

Bachelor of Science in Computer Science and Engineering

Enhancement of Bluetooth Security by Improving RSA Algorithm

Md. Akibul Alam

October, 2013

Department of Computer Science & Engineering
Chittagong University of Engineering & Technology
Chittagong-4349, Bangladesh.

Enhancement of Bluetooth Security by Improving RSA Algorithm

This thesis is submitted in partial fulfillment of the requirement for the degree of
Bachelor of Science in Computer Science & Engineering.

Md. Akibul Alam
ID: 0804060

Supervised by
Thomas Chowdhury
Assistant Professor
Department of Computer Science & Engineering (CSE)
Chittagong University of Engineering & Technology (CUET)

Department of Computer Science & Engineering
Chittagong University of Engineering & Technology
Chittagong-4349, Bangladesh.

The thesis titled “**Enhancement of Bluetooth Security By Improving RSA Algorithm**” submitted by Roll No. 0804060, Session 2008-2009 has been accepted as satisfactory in fulfillment of the requirement for the degree of Bachelor of Science in Computer Science & Engineering (CSE) as B.Sc. Engineering to be awarded by the Chittagong University of Engineering & Technology (CUET).

Board of Examiners

1. _____

Chairman

Thomas Chowdhury

Assistant Professor

Department of Computer Science & Engineering (CSE)

Chittagong University of Engineering & Technology (CUET)

2. _____

Member

Dr. Kaushik Deb

(Ex-officio)

Head

Department of Computer Science & Engineering (CSE)

Chittagong University of Engineering & Technology (CUET)

3. _____

Member

Department of Computer Science & Engineering (CSE)

(External)

Chittagong University of Engineering & Technology (CUET)

Statement of Originality

It is hereby declared that the contents of this project is original and any part of it has not been submitted elsewhere for the award of any degree or diploma.

Signature of the Supervisor

Date:

Signature of the Candidate

Date:

Acknowledgment

First of all I would like to thank almighty Allah for successful completion of this project. Then after I convey my sincerest thanks and gratitude to my honorable project supervisor Thomas Chowdhury, Assistant Professor, Department of Computer Science & Engineering, Chittagong University of Engineering & Technology, for his valuable suggestion, constructive advice, encouragement and sincere guidance throughout my project work. I also convey special thanks and gratitude to all my respected teachers of the department. I would like to thank all my friends and the staffs of the department for their valuable suggestion and assistance. Finally, I would like to thank my parents for their constant love and support during my study period.

Abstract

Bluetooth is a wireless PAN communication that represents a way of gaining easiness for sharing electronics data. Data security is an important issue in Bluetooth communication, where cryptography has a remarkable contributions to secure it. Current 128 bit E_0 stream cipher for Bluetooth security system can be broken under some attacks or conditions. The popular implementation of public key infrastructure in cryptography, composition of different algorithm like RSA, DES, triple DES, MD5 and several key agreement protocols are rarely maintained because of their vulnerability of attacks. This paper gives a Bluetooth security system comparing with all existing systems and illustrates an efficient implementation of RSA for Bluetooth communication for its security enhancement in network so that it is free from several kinds of attacks, attackers and cryptanalysts. The performance of the proposed system shows that, it is superior to all other existing Bluetooth communication system.

Contents

Chapter 1	1
1.1 Background and Present State of the Problem	2
1.2 Motivation of the Research	3
1.3 Objectives	4
1.4 Organization of the Paper	4
Chapter 2	5
2.1 Bluetooth	5
2.1.1 Radio Interface	5
2.1.2 Physical Link	6
2.1.3 Interface	6
2.1.4 Scatter Net	7
2.2 Cryptology	7
2.2.1 Plain Text	7
2.2.2 Cipher Text	7
2.2.3 Encryption	7
2.2.4 Decryption	7
2.2.5 Key	8
2.3 Cryptanalysis	8
2.3.1 Symmetric Key Cryptography	8
2.3.2 Asymmetric Key Cryptography	9
2.3.3 Advantage of Public Key Cryptosystem	10
2.4 Fundamental Theorem of Arithmetic	10
2.4.1 Mathematics Involved in RSA	11
2.4.2 Fermat's Little Theorem	11
2.4.3 Euler's Totient Function	11
2.4.4 Property of Modular Arithmetic	12
2.4.5 Linear Congruence	12

2.4.6 Property of Odd Integer -----	12
2.4.7 Properties of Prime Number -----	13
2.5 RSA Algorithm-----	13
2.5.1 Reasons RSA Algorithm -----	15
2.5.2 Improvements of RSA -----	15
2.5.2.1 CRT-RSA-----	16
2.5.2.2 Rebalanced RSA –CRT -----	16
2.5.2.3 Dual RSA ---- -----	17
2.6 Security of RSA -----	17
2.6.1 Brute Force Attack -----	17
2.6.2 Mathematical Attack -----	17
2.6.3 Timing Attacks -----	18
2.6.4 Chosen Cipher text Attack -----	20
2.6.5 Winners attack on RSA algorithm -----	21
2. 7 Coppersmith theorem -----	21

Chapter 3 ----- 22

3.1 System Model-----	22
3.2 Proposed System -----	23
3.3 Bluetooth security improvements through RSA algorithm-----	23
3.3.1 Sender -----	23
3.3.2 Receiver -----	24
3.4 Verification Process -----	25

Chapter 4	26
4.1 Sender	26
4.2 Receiver	27
4.3 Performance Evaluation	28
4.3.1 Winners attack removing	28
4.3.2 Timing attack removing	29
4.4 Comparison of Securities	30
4.4.1 RSA versus proposed algorithm securities analysis	30
4.4.2 Latest Bluetooth security system versus proposed system	31
4.4.3 Bluetooth security Analysis through all system	31
Chapter 5	33
Bibliography	34
Appendix	35

List of Figures

FIGURE 1.1: BLUETOOTH NETWORK USING DIFFARENT KINDS OF DEVICES-----	3
FIGURE 2.1: BLUETOOTH NETWORK MODEL -----	5
FIGURE 2.2: CATEGORIES OF CRYPTOGRAPHY-----	8
FIGURE 2.3: SYMMETRIC KEY CRYPTOGRAPHY -----	9
FIGURE 2.4: ASYMMETRIC KEY CRYPTOGRAPHY-----	9
FIGURE 3.1: EXISTING BLUETOOTH DATA TRANSFER POLICY -----	22
FIGURE 3.2: PROPOSED COMMUNICATION SYSTEM FOR BLUETOOTH NETWORK. -----	23
FIGURE 3.3: BLUETOOTH ENCRYPTION IMPROVEMENTS FOR SENDER -----	24
FIGURE 3.4: BLUETOOTH DECRYPTION IMPROVEMENTS FOR RECEIVER -----	24
FIGURE 4.1: ENCRYPTED BY SENDER PUBLIC KEY -----	26
FIGURE 4.2: ENCRYPTED BY RSA. -----	26
FIGURE 4.3: DECRYPTED BY RSA. -----	27
FIGURE 4.4: GUI REPRESENTATION OF BLUETOOTH SECURITY-----	27
FIGURE 4.5: ENCRYPTED BY SENDER PUBLIC KEY -----	28
FIGURE 4.6: ENCRYPTED BY RSA ALGORITHM -----	28
FIGURE 4.7: FINAL RESULT OF SUCCESSFUL WINNER ATTACK -----	29
FIGURE 4.8: FIRST SUCCESSFUL APPROACH TO TIMING ATTACK -----	29
FIGURE 4.9: FINAL RESULT OF SUCCESSFUL TIMING ATTACK -----	30
FIGURE 4.10: RSA VERSUS PROPOSED SYSTEM TIME COMPLEXITY -----	30
FIGURE 4.11: LATEST EXISTING VERSUS PROPOSED SYSTEM -----	31

List of Tables

TABLE 2.2: YEAR WISE DEVELOPMENT OF RSA ALGORITHM -----	18
TABLE 4.1: COMPARISON BETWEEN PROPOSED AND ALL OTHER EXISTING BLUETOOTH SECURITY SYSTEMS -----	32

Chapter 1

Introduction

Bluetooth is a flexible data communication system implemented as an extension to, or as an alternative for a wired communication. Bluetooth have gained a strong popularity in a number of vertical markets, including the health care, retail, manufacturing, warehousing and academia. These industries have profited from the productivity gains of using hand-held terminals notebook computer to transmit real time information to centralized host for processing.

Today Bluetooth is becoming more widely recognized as a general-purpose connectivity alternative for a broad range of business customers. The IEEE 802.15 protocol have become the standard protocol for Bluetooth. The 802.15 have been deployed in various types of locations including homes, school, airports, business offices, government buildings, military facilities, coffee shops, book stores as well as many others venue. One of the primary advantage offered by Bluetooth is its ability to provide unfettered connectivity to portable devices, such as wireless laptop and PDAs. The further widespread deployment of Bluetooth, however depends on whether secure network can be achieved. In order for critical data and services to be delivered over Bluetooth, reasonable level of security must be guaranteed. The current Bluetooth security system based on E_0 stream cipher can be easily cracked by commonly available hacking software. Bluetooth security is suffering from various security vulnerabilities such as winner attack, slide channel or timing attack, mathematical attack, denial of service attacks, absence of mutual authentication and session hijacking etc. Today it is important that information is sent confidentially over the network.

1.1 Background and Present State of the Problem

Bluetooth is a short range wireless radio specification adhoc network designed to replace wire as the medium for data and voice signal between electronic devices operates in the unlicensed 2.4000 gigahertz (GHz) to 2.4835 GHz Industrial, Scientific, and Medical (ISM) frequency band planned and implemented by Bluetooth Special Interest Group.

Current Bluetooth communication uses 128 bit E_0 stream symmetric cipher, which may be broken under certain conditions. A solution [2] based on DES to the short coming 128-bit E_0 stream ciphers in some cases can be cracked. This security system of Bluetooth has low credibility of PIN, High probability of non-link key cheat, vulnerability of attacks & the present of address spoofing. The problem with this approach is the distribution of encryption key used in DES, where both communication parties agree on one shared secret key that is known as symmetric key. There have a confusion that, how one party exchange this secret key with other party? It is also a problem, because it is possible that opponent can intercept the key during transmission of symmetric key. Another limitation of DES [2] is small key size, which is highly vulnerable to brute force attack. There have a hybrid system [1] based on DES and RSA is the solution of DES [2] but has small key size also. DES is a symmetric key cryptographic algorithm and RSA is an asymmetric key cryptographic algorithm in which public and private key pair is used. Here [1] DES use symmetric key and the size of the key is 56-bit only that is more vulnerable to attacks like brute force attack, man-in-middle attack etc. but it has no process for verifying the integrity of data.

Key agreement protocols [3] used in Bluetooth communication which gives an outline about generation of keys that are used to implement security in Bluetooth communication like encryption key generation, link key generation, unit key generation, initialization key generation and combination key generation.

The security issues of Bluetooth standard [4] introduced security frame work which includes both link level and service level security schemes [5]. Flexible security architecture is implemented at service level security. In the security frame work security modes can be defined for each Bluetooth device. There is an analysis of potential risks, attacks against the vulnerabilities like DOS, man-in- middle attack, spoofing, session

hijacking, eavesdropping etc. A system for Bluetooth communication is using Triple DES (with 2-keys), RSA, MD5 [6] has gave a volatile solution of those problem [4] [5] by solving brute force attack temporarily in worst case. There have also well-known timing attack [7] [8] and winners attack on RSA algorithm for Bluetooth network.



Figure 1.1: Bluetooth network using different kinds of devices.

1.2 Motivation of the Research

The aim of Bluetooth security through cryptography is to secure information so that only the intended parties can read data. Bluetooth had been developed from an aeon where cryptography had been developed for centuries. The advance of computer technology and popularity of personal computers provide a large base on which cryptographic applications are installed. The recent popularity of the internet and e-commerce have made strong demands on cryptography. Cryptosystem gives Bluetooth network's security from several kinds of attacks, attackers & vulnerabilities.

1.3 Objectives

In this project, we give our attention on the following specification & has the following features:

- Analysis of existing Bluetooth security.
- Analysis of RSA algorithm.
- Analysis of RSA for Bluetooth security
- Enhancement of RSA algorithm through asymmetric way.
- Simulate the Bluetooth security by using enhancement of RSA algorithm.
- Analysis & compare the RSA algorithm vs enhanced RSA algorithm
- Analysis & compare the existing Bluetooth security vs enhanced RSA security.

1.4 Organization of the Paper

This paper is organized into five chapters. Chapter one contains some introductory texts on Bluetooth security in cryptography, background and present state of the problem, motivation of the research and objectives. Chapter two contains brief discussion on Bluetooth security and RSA algorithms. Chapter three deals with the overall process of working, proposed system. In chapter four, the simulations of work and result is explained. Chapter five concludes our overall work.

After that there is a list of books and papers which assist throughout the thesis work. Last of all, the Appendix contains the code of simulation work.

Chapter 2

Literature Review

2.1 Bluetooth

The Bluetooth technology enables devices equipped with Bluetooth interfaces to in short range and wireless connect with each other and form an Ad Hoc network [1]. Each unit can simultaneously communicate with up to seven other units per piconet, i.e. a small network that only contains seven members. A unit can also belong to other piconets.

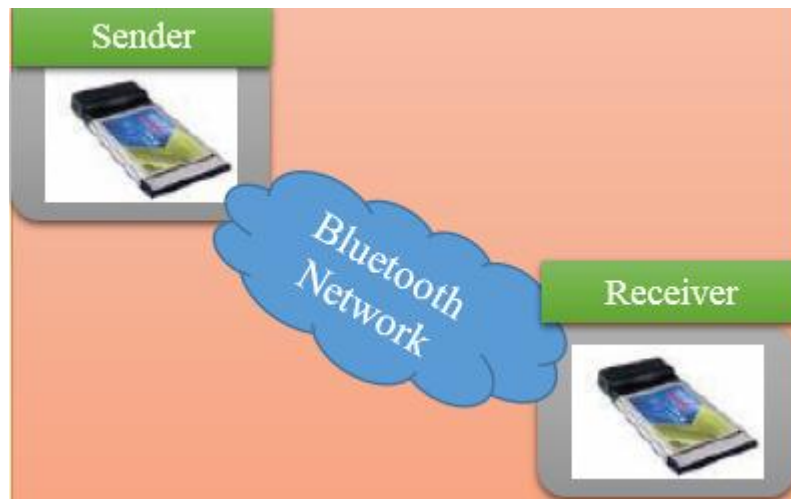


Figure 2.1: Bluetooth Network Model

It consists of patterns of raised dots arranged in cells of up to 6 dots in a 3 x 2 configuration. Each cell represents a letter, numeral or punctuation mark. Braille characters are small rectangular blocks called cells that contain tiny palpable bumps called raised dots. The mappings of Braille system vary from language to language.

2.1.1 Radio Interface

The initial requirements that was defined for the Bluetooth interface were:

- It must operate world-wide.
- The connection must support voice and data.

- The radio transceiver must be small and run on low power.

To be able to operate world-wide the frequency band must be license-free and open to any radio system. The only frequency band that satisfies these requirements is the 2.45 GHz band or the Industrial-Scientific-Medical (ISM) band.

2.1.2 Physical Link

There are two types of physical links defined, which support multimedia applications:

- Synchronous connection-oriented (SCO) link
- Asynchronous connectionless (ACL) link

When transmitting voice, the SCO link is typically used. SCO supports symmetrical, circuit-switched, point-to-point connections.

ACL link handles typically peak-flow transmission and supports symmetrical or asymmetrical, packet-switched, point-to-multipoint connections.

2.1.3 Interference

When choosing an open frequency band, the radio systems must cope with several uncontrolled sources of interference. The following optimizations have been done to prevent those sources of interference:

- Frequencies hopping with high hopping rate and short packet lengths.
- Forward error code.
- Using automatic-retransmission-query (ARQ) to achieve short delays and shorten transmission ratio, i.e. if an error is detected the receiver at once indicates that in the next packet.
- Never retransmit voice. Instead, a robust voice-encoding scheme is used error code.

2.1.4 Scatter Net

As discussed above, units within range can establish Ad Hoc connections between them. If one member of this newly created piconet also belongs to another piconet an overlapping pattern will occur. If all members of the two piconets share the same hop-channel that will result in a drastically decrease of throughput in the network. Therefore, another solution was adopted – scatter net. This solution simply states that only devices that wants to communicate with other piconet members should be using the same hop channel. In spite of this approach collision do occur, because the piconets hop independently.

2.2 Cryptology

Cryptology is formed from the two Greek words –Kryptos which means secret and Logos which means word. It generally refers to the study of secret communication [2]. Cryptology is an area which is comprised of cryptography and cryptanalysis [9].

2.2.1 Plain Text

An original intelligible message which is fed as input before being transformed is called plain text [9].

2.2.2 Cipher Text

The coded or scrambled message after transformation produced as an output is known as the cipher text. It depends upon the plain text and the key used for encryption [9]

2.2.3 Encryption

The process of converting plain text to cipher text is known as encryption. It is also known as enciphering [9]. Any algorithm which encrypts the data is known as encryption algorithm. The sender uses the encryption algorithm [2].

2.2.4 Decryption

Restoring the plain text from the cipher text is known as deciphering or decryption [9]. Any algorithm which decrypts the data is known as decryption algorithm. The receiver uses the decryption algorithm.

2.2.5 Key

A key is a number (or a set of numbers) that the cipher as an algorithm operates on [2]. To encrypt a message, an encryption algorithm, an encryption key and the plain text are needed. These create the cipher text. To decrypt a message, a decryption algorithm, a decryption key and the cipher text are needed. These reveal the original plain text.

2.3 Cryptanalysis

Techniques that are used for deciphering a message without the prior knowledge of the enciphering details fall into the area of cryptanalysis [9].

The cryptographic algorithms are divided into two groups: symmetric key cryptography algorithms and asymmetric key cryptography algorithms.

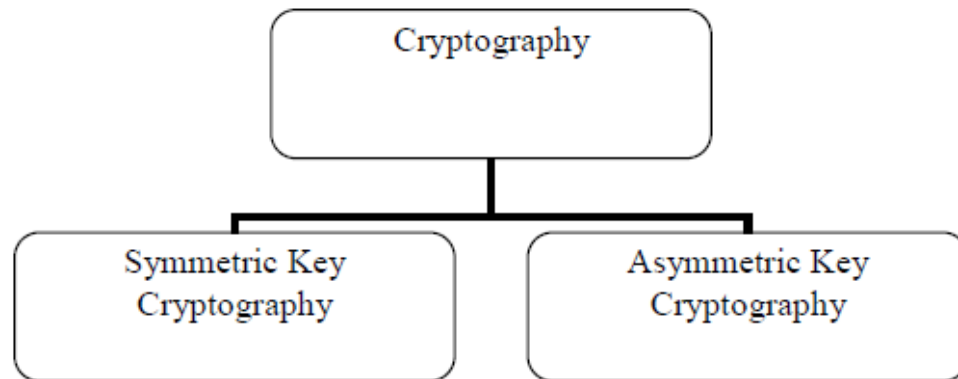


Figure 2.2: Categories of Cryptography [2]

2.3.1 Symmetric Key Cryptography

In a symmetric key cryptography, the same key is used by both the parties. The sender uses this key and encryption algorithm to encrypt the data; the receiver uses the same key and the corresponding decryption algorithm to decrypt the data. Symmetric key cryptography are DES (Data Encryption Standard), IDEA (International Data Encryption Algorithm), Blowfish, RC5 (Rivest Cipher 5), AES (Advanced Encryption Standard) etc.

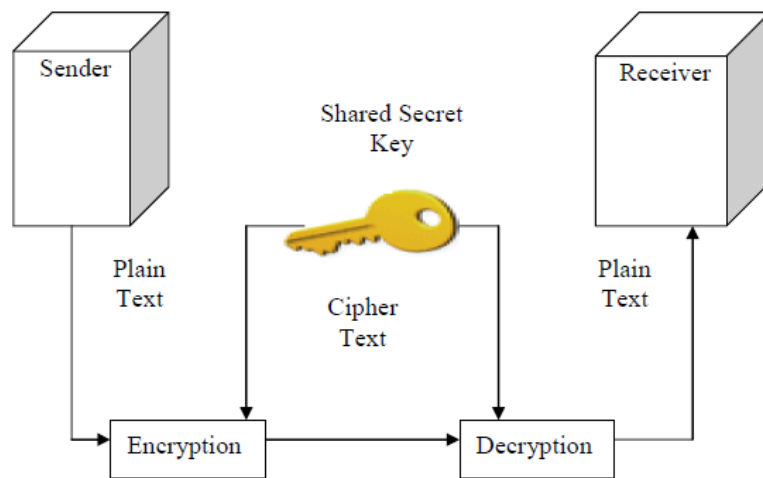


Figure 2.3: Symmetric - Key Cryptography [2]

2.3.2 Asymmetric Key Cryptography

RSA is the kind of asymmetric key cryptographic algorithm. The asymmetric process can be illustrated in the following fig. 2.4 below.

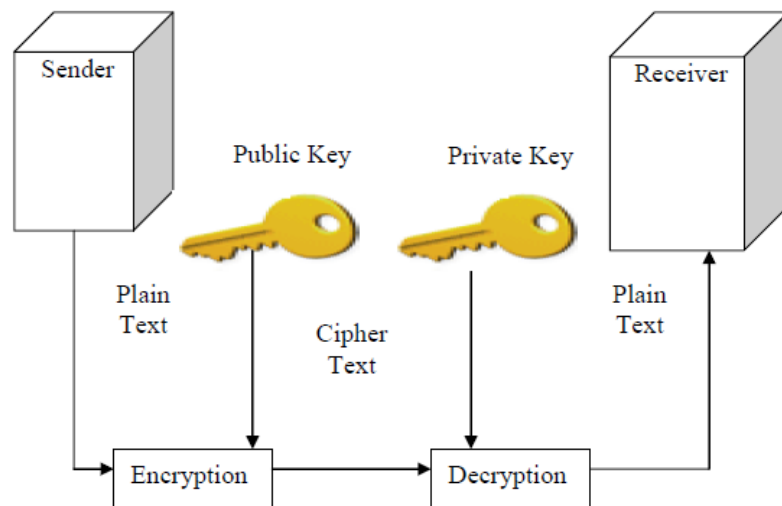


Figure 2.4: Asymmetric Key Cryptography [2]

2.3.3 Advantages of Public-Key Cryptosystems

Public-Key Cryptosystem was developed to address mainly two key issues:

- Key distribution: how to have secure communications in general without having to trust a KDC with one's key.
- Digital signatures: how to verify a message that comes intact from the claimed sender. It was invented by Diffie & Hellman at Stanford University in 1976.

The advantages of public-key cryptosystems can be listed as below [9]:

- a) Increased security and convenience

It is the chief advantage of the public-key cryptosystems. Private keys are for no reason needed to be transmitted or revealed to anyone. By contrast, in a secret-key system, the secret keys must be conveyed (either manually or through a communication channel) as the same key is used for encryption and decryption.

- b) Offer digital signatures that cannot be repudiated

Authentication via secret-key systems requires the sharing of secret and sometimes requires the confidence from a third party as well. As a consequence, a sender can deny a previously authenticated message by declaring that the shared secret was in some way compromised by one of the parties sharing the secret.

2.4 Fundamental Theorem of Arithmetic

Fundamental theorem of arithmetic [5] [8] in asymmetric or public key cryptography, there are two keys: a private key and a public key. The private key is kept by the receiver. The public key is announced to the public. It is also known as public-key cryptography [2]. Some of the cryptographic technologies used in asymmetric key cryptography are RSA (Rivest, Shamir, Adleman), DH (Diffie-Hellman Key Arrangement Algorithm), ECDH (Elliptic Curve Diffie-Hellman Key Arrangement Algorithm), RPK (Raiké Public Key), ElGamal, IES (Integrated Encryption Scheme), CEILIDH etc.

2.4.1 Mathematics Involved in RSA

Simple mathematics concepts like prime numbers, modular exponentiation, Euler's theorem had a dramatic impact on computer security. Cryptography is considered not only a part of the branch of computer science, but also a branch of mathematics. The strength of the RSA algorithm lies in the mathematics that is involved in it [8]. Before proceeding with the RSA algorithm, there is a need to know the mathematics on which RSA is based

Every positive integer $n > 1$ can be expressed as a product of primes; this representation is unique, apart from the order in which the factors occur. It can be written in canonical form as

$n = p_1^{a_1} p_2^{a_2} p_3^{a_3} \dots p_t^{a_t}$ where, for $i = 1, 2, \dots, t$, each a_i is a positive integer and each P_i is a prime number, with $P_1 < P_2 < P_3 < \dots < P_t$ Example: $91 = 7 \times 13$, $11011 = 7 \times (11)^2 \times 13$

Example: $91 = 7 \times 13$,

$11011 = 7 \times (11)^2 \times 13$

2.4.2 Fermat's Little Theorem

For Fermat's Little Theorem [5][8] if p is a prime number and a is a positive integer, then $a^p \equiv a \pmod{p}$ If p is a prime number and p is not a divisor of a , then $a^{p-1} \equiv 1 \pmod{p}$

Example: $2^{11} - 1 \pmod{11} = 2^{10} \pmod{11}$

$= 1024 \pmod{11}$

$= 1$

2.4.3 Euler's Totient Function

For Euler's Totient [8][9] function $\phi(n)$ denotes the number of positive integers not exceeding n that are relatively prime to n . It is also known as Euler phi-function. For a prime number p , the Euler Totient function would be:

$\Phi(p) = p-1$

For two prime numbers p and q , $n = p \times q$ then

$$\Phi(n) = (p-1) \times (q-1)$$

Example: $\phi(30) = \phi(2 \times 3 \times 5)$

$$= (2-1) \times (3-1) \times (5-1)$$

$$= 1 \times 2 \times 4$$

$$= 8$$

2.4.4 Property of Modular Arithmetic

Property of Modular Arithmetic [9] for finding the modulus of an integer number raised to an integer power when divided by n is involved both in encryption and in decryption.

So we make use of the property:

$$(a \bmod n) \times (b \bmod n) = (a \times b) \bmod n \text{ provided } a > n \text{ and } b > n$$

2.4.5 Linear Congruence

An equation of the form $ax \equiv b \pmod{n}$ is called a linear congruence. The linear congruence has a solution if and only if d is a divisor of b , where $d = \gcd(a, n)$. If d is a divisor of b , then it has d mutually incongruent solutions modulo n [5].

Example: Consider the linear congruence $18x \equiv 30 \pmod{42}$ $\gcd(18, 42) = 6$ and 6 divides 30

According to linear congruence theorem, there are exactly 6 solutions which are incongruent to modulo 42

The six solutions are $x \equiv 4 + (42/6)t$

$$\equiv 4 + 7t \pmod{42},$$

Plainly enumerated as $x \equiv 4, 11, 18, 25, 32, 39 \pmod{42}$

2.4.6 Property of Odd Integer

Any positive odd integer $n \geq 3$ can be expressed $n-1 = 2^k q$ where $k > 0$, q is odd. $n-1$ is an even integer. Then, divide $n-1$ by 2 until the result is an odd number q , for a total of k divisions [9].

For example: $n = 15$ then $15-1 = 14$

$$= 2^1 * 7$$

2.4.7 Properties of Prime Number

The two properties of prime number are stated below [5]:

- i. If p is prime and a is a positive integer greater than p , then $a^2 \bmod p = 1$ if and only if either $a \bmod p = 1$ or $a \bmod p = -1$
- ii. Let p be a prime number greater than 2. It can be written as $p - 1 = 2kq$ with $k > 0$, q odd. Let a be any integer in the range $1 < a < p - 1$. Then one

of the following conditions is true:

- a) a^q is congruent to 1 modulo p . That is, $a^q \bmod p = 1$, or equivalently, $a^q \equiv 1 \pmod{p}$
- b) One of the numbers $a^q, a^{2q}, a^{4q}, \dots, a^{2^{k-1}q}$ is congruent to -1 modulo p . That is, there is some number $2^{j-1}q \bmod p = -1 \bmod p = p - 1$ or equivalently $a^{2^{j-1}q} \equiv -1 \pmod{p}$

2.5 RSA Algorithm

There are three main operations which are to be performed in the algorithm. The three operations are: key generation, encryption and decryption.

a) Key Generation

RSA comprises of two keys –public key and private key. The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted by using the private key. e is released as the public key exponent and d is kept as the private key exponent. The steps for key generation are explained below [11]:

- i. Select p and q where $p \neq q$ and both p and q are prime numbers.
- ii. Determine $n = p \times q$
- iii. Compute $\phi(n) = (p-1) \times (q-1)$
- iv. Choose an integer e such that $\gcd(\phi(n), e) = 1$ and $1 < e < \phi(n)$
- v. Evaluate d as $d \equiv e^{-1} \pmod{\phi(n)}$
- vi. Public Key (PU) = $\{e, n\}$

Private Key (PR) = $\{d, n\}$

b) Encryption

The steps for encryption of message in order to get the cipher-text are explained below [11]:

- i. Obtain a plain text M such that $M < n$.

- ii. Compute the cipher text as $C = M^e \bmod n$

c) Decryption

The steps for decryption of cipher-text in order to get the original message are explained below [11]:

- i. Get the cipher text C.
- ii. Calculate the plain text as $M = C^d \bmod n$

Take an example where the message is 88 which needs to be encrypted and then decrypted using RSA algorithm. The example is explained by using the three operations as follows

a) Key Generation

According to the steps, the private and public keys are to be generated as:

- i. Select primes: $p = 17$ & $q = 11$
- ii. Compute $n = p \times q = 17 \times 11 = 187$
- iii. Compute $\phi(n) = (p-1) \times (q-1) = 16 \times 10 = 160$
- iv. Select e : $\gcd(e, 160) = 1$. Choose $e = 7$
- v. Determine d : $d \times e = 1 \pmod{160}$ and $d < 160$. Value of $d = 23$
Since $23 \times 7 = 161 = 10 \times 160 + 1$
- vi. Publish public key $PU = \{7, 187\}$
- vii. Keep secret private key $PR = \{23, 187\}$

Sample RSA encryption/decryption:

The given message M is 88. The encryption and decryption operation on the message is performed as:

b) Encryption

$$C = 88^7 \bmod 187 = 11$$

c) Decryption

$$M = 11^{23} \bmod 187 = 88$$

2.5.1 Reason for RSA Algorithm

The main benefit of RSA comes from the information that while it is easy to multiply two huge prime numbers collectively to get the product, it is computationally hard to do the reverse.

The RSA system is presently been used in a wide variety of products, platforms, and industries throughout the world. It has been observed in many commercial software products and is planned to be in many more. The RSA algorithm is put together into the current operating systems by Microsoft, Apple, Sun, and Novell. In hardware, the RSA algorithm can be noticed in secure telephones, on Ethernet network cards, and on smart cards. In addition, the algorithm is even included in all of the major protocols for safe Internet communications, including S/MIME, SSL and S/WAN. It is also used internally in many of the institutions including branches of the U.S. government, major corporations, national laboratories, and universities [11].

In today's, technology scenario using the RSA algorithm is licensed by over 700 companies. The ISO (International Standards Organization) 9796 standard records RSA as a compatible cryptographic algorithm, as does the ITU-T X.509 security standard. The RSA system is an element of the Society for Worldwide Interbank Financial Telecommunications (SWIFT) standard, the French financial industry's ETEBAC 5 standard, the ANSI X9.31 r DSA standard and the X9.44 draft standard for the U.S. banking industry. The Australian key management standard, AS2805.6.5.3, also lists the RSA system. The RSA algorithm is found in Internet standards and projected

Protocols including S/MIME, IPsec and TLS as well as in the PKCS standard for the software industry. The OSI Implementers Workshop (OIW) has issued implementers agreements referring to PKCS, which includes RSA [11].

2.5.2 Improvements of RSA

In order to improve the original RSA, different investigations were carried out on it. The researches lead to many variants which were proposed to enhance the original RSA. These variants are explained one by one below:

2.5.2.1 CRT –RSA

The key generation and encryption algorithm is identical to that of the original RSA, except that the private key is the tuple (d_p, d_q, p, q) where

$$d_p = d \bmod (p-1) \quad d_q = d \bmod (q-1)$$

Obtain a cipher-text c subset of \mathbb{Z}_N . The decipher can first compute

$$m_p = c^{d_p} \bmod p \quad \text{and} \quad m_q = c^{d_q} \bmod q$$

Next, using the Chinese Remainder Theorem (CRT) in order to obtain

$$m = (m_p \cdot q \cdot (q^{-1} \bmod p) + m_q \cdot p \cdot (p^{-1} \bmod q)) \bmod (p \times q) = c^d \bmod N, \text{ due to}$$

$$m = m_p \bmod p \quad \text{and} \quad m = m_q \bmod q$$

2.5.2.2 Rebalanced RSA –CRT

The rebalanced RSA-CRT uses two prime numbers with $n/2$ -bit size. The difference in the algorithm lies in the decryption. The algorithm can be described as:

a) Key Generation Operation [16] [17]

- i. At random select any two large primes p and q , each of which is $n/2$ -bit long such that $\gcd(p-1, q-1) = 2$
- ii. Compute $N = p \times q$ and $\phi(N) = (p-1) \times (q-1)$
- iii. After that, randomly choose any two 160-bit integers r_1 and r_2 such that $\gcd(r_1, p-1) = 1$ and $\gcd(r_2, q-1) = 1$
- iv. Then obtain an integer d such that $d = r_1 \bmod (p-1)$ and $d = r_2 \bmod (q-1)$
- v. Ultimately compute $e = d^{-1} \bmod \phi(N)$
- vi. The public key is (e, N) and the private key is (r_1, r_2, p, q)

b) Encryption Operation

The encryption algorithm is similar to the original RSA.

c) Decryption Operation [16] [17]

The decryption process to decrypt a cipher-text c with the private key (r_1, r_2, p, q) is carried out as shown in the subsequent steps:

- i. Compute $m_1 = c^{r_1} \bmod p$ and $m_2 = c^{r_2} \bmod q$
- ii. Using the CRT m can be obtained as $m = c^d \bmod N$, due to the fact that $m = m_1 \bmod p$ and $m = m_2 \bmod q$

2.5.2.3 Dual RSA

It is a variant of RSA in which there are two different instances of RSA which share the same public and private key exponents. The public key is (e, N_1, N_2) and the private key is (d, p_1, p_2, q_1, q_2) where e and d satisfy the relation

$(e \times d) \equiv 1 \pmod{\phi(N_1)}$ and $(e \times d) \equiv 1 \pmod{\phi(N_2)}$ There exist two relations:

$e \times d = 1 + k_1 \times \phi(N_1)$ and $e \times d = 1 + k_2 \times \phi(N_2)$

The basic idea is to construct k_1, k_2, k_3 such that $k_2 \times k_3 = (p_1-1) \times (q_1-1)$ and

$k_1 \times k_3 = (p_2-1) \times (q_2-1)$

2.6 Security of RSA

There are four possible methodologies to attack the RSA algorithm. The four approaches are [9]

2.6.1 Brute Force

A brute force attack or exhaustive key search is an approach that can be used against any encrypted data by an attacker who is not capable of taking benefit from any kind of flaw that exist in an encryption system which would make his/her task easier. It comprises of methodically examining all the possible keys in anticipation of the exact key to be found. In the worst situation, it would include traversing through the whole search space [9].

2.6.2 Mathematical Attack

There are three different approaches to hit RSA mathematically. They are:

- i. Dividing n into its two prime factors. It would facilitate estimation of $\phi(n) = (p - 1) \times (q - 1)$ which would sequentially allow to resolve

$$d \equiv e^{-1} \pmod{\phi(n)}.$$

- ii. Finding out $\phi(n)$ straightforwardly devoid of discovering p and q . Yet again, this would permit the resolving of $d \equiv e^{-1} \pmod{\phi(n)}$.
- iii. Resolve d directly without finding out $\phi(n)$ first.
- iv. Determining d given e and n seems to be as time-consuming as the factoring problem by means of current well-known algorithms. So, factoring performance can be used as a point of reference for evaluating the security of RSA. The intensity of effort is measured in MIPS-years: a million-instructions-per-second processor running for one year which is about 3×10^{13} instructions executed. A 1 GHz Pentium is about a 250-MIPS -years machine [9].

N	Year	Algorithm
RSA-120 (399 bits)	1993	MQPS
RSA-129 (429 bits)	1994	MPQS
RSA-130 (432 bits)	1996	NFS
RSA-140 (466 bits)	1999	NFS
RSA-155 (512 bits)	1999	NFS
RSA-160 (532 bits)	2003	NFS
RSA-200 (665 bits)	2005	NFS
RSA-768	2010	NFS
RSA-1024	2011	NFS
RSA-2048	2030	??

Table 2.2: Year wise development of RSA algorithm [3]

2.6.3 Timing Attacks

Kocher [7] gives an idea for timing attack known as Kocher timing attack that is theoretically feasible where it presents a lot of data suggesting with its possibility. There is no evidence that Kocher actually performed the attack himself. A known practical attack [8] was developed that uses the same general idea as Kocher's work, but attempts to simplify both the timing and the calculations performed. Computer Scientists were in

fact unable to implement Kocher's idea that attacking the entire loop, they decided to attack the multiplication. Using a cryptographic library developed for the CASCADE smart card, they attacked the decryption algorithm shown below,

Where k is the private key and m is the cipher text. Here the algorithm for practical timing attack

```
x = m
for i = n-2 downto 0
  x = x^2
  if (ki == 1) then
    x = x * m
  endfor
return x
```

The modular multiplication and squaring performed in this algorithm are done using the Montgomery method that has small inconsistency in the multiplication method. The method performs an extra subtraction when the intermediary result of the multiplication is greater than the value of the modulus. Thus the cipher texts can be separated into two groups, those that require the extra subtraction during Montgomery multiplication (C1) and those that do not (C2).

In experimental results researchers found that when RSA is allowed to operate as it should the extra reduction is only performed only 17% of the time. They were able to increase this probability to numbers as high as 50% by fixing the modulus and one of the factors, however these modifications would not be performed in practice and thus compromises the effectiveness of the attack [8]. Looking back to the algorithm we can see that the multiplication step is performed only if bit i of the private key is a '1'. Using this knowledge, and the inconsistencies of the Montgomery multiplication we can see that when the private key is a '1' there should be a difference between the execution time of cipher texts in group C1 and the execution times of the ciphertexts in group C2. Whereas if bit i of the private key is a '0' we would expect to see no timing difference between the two groups. Although in theory the algorithm should run in constant time, in reality this is certainly not the case. This being the case we now have a difficult time identifying not only whether or not a reduction was performed, but also while running

the actual attack we must decide how different the timing of group C1 must be from group C2 in order to assign the bit i of the private key to be a '1'. The second problem is inherent to the Montgomery multiplication and impossible to correct without modifying components to the RSA algorithm. The attack first simulates a guess of '1', where the multiplication is performed, dividing the results of each ciphertext into two groups, M1 and M2, just with the previous multiplication attack. This timing is then followed up by a timing of a guess of '0', separating the timing results into groups, M3 and M4. Finally, after timing the actual private key execution and gathering the same two groups we compare these to the M1 and M2 set as well as the M3 and M4 set. While very excited about the new and seemingly more successful timing attack [8], we realized that the issue of accurate timing results had not gone away. Researchers suggest that the attack is based on a variation in timing of 422 clock cycles out of 7,400,000, so it was clear to us that the accuracy of measurements was still crucial to the success of the attack. So we decided that prior to any attempts at implementing the attack we should first secure accurate timing results.

2.6.4 Chosen Cipher text Attack

The fundamental RSA algorithm is exposed to a chosen cipher-text attack (CCA). CCA is described as an attack in which opponent picks up a number of cipher texts and is then given the equivalent plain texts which are decrypted with the target's private key. Hence, the adverse means of the target's public key and the having it decrypted by means of the private key. Evidently, this offers the opponent with no new information. As an alternative, the adversary takes advantage of properties of RSA and opts for blocks of data which are processed using the target's private key in order cryptanalysis.

An uncomplicated example of CCA in opposition to RSA takes benefit from the following property of RSA:

$$E(PU, M1) \times E(PU, M2) = E(PU, [M1 \times M2])$$

$C = Me \bmod n$ can be decrypted using a CCA as follows [13]:

- i. Work out $X = (C \times 2e) \bmod n$.
- ii. Given X as a chosen cipher-text and obtain back $Y = Xd \bmod n$.

It is found that X can be written as

$$\begin{aligned} X &= (C \bmod n) \times (2e \bmod n) \\ &= (Me \bmod n) \times (2e \bmod n) \\ &= (2M) e \bmod n \end{aligned}$$

Therefore, $Y = (2M) \bmod n$. From this, M can be deduced.

2.6.5 Winners attack on RSA algorithm

1) Algorithm Generalized winner Attack

a) Input: (N, e), where $N = p \cdot q$ and $ex + y = 0 \bmod \phi(N)$ for some

Unknown, $0 < x < \frac{x N^{\frac{1}{2}}}{3}$ and $|y| \leq c N^{\frac{3}{2}ex}$

- Compute the continued fraction function of $\frac{e}{N}$
- For every convergent $\frac{K}{x}$ of the expansion
 Compute $S = N+1 - \frac{ex}{K}$, $t = \sqrt{S^2 - 4N}$ and
 $p = \frac{1}{2}(s+t)$
 Apply coppersmith algorithm to the candidate
 $p' + (2k+1) N^{\frac{1}{2}}$
 For $k = -3, -2 \dots 2$. If coppersmith algorithm outputs the factorization of N then stop.

b) Output: p, q

2.7 Coppersmith theorem

Let $N = pq$ be an RSA-modulus, where p and q are of the same bit-size. Suppose we are given an approximation of p with additive error at most $N^{\frac{1}{2}}$. Then N can be factored in time polynomial in $\log N$. We are now able to state our main theorem. Here we consider the normal RSA case

Where $p - q = \Omega\sqrt{N}$

Chapter 3

Methodology

This paper describes the security of Bluetooth network and propose a new algorithm to secure the Bluetooth in order to reduce the weakness of the RSA algorithm. Here RSA is modified by using public key and private key, then the algorithm perform the task of security. This modification is advantageous because the fact the public and private keys in the public key system are related in such a way that only the public key can be used to encrypt the messages and only the corresponding private key can be used to decrypt them. Moreover, it is virtually impossible to deduce the private key if you know the public key.

3.1 System Model

Our proposed system can help the visually impaired people by providing a way of communication with others by using smart phones and Braille method. We represents above six options by six dots like Braille method.

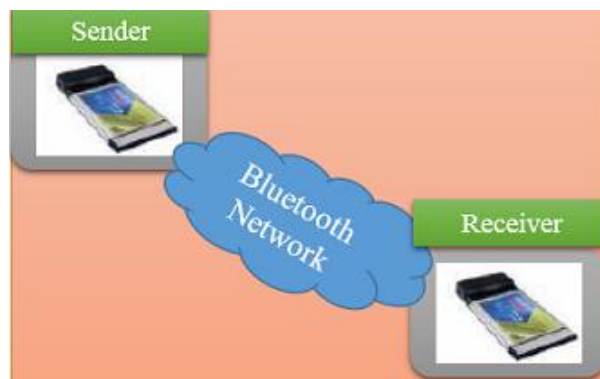


Figure 3.1: Existing Bluetooth data transfer policy.

3.2 Proposed System

In this thesis paper, a system is developed to improve the security and the performance of the Bluetooth network. Figure 3.3 shows the proposed model system.



Figure 3.2: Proposed communication system for Bluetooth communication.

3.3 Bluetooth security improvements through RSA algorithm

The improvements of Bluetooth security systems through RSA are given below

3.3.1 Sender

At first sender encrypt the original message with the help of public key that is known to all and get the cipher text 1, then again the encrypted message i.e. cipher text 1 by RSA algorithm then cipher text 2 can be found. After this the sender send this data to the network for receiver. Figure 3.3.1 is the illustration of encryption improvements for Bluetooth security.

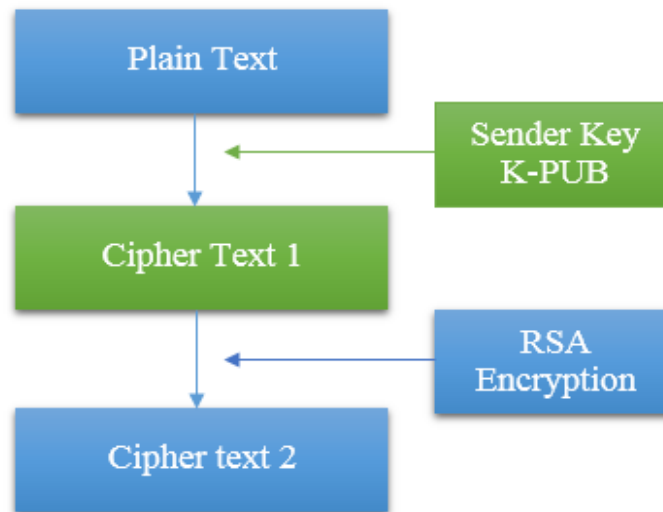


Figure 3.3: Bluetooth Encryption Improvements for sender

3.3.2 Receiver

After receiving the encrypted message the receiver first decrypt the message by RSA algorithm and can find the cipher text 1. Then receiver again decrypt the cipher text 1 by private key as asymmetric process. The figure is the illustration of decryption improvements of Bluetooth security.

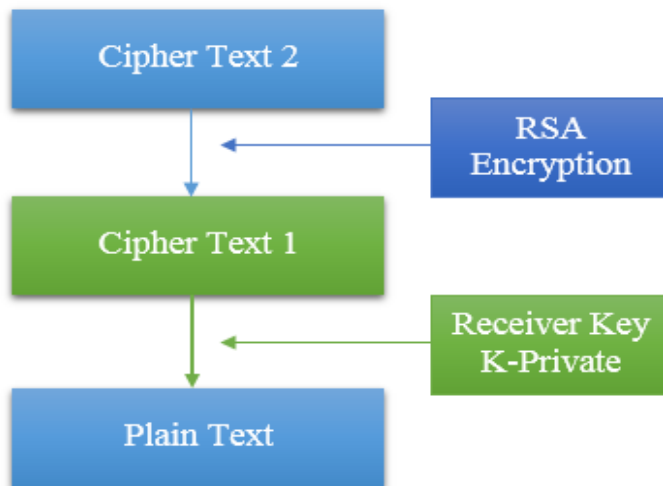


Figure 3.2: Bluetooth Decryption Improvements for receiver

3.4 Verification Process

- Encrypt the original messages using sender private key and create ciphertext1.
- Calculate RSA for the ciphertext2 with RSA public key.
- Ciphertext2 of key decrypts RSA with the help of corresponding RSA private key to find out ciphertext1 for receiver.
- Compute the receiver's decryption to find out the original messages using the receiver's private key.

Chapter 4

Simulations and Results

The task of our algorithm performs by several functions. Suppose sender would like to send a message to the receiver.

4.1 Sender

Original Message: “RSA Bluetooth”

The sender follows the following steps to encrypt the message

Step 1: At first sender generate public and private key. Sender encrypt the message with the help of public key and creates a cipher text that is illustrated in the figure 4.1

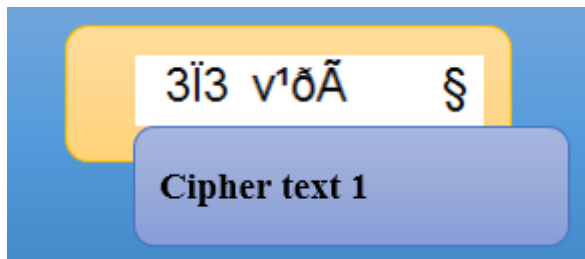


Figure 4.1: Encrypted by Sender public key.

Step 2: Then the message is encrypted by RSA algorithm and another cipher text is found that is depicted in the figure 4.2

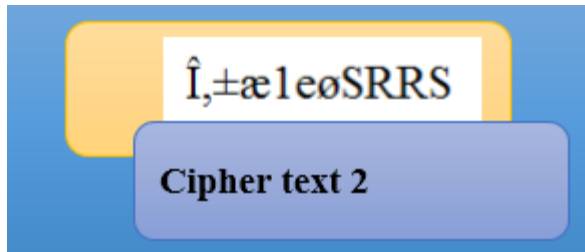


Figure 4.2: Encrypted by RSA.

4.2 Receiver

Step 1: After getting the cipher text from the sender receiver first decrypt the message by using RSA algorithm. The output message of the RSA decryption is depicted in figure 4.3 below

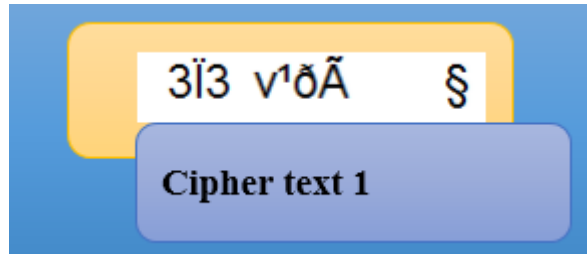


Figure 4.3: Decrypted by RSA.

Step 2: Then the receiver again decrypt the RSA decrypted cipher text by using private key. If the key is true then the receiver found the original message “RSA Bluetooth” if not true then fail to decrypt.

The GUI representation of the sender portion is depicted in figure 4.4

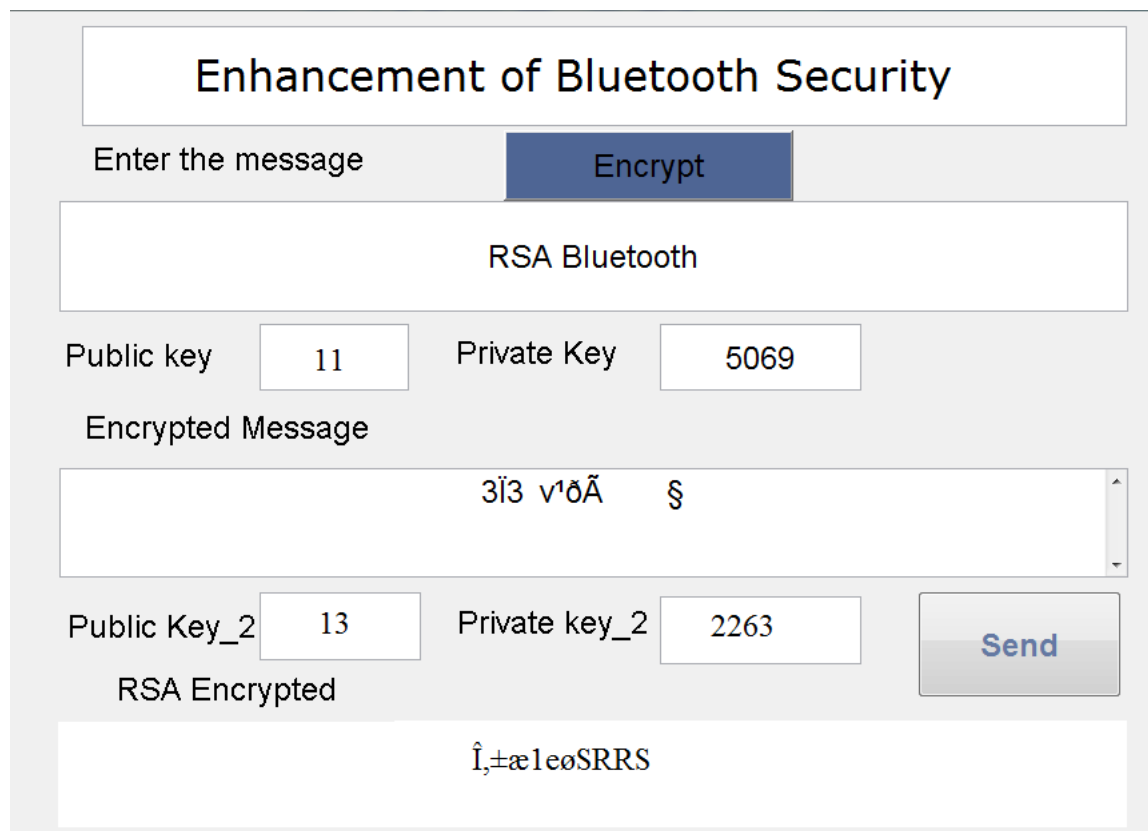


Figure 4.4: GUI representation of Bluetooth Security

4.3 Performance Evaluation

In this thesis work, several kinds of attack of RSA that is described on chapter 2 has removed by the propose system. These are

4.3.1 Winners attack removing

Consider a plaintext “RSA Bluetooth” for the winner’s attack.

Now from the simulation

Sender public key= 7

Sender private key for receiver = 5069

First encrypt sender public key and found the ciphertext1 is illustrated in figure 4.5.

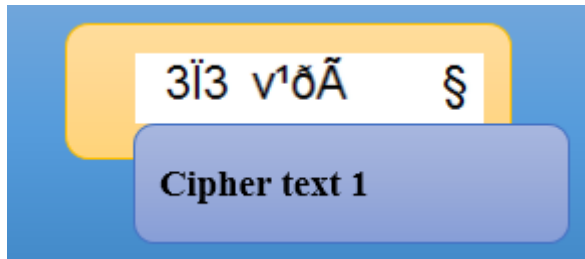


Figure 4.5: Encrypted by sender public key

This ciphertext1 goes to RSA algorithm for RSA encryption.

For RSA

$p = 79$

$q = 53$

Public key = 13

Private Key = 2263

Now the ciphertext2 from ciphertext1 using RSA algorithm is illustrated in figure 4.6.

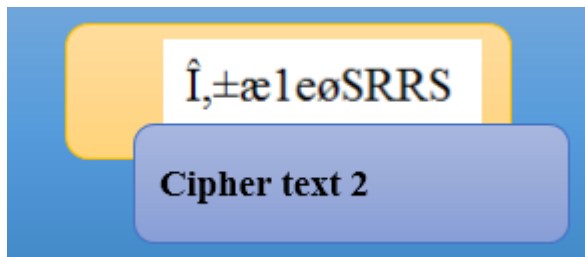


Figure 4.6: Encrypted by RSA algorithm

From the generalize winners attack algorithm the output of the attack found

$p=79$

$q=53$.

For a successful winner's attack. The interesting point is that the attacker didn't know, this attack is successful if the data is not readable. For this the attacker will find the ciphertext1 that is illustrated in figure 4.7.

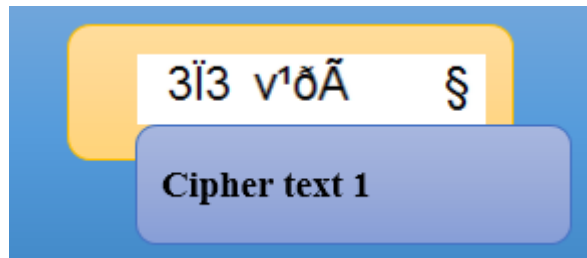


Figure 4.7: Final Result of successful winner attack

So attacker decide that the attack is not successful.

4.3.2 Timing attack removing

From the successful timing attack, it first separate the ciphertext2 into two parts hence it found that illustrated in figure 4.8.

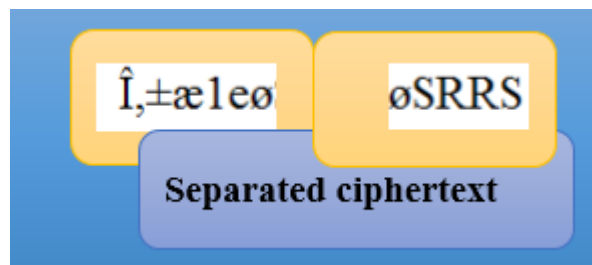


Figure 4.8: First successful approach to timing attack

After this simulates a guess of '1', where the multiplication is performed i.e. $79 \times 53 = 4187$.

Here execution time of those two cipher text is different. Dividing the results of each cipher text into two groups, M1 and M2. Just with the previous multiplication attack this timing is then followed up by a timing of a guess of '0', separating the timing results into groups, M3 and M4. Here it is found that the execution time of both cipher text is equal. Finally, after timing the actual private key execution and gathering the same two groups we compare these to the M1 and M2 set as well as the M3 and M4 set. At last the successful timing attack result is ciphertext1 that is illustrated in figure 4.9

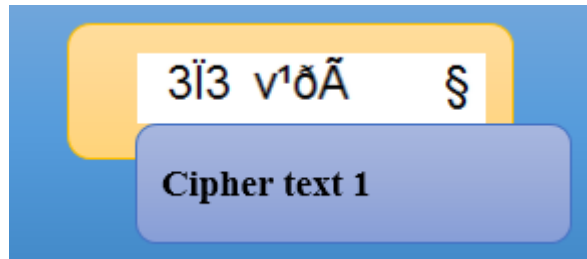


Figure 4.9: Final result of successful timing attack

Here the attacker don't find the original messages. So the attacker guess that the attack is not successful.

4.4 Comparison of Securities

Securities in various aspects are given below

4.4.1 RSA versus proposed algorithm securities analysis

From the analysis of the proposed and the RSA algorithm here found the table that compares the RSA versus proposed system analysis. From the analysis time complexity of the proposed system is greater but security is the greater in proposed system. RSA has timing, winner attack but proposed system overcome this limitations. The time complexity of RSA versus proposed system is drawn in figure 4.10

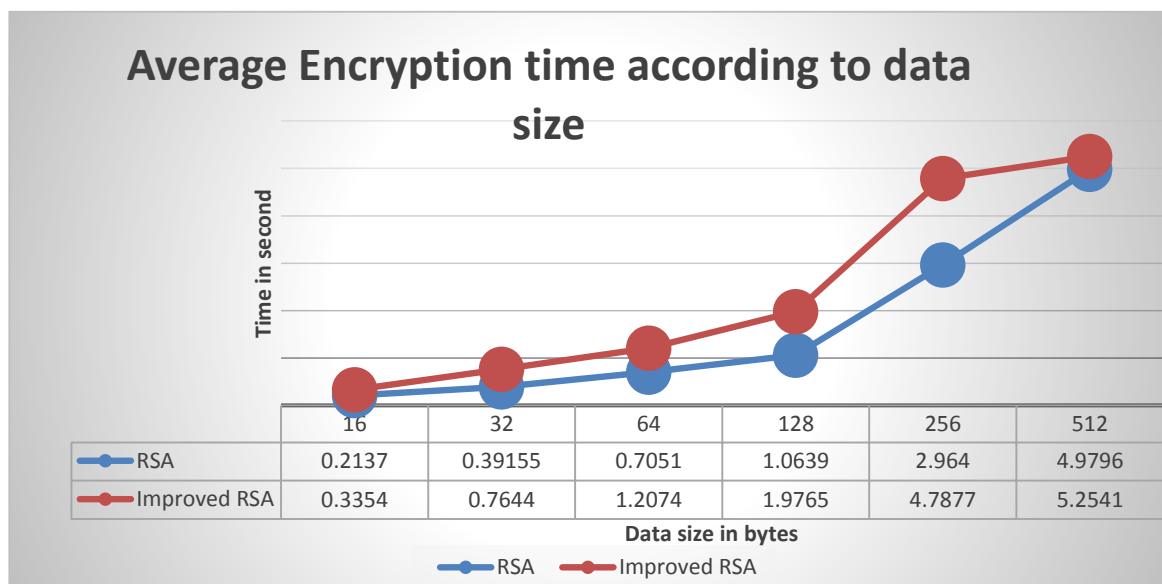


Figure 4.10: RSA versus proposed system time complexity

4.4.2 Latest Bluetooth security system versus proposed system

According to the existing security system for the Bluetooth network security the analysis result shows that the proposed system is superior to any other existing Bluetooth security system. The comparison of complexity according to the time and data size graph between existing latest system versus proposed Bluetooth system are drawn in figure 4.11 and table 4.1 shows the summary of those system and proposed system.

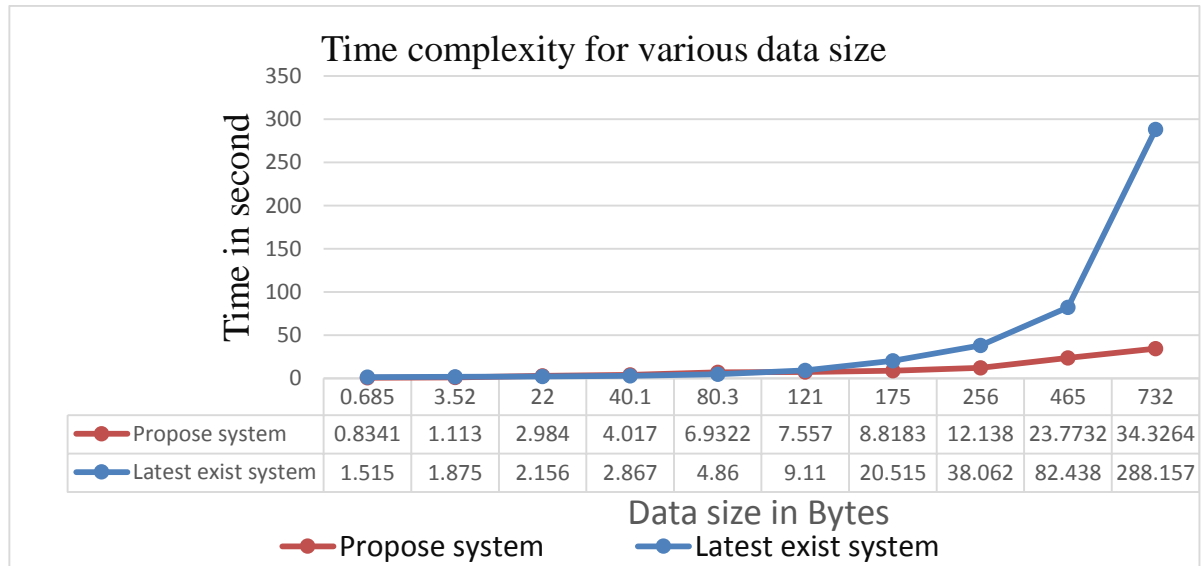


Figure 4.11 latest existing versus proposed system

4.4.3 Bluetooth security Analysis through all systems

Bluetooth Security System	Security Mechanism	Types & Size of keys	Advantage	Limitation	Countermeasure
Existing System	-E0 stream cipher - Challenge response technique	Link key, 128-bit	Low complexity due to absence of cryptographic technique	-Low credibility of PIN -Address spoofing -Non link key cheat because same keys is use for different parties.	-Increase the PIN code length.[6] -Need user authentication and application level security
System proposed by Li Juan Chen Bin,	Use single DES algorithm	Symmetric key 56-bit	Fast Encryption/Decryption	-Key distribution problem -More vulnerable to attack because of small key size	-Use asymmetric cryptographic algorithm -Use Public and

Li Kun [2]					Private key pair.
System proposed by Wuling Ren, Zhejinag Gongshang [1]	Use DES & RSA algorithm	Symmetric and Asymmetric key, 56-bit Symmetric key	Key distribution Problem solved	<ul style="list-style-type: none"> - Brute force attack is possible on small key size(56-bit only) - No any process proposed for verifying the integrity of message 	<ul style="list-style-type: none"> -Increase the size of the key like 112 bit and 168 bit key. -Proposed a system that include integrity check also.
System proposed by Trishna V Kapoor [7]	Use triple DES (with 2-keys), RSA, MD5	Symmetric and Asymmetric key, 112-bit Symmetric key	<ul style="list-style-type: none"> -Key distribution problem solved -Brute force attack problem is solved 	<ul style="list-style-type: none"> Only text file is encrypted. -Time complexity is high -memory Space complexity may exceed 	<ul style="list-style-type: none"> -use some advanced technique for encryption of file other than text for example PDF file, Image file etc.
Proposed System	Use public, private key & RSA	Asymmetric & asymmetric key	<ul style="list-style-type: none"> Higher speed for Bluetooth data transfer. -Higher security 	Only text file is encrypted.	Use some advanced technique for encryption of file other than text for example PDF file, Image file etc.

Table 4.1: Comparison between proposed latest and all other existing Bluetooth security system.

Chapter 5

Conclusion

Nowadays data sharing in Bluetooth is largely used but its security, confidentiality and integrity are rarely maintained. The main objectives of this thesis is to encrypt data for its security enhancement, when it is on the Bluetooth network for sharing. But here encryption and decryption time is greater than RSA algorithm. Hardware implementation makes complex than the same case of RSA algorithm. Here also sender public key can be increased. Sender and receiver public key can be applied to any other symmetric or asymmetric algorithm. This thesis shows in the simulations and results that it is efficient than all other existing Bluetooth system by comparing the complexity and proficiency of Bluetooth security systems. This work would be inspiring for advance research such as secure Bluetooth transmission of PDF file, video file, image file etc. that can be our future research topics.

Bibliography

- [1] Wuling Ren, Zhiqian Miao, “*A Hybrid Encryption Algorithm Based on DES and RSA in Bluetooth Communication*” In Proceedings of the Second International Conference on Modeling, Simulation and Visualization Methods. Sanya, pp. 221-225, May 2010.
- [2] Li Juan, Chen Bin, Li Kun “*Study on the Improvement of Encryption Algorithm of Bluetooth*” In Proceedings of the 2009 International Conference on Networking and Digital Society (ICNDS '09). ACM, Volume 01, pp. 88-92, May 2009.
- [3] Trishna Panse, Vivek Kapoor, Prashant Panse, “*A Review on Key Agreement Protocols used in Bluetooth Standard and Security Vulnerabilities in Bluetooth Transmission*” International Journal of Information and Communication Technology Research, March 2012 Volume 2, Issue 3, pp. 315-318.
- [4] Sun J, Howie D, Koivisto A & Sauvola J “*Design, implementation, and evaluation of Bluetooth security*” In Proceedings of the IEEE International Conference on Wireless LANs and Home Networks, Singapore, 2001, pp. 121 - 130.
- [5] Trishna Panse, Vivek Kapoor, “*A Review on Security Mechanism of Bluetooth Communication*”, International Journal of Computer Science and Information Technologies, 2012, Vol. 3 (2) , pp. 3419-3422.
- [6] Trishna Panse, Vivek Kapoor “*An Integrated Scheme based on Triple DES, RSA and MD5 to Enhance the Security in Bluetooth Communication*”, International Journal of computer Application, July 2012. (0975-8887) volume 50- No.7.
- [7] P. C. Kocher “*Timing Attacks on Implementations of DiffieHellman, RSA, DSS and Other Systems*” International Journal of computer Application Volume 1109 pp. 104-113.
- [8] Dhem, Koeune, Leroux, Mestre, Quisquater, and Willems. “*A Practical Implementation of the Timing Attack*” In Proceedings of the third International Conference, CARDIS'98, Louvain-la-Neuve, Belgium, September, 1998 volume 1820 pp. 167-182.
- [9] Sun H., “*Dual RSA and its Security Analysis*”, IEEE Transactions on Information Theory, 2007, pp. 2922-2933.

Appendix

Main .m

```
clc;
disp('RSA Algorithm improvement');
clear all; close all;
M = input('\nEnter the message: ','s');
x=length(M);
[Pp,Pz,de,en] = initializeme(r,s);
[Pk,Phi,d,e] = initialize(p,q);
c=0;
p=1;
q=1;
while p==q || state1==state2 || r==s || state3==state4 || p==r || q==s
state1=randi(1000);
state2=randi(1000);
state3=randi(1000);
state4=randi(1000);
if state1~=state2 || state3~=state4 || state1~=state4 || state2~=state3
p= randseed(state1,1,1,30,80);
q= randseed(state2,1,1,30,80);
r= randseed(state3,1,1,30,100);
s= randseed(state4,1,1,30,100);
end
end

for j= 1:x
    for i=0:255
        if strcmp(M(j),char(i))
            c(j)=i;
        end
    end
end
disp('ASCII Code of the entered Message:');
disp(c);
fprintf('\n.....Sender Encryption.....\n');
% % %Encryption
for j= 1:x
    sender_cipher_1(j)= crypt(c(j),Pp,en);
    if sender_cipher_1(j)>255
        agcipher(j)=mod(sender_cipher_1(j),255);%again cipher
    else
        agcipher(j)=sender_cipher_1(j)+32;
    end
    if agcipher(j)<32
        agcipher(j)=agcipher(j)+32;
    end
    sender_cipher_2(j)= crypt(sender_cipher_1(j),Pk,e);

    if sender_cipher_2(j)>255
        aglcipher(j)=mod(sender_cipher_2(j),255);%again cipher
    else
        aglcipher(j)=sender_cipher_2(j)+32;
```

```

        end
        if aglcipher(j)<32
            aglcipher(j)=aglcipher(j)+32;
        end
    end
    disp('first (public key) encrypted ASCII of the entered Message:');
    disp(sender_cipher_1);
    disp(['Encrypted on pulic key  Message is: ' aglcipher]);

    disp('First encrypt+ RSA ASCII of the  Message:');
    disp(sender_cipher_2);
    disp(['Encrypted Improved RSA Message is: ' aglcipher]);
    % % %Server Decryption
    server_cipher=sender_cipher_2;

    disp('-----Server decryption-----');
    disp(['Server received Data ' aglcipher]);
    disp(' Received cipher ASCII: ');
    disp(server_cipher);
    disp(['The privatel key (d) is: ' num2str(d)]);
    disp(['The private2 key (de) is: ' num2str(de)]);
    for j= 1:x
        server_cipher_1(j)= crypt(server_cipher(j),Pk,d);
        if server_cipher_1(j)>255
            ag2cipher(j)=mod(server_cipher_1(j),255);%again cipher
        else
            ag2cipher(j)=server_cipher_1(j)+32;
        end
        if ag2cipher(j)<32
            ag2cipher(j)=ag2cipher(j)+32;
        end
        server_cipher_2(j) = crypt(server_cipher_1(j),Pp,de);
    end
    disp(' First decrypted ASCII of Message:');
    disp(server_cipher_1);
    disp(['Decrypted Message is: ' ag2cipher]);
    disp(' Decrypted ASCII of Message:');
    disp(server_cipher_2);
    disp(['Decrypted Message is: ' server_cipher_2]);

    receiver_cipher=server_cipher;
    fprintf('\n-----Receiver decryption-----\n');
    disp(['Receiver received Data (Cipher text)' aglcipher]);
    disp(' Received cipher ASCII: ');
    disp(receiver_cipher);
    a= '0';
    trail=3;
    while a =='0'&& trail<4
        fprintf('trail %d remain, \n Enter The privatel (Decryption)key:',
            trail);
        rd = input('');
        if (d==rd)
            a='1';
            trail=4;
        for j= 1:x
            receiver_cipher_1(j)= crypt(receiver_cipher(j),Pk,d);
            if receiver_cipher_1(j)>255

```

```

re2cipher(j)=mod(receiver_cipher_1(j),255);%again cipher
else
re2cipher(j)=receiver_cipher_1(j)+32;
end
if re2cipher(j)<32
re2cipher(j)=re2cipher(j)+32;
end
end
disp('Decrypted ASCII of Message:');
disp(receiver_cipher_1);
disp(['Rsa Decrypted Message is: ' re2cipher]);
a_1='0';
trail_1=3;
while a_1=='0' && trail_1 <4
fprintf('trail %d remain, \n Enter The private 2
(Decryption)key:', trail_1);
rde=input('');
if de==rde
a_1='1';
trail_1=4;
for j= 1:x
receiver_cipher_2(j)
=crypt(receiver_cipher_1(j),Pp,de);
end
fprintf('\nRSA improved decrypted (Original) Message
ASCII\n');
disp(receiver_cipher_2);
disp([' RSA improved Decrypted(Original) Message is: '
receiver_cipher_2]);
else
a_1='0';
disp('Private2 key mismatch.....\n Try again ');
trail_1=trail_1-1;
if trail_1==0
return;
end
end
end
end
else
a='0';
disp('Private1 key mismatch.....\n Try again ');
trail=trail-1;
if trail == 0
return;
end
end
end
end
end

```

crypt.m

```

function mc = crypt(M,N,e)
e=dec2bin(e);
k = 65535;
c = M;
cf = 1;
cf=mod(c*cf,N);

```



```

for i=k-1:-1:1
    c = mod(c*c,N);
    j=k-i+1;
    if e(j)==1
        cf=mod(c*cf,N);
    end
end
mc=cf;

```

Initialize.m

```

function [Pk,Phi,d,e] = initialize(p,q)
Pk=p*q;
Phi=(p-1)*(q-1);
x=2;e=1;
while x > 1
    e=e+1;
    x=gcd(Phi,e);
end
i=1;
r=1;
while r > 0
    k=(Phi*i)+1;
    r=rem(k,e);
    i=i+1;
end
d=k/e;

```

dec2bin.m

```

function a = dec2bin(d)
i=1;
a=zeros(1,65535);
while d >= 2
    r=rem(d,2);
    if r==1
        a(i)=1;
    else
        a(i)=0;
    end
    i=i+1;
    d=floor(d/2);
end
if d == 2
    a(i) = 0;
else
    a(i) = 1;
end
x=[ a(16) a(15) a(14) a(13) a(12) a(11) a(10) a(9) a(8) a(7) a(6) a(5)
a(4) a(3) a(2) a(1)];

```