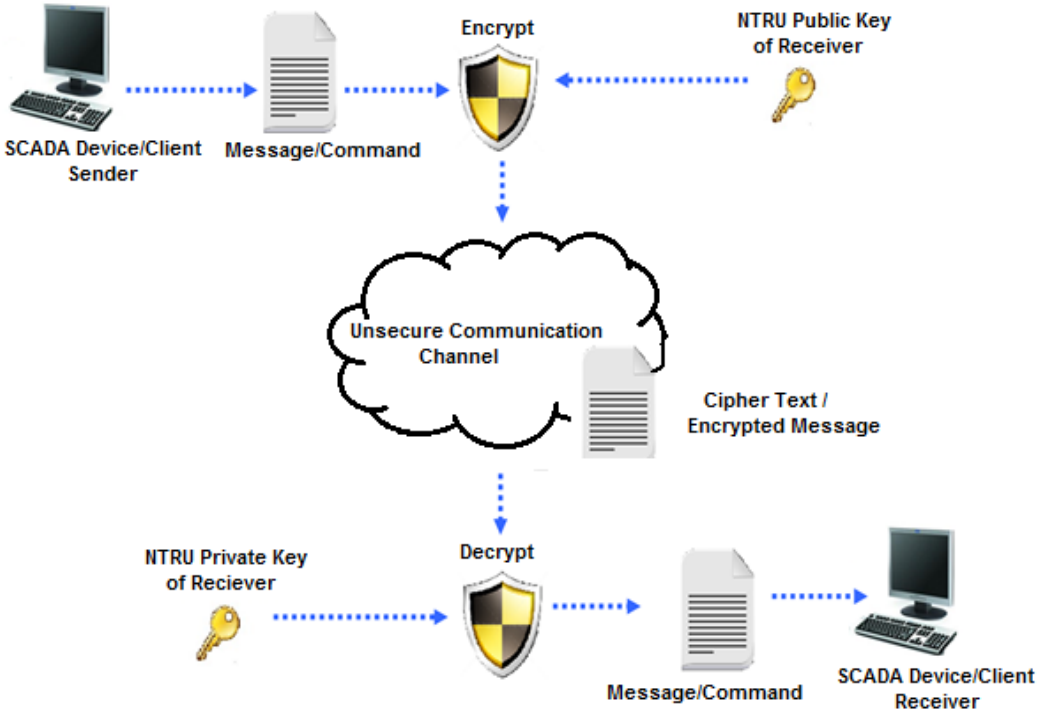

Figure 5.2 Encrypting and decrypting SCADA messages with NTRU Keys

For SCADA communication involving lengthy message transmission, the NTRU asymmetric encryption can be used to distribute the secret symmetric session keys with which the actual message encryption takes place. A detailed explanation of it can be seen in Figure 5.3

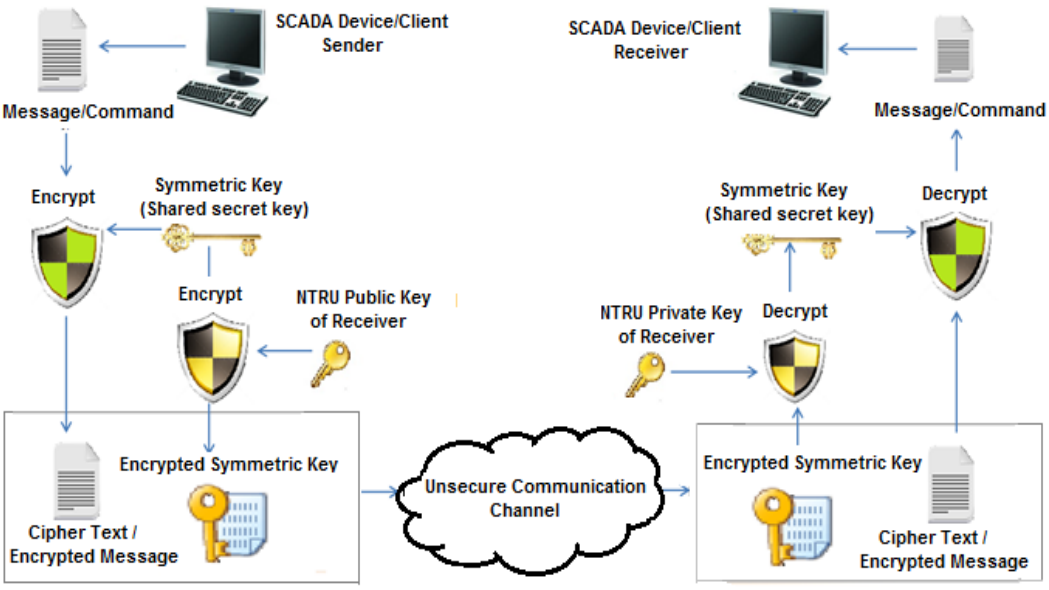

Figure 5.3 NTRU used in distributing secret symmetric key

For each new session, a new session key will be established. This is mainly adopted for bulk message transmission because symmetric key encryption is much faster than asymmetric key encryption. The secret key used can be generated using any symmetric algorithm approved by IEC 62351 such as AES, DES etc.

5.5.2.3 NTRU Based Authentication in SCADA communication

In this approach NTRU Digital Signature scheme is chosen in SCADA systems for ensuring authenticity. The message to be transmitted is encoded into a polynomial ring R. To verify the integrity of the data that is transmitted, the data is subjected to non-keyed hash algorithm such as SHA-1, SHA-256 etc. The message digest obtained by this process is signed using the sender's NTRU private key. The encrypted message digest is sent to the receiver along with the encoded message that is encrypted using the receiver's NTRU public key. Thus the sender sends 1) Encrypted message 2) Encrypted message digest which is the digital signature.

The receiver upon receiving them decrypts the message using its own NTRU private key and computes the message digest using the same hash algorithm. It then verifies the digital signature using the publicized NTRU public key of the sender and the computed message digest. In our proposed approach, NTRU digital signature algorithm is used rather than the slow RSA which makes it convenient for use in SCADA applications involving real-time constraints as seen in Fig 5.4.

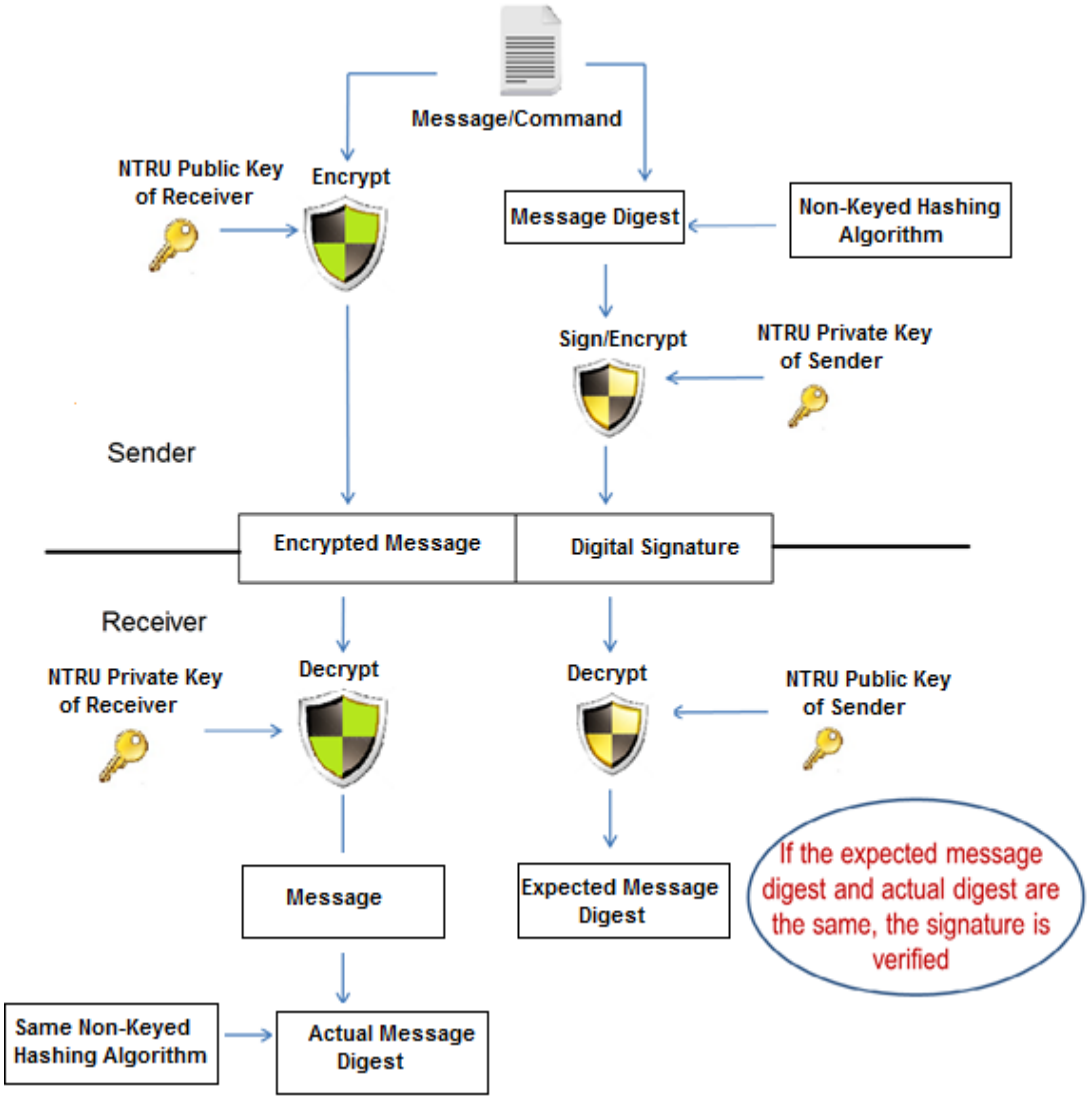

Figure 5.4 NTRU based Authentication in SCADA communication

Previously, the security of real-time traffic was limited to message authentication with no encryption specified. However since implementing NTRU digital signature takes less time compared to its RSA counterpart, encryption can also be specified for such applications with real-time constraints ensuring data-integrity of the message as well. We can even consider a hybrid approach that uses NTRU for digital signature and a symmetric key (such as AES, DES) for encryption purpose.

CHAPTER 6

EXPERIMENTATION AND TEST RESULTS

This chapter focuses on the experimental case study conducted on Raspberry Pi and Intel Core machine for the comparison of NTRU asymmetric cryptosystem with others used in SCADA system such as RSA and ECC. The first section of this chapter gives an introduction to Raspberry Pi and the second section gives a detailed description of the experimentation conducted and the test results observed.

6.1 Introduction to Raspberry Pi

The tests for comparing the performance of NTRU, RSA and ECC with respect to their encryption, decryption and digital signature speeds were conducted on Raspberry Pi. Raspberry Pi is a credit-card sized single board computer developed in UK. It has a Broadcom BCM2835 system on a chip which includes ARMv6k 700 MHz processor. Figure 6.1 shows the Raspberry Pi kit used for experimentation.

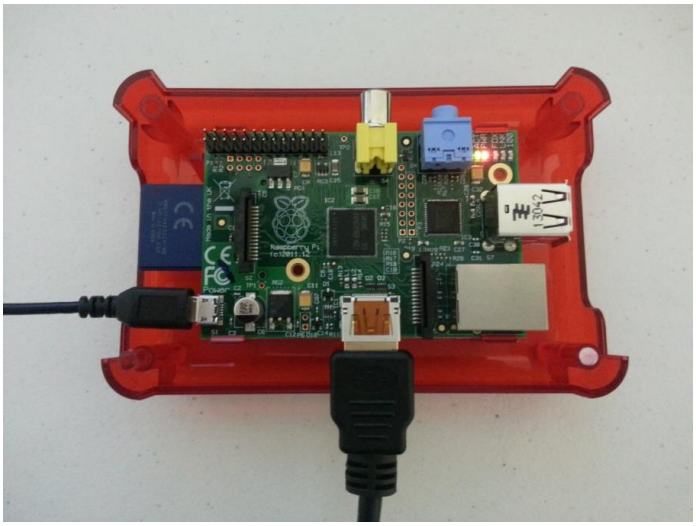

Figure 6.1 Raspberry Pi

There are two models of Raspberry Pi available mainly based on the size of memory. Model B with 512 MB SDRAM @ 400 MHz was chosen for our experimentation purpose. The Raspberry Pi was chosen to run on Debian Linux operating system. It also supports other OS such as RISC OS, FreeBSD, NetBSD and Plan 9. It does not include a built-in hard disk or solid-state drive, but uses an SD card for booting and long-term storage.

6.2 Experimentation

To motivate our research, the performance characteristics of NTRU, RSA and ECC are observed by implementing the algorithms for computation using the open source Bouncy Castle 1.47 Java library and comparing their experimental run times. Open JDK-7 with Cacao Virtual Machine was used for faster execution. Our experiments were conducted on 700 MHz Raspberry Pi running Linux and Intel(R) Core(TM) i3 CPU @ 2.27GHz to facilitate performance comparison.

For the first experiment, the run times for three fundamental primitives of a cryptosystem: encryption, decryption and key generation was chosen as comparison parameters for the two algorithms (NTRU and RSA) for different key-sizes. The test was done for randomly generated message of size 32 bytes. Table 6.2 and 6.3 shows the comparison of computation times of encryption algorithms between Intel machine and Raspberry Pi. The result shows that RSA leads to a bad performance while generating asymmetric keys, and worst when the CPU speed is as low as 700 MHz. Also at equivalent cryptographic strength, NTRU performs costly private key operations much faster than RSA. As key sizes increase, RSA's operations per second decrease cubicly, whereas NTRU's operations per second decrease

quadratically (RSA-2048 can be compared to NTRU-439, RSA-3072 can be compared to NTRU-743). RSA Decryption is expensive because it involves modular exponentiation of huge numbers. Though there wasn't any huge difference in the encryption speed between NTRU and RSA in Intel core machine, NTRU encryption was 2-3x faster than RSA encryption. The tabular results show that NTRU would be a better choice for encryption in SCADA systems were key management of symmetric keys become difficult.

Asymmetric Algorithm	**Key Generation (ms)**	**Encryption (ms)**	**Decryption (ms)**
RSA-1024	21.27	0.45	1.25
RSA-2048	90.51	0.52	3.03
RSA-3072	233.68	0.59	9.48
NTRU-439	6.27	0.28	0.22
NTRU-743	9.94	0.32	0.25

Table 6.1 Comparison of Key Generation, Encryption and Decryption speed on Intel Core @ 2.27 GHz

Asymmetric Algorithm	**Key Generation (ms)**	**Encryption (ms)**	**Decryption (ms)**
RSA-1024	3701.53	9.23	160.55
RSA-2048	24714.24	72.80	1123.98
RSA-3072	69522.21	172.23	3618.86
NTRU-439	1173.29	15.98	20.59
NTRU-743	2970.41	160.35	46.12

Table 6.2 Comparison of Key Generation, Encryption and Decryption speed on Raspberry Pi @ 700 MHz

Secondly, the performance of RSA, ECC and NTRU digital signature schemes were compared using Java. After performing several tests, the average time taken for signing and verification for various algorithms can be seen in Table 6.4. Clearly the total time taken by NTRU signature scheme is apparently very less when compared to RSA and ECC making it an appropriate choice for providing authentication in SCADA systems with real-time constraints.

Asymmetric Algorithm	**Signing speed (ms)**		**Verification Speed (ms)**		**Total Digital Signature Speed (ms) (~)**	
	Intel Core	**Rasp. Pi**	**Intel Core**	**Rasp. Pi**	**Intel Core**	**Rasp. Pi**
RSA-2048	40.47	1324	1.64	48.81	42	1372
RSA-3072	63.37	3962.9	1.97	85.34	65	4048
ECDSA-256	9.41	437.3	4.48	336.7	13	773
ECDSA-512	21.55	926.5	6.30	468.8	27	1394
NTRU-439	5.16	402.6	4.18	206.4	9	608
NTRU-739	6.23	532.4	5.74	396.2	12	928

Table 6.3 Comparison of Signing and Verification speed on Intel Core @ 2.27GHz and on Raspberry Pi @ 700 MHz

6.3 Test Results

The following results were observed based on the experimentation conducted. Figure 6.2 shows the performance of RSA and NTRU key generation operation on Raspberry Pi. The time taken for generating the private and public key for each algorithm with different key sizes is represented as a bar graph. Results indicate that RSA key generation is 20 to 25 times slower than the corresponding NTRU key generation operation for the same level of security.

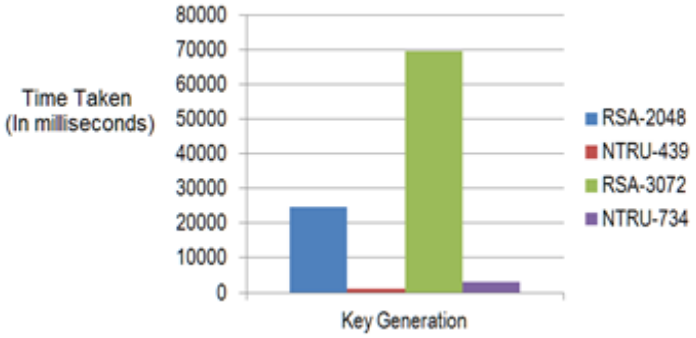

Figure 6.2 Performance Comparison of RSA and NTRU Key Generation

Based on the encryption and decryption runtimes of NTRU and RSA on Raspberry Pi, Figure 6.3 shows the graphical comparison of their total runtime for different key sizes of NTRU and RSA. Observed results show that total time taken for encryption and decryption by RSA is 18 to 33 times more than that of NTRU for the same level of security.

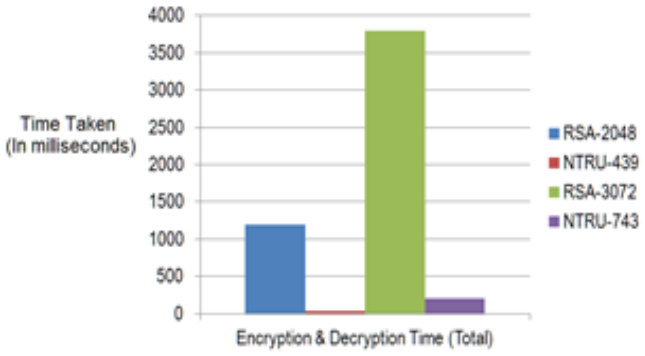

Figure 6.3 Performance Comparison of RSA and NTRU Encryption and Decryption

The experimental evaluation of digital signature schemes of all three algorithm results in the following graph. Figure 6.4 shows the number of operations that can be

performed per second of each algorithm based on the Intel core results. Clearly NTRU based digital signature scheme takes less time when compared to its counterparts for the same level of security.

Figure 6.4 Number of Digital Signature operations per second of RSA, ECC and NTRU

CHAPTER 7

CONCLUSION AND FUTUREWORK

In previous sections we summarized the need for securing SCADA communication and about the practical difficulties in implementing the currently used cryptographic standards (such as RSA) in applications involving real-time constraints and devices with hardware limitations. Thus it becomes necessary and useful to devise an alternative asymmetric cryptography solution to enable end-to-end security in all SCADA systems/devices irrespective of any constraints.

In this research, a new alternative solution for the above said issue was proposed by employing NTRU-based encryption and authentication schemes in SCADA that addresses the Data-Integrity, Confidentiality, Authentication and Non-repudiation issues. The performance evaluation of different asymmetric algorithm such as RSA, NTRU and ECC was done in Java. Their encryption, decryption and key-generation speeds were compared. The results show that cryptographic operations of NTRU are indeed faster than RSA & ECC for the same level of security (around 2 to 10 times faster). The time taken for RSA and NTRU digital signature algorithm was compared along with the time for encrypting and decrypting data using the hybrid solution. While a more optimized version of NTRU in C would yield faster times when compared to its counterparts' optimized version. Since NTRU is not based on any factorization or discrete logarithmic problems allowing it to achieve high speeds with the use of minimal computing power. This shows that usage of a light-weight asymmetric key protocol like NTRU is necessary for supporting a secure and faster real-time critical application in SCADA systems.

To further motivate our research, in the future we intend to integrate and compare the performances of these cryptographic operations in real-time SCADA protocols and to evaluate the results in a simulated SCADA device.

BIBLIOGRAPHY

[1] Wenye Wang, Zhuo Lu., "Cyber Security in the Smart Grid: Survey and challenges", *Computer Networks: The International Journal of Computer and Telecommunications Networking*, (April, 2013): Vol.57 Issue 5.

[2] Keith Stouffer, Joe Falco and Karen Scarfone., *Recommendations of National Institute of Standards and Technology,* June, 2011.

[3] P.Blomgren and S.M Kotronx., "Cryptographic Protection of SCADA Communications Part 1: Background, Policies and Test Plan," *American Gas Association* (2006): Draft 4, AGA Report 12.

[4] S.C.Patel and G.D. Bhatt and J.H.Graham., "Improving the cyber security of SCADA communication network", *Communication of ACM*,(July, 2009): Vol.52 No.7

[5] S.Sridhar, G.Manimaran., "*Data integrity attacks and their impacts on SCADA control system*", Proceedings of IEEE power and Energy Society General Meeting (PES'10), 2010.

[6] S.Fries et al., "Security for the Smart Grid- Enhancing IEC 62351 to Improve Security in Energy Automation Control," *Int'l. J. Advances in Security*,(2010): Vol.3, No.3-4, pp. 169-83.

[7] Martin Drahansky and Maricel Balitanas., "Cipher for Internet-based Supervisory Control and Data Acquistion Architecture," *Journal of Security Engineering*, (June, 2011).

[8] Aamir Shahzad and Shahrulniza Musa., "Cryptography and Authentication Placement to Provide Secure Channel for SCADA Communication", *International Journal of Society (IJS),* (2012): Vol.6, Issue.3.

[9] Hoffstein et al., *NTRU: A Ring-based Public Key Cryptosystem*, Appendix to U.S. Patent 6,081,597.

[10] Hoffstein et al., *NSS: The NTRU Signature Scheme*.

[11] Core IEC standards, IEC 61850: Power Utility Automation., *IEC 62351:Security*.Available: http:// www.iec.ch/smartgrid/standards/

[12] *1815-2012-IEEE Standard for Electric Power Systems Communications*-Distributed Network Protocol (DNP3)(2012)

[13] Clarke, Gordon; Reynders, Deon., "Practical Modern Scada Protocols: Dnp3, 60870.5 and Related Systems." *Newnes*. pp. 47–51.

[14] "Modbus Application Protocol V1.1b3", *Modbus Organization, Inc*.(August, 2013). Available: http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf.

[15] Frank Hohlbaum, Markus Braendle, Fernando Alvarez., "Practical considerations for implementing IEC 62351": *ABB Technical Report*.

[16] Rivest, R.; A. Shamir; L. Adleman., "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", *Communications of the ACM 21 (2)*: 120–126.

[17] *Standards for Efficient Cryptography Group (SECG), SEC 1: Elliptic Curve Cryptography*, Version 1.0, (September, 2000).

[18] Keith Stouffer, Joe Falco, Karen Kent., "Guide to Supervisory Control and Data Acquisition(SCADA) and Industrial Control System(ICS) security*". National Institute of Standards and Technology (NIST):* Special Publication 800-82.

[19] Boyer, Stuart A., "SCADA Supervisory Control and Data Acquisition". *USA: ISA - International Society of Automation*.(2010) p. 179.

[20] *NIST Industrial Control System Security (ICS).* Available: http://csrc.nist.gov/groups/SMA/fisma/ics/index.html

[21] Bellare, Mihir; Canetti, Ran; Krawczyk, Hugo., "Keying Hash Functions for Message Authentication", (1996).

[22] L.Reyzin, N.Reyzin, "Better than BiBa: short one-time signatures with fast signing and verifying", *Proc. Of Seventh Australasian Conference on Information Security and Privacy*,(2002).

[23] Hermans, J, Vercauteren F, Preneel B ., “Speed records for NTRU.” *Topics Cryptol* . (2010) *CT-RSA*: 73-88.

[24] Karu, P., Loikkanen, J.: “Practical Comparison of Fast Public-key Cryptosystems” (2001)

[25] Adams, Carlisle & Lloyd, Steve. “Understanding PKI: concepts, standards, and deployment considerations.” *Addison-Wesley Professional*.(2003) pp. 11–15

[26] “Secure Authentication for DNP3”, *Proc. IEEE Power and Energy Society General Meeting,* (2008)

[27] S.Mohagheghi, J.Stoupis, Z.Wang., “Communication Protocols and Network for Power Systems- Current status and future trends,” *Proc. Of Power Systems Conference and Exposition (PES’09)*, (2009)

VITA

Graduate College

University of Nevada, Las Vegas

Amritha Puliadi Premnath

Degrees:

Bachelor of Engineering in Computer Science, 2010

Thiagarajar College of Engineering

Master of Science in Computer Science, 2013

University of Nevada Las Vegas

Thesis Title: Application of NTRU Cryptographic Algorithm for Securing SCADA Communication

Thesis Examination Committee:

Chair Person, Dr. Ju-Yeon Jo, Ph.D.

Committee Member, Dr. Yoohwan Kim, Ph.D.

Committee Member, Dr. Laxmi Gewali, Ph.D

Graduate College Representative, Dr. Venkatesan Muthukumar, Ph.D.