# SECURE MESSAGING SYSTEM USING RC4-2S

HNIN HSU HLAING

M.C.Sc. JUNE 2022

# SECURE MESSAGING SYSTEM USING RC4-2S

BY

HNIN HSU HLAING

(B.C.Sc.)

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Computer Science

(M.C.Sc.)

University of Computer Studies, Yangon

JUNE 2022

# ACKNOWLEDGEMENTS

To complete this thesis, many things are needed like my hard work and the support of many people.

Firstly, I would like to express my appreciation and sincere thanks to **Dr. Mie Mie Khin**, the Rector of the University of Computer Studies, Yangon, for allowing me to submit this dissertation.

I would like to thank course coordinators, **Dr. Si Si Mar Win** and **Dr. Tin Zar Thaw**, Professors, Faculty of Computer Science, the University of Computer Studies, Yangon for their superior suggestions and administrative support during my academic study.

I am extremely grateful to my supervisor **Dr. Cho Cho San**, Lecturer, Faculty of Computer Systems and Technologies, the University of Computer Studies, Yangon for her valuable guidance, supervision, patience, encouragement, excellent ideas, and editing during the period of study towards completion of this piece of research work.

I wish to thank **Daw Hnin Yee Aung**, Lecturer, English Department, the University of Computer Studies, Yangon for her guidance and editing of my thesis writing from the language point of view.

I also thank my respectful my Professor **Dr. Hnin Wint Khaing** for her valuable comment and advice which are invaluable to me.

I would like to give my sincere thanks to all my teachers who taught me throughout the master's degree course and my friends for their corporation.

Finally, I would like to express my special thanks to my beloved family. Because of it is their love, understanding and financial support, I am able to accomplish this thesis.

# ABSTRACT

Nowadays, telecommunication technologies are developing rapidly and many organizations are mostly using these technologies. The well-known telecommunication technology is the short message service (SMS). It plays a very vital role in the business area such as mobile banking, organizational marketing system, etc. A simple SMS system doesn’t have the built-in procedure to offer data security and attackers can easily intercept it. This problem can be solved by using the encrypted message. This system developed secure SMS application for android smartphone with an RC4-2S algorithm for SMS data confidentiality on mobile networks. This system is developed as an online chat application therefore the user can accept this application anywhere and anytime over the internet without paying SMS fees. The user can send both English language and Myanmar language. The proposed system developed android secure messaging application and provided the performance evaluation results through the use of NIST tests, confusion and diffusion properties to prevent the statistics and other cryptanalysis attacks.

Keywords: SMS, Cryptography, Randomness, Encryption, Decryption

# CONTENTS

# LIST OF FIGURES

# LIST OF FIGURES

**Figure** **Page**

# LIST OF TABLES

# LIST OF EQUATIONS

# CHAPTER 1

# INTRODUCTION

Nowadays, telecommunication technology is developing rapidly; many organizations are mostly using this technology. Short Message Service (SMS) is the top application in the telecommunication technology environment. SMS is a communication method between mobile phones or personal computers (PCs) and the devices that hold the SIM card. SMS message can accept only 160 characters as its maximum size. When the sender sends the SMS message, it is initially stored at the short message service center (SMSC). Then, SMSC forwards these messages to the receiver. The message will keep at the SMSC unless the receiver is in use. When confidential data is sent by using the simple SMS system, it is not safe and secure. A secure messaging service has been developed in this system by encrypting the plain text message to solve these security issues.

## 1.1 Importance of Data Confidentiality in SMS

A simple SMS communication system over the GSM network is not secured. As a result, the confidential data of the SMS can be easily intercepted by unauthorized parties. Therefore, data security in SMS is very important. Simple SMS systems can solve these security issues by combining them with the cryptosystem. In this system, secure messaging for SMS over mobile networks has been implemented by using the RC4-2S stream cipher to provide the data confidentiality service.

Cryptography is the art of maintaining data confidentiality from unauthorized people by changing it into a form of data storing and transmitting the process. The symmetric key encryption algorithm and the asymmetric key encryption algorithm are two types of cryptographic algorithms. Symmetric key encryption algorithms use only one key for both the encryption and decryption processes. An asymmetric key encryption algorithm uses a pair of keys, one for the public and another for the private key.

By using a cryptosystem, the message that the sender sends is encrypted into cipher text. If the attackers have intercepted the SMS system, they would not have changed the cipher text to the plain text without using the correct key. The proposed system developed the mobile phone application to convert the SMS plain text message into cipher text therefore that the data of the SMS message can’t be known by unauthorized parties [3].

## 1.2 Motivation of the Thesis

The main significance of the thesis is to reduce the risks of SMS security by using the encryption and decryption process. Message encryption and decryption processes do not take a long time and these processes are more secure. When the SMS system sends the plain text over the network, it is easy to make criminality by intercepting the data of SMS. The attackers can be monitoring the text conversation and can change the SMS data for many reasons.

## 1.3 Objective of the Thesis

This system proposed as a secure messaging service that encrypts the message to provide the security from anyone because a simple SMS system can be easily intercepted by unauthorized parties. This system is developed

- To offer the data security during the transmission of SMS message.
- To reduce the risks by preventing hackers from accessing confidential data and messages.
- To prevent from the cyber-criminal activity.

This system developed as a simple online chat application and secure message system. Therefore, it is secure, flexible and it sends the secure message at the same time.

## 1.4 Overview of the Thesis

The proposed system provides the security of the simple SMS system. In this system, the messages are encrypted into cipher text before being sent over the communication channel. Therefore, this system provides the confidentiality service to improve security. The system automatically generates the random initial key for the encryption and decryption process. To provide the confidentially data, the proposed system uses the RC4-2S stream cipher algorithm. This proposed system uses the Firebase authentication feature and Firebase real-time database feature for user authentication and data storage. Online Messaging Application has been proposed in this system to provide the security in simple SMS. The user can use this application everywhere and they can access the internet every time without paying SMS fees.

## 1.5 Organization of the Thesis

The system designs to provide data security and protect the SMS's confidential data of the SMS from attackers. The thesis is organized into five main parts.

The importance of data confidentiality, motivation, objectives of the thesis, an overview of the thesis, and organization of the thesis are expressed in this Chapter 1. Chapter 2 describes the background theory, along with recent research works, and also discuss the cyber security goals, cryptography, Short Message Service (SMS), SMS security, working of the SMS, cryptographic algorithm, and the randomness key. The detail discussion of proposed system is presented in Chapter 3. Chapter 4 describes system design, user interface of the developed application, the randomness of the key-stream is tested using the frequency test and run test of the NIST statistical test suit. Chapter 5 concludes the proposed system and discusses the future extension of this proposed system.

# CHAPTER 2

# BACKGROUND THEORY

Instant messaging is one of the most popular forms of communication used today. The instant messaging system is ripe for hacking and cybercrime. As a result, secure message transmission is critical. The next section describes various methods how a secure messaging system is implementing for SMS.

## 2.1 Short Message Service

Short Message Service (SMS) is a mobile messaging technology that enables to send and receive messages between mobile phones. The ‘short’ part refers to the maximum length of the text message at 160 characters (SMS systems are not encrypted when sending the SMS data) [10]. SMS sends plain text without encryption. Attackers can easily intercept SMS systems. SMS communication is not secure; therefore, the author in [3] proposed a secure SMS system.

### 2.1.1 SMS Architecture

When the user sends a message using their cell phone, it is received by short message service center (SMSC). Figure 2.1 describes the SMS architecture. SMSC finds the destination (receiver) and then sends this message to the destination device (mobile phone). SMSC is installed on mobile carrier core networks. Before forwarding these messages to the destination device, SMSC also acts as a temporary store for SMS messages. If the destination device is not in use, SMSC will store the SMS message and then forward this message when the destination device is active. The SMSC can also notify whether the delivery process is successful or not. SMSC can’t store the SMS messages for a long time. Therefore, the storage capacity is limited. During the SMS delivery, the sender’s mobile phone and SMSC are actively communicating. If the destination device is not available for a time, the SMSC will directly notifies the sender’s mobile phone and tells that the message delivery was not successful [10].

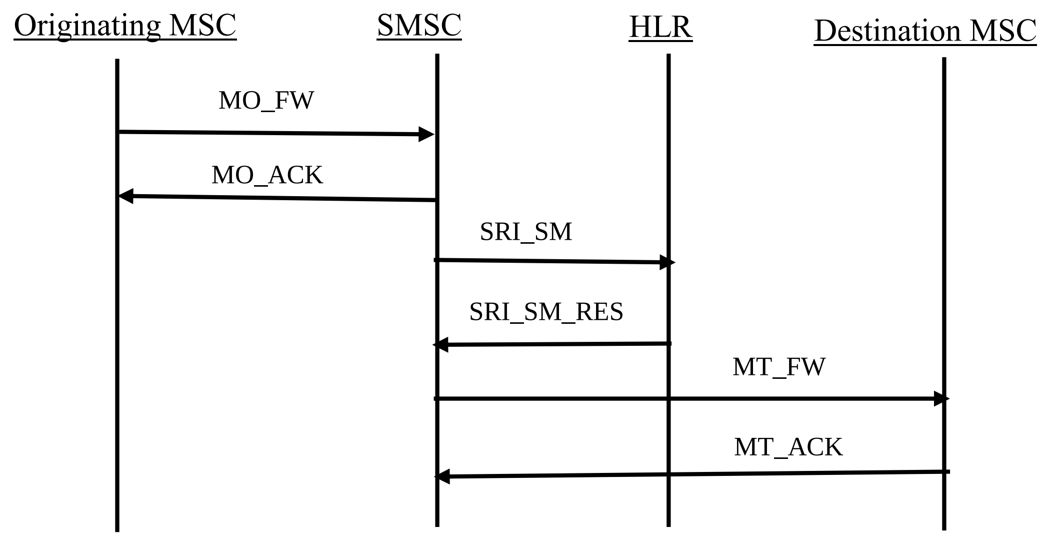


**Figure 2.1 SMS Architecture**

MO_FW **:** The MSC forwards an SMS to the SMSC that has been submitted a subscriber

MO_ACK **:** The SMSC acknowledges the reception of the SMS

SRI_SM **:** The SMSC sends a “Send Routing Info for Short Message” message to the HLR, requesting the destination’s location

SRI_SM_RES **:** The HLR responds the routing info

MT_FW **:** The SMSC route the SMS to the destination MSC

MT_ACK **:** The destination MSC acknowledges the reception of the SMS.

Where;

| | |
|---|---|
| MSC | : Mobile switching center |
| SMSC | : Short message service center |
| HLR | : Home location register |
| MO_FW | : Mobile originating short message transfer message |
| MO_ACK | : Mobile originating short message acknowledgement |
| SRI_SM | : Send routing information for short message |
| SRI_SM_RES | : Send routing information for short message response |
| MT_FW | : Mobile terminating short message transfer message |
| MT_ACK | : Mobile terminating short message acknowledgement |

## 2.2 Security of SMS

All over the world, mobile phone networks are connected to each other through the SS7 protocol (Signaling System No. 7 protocol). This is the pathway through which the user’s phone can connect to a cellular network and make receive calls. The SS7 system can be attacked easily by attackers, who have peaked at SMS messages or intercepted them by unauthorized parties. This is especially useful when cooperating with bank accounts. For example, the attackers can snoop on the SMS verification codes and use them to gain access bank accounts, and drain them [4] [9] [12].

Therefore, this system proposes to protect the security of the SMS. The proposed system is based on the simple SMS architecture, but only the cipher text is passed around the communication channel. This system used the Firebase real-time database as a short message service center.

### 2.2.1 Information Security Goals

There are three major goals for information security. The cyber security goal is to defend the data from being intercepted by unauthorized people. Three types of security goals are confidentiality, integrity, and availability (CIA) [2] [6] [22].

#### 2.2.1.1 Data Confidentiality

Data Confidentiality deals with protecting against the disclosure of information by ensuring that access to the data is restricted to those who are not authorized. It is provided to access the data if the user is authorized. Otherwise, the data needs to be protected from unauthorized users. In such a way, its semantics remain accessible only to unauthorized users who possess some critical information [2] [6] [22].

#### 2.2.1.2 Data Integrity

Data integrity is ensuring the authenticity of the data or information that does not alter between sender and receiver. Data integrity is the availability of data created by the source that does not alter by an unauthorized person in an unauthorized way. Data integrity is important because the receiver needs to access the data without changing the data [2] [6] [22].

#### 2.2.1.3 Data Availability

Data availability is a measure of how data is available to be used, whether by own organization or partner. Data Availability concerns both the accessibility and continuity of information. Information needs to be available to an authorized entity when it is needed. The user can access the data as much as possible without unexpected issues and interruptions [2] [6] [22].

## 2.3 Cryptography

Cryptography is the art of protecting data security by turning it into another form that is human unreadable form. It can provide secure communication by preventing malicious threads, the attackers can't access the information. Cryptographic systems change the plain text into cipher text by applying an encryption algorithm when transmitting the data or information.

Encryption is the process to transform the plain text into cipher text using a secret key known only by the sender and receiver. The secret key value is typically known only by the data owner. The decryption process is changing the cipher text to plain text by using the secret key value and the receiver must use the correct key to see the original text. Cryptographic algorithms are used privacy protection methods for credential data [2] [6].

Confusion and diffusion properties are important to secure the message in cryptography. Both confusion and diffusion are used to prevent the encryption key from its deduction or ultimately to prevent the original message.

Confusion is the correlation between the key and the cipher text but they does not depend each other. This means that each binary digit of the cipher text should depend on several parts of the key.

Diffusion means that if we change a character in the plain text, several characters in the cipher text will change [13] [23].

In this proposed system, the generated cipher text is not related to the key, therefore the system also provides the confusion properties and when the plain text is changed a little, the cipher text is clearly different that this proposed system also provided the confusion properties.

Cryptography provides three techniques:

1. Symmetric Key Encryption
2. Asymmetric Key Encryption
3. Hashing Algorithm

### 2.3.1 Symmetric Key Algorithm

Symmetric key encryption can be defined as a secret key algorithm. Figure 2.2 describes the process of the symmetric key encryption algorithm. This used only one key for both the encryption and decryption processes. The symmetric key algorithm can be considered very secure. The requirement that the sender and the receiver have access to the secret key is one of the most vulnerable of the symmetric key encryption algorithms. Some of the symmetric key algorithms are Data Encryption Standard (DES), Advanced Encryption Process (AES), Rivest Cipher (RC4), and RC4-2S [6].

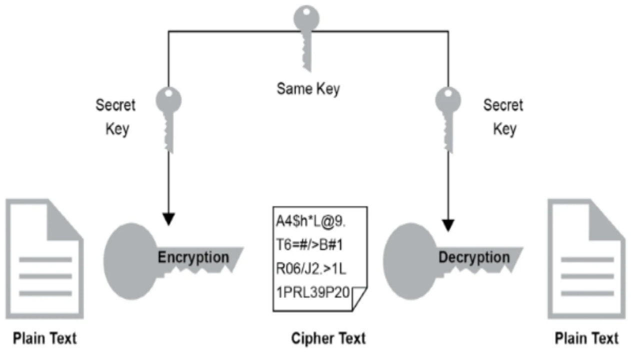
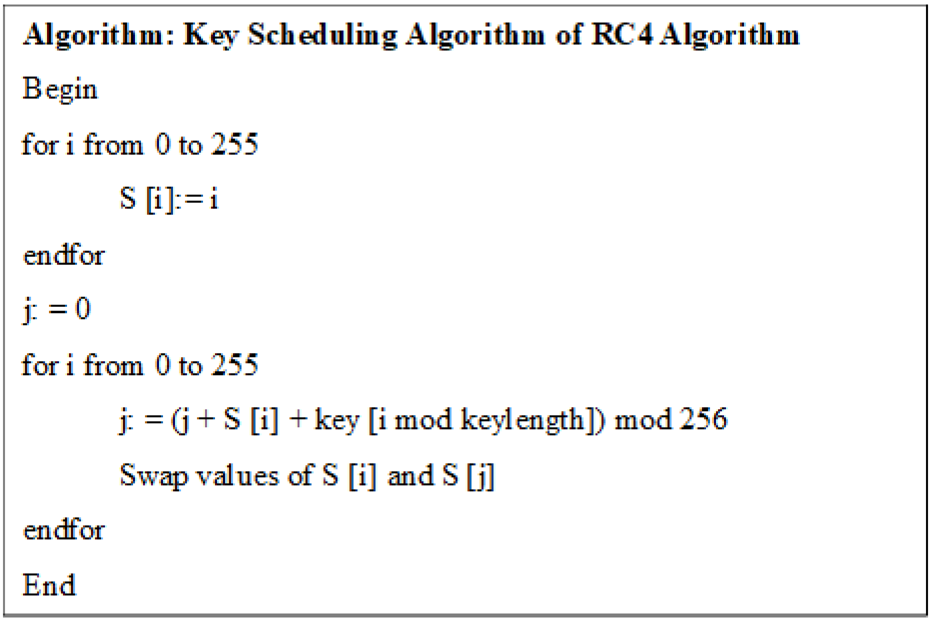


**Figure 2.2 Symmetric Key Encryption Algorithm**

#### 2.3.1.1 Rivest Cipher 4 (RC4)

RC4 is the stream cipher algorithm, invented by Ron Rivest. This is one of the symmetric key algorithms, it uses only one key for both the encryption and decryption process [1] [5] [19].

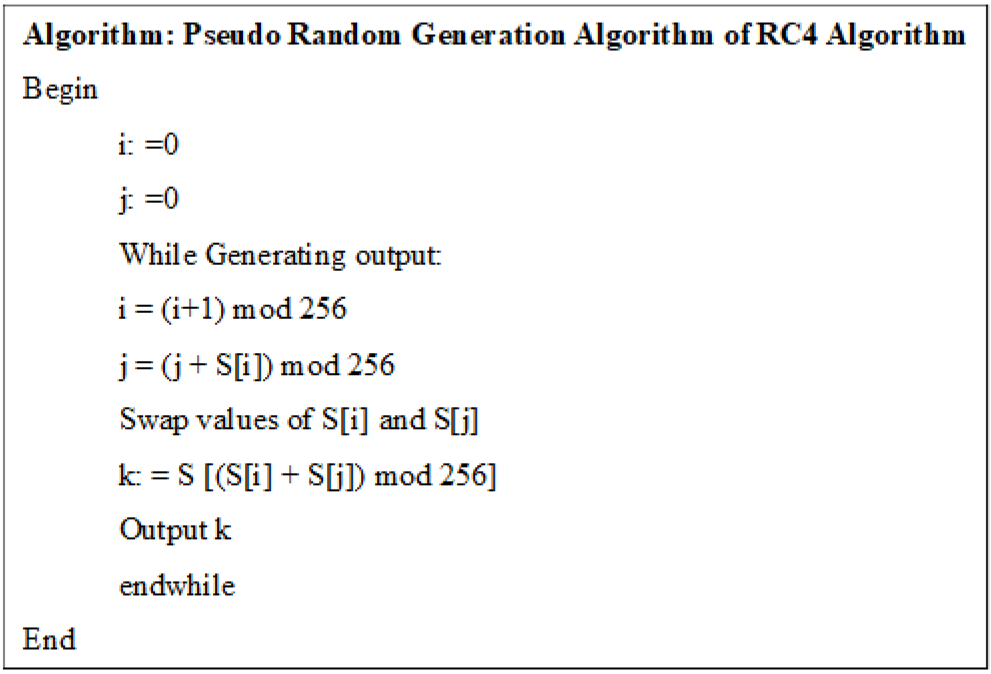
The generated key stream was growing based on the plain text. A variable key length can be used between 1 and 256 bits. However, the minimum number of the key length must be 128 bits to keep the security of the data. Among the stream cipher algorithms, the RC4 stream cipher is widely used because the encryption time is faster than the other stream cipher algorithms.



```
Algorithm: Key Scheduling Algorithm of RC4 Algorithm
Begin
for i from 0 to 255
        S [i]:= i
endfor
j: = 0
for i from 0 to 255
        j: = (j + S [i] + key [i mod keylength]) mod 256
        Swap values of S [i] and S [j]
endfor
End
```

**Figure 2.3 KSA Phase of the RC4 Algorithm**

It has two phases to generate the key stream. The first one is the Key Scheduling Algorithm (KSA). Figure 2.3 describes the process flow of the Key Scheduling Algorithm of the RC4 encryption algorithm. It is the initialization state, initialized a 256-byte array S, with the elements from 0 to 255. And it defines the key with another state vector. After performing the swapping operations that generate the random state vector S, that is the input vector for the PRGA phase.



```
Algorithm: Pseudo Random Generation Algorithm of RC4 Algorithm
Begin
        i: =0
        j: =0
        While Generating output:
        i = (i+1) mod 256
        j = (j + S[i]) mod 256
        Swap values of S[i] and S[j]
        k: = S [(S[i] + S[j]) mod 256]
        Output k
        endwhile
End
```

**Figure 2.4 RC4 Pseudo-Random Generation Algorithm**

The second phase is the Pseudo-Random Generation Algorithm (PRGA). Figure 2.4 described the process flow of the PRGA phase. Firstly, it initializes two values and then takes the loop cycles until the end of the plain text is reached. In this state, each element swaps with another element in the state vector S. After performing the swapping operation that generates the random key stream. Finally, convert the plain text to cipher text by XOR-ing the key-stream with the plain text [9] [10].

#### 2.3.1.2 Weakness of the RC4

The RC4 algorithm has some weak points, consisting a correlation problem. RC4 algorithm has two types of problems: the weakness of the KSA phase and the weakness of relations between the S-box in different time [19]. The generated key-stream of RC4-2S is more random than the RC4 generated key-stream. Furthermore, the RC4-2S algorithm generates keys faster than the RC4 algorithm.

### 2.3.2 Asymmetric Key Algorithm

An asymmetric key algorithm is also known as a public key encryption algorithm. In Figure 2.5 describes the process flow of the Asymmetric Key Encryption Algorithm. The asymmetric key encryption process has a pair of keys. One is for public keys and another for private keys. The public key is known to users, but the private key is controlled by the owner. The private key cannot be calculated through the use of a public key because they are related cryptographically. An asymmetric key algorithm is typically used in computing digital signatures. Some asymmetric key encryption algorithms are RSA, DSS, and Diffie-Hellman exchange method [6].

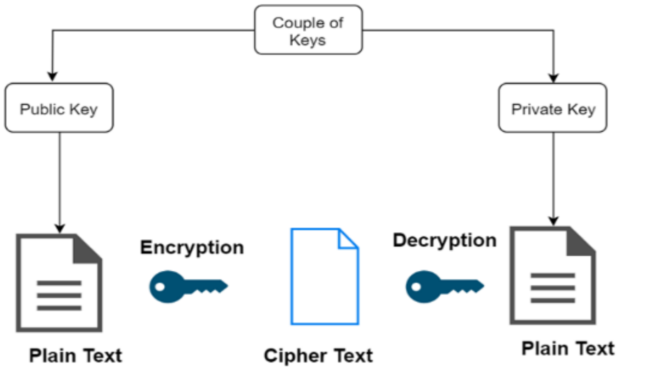


**Figure 2.5 Asymmetric Key Encryption Algorithm**

### 2.3.3 Hashing Algorithm

The hashing algorithm is also known as a hash function. Figure 2.6 expresses about the hashing algorithm. A hash function can take an arbitrary amount of input data and produce fixed-length cipher text by applying a mathematical formula [6]. The Message-digest algorithm (MD5) and the Secure Hashing Algorithm (SHA) are well-known algorithms.

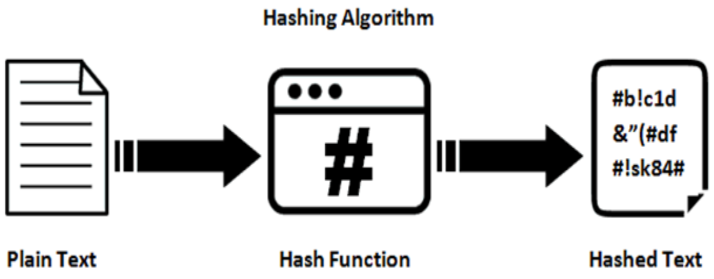


**Figure 2.6 Hashing Algorithm**

## 2.4 Secure Messaging Techniques

R.Rifki, A.Septiarini and H.R.Hatta [19] developed secure messaging methods based on encryption and decryption time, which are influenced by the characters, the number of the SMS message, and the key as well as the smartphone specification. The author mentioned that the correlation value was only affected by the characters, the number of the message, and the key. The author showed that the fineness of RC4 by comparing it with the method of Vigenere and the Playfair Algorithm.

M.S.Novelan, A.M.Husein, M.Harahap and S.Aisyah [16], the author used a block cipher such as the Tiny Encryption Algorithm (TEA). The block size is 64 bits and has a 128-bit key length. The number of rounds is 32 and is based on the Feistel network. The encryption and decryption processes are done by using a one-time pad algorithm, and then the same key is used for enciphering the message. Furthermore, the author described how the speed of the encryption and decryption process depends on the size of the file, the larger the file size the more time it takes for encryption and decryption.

A.F.Doni, O.A.H.Maria and S.Hanif [1], the author applied the RC4 algorithm to encrypt the pdf, Docx, Xls, xlsx, or text files with at least 8 characters password. The experiment demonstrated that the provided data was authentic in their approach, such that it could not be easily changed in the form of text by people who are not responsible. The encryption and decryption processes of the files are direct to the file execution time in accordance with the test that has been carried out.

M.S.Novelan, A.M.Husein, M.Harahap and S.Aisyah [16], the other researchers claimed that RC4's performance is better than AES, which is based on the CPU processing time, encryption and decryption time, throughput, and memory utilization. In the comparison with the Blowfish algorithm, the encryption performance of RC4 is better than Blowfish but the decryption performance of Blowfish is better than the RC4 for a small message. For performance in power consumption, RC4 is better than Blowfish.

E.N.Ekwonwune and V.C.Enyinnaya [9], described the processing time of decipherment as directly proportional to the size of key length and data size if the data is large enough. The authors also mentioned the importance of data type because the image data need more time to process than text or sound data.

K.H.Myint [12], the author developed the SMS architecture by using the RC4 encryption and decryption algorithm. The author developed it as an offline application and split it into two applications. One is the senderSMS application and the other is the receiverSMS application. And the users type the encryption key manually.

O.S.Sitompul, N.H.Pasaribu and E.B.Nababan [18], the author developed the application using a hybrid approach of RC4 and the Affine cipher to secure the SMS system. The testing was done by using two smartphones, android OS, to ensure the transmission of SMS messages across platforms. The key stream of the RC4 cipher was used to encrypt the plaintext, and the Affine algorithm was used to save the secret key of RC4. The author showed that the length of the plain texts and the cipher texts are the same before and after encryption and decryption.

Almost all recent research works that are mentioned above developed on Android phones and require the user to manually type the encryption key. This system is also implemented on Android Phone by using the RC4-2S algorithm to implement the secure messaging system. The key generation process is automatically generated using the random key generator and then the encryption and decryption processes are run in the background. The proposed system can reduce the charge for SMS transmission by using an online messaging system.

This section discusses the recent related works for securing the SMS system, as well as the advantages and disadvantages of encryption algorithms in term of processing time. The next section describes the concepts of SMS and the working procedure of SMS.

## 2.5 Randomness

Randomness is the non- predictability of the elements in the sequence. The sequence's elements are generated independently of one another, and the value of the next element in the sequence cannot be predicted regardless of how many elements have already been generated. This sequence is randomized. In cryptography, randomness is very important because the secret key can be guessed by attackers. The security of the information must be ensured when the secret key has a high randomness level that is used in any algorithm [8] [20] [21].

### 2.5.1 NIST Statistical Tests

The binary generated sequences are tested using the National Institute of Standard and Technology (NIST) test suite, which is a statistical package for random number generation tests and that includes 15 statistical tests to measure the randomness of the output sequence of true random number generators or pseudo random number generators [5].

The 15 tests are

1. The frequency (monobit) test
2. Frequency Test within a block
3. The run test
4. Test for the longest-Run-of-Ones in a block
5. The Binary Matrix Rank Test
6. The Discrete Fourier Transform
7. The Non overlapping Template Matching
8. The overlapping Template Matching Test
9. Maurer’s “Universal Statistical” Test
10. The linear Complexity Test
11. The Serial Test
12. The Approximate Entropy Test

13. The Communication Sum (Cusums) Test

14. The Random Excursion Test and

15. The Random Excursions Variant Test

#### 2.5.1.1 The Frequency Monobit Test

The focus of the test is the proportion of zeroes and ones for the entire sequence. The test determines whether the number of ones and zeroes in a sequence is roughly the same as would be expected for a truly random sequence.

**Function call**

Frequency (*n*), where:

n : The length of the bit string.

ε : The sequence of bits as generated by the RNG or PRNG being tested.

$S_{obs}$ : The absolute value of the sum of the $X_i$ (where, $X_i = 2\varepsilon - 1 = \pm 1$) in the sequence divided by the square root of the length of the sequence.

$$S_{obs} = \frac{S_n}{\sqrt{n}}$$

**Equation 2.1**

$$P_{value} = erfc(\frac{S_{obs}}{\sqrt{2}})$$

**Equation 2.2**

**Decision Rule (at the 1% Level)**

If the computed P-value is less than 0.01, the sequence is not random. Otherwise, assume the sequence is random.

#### 2.5.1.2 Frequency Test within a Block

The focus of the test is the proportion of one within M-bit blocks. The goal of this test is to determine if the frequency of ones in an M-bit block is approximately M/2, as would be expected under a randomness assumption.

**Function call**

BlockFrequency (*M*, *n*), where:

M : The length of each block.

n : The length of the bit string.

$\varepsilon$ : The sequence of bits as generated by the RNG or PRNG being tested

$\chi 2_{(obs)}$ : A measure of how well the observed proportion of ones within a given M-bit block matches the expected proportion (1/2).

$$X_{obs}^{2} = 4M \sum_{i=1}^{N} \left(\pi_i - \frac{1}{2}\right)^2$$

**Equation 2.3**

$$P_{value} = igamc\left(\frac{N}{2}, \frac{X_{obs}^{2}}{2}\right)$$

**Equation 2.4**

### 2.5.1.3 The Run Test

The total number of runs on the sequence is the focus of this test, where a run is an uninterrupted sequence of identical bit. A run of length k consists of exactly k identical bits and is bounded before and after with a bit of the opposite value. The purpose of the test is to determine whether the number of runs of ones and zeros of varying lengths is as expected for a random sequence.

**Function call**

Runs (*n*), where:

n : The length of the bit string.

$\varepsilon$ : The sequence of bits as generated by the RNG or PRNG being tested

$V_{n(obs)}$: The total number of runs across all n bits.

$$V_{n(obs)} = \sum_{k=1}^{n-1} r(k) + 1$$

**Equation 2.5**

Where r (k) = 0 if $\varepsilon_k = \varepsilon_{(k+1)}$ otherwise r (k) =1

$$P_{value} = erfc\left( \frac{V_{n(obs)-2n\pi(1-\pi)}}{\sqrt[2]{2n}\,\pi(1-\pi)}\right)$$

**Equation 2.6**

**Decision Rule (at the 1% Level)**

If the calculated P-value is less than 0.01, the sequence is not random. Otherwise, regarded the sequence to be random.

### 2.5.1.4 Test for the Longest-Run-of-Ones in a Block

The focus of the test is the longest run of ones within M-bit blocks. The purpose of this test is determined whether the length of the longest run of ones within the tested sequence matches the length of the longest run of ones expected in a random sequence.

**Function Call**

LongestRunOfOnes ($n$), Where:

n : The length of the bit string

$\varepsilon$ : The sequence of bits as generated by the RNG or PRNG being tested

M : The length of each block. The test code has been pre-set to accommodate three values for M: M = 8, M = 128 and M = 104 in accordance with the following values of sequence length, n:

N : The number of blocks; selected in accordance with the value of M

$V_i$ : The the longest runs of ones in each block into categories, where each cell contains the number of runs of ones of a given length.

$\chi^2_{(obs)}$: A measure of how well the observed longest run length within M-bit blocks matches the expected longest length within M-bit blocks.

$$X_{obs}^2 = \sum_{i=0}^{K} \frac{(V_i - N\pi_i)^2}{N\pi_i}$$

**Equation 2.7**

$$P_{value} = igamc\left(\frac{K}{2}, \frac{X_{obs}^2}{2}\right)$$

**Equation 2.8**

### 2.5.1.5 The Binary Matrix Rank Test

The focus of the test is the rank of disjoint sub-matrices of the entire sequence. The goal of this test is to check if there is any linear dependence between fixed length substrings of the original sequence.

**Function Call**

Rank ($n$), where:

n : The length of the bit string.

$\varepsilon$ : The sequence of bits as generated by the RNG or PRNG being tested

M : The number of rows in each matrix. For the test suite, M has been set to 32. If other values of M are used, new approximations need to be computed.

Q : The number of columns in each matrix. For the test suite, Q has been set to 32. If other values of Q are used, new approximations need to be computed

$X^2_{(obs)}$ : A measure of how well the observed number of ranks of various orders matches the expected number of ranks under an assumption of randomness.

$$X^2_{obs} = \frac{(F_M - 0.2888N)^2}{0.2888N} + \frac{(F_{M-1} - 0.5776N)^2}{0.5776N} + \frac{(N - F_M - F_{M-1} - 0.1336N)^2}{0.1336N}$$

**Equation 2.9**

$$P_{value} = e^{-X^2_{obs}/2}$$

**Equation 2.10**

### 2.5.1.6 The Discrete Fourier Transform (Spectral) Test

The focus of this test is the peal height in the sequence. The purpose of this test is to detect periodic features (i.e., repetitive patterns that are near each other) in the tried arrangement that would indicate a deviation from the randomness hypothesis.

**Function Call**

DiscreteFourierTransform (*n*), where:

n : The length of the bit string.

$\varepsilon$ : The sequence of bits as generated by the RNG or PRNG being tested;

d : The normalized difference between the observed and the expected number of frequency components that are beyond the 95 % threshold.

$N_0$ *: .95n/2*

$N_1$ : The actual observed number of peaks in *M* that are less than *T.*

$$d = \frac{(N_1 - N_{0)}}{\sqrt{n(.95).05)}/4}$$

**Equation 2.11**

$$\mathrm{P_{value}} = erfc(\frac{|d|}{\sqrt{2}})$$

**Equation 2.12**

### 2.5.1.7 Non-Overlapping Template Matching Test

The number of occurrences of pre-specified target string is the focus of this test. The purpose of this test is to detect generators that produce an excessive number of occurrences of a given non-periodic pattern.

**Function Call**

NonOverlappingTemplateMatching (*m, n*)

m : The length in bits of each template

n : The length of the entire bit string under test.

$\varepsilon$ : The sequence of bits as generated by the RNG or PRNG being tested

B : The m-bit template to be matched; B is a string of ones and zeros (of length m) which is defined in a template library of non-periodic patterns contained within the test code.

M : The length in bits of the substring of ε to be tested.

N : The number of independent blocks. N has been fixed at 8 in the test code.

$Wj$ : (*j* = 1, …, *N*) be the number of times that *B* (the template) occurs within the block j

$X_{0bs}^2$ : A measure of how well the observed number of templates

$$X_{0bs}^2 = \sum_{j=1}^{N} \frac{(W_j - \mu\,)^2}{\sigma^2}$$

**Equation 2.13**

$$P_{Value} = igamc(\frac{N}{2}, \frac{X_{obs}^2}{2})$$

**Equation 2.14**

### 2.5.1.8 Overlapping Template Matching Test

The focus of this test is the number of occurrences of pre-specified target strings. The purpose of this test and the Non-Overlapping Template Matching test use an m-bit window to search for a specific m-bit pattern.

**Function Call**

OverlappingTemplateMatching (*m, n*)

m : The length in bits of the template – in this case, the length of the run of ones.

n : The length of the bit string.

$\varepsilon$ : The sequence of bits as generated by the RNG or PRNG being tested

*B* : The m-bit template to be matched.

*K* : The number of degrees of freedom. K has been fixed at 5 in the test code.

*M* : The length in bits of a substring of ε to be tested. M has been set to 1032 in the test code.

*N* : The number of independent blocks of n. N has been set to 968 in the test code.

$V_i$ : The number of occurrences of B in each of the N blocks.

$\chi^2_{(obs)}$ : A measure of how well the observed number of template "hits" matches the expected number of template "hits" (under an assumption of randomness).

$$\mathrm{X}^2_{obs} = \sum_{i=0}^{5} \frac{(V_i - N\pi_i)^2}{N\pi i}$$

**Equation 2.15**

$$P_{value} = igamc(\frac{5}{2}, \frac{X^2_{obs}}{2})$$

**Equation 2.16**

### 2.5.1.9 Maurer's "Universal Statistical" Test

The focus of the test is the number of bits between matching patterns. The purpose of the test is to detect whether or not the sequence can be significantly compressed without losing information.

**Function Call**

Universal (*L, Q, n*), where

*L* : The length of each block.

*Q* : The number of blocks in the initialization sequence.

n : The length of the bit string.

$\varepsilon$ : The sequence of bits as generated by the RNG or PRNG being tested.

$f_n$ : The sum of the log2 distances between matching L-bit templates.

$$f_n = \frac{1}{k}\sum_{i=Q+1}^{Q+k} \log_2(i - T_j)$$

**Equation 2.17**

$$P_{value} = erfc(\left|\frac{f_n - expectedValue(L)}{\sqrt{2}\,\sigma}\right|)$$

**Equation 2.18**

### 2.5.1.10 Linear Complexity Test

The focus of this test is the length of a linear feedback shift register (LFSR). The purpose of the test is to determine whether the sequence is complex enough to be considered random.

**Function Call**

LinearComplexity (*M, n*), where:

M : The length in bits of a block.

n : The length of the bit string.

$\varepsilon$ : The sequence of bits as generated by the RNG or PRNG being tested.

$K$ : The number of degrees of freedom. K = 6 has been hard coded into the test.

$V_i$ : The number of occurrences of B in each of the N blocks.

$\chi^2$ *(obs)*: A measure of how well the observed number of occurrences of fixed length LFSRs matches the expected number of occurrences under an assumption of randomness.

$$X_{obs}^2 = \sum_{i=0}^{K} \frac{(V_i - N\pi_i)^2}{N\pi_i}$$

**Equation 2.19**

$$P_{value} = igamc\left(\frac{K}{2}, \frac{X_{obs}^2}{2}\right)$$

**Equation 2.20**

### 2.5.1.11 Serial Test

The focus of the test is the frequency of all possible overlapping m-bit patterns across the entries sequence. The goal of this test is to see if the number of occurrences of the 2m m-bit overlapping pattern is close to what would be expected for a random sequence.

**Function Call**

Serial (*m, n*), where:

m : The length in bits of each block.

n : The length in bits of the bit string.

$\varepsilon$ : The sequence of bits as generated by the RNG or PRNG being tested

$\nabla\psi^2_m(obs)$ *and* $\nabla^2\psi^2_m(obs)$ : A measure of how well the observed frequencies of m-bit patterns match the expected frequencies of the m-bit patterns.

$$P_{value1} = igamc(2^{m-2}, \nabla\psi^2{}_m)$$

**Equation 2.21**

$$P_{value2} = igamc(2^{m-3}, \nabla\psi^2{}_m)$$

**Equation 2.22**

### 2.5.1.12 Approximate Entropy Test

The focus of the test is the frequency of all possible overlapping m-bit pattern across the entire sequence. The goal of the test is comparing the frequency of overlapping blocks of two consecutive adjacent lengths (m and m+1) to the expected result for a random sequence.

**Function Call**

ApproximateEntropy (*m, n*), where:

m : The length of each block- in this case, the first block length used in the test. m+1 is the second block length used.

n : The length of the entire bit sequence.

$\varepsilon$ : The sequence of bits as generated by the RNG or PRNG being tested.

$\chi^2$ *(obs):* A measure of how well the observed value of *ApEn(m)*

$$X^2 = 2\mathrm{n}[\log 2 - ApEn(m)]$$

**Equation 2.23**

$$P_{value} = igamc\left(2^{m-1}, \frac{x^2}{2}\right)$$

**Equation 2.24**

### 2.5.1.13 Cumulative Sums (Cusum) Test

The focus of this test is the maximal excursion (from zero) of the random walk define by the cumulative sum of adjusted (-1, + 1) digits in the sequence. The purpose of the test is to determine whether the cumulative sum of the partial sequences occurring in the tested sequence is too small relative to the expected behavior of that cumulative sum for random sequence.

**Function Call**

CumulativeSums (*mode*, *n*), where:

n : The length of the bit string.

$\varepsilon$ : The sequence of bits as generated by the RNG or PRNG being tested.

mode : A switch for applying the test either forward through the input sequence (mode = 0) or backward through the sequence (mode = 1).

z : The largest excursion from the origin of the cumulative sums in the corresponding (-1, +1) sequence.

$$z = max_{1\leq k\leq n}|S_k$$

**Equation 2.25**

$$P_{value} = 1 - \sum_{k=\left(\frac{-n}{2}+1\right)/4}^{\left(\frac{n}{z}-1\right)/4} \left( \emptyset\left(\frac{(4k+1)z}{\sqrt{n}}\right) - \emptyset\left(\frac{(4k-1)z}{\sqrt{n}}\right) \right) + \sum_{k=\left(\frac{-n}{2}+3\right)/4}^{\left(\frac{n}{z}-1\right)/4} \left( \emptyset\left(\frac{(4k+3)z}{\sqrt{n}}\right) - \emptyset\left(\frac{(4k+1)z}{\sqrt{n}}\right) \right)$$

**Equation 2.26**

### 2.5.1.14 Random Excursion Test

The focus of this test is the number of cycles having exactly K visits in a cumulative sum random walk. After transferring the (0,1) sequence to the appropriate (-1, +1) sequence, the cumulative sum random walk is derived from partial sums. A cycle of a random walk consists of a sequence of steps of unit length taken at random that begin at and return to the origin.

**Function Call**

RandomExcursions (*n*), where:

n : The length of the bit string.

$\varepsilon$ : The sequence of bits as generated by the RNG or PRNG being tested.

$\chi^2(obs)$ : For a given state x, a measure of how well the observed number of state visits within a cycle match the expected number of state visits within a cycle, under an assumption of randomness.

$$x_{obs}^{2} = \sum_{k=0}^{5} \frac{(V_k(x) - J\pi_k(x)^2)^2}{J\pi_k(x)}$$

**Equation 2.27**

### 2.5.1.15 Random Excursions Variant Test

The focus of the test is the total number of times that a particular state is visited (i.e., occurs) in a cumulative sum random walk. The purpose of this test is to identify deviations from the expected number of visits to different states on the random walk.

**Function Call**

RandomExcursionsVariant (*n*), where:

n : The length of the bit string; available as a parameter during the function call.

$\varepsilon$ : The sequence of bits as generated by the RNG or PRNG being tested.

$\xi$ : For a given state x, the total number of times that the given state is visited during the entire random walk.

$$P_{value} = erfc\left(\frac{|\mathrm{\xi(x)} - \mathrm{J}|}{\sqrt{2J(4|x| - 2)}}\right)$$

**Equation 2.28**

## 2.6 Chapter Summary

This chapter mentions the background theories, SMS architecture, how important the security of instant messaging, cyber security goals, cryptologic, NIST statistical test suite for randomness and also express the related works of the previous user. The next chapter describes the proposed system.

# CHAPTER 3

# PROPOSED SECURE MESSAGING SYSTEM

This section will describe the detail of the proposed secure messaging system and express the user interface of the proposed system. Simple SMS is not secured for confidential data transmission. Therefore, this system proposed to offer the data security by using the encrypted messages.

## 3.1 Proposed System Methodology

This system develops an online chat application by using the end-to-end encryption mechanism. The simple SMS system is not secure for sending confidential data. As a result, the RC4-2S encryption algorithm has been applied to ensure the security of the message in the original SMS service. The proposed system used the original SMS architecture (described in Chapter 2) by combining the RC4-2S stream cipher algorithm. Overall architecture of the proposed system is described in Figure 3.1.

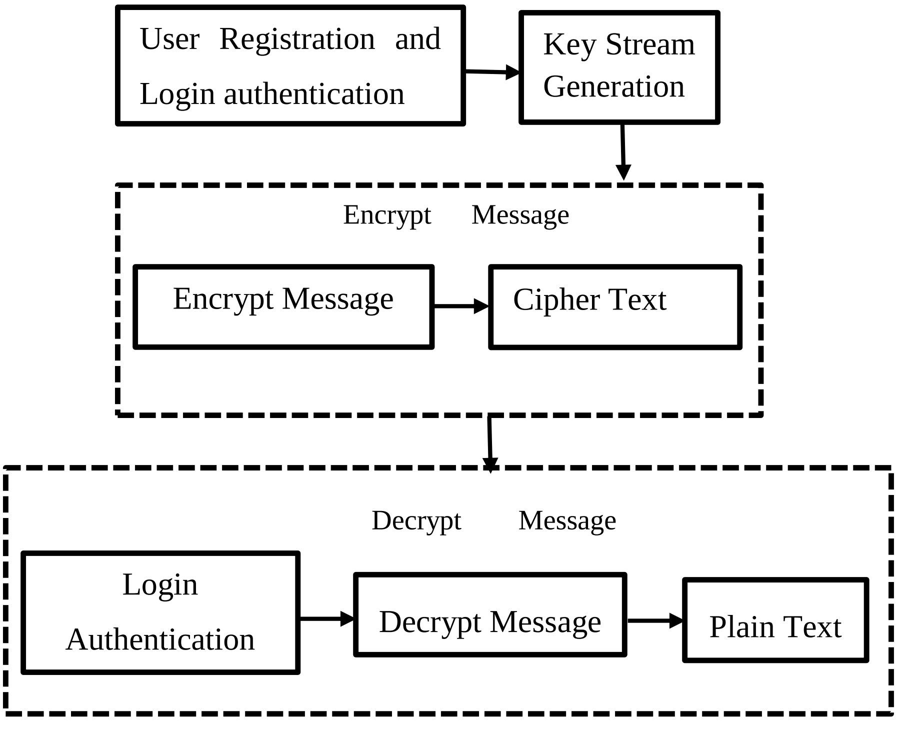


**Figure 3. 1 Overall Architecture of Secure SMS System**

This proposed system can be differentiated into four main parts:

- User registration and login authentication
- Key stream generation based on RC4-2S algorithm
- Message Encryption
    - Encrypt message
    - Send the cipher text
- Message Decryption
    - Login authentication
    - Decrypt message

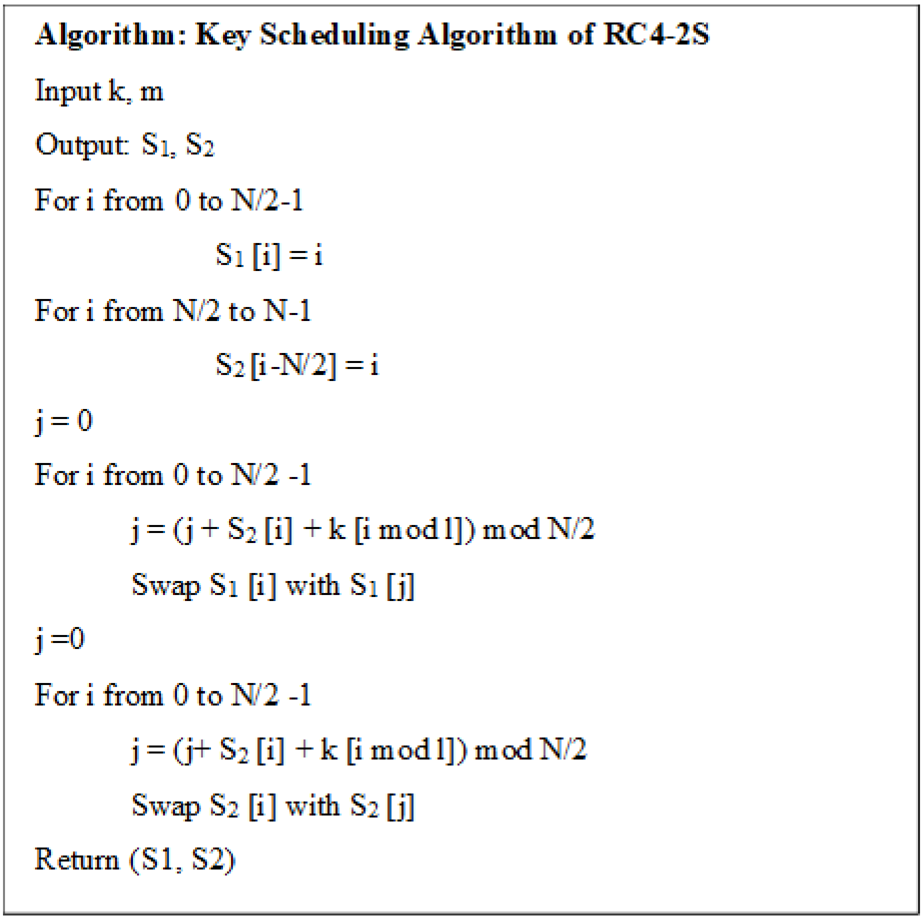
This proposed system is used the initial secret key value is a 128-bit key stream generated randomly. When the user sends the message to the receiver, the sender needs to type only the message. The user does not need to type the initial secret key manually. This system is automatically generating the randomness key for the initial key value for the KSA phase. This proposed system is used the encryption key as a session key, which is one random secret key used for only one text message to be encrypted. This does not use the same secret key rapidly for the message encryption process.

RC4-2S's PRGA phase generate the random key stream, and then the key stream is XOR-ed with the plain text. Then the cipher text will be generated after the encryption process. The cipher text has been sent over the network. The whole encryption process has been performed on the background automatically. The users do not need to perform manually encrypt and decrypt process.

### 3.1.1 Rivest Cipher 4 with State Tables (RC4-2S)

RC4-2S is an updated version of the original RC4 to improve data security [14]. The key generation time of the RC4-2S is faster than the original RC4 because it generates a pair of key streams after taking two swaps and five modulo functions. This takes a loop cycle until half of the plain text sequences are reached. RC4 can generate only one key stream after one swap and taking three modulo functions. RC4-2S also contains the Key Scheduling Algorithm (KSA) phase and the Pseudo Random Generation Algorithm (PRGA) phase.

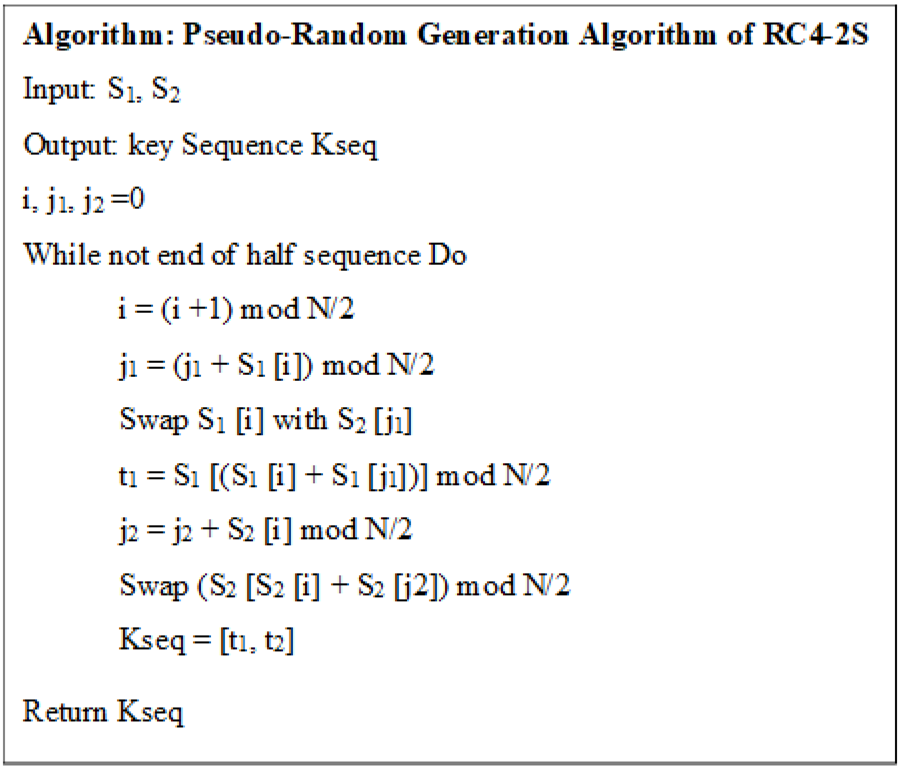
Swap elements by using two arrays in the KSA phase of the RC4-2S. Figure 3.2 describes the process of the KSA stage. The first array, $S_1$ is filled with the elements from 0 to 127, and the second array $S_2$ is filled with the remaining elements from 128 to 255. The initial key-value K is set with another array. After the swap operations, the generated S1 and S2 become two secret random inputs for the second phase (PRGA) of RC4-2S.



```
Algorithm: Key Scheduling Algorithm of RC4-2S
Input k, m
Output: $S_1, S_2$
For i from 0 to N/2-1
        $S_1 [i] = i$
For i from N/2 to N-1
        $S_2 [i-N/2] = i$
j = 0
For i from 0 to N/2 -1
    $j = (j + S_2 [i] + k [i \bmod l]) \bmod N/2$
    Swap $S_1$ [i] with $S_1$ [j]
j =0
For i from 0 to N/2 -1
    $j = (j+ S_2 [i] + k [i \bmod l]) \bmod N/2$
    Swap $S_2$ [i] with $S_2$ [j]
Return (S1, S2)
```

**Figure 3. 2 RC4-2S Key Scheduling Algorithm**

Figure 3.3 described the process of the PRGA phase. In the PRGA phase, swap the elements of $S_1$ and $S_2$ by using the three-pointers. Each loop cycle produces two keys. The encryption process is done that creates the cipher text by XOR-ing the plaintext and the generated key-stream. Block diagram of the RC4-2S is described in Figure 3.4



```
Algorithm: Pseudo-Random Generation Algorithm of RC4-2S
Input: $S_1, S_2$
Output: key Sequence Kseq
$i, j_1, j_2 = 0$
While not end of half sequence Do
    $i = (i + 1) \bmod N/2$
    $j_1 = (j_1 + S_1[i]) \bmod N/2$
    Swap $S_1[i]$ with $S_2[j_1]$
    $t_1 = S_1[(S_1[i] + S_1[j_1])] \bmod N/2$
    $j_2 = j_2 + S_2[i] \bmod N/2$
    Swap $(S_2[S_2[i] + S_2[j2]) \bmod N/2$
    $Kseq = [t_1, t_2]$

Return Kseq
```

**Figure 3.3 RC4-2S Pseudo-Random Generation Algorithm**

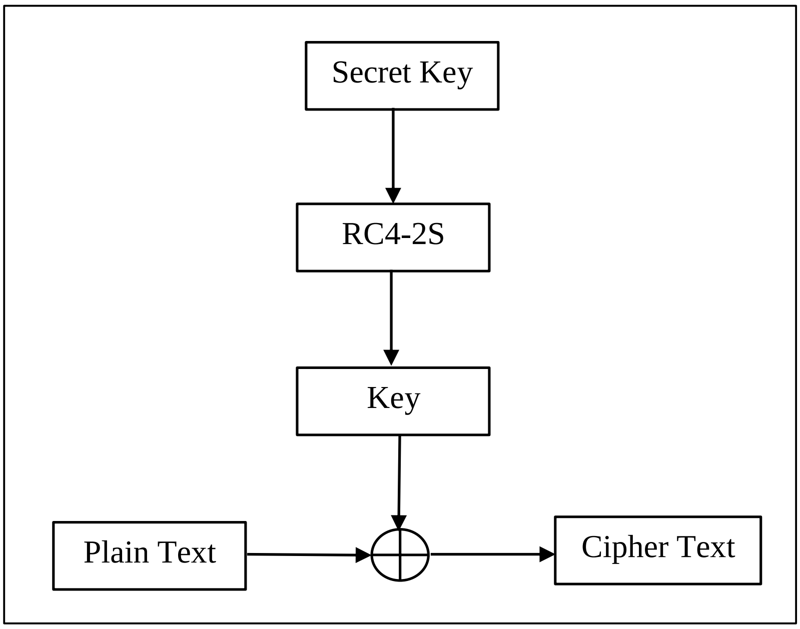


**Figure 3.4 Block Diagram of the RC4-2S Algorithm**

## 3.2 Proposed System Design

SMS communication over the network is not secure, therefore it is needed to reduce the risks by preventing hackers from accessing secure data and message. This proposed system is developed as an online chatting application, which use the encrypted messages to improve the data security by using the RC4-2S stream cipher algorithm. The detail system flow diagram describes in Figure 3.5.

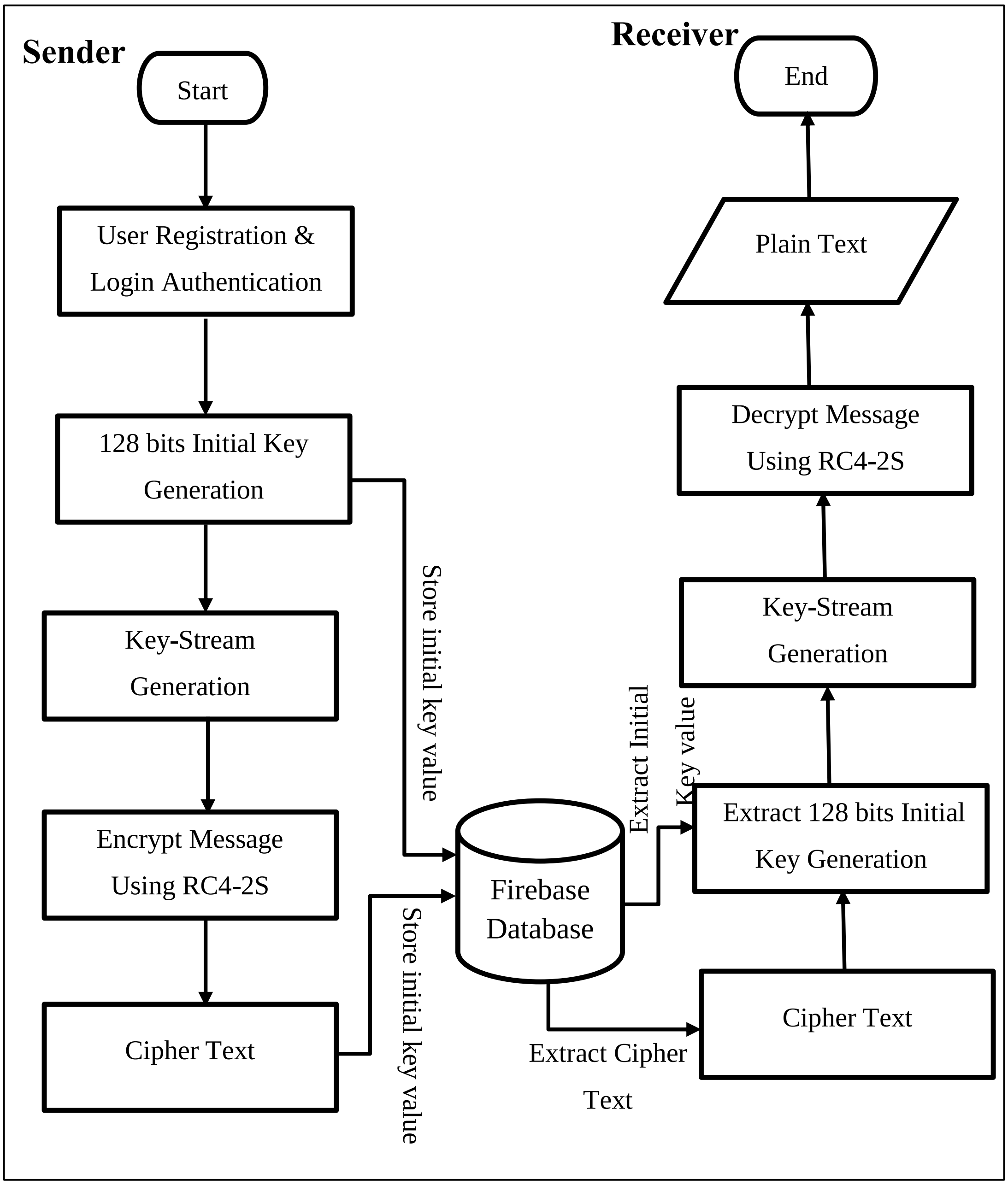
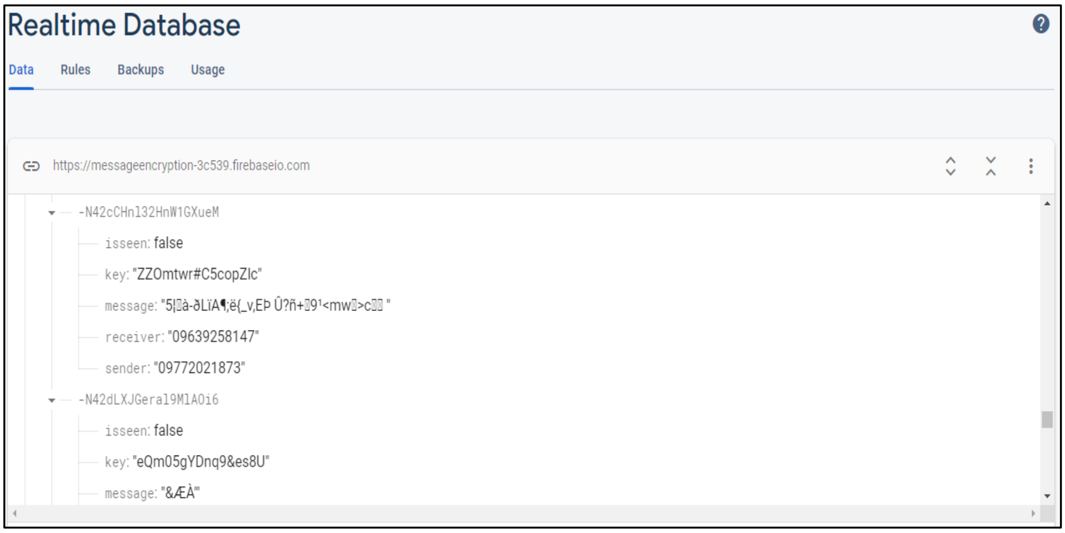

Sender
Start
User Registration & Login Authentication
128 bits Initial Key Generation
Key-Stream Generation
Encrypt Message Using RC4-2S
Cipher Text
Store initial key value
Store initial key value
Firebase Database
Extract Initial Key value
Extract Cipher Text
Receiver
End
Plain Text
Decrypt Message Using RC4-2S
Key-Stream Generation
Extract 128 bits Initial Key Generation
Cipher Text

**Figure 3.5 Flow Diagram of the Proposed System**

If the user is a first-time new user, the user will need to register this application using some information (email, phone number, and password). Otherwise, the user can directly log in with the registered information. In the registration phase, the user must not leave any text boxes empty. After the registration steps are successfully completed, the user must log in using the registered information.

If the user's authentication is completely successful in the login stage, the user can see the contact list and chat list. The contact list is the user that uses this secure SMS application. The sender can choose the receiver from this contact list. And then the sender can send the automatic encrypted message directly to the receiver without paying the SMS charge. The sender does not need to type the initial secret key value, which is automatically taken by the system background.

This system generates the 128-bit random key to use as an initial secret key for the KSA phase. The RC4-2S KSA phase swaps the array elements based on the initial key value. After the KSA swapping phase, that generates two arrays to take as an input for the PRGA phase. In the PRGA phase, that takes permutation operations and generates the random key-stream. The plain text is changed to the cipher text by XOR-ing with the key-stream. The secure SMS system sends this cipher text to the recipient and stores it, and along with the initial key value, in the firebase real-time database for decryption. Figure 3.6 describes the data storage format in the firebase real-time database.



```
https://messageencryption-3c539.firebaseio.com
-N42cCHnl32HnW1GXueM
    isseen: false
    key: "ZZOmtwr#C5copZlc"
    message: "5¦▯à-ðLïA¶;ë{_v,EÞ Û?ñ+▯9¹<mw▯>c▯▯ "
    receiver: "09639258147"
    sender: "09772021873"
-N42dLXJGeral9MlA0i6
    isseen: false
    key: "eQm05gYDnq9&es8U"
    message: "&ÆÀ"
```

**Figure 3.6 Chat Message in Firebase Database**

The receiver must log in with the information used during the registration stage. If the receiver is an authorized user, the receiver accepts the message as cipher text. When the user taps this cipher text, the background system automatically extracts the key and the cipher text from the firebase database for the decryption process. After that, use this cipher text and this initial key value as an input for the KSA phase and then generate the random key stream after taking the PRGA phase. The cipher text is changed to plain text by XOR-ing the cipher text and the generated random key stream.

### 3.2.1 User Registration and Login Authentication

The proposed system can divide into four parts: the first one is user registration and login authentication, the second is the key-stream generation, and the third one is message encryption and the last is the message decryption. User registration and login authentication processes are described in Figure 3.7.

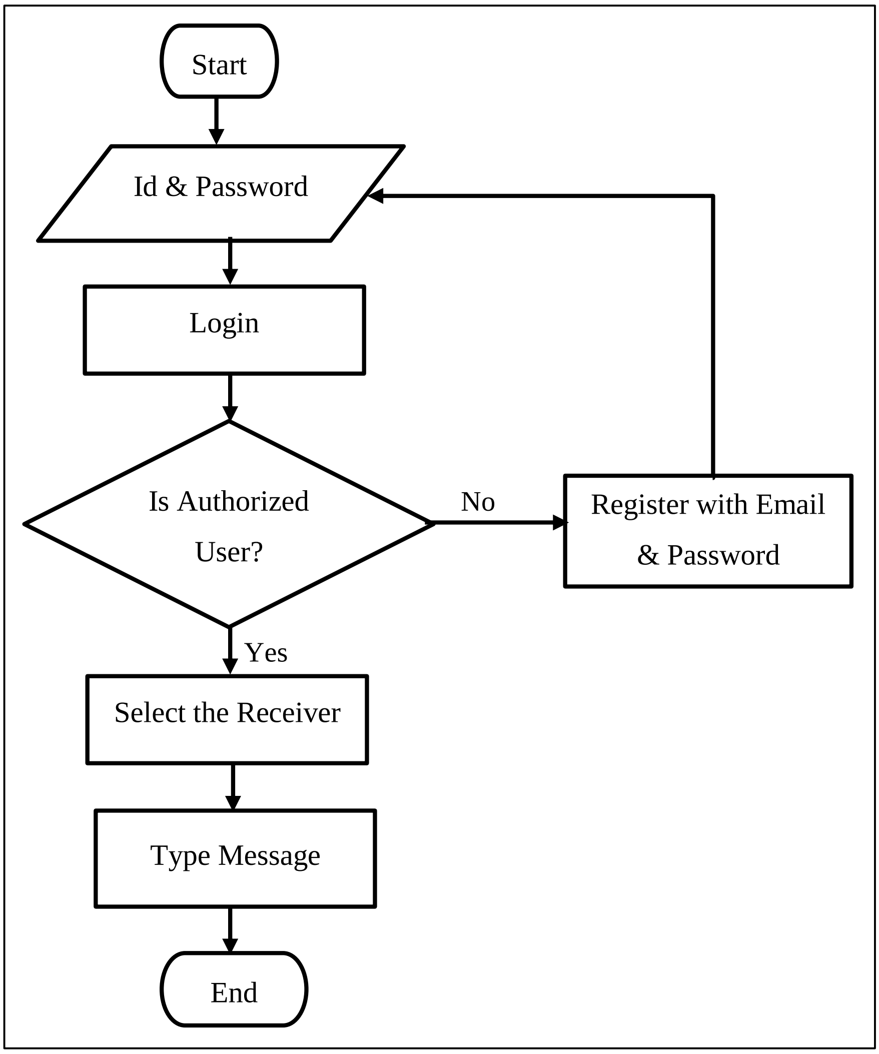


**Figure 3.7 Flow Diagram for the User Registration and Login Phase**

When the user registers this application, the user must not leave any empty text box and must fill with your email address, password, and a phone number that starts with “09” and does not exceed 11 numbers. And then the password must have at least 6 characters (that can be used with any letter character to create a stronger password). This password is only used to launch the application by logging in with this password, which does not relate to the encryption key in the KSA phase. User registration and log-in authentication processes are applied to the Firebase authentication feature. If the user registration process is successfully completed, the user can login using email and password. If the user enters correct id and password, the login authentication process is successful. Then, the user can now send the message to the receiver. The proposed system used the email password login method of the Firebase authentication feature for user login process and the user registration process. This feature can offer the account recovery process and generate the user ID for the registered user.

### 3.2.2 Key Stream Generation

The initial secret key is automatically generated by the system that has 128 bits (16 characters). This secret key and the plain text are fed into the KSA phase, which performs many swapping operations based on the secret key. After the operation is successfully completed, two arrays are generated. A flow diagram of the KSA phase is shown in Figure 3.8. In the KSA phase, firstly initialize the two pointers value, and then two arrays are filled with the elements from 0 to 127 and 128 to 255 respectively. After that the KSA phase update the initialized value and take the swapping operations until the 128 times. After all swapping operations are successfully completed that generate the two random states.

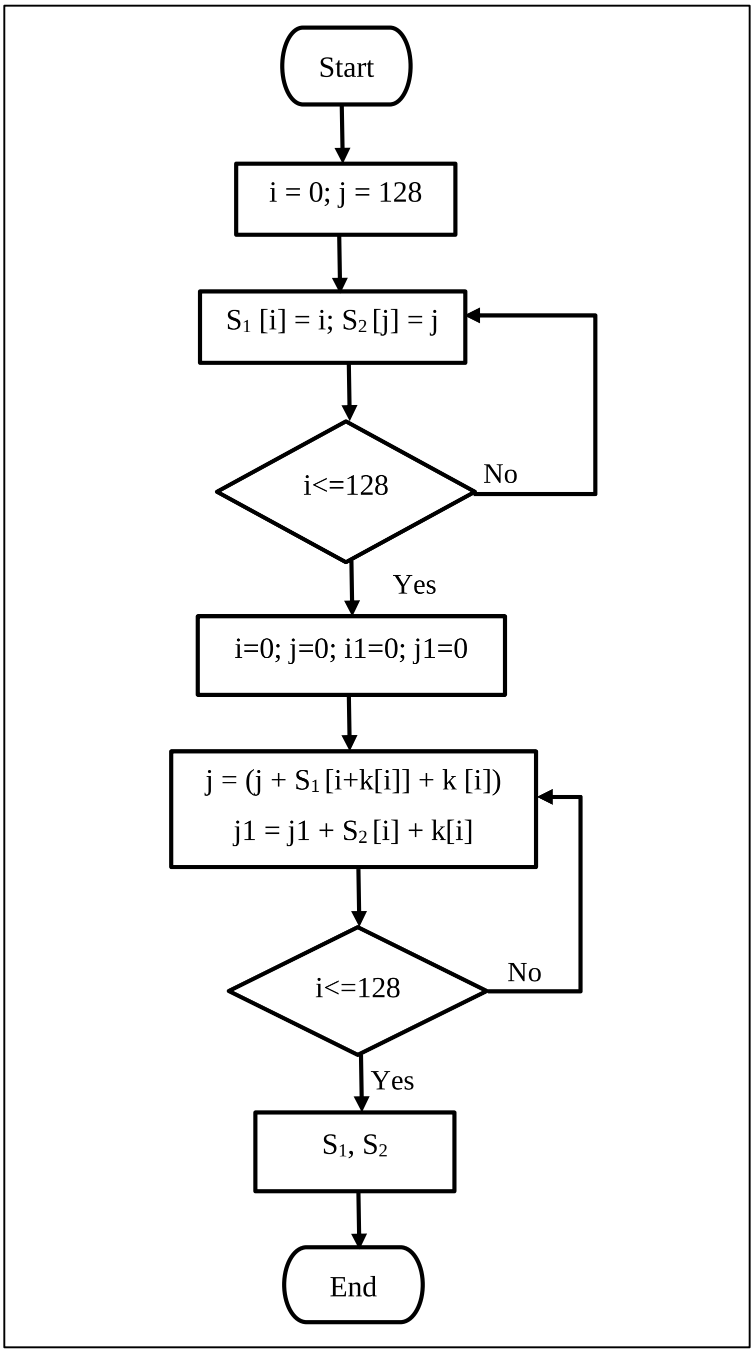


**Figure 3.8 Flow Diagram of the KSA phase**

The PRGA phase uses these two arrays as input. Figure 3.9 describes the flow diagram of the PRGA phase. The PRGA phase performs the permutation and swapping operation are based on the three pointers. Firstly, initialize these three pointers. It takes the swapping operation on the array $S_1$ and then generate the first one-bit key stream. And also take theses swapping operations on the second arrays $S_2$. After one loop cycle that generates a pair of keys. The random key-stream is generated after taking a loop cycle until half of the plain text sequence is reached.

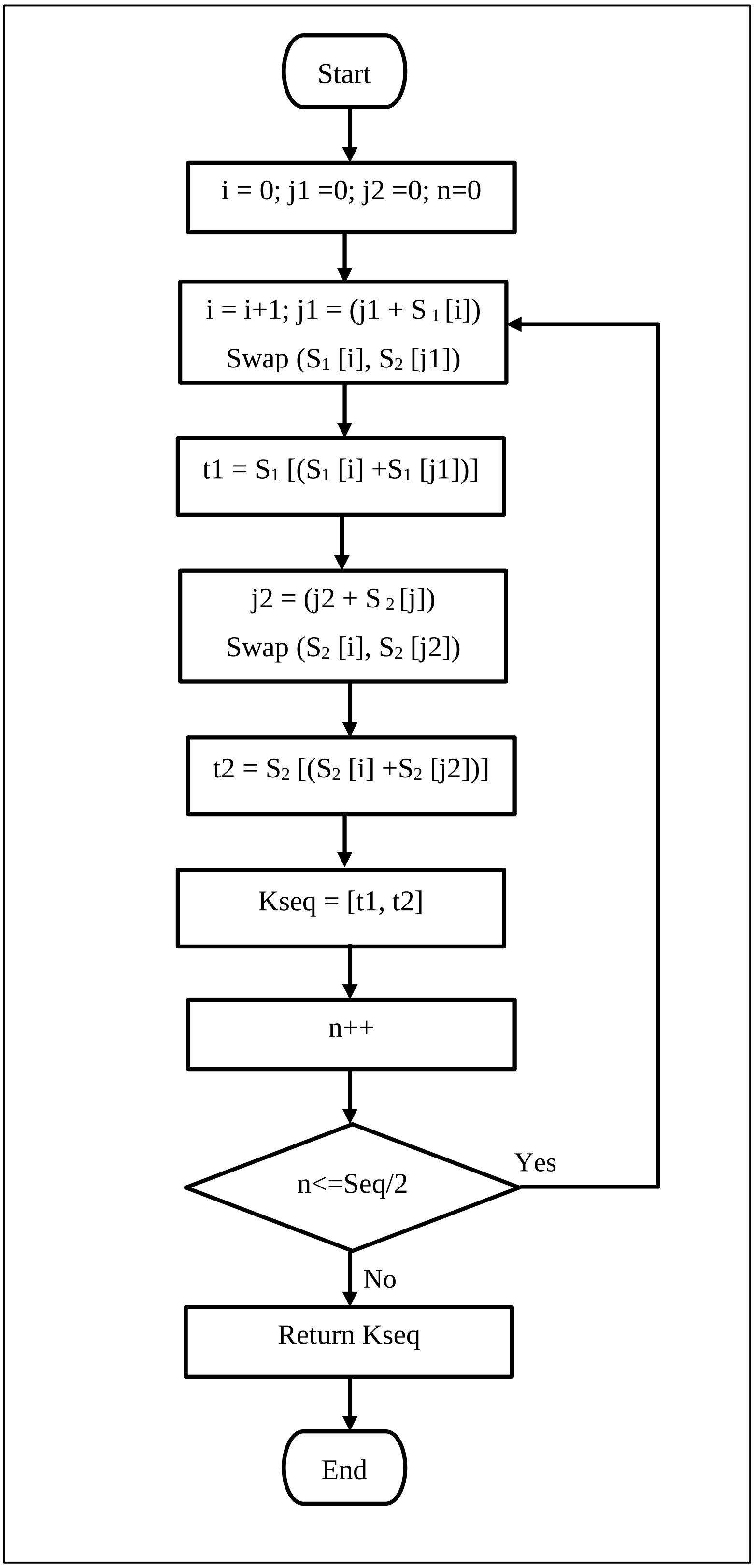


**Figure 3.9 Flow Diagram of the PRGA Phase**

### 3.2.3 Message Encrypting

This phase has two sub-portions. The first is the encrypted message, and the second is sending the cipher text. . The flow diagram for the encrypting message phase is shown in Figure 3.10.

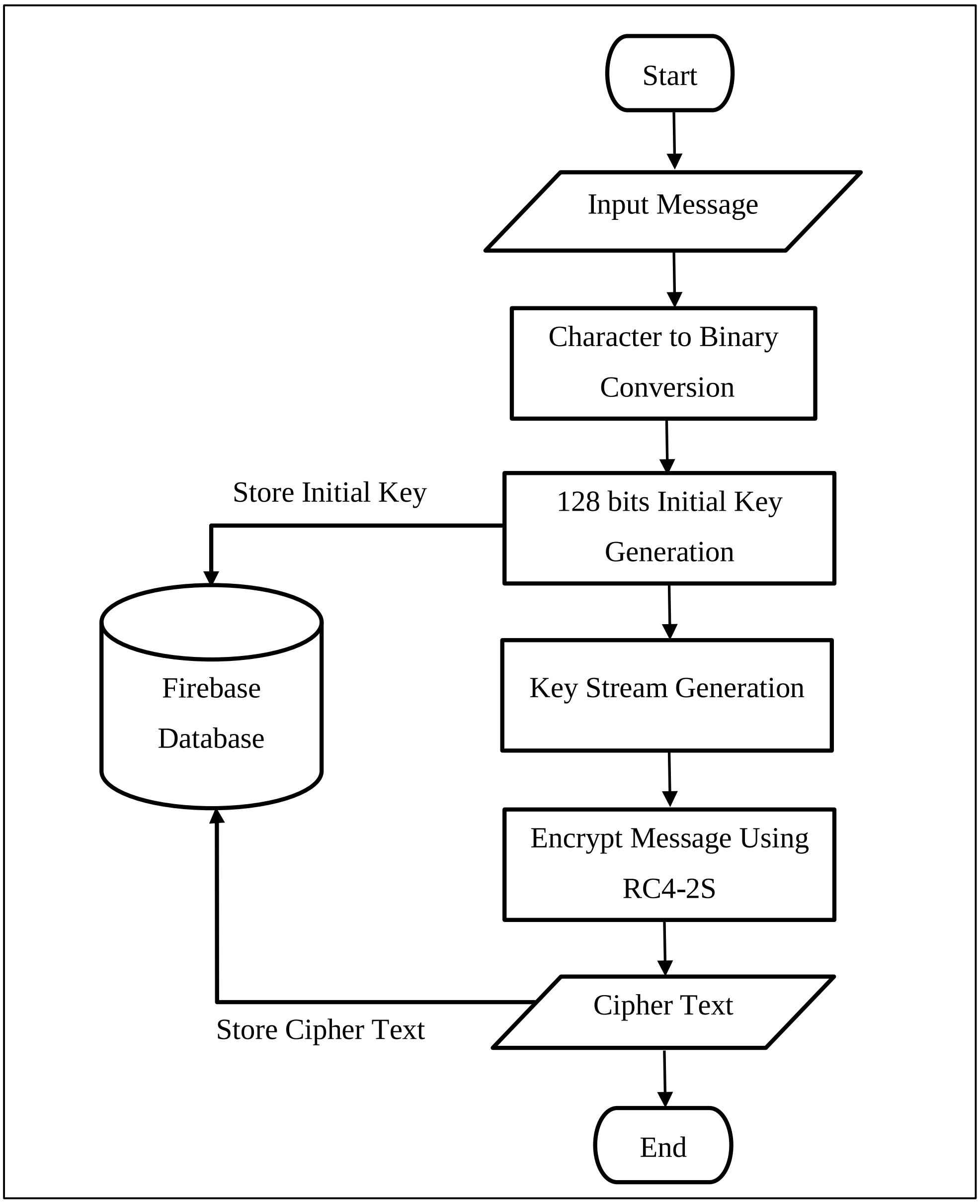


**Figure 3.10 Flow Diagram of the Message Encryption Process**

After the key stream generation step is successfully completed, the PRGA phase generates the random key stream. The plain text string value is converted into the decimal value of each character. After that, the key- stream is used to generate the cipher text by XOR-ing with the plain text. And then, the generated cipher text is sent directly to the receiver. The generated cipher text and the initial key values are store in firebase database for the decrypting process

### 3.2.4 Decrypting Message

Finally, the receiver needs to login to read the message from sender. Figure 3.11 describes the flow diagram of the decryption message. If login authentication is successfully complete, the user can decrypt the message.

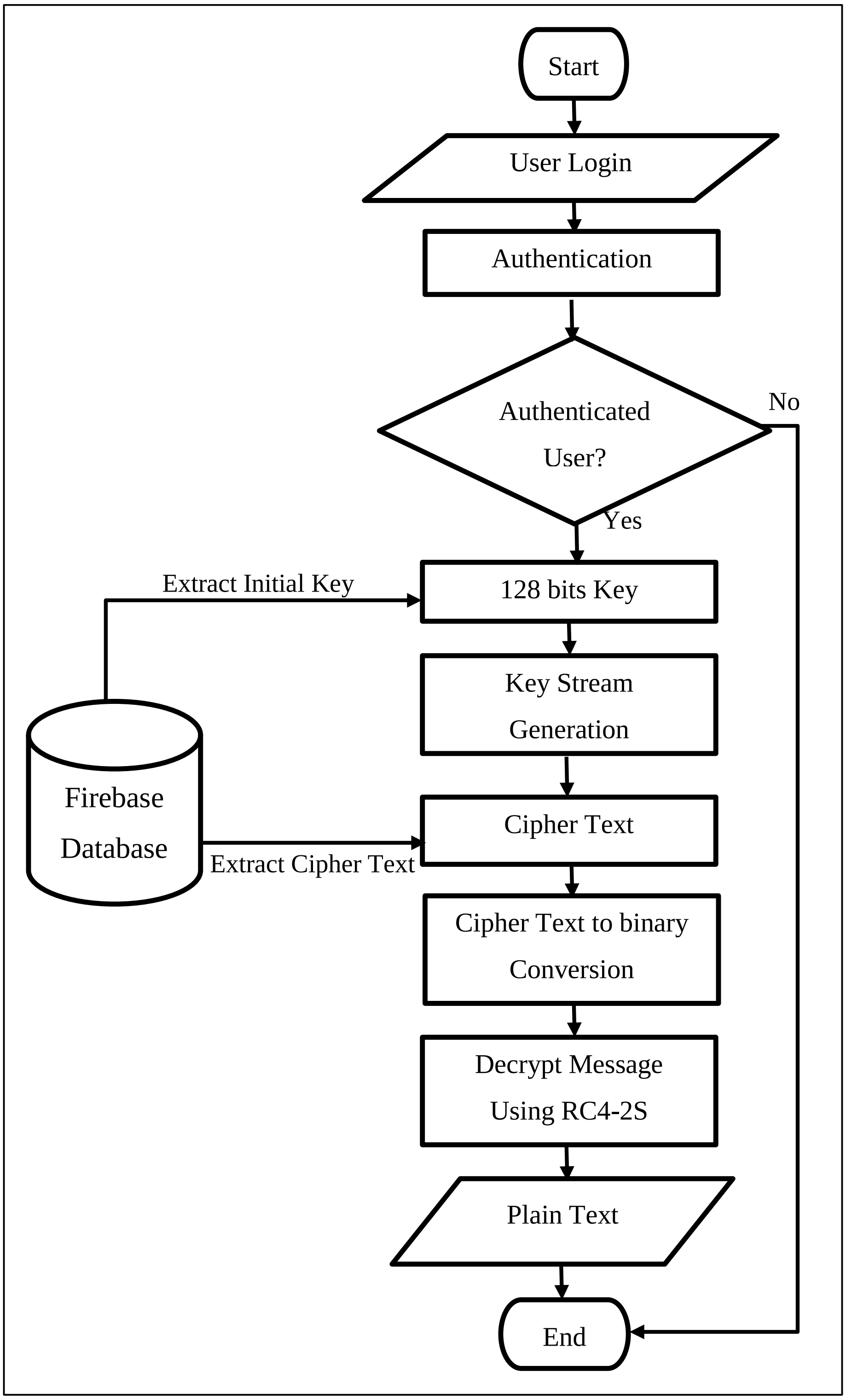


**Figure 3.11 Flow Diagram of the Decrypting Message Process**

The receiver received the cipher text, it must be decrypted to see the original text message. In this decrypting process, the cipher text and the initial key value are extracted from the Firebase database and used them to perform the KSA phase and PRGA phase based on these two inputs. The cipher text is also converted to the decimal value for each character. After that, the plain text is generated by XOR-ing the cipher text and the generated key stream. At that time, the user can see the original plain text.

## 3.3 Chapter Summary

This chapter explains the detailed processes of the proposed system, the main methodology of this proposed system, and the four parts of the proposed secure messaging system in detail. These parts are explained clearly with the diagram. The next chapter will show the results of the evaluation of the system.

# CHAPTER 4

# IMPLEMENTATION AND EXPERIMENTAL RESULTS

SMS security has become critical in SMS communication and transactions because of the advancement of technology for many chat applications that cannot be made into an SMS service to late. This system is implemented as a secure message chatting application to provide the security of the original SMS system. This section describes the system implementation and user interface for secure message transmission.

## 4.1 System Implementation and User Interface Design

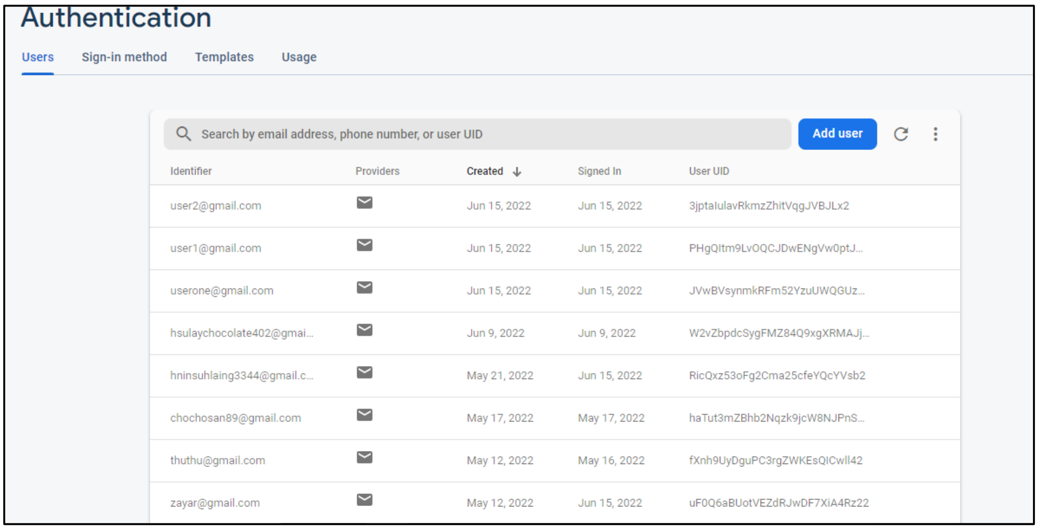
This proposed system is implemented using Java programming language and can run on an Android smart phone. Based on the simple SMS architecture, this system combined with the RC4-2S stream cipher encryption algorithm. Table 4.1 describe the software requirements and hardware requirements for developing this software.

**Table 4.1 System Specification for Developing Secure SMS Application**

| Name | Specification |
|---|---|
| Android Studio | Version 4.1 |
| PC | ▪ Acer Intel® Core i3<br>▪ RAM :8GB<br>▪ 64 bits operating system |
| Android Phone | Oppo A54 (Android Version:11) |

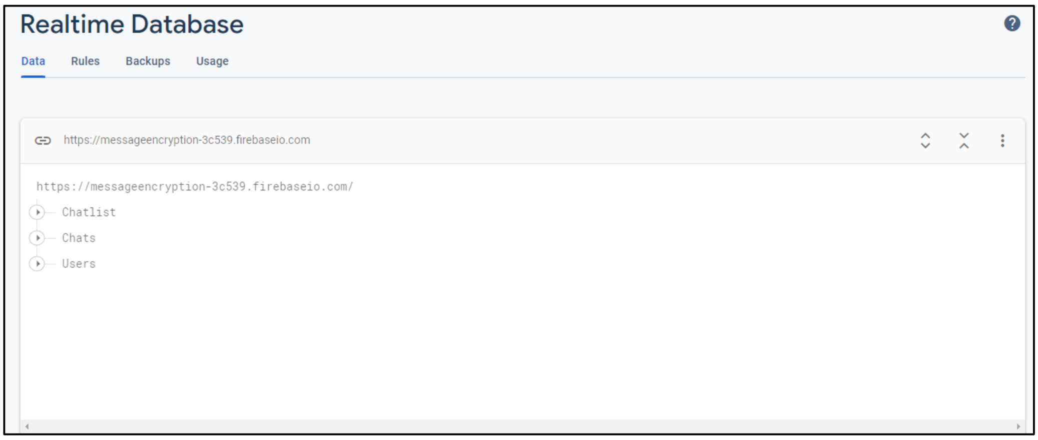
This system is developed on these requirements by using the Java programming language as an android application. And then this proposed system used two Firebase features: Firebase authentication feature and Firebase real-time database feature. Firebase is an app development platform developed by Google that enables developers to develop mobile and web applications Analysis, Authentication, Cloud Computing, Real-time Database, Crashlytics, Performance, and Test Lab are just a few of the services provided by Firebase.

The user authentication feature provides backend services, that can be used in many types of authentication ways, such as password login, phone number login, Google login, and more. This is easy to use for developers to develop the application. This feature can provide the user registration and login process and then this also handles sending password reset email. This proposed system is used the email password login authentication feature, thus the user can authenticate with their email and password. Shown in Figure 4.1 the authentication feature UI of the Firebase app platform.



| Identifier | Providers | Created ↓ | Signed In | User UID |
|---|---|---|---|---|
| user2@gmail.com | ✉ | Jun 15, 2022 | Jun 15, 2022 | 3jptaIulavRkmzZhitVqgJVBJLx2 |
| user1@gmail.com | ✉ | Jun 15, 2022 | Jun 15, 2022 | PHgQltm9LvOQCJDwENgVw0ptJ... |
| userone@gmail.com | ✉ | Jun 15, 2022 | Jun 15, 2022 | JVwBVsynmkRFm52YzuUWQGUz... |
| hsulaychocolate402@gmai... | ✉ | Jun 9, 2022 | Jun 9, 2022 | W2vZbpdcSygFMZ84Q9xgXRMAJj... |
| hninsuhlaing3344@gmail.c... | ✉ | May 21, 2022 | Jun 15, 2022 | RicQxz53oFg2Cma25cfeYQcYVsb2 |
| chochosan89@gmail.com | ✉ | May 17, 2022 | May 17, 2022 | haTut3mZBhb2Nqzk9jcW8NJPnS... |
| thuthu@gmail.com | ✉ | May 12, 2022 | May 16, 2022 | fXnh9UyDguPC3rgZWKEsQICwll42 |
| zayar@gmail.com | ✉ | May 12, 2022 | Jun 15, 2022 | uF0Q6aBUotVEZdRJwDF7XiA4Rz22 |

**Figure 4.1 User Authentication Feature of the Firebase**



```
https://messageencryption-3c539.firebaseio.com/
  Chatlist
  Chats
  Users
```

**Figure 4.2 Real-time Database Feature of the Firebase**

Firebase Real-time database feature is shown in Figure 4.2. The real-time database feature is a cloud-hosted database. Data is stored as JSON and synchronized in real-time to every connected device. All of the clients share one real-time database instance and automatically receive updates with the newest data.

## 4.2 User Interface of the Proposed System

If the user is a new user, the new user needs to be registered by clicking the register button. Figure 4.3 shown the registration form of this proposed system.

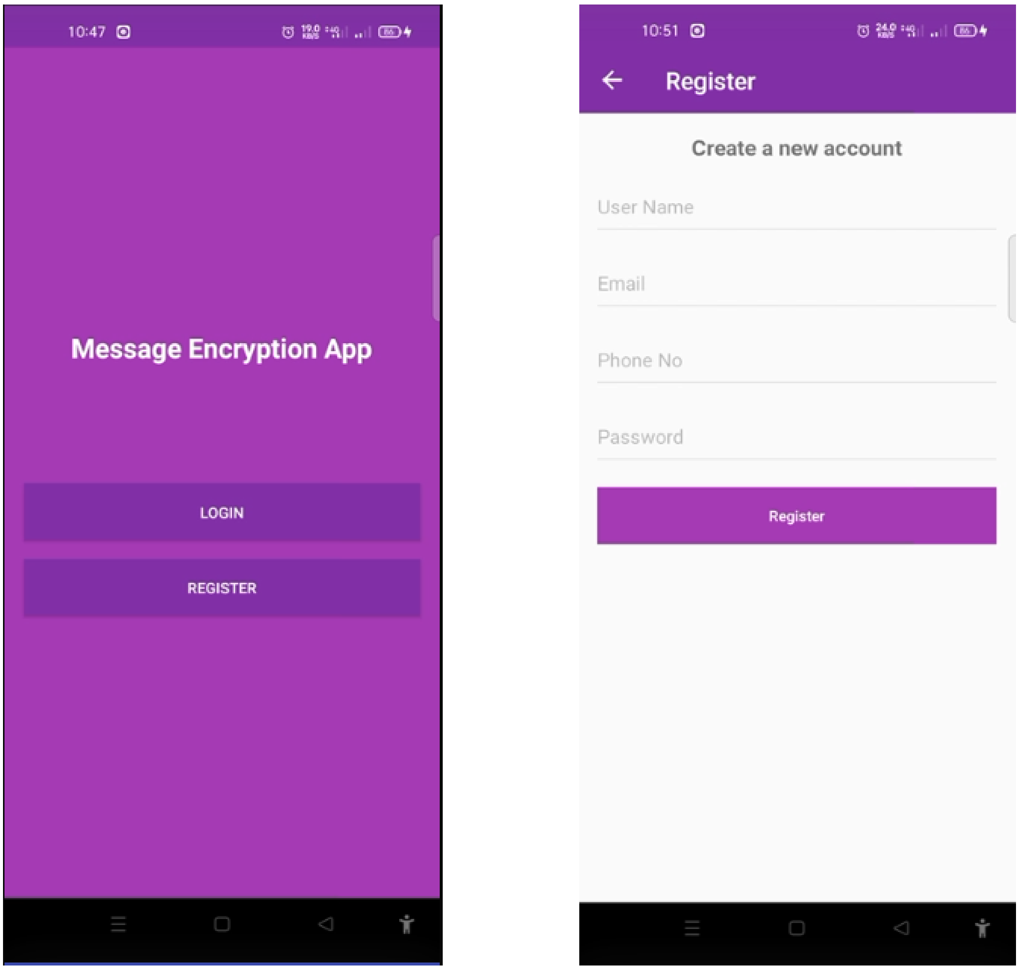
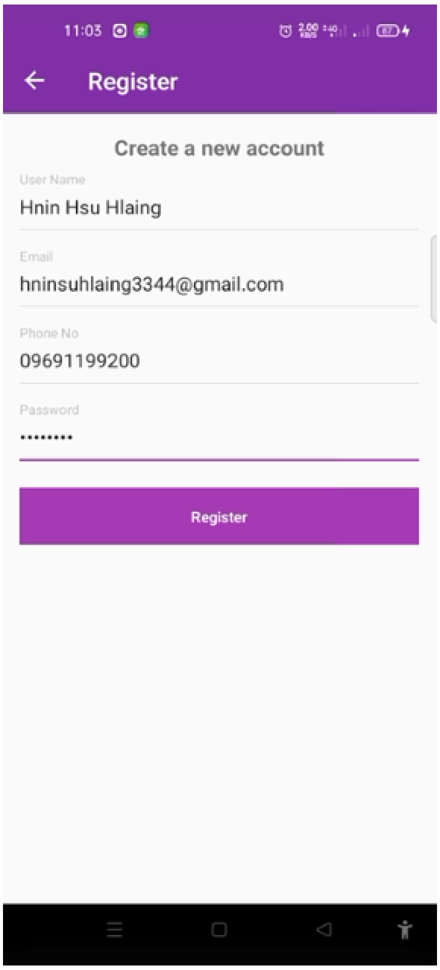


**Figure 4.3 User Registration Page**

After that, the user needs to fill in all the text boxes on the registration page. If all the fields are not filled in, it cannot complete the registration state. The first text box is filled with the user name (that is displayed on your chatting screen), the second is filled with email, and then the third is the user’s phone number (in this case, the user’s phone number exactly starts with 09), and the last text box is the password that is used to enter the application (the password is at least 6 characters). Shown in Figure 4.4 the registration form with the user’s information.



Create a new account

User Name
Hnin Hsu Hlaing

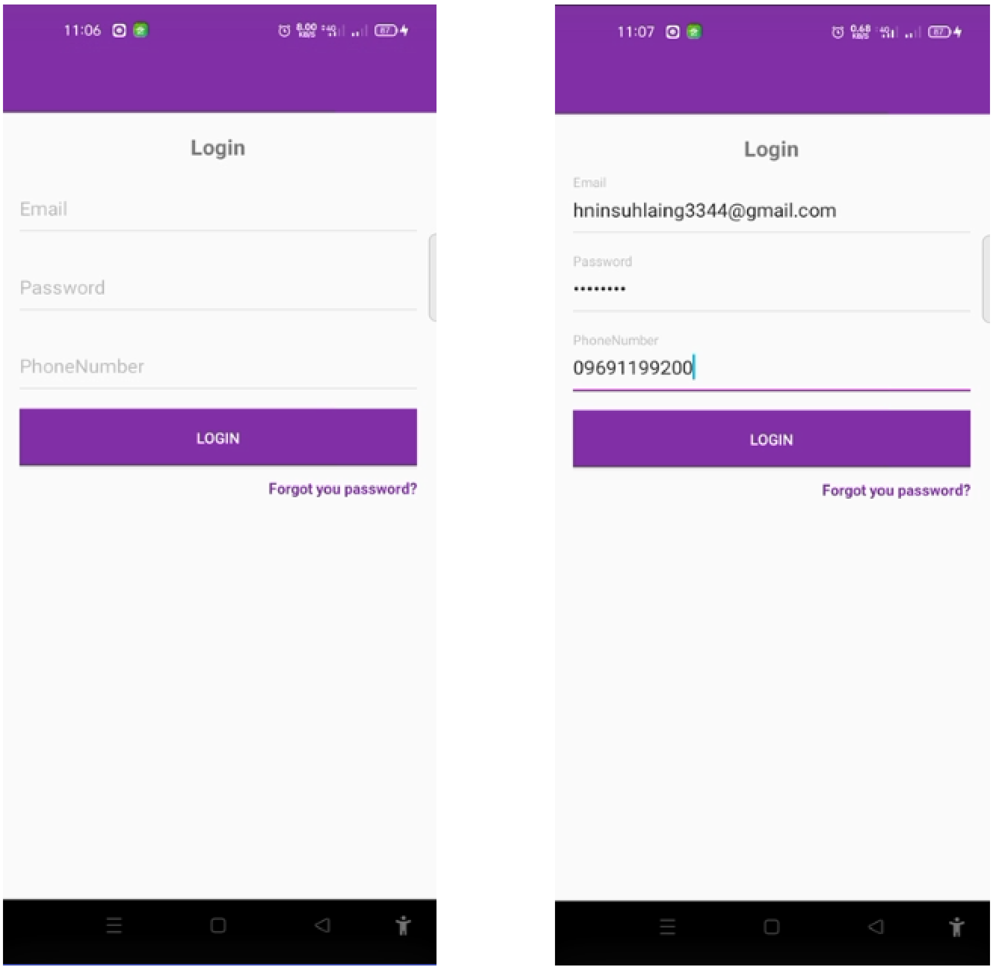
Email
hninsuhlaing3344@gmail.com

Phone No
09691199200

Password
••••••••

**Figure 4.4 User Registration Page with User's Information**

After the user completes the registration task, the log in page will appear. Shown in Figure 4.5 the login form of the proposed system. Then the user can login with their username and password. If the user is not a new user, skip the registration task. User can login directly by clicking the login button.



Email
hninsuhlaing3344@gmail.com

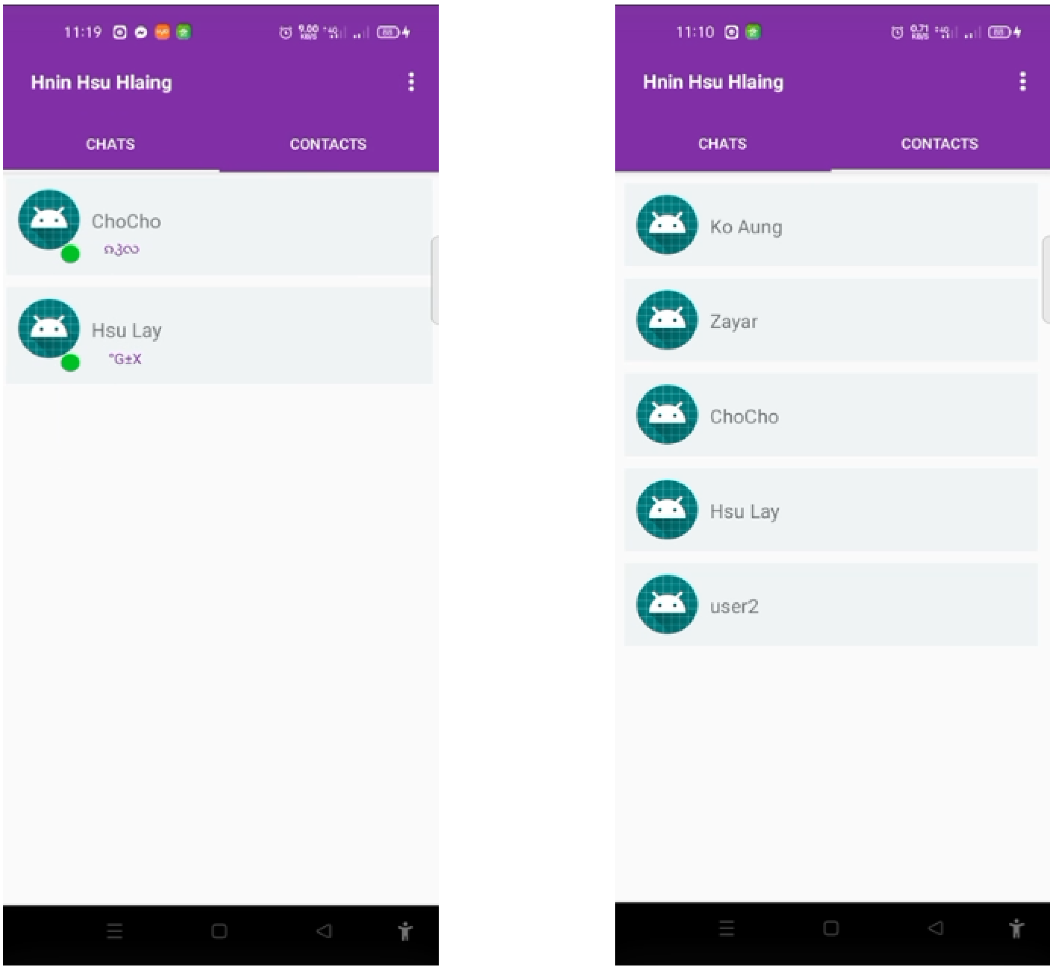
Password
••••••••

PhoneNumber
09691199200

**Figure 4.5 Login Page and fill with the Registered Information**

On the login task, the user needs to login with email, phone and password (with the registered email, phone number, and password). In this state, the user cannot leave empty in all text boxes. And click the login button. If the user's info is correct, the user can send the message. This proposed system used the email password login feature of the Firebase Authentication Feature.

After completing the previous steps, the authorized user can send the message securely to the receiver. The user can see the contact list and chat list. Contact list shows the entire user using this app. And in the chat list, it shows the conversation list of the user. The user chooses the receiver from either the contact list or chat list. Figure 4.6 show the content list and chat list form of the proposed system.



11:19

Hnin Hsu Hlaing

CHATS CONTACTS

ChoCho
ฑ3cc

Hsu Lay
°G±X

11:10

Hnin Hsu Hlaing

CHATS CONTACTS

Ko Aung

Zayar

ChoCho

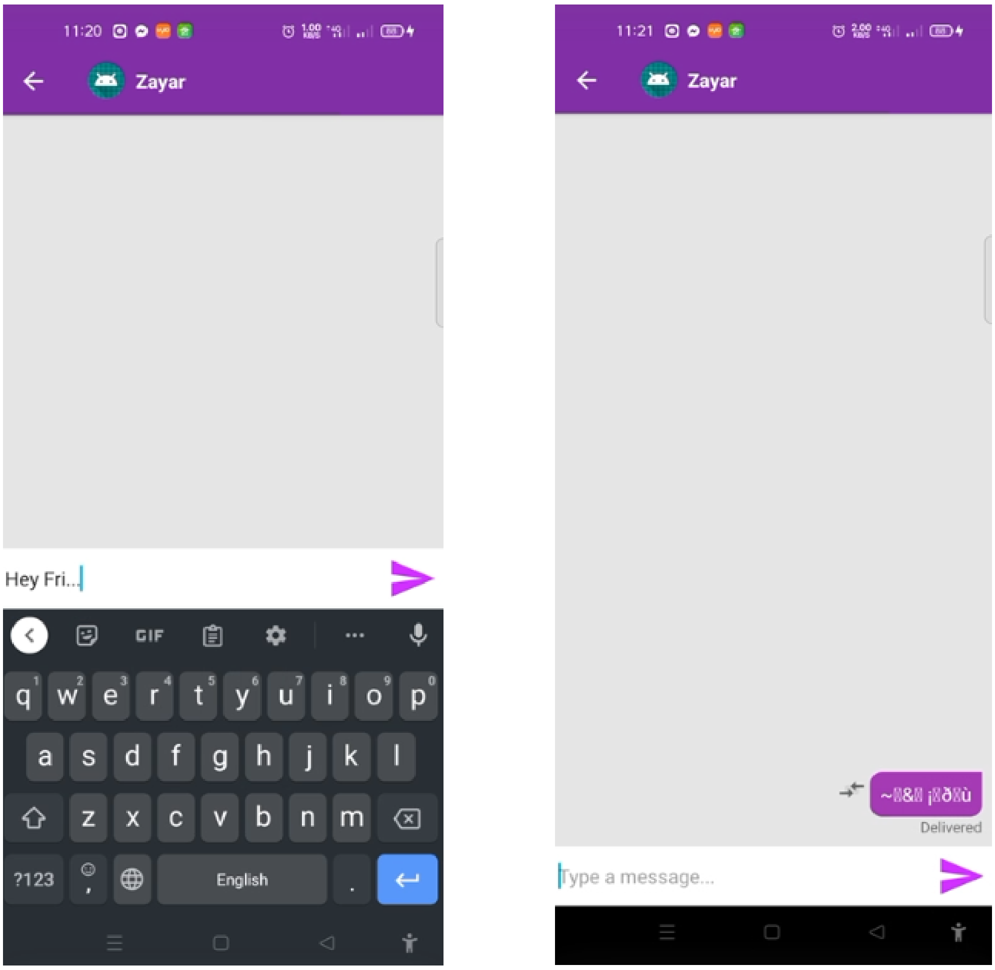
Hsu Lay

user2

**Figure 4.6 User Contact List and Chat List**

When the user taps the receiver on the chat list or contact list, the user can send a message. Figure 4.7 shows the message activity of the proposed system. In this stage, which is very simple, the user types the text message that they want to send and then clicks the send button.

If the user clicks the send button with the empty message text, the system shows the alert with the information that the user needs to type the message in the text box. And the system does not process the encryption process. Otherwise, an automatic encrypted message is sent directly to the receiver. The sender must see the cipher text in his chat list. After that, this user’s conversation is bind in the chat list.



Hey Fri...

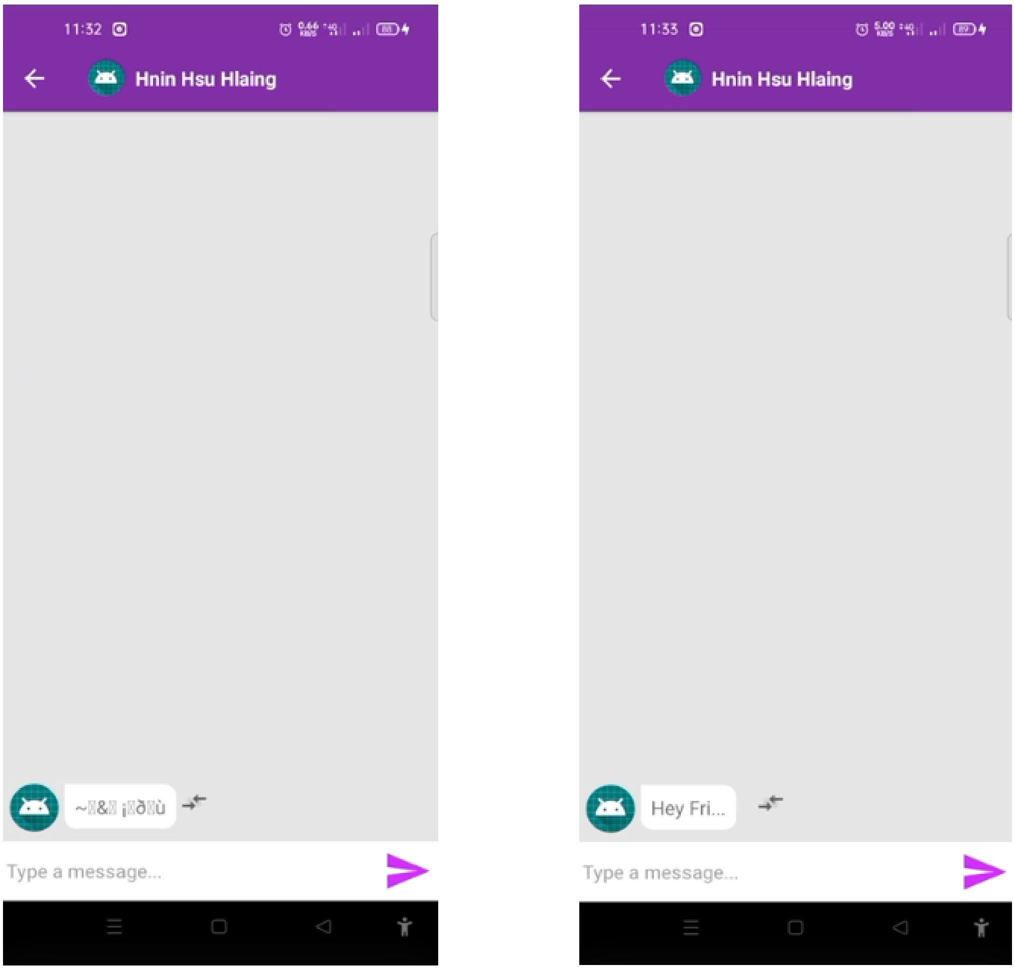
~▯&▯ ¡▯ð▯ù

Delivered

Type a message...

**Figure 4.7 Send Message Activity**

On the receiver side, the receiver receives the message as cipher text on the chat list. Figure 4.8 shows the receive message form of the proposed system. When the receiver taps these messages, they reach the message activity. When taping this cipher text, the background process extracts the initial key value and cipher text from the firebase database for the KSA stage and then generates plain text using the RC4-2S algorithm. The message delivery status is updated to the seen status. In the chat list, show the cipher text of the last message. Another person can’t see the message clearly a few times.



11:32

Hnin Hsu Hlaing

~▯&▯ ¡▯ð▯ù

Type a message...

11:33

Hnin Hsu Hlaing

Hey Fri...

Type a message...

**Figure 4.8 Receive Message at the Receiver Side**

## 4.3 Performance Evaluation

The quality of the data confidentiality relies on the randomness of the pseudorandom number. NIST test was developed at the National Institute of Standards and Technology that is used to validate the random number generator and pseudo-random number generator for cryptographic application. NIST randomness tests are developed to test the randomness of the binary sequence produced by the pseudo-random number generator. There are 15 statistical tests recognized by NIST to measure the randomness of the key stream [3]. NIST tests can be divided into three classes: The first is the test that processes each bit of the bit stream. Frequency, Block Frequency, Runs, Longest run, Cumulative sums, Random Excursion and Random Excursion Variant: these tests are carried out for each bit in the bit stream. This type is the fastest test. The second type of test is one that processes an m-bit block; this type of test is dependent on the m because each bit of the m-bit is tested, and the last is the slow and complicated test. This proposed system is tested with the frequency monobit test and run test of the first class.

### 4.3.1 Implementation of Frequency (Monobit) Test and Run Test

In many cryptographic applications, the randomness of the key stream is very prominent. Many cryptographic protocols require randomness input at various point of view. This section discusses the randomness testing of the key stream generated by the pseudo random number generator of RC4-2S.

**The Purpose of the Frequency (Monobit) Test**

The focus of this test is the component of zeros and ones for the entire sequence. The main purpose of this test is to decide whether the number of ones and zeros in a sequence are approximately the same would be expected for a truly random sequence. This is the number of zeros and ones in a sequence that should be the same foe random sequence [3]. All subsequent tests are depended on the passing of this test.

**Test Description**

Transformation to ±1: the zero and one of the binary sequences (Ɛ) are transform to values of -1 and +1. That values are added to produce $S_n = X_1 + X_2 + X_3 + \ldots + X_n$ .
For example: if Ɛ= 01010001 11011001, then n = 16
Compute $S_n$ is assumed by

$$\begin{aligned} S_n &= (-1) +(+1) + (-1) +(+1) +(-1) +(-1) +(-1) +(+1) +(+1) +(+1) +(-1) +(+1) \\ &\quad +(+1) +(-1) +(-1) +(+1) \\ &= (-10) + (6) \\ &= -4 \end{aligned}$$

Compute the test statistic is assumed by

$$S_{obs} = \frac{|S_n|}{\sqrt{n}}$$

$$S_{obs} = \frac{|-4|}{\sqrt{16}}$$

$$S_{obs} = 1$$

Compute the $P_{value}$ is assumed by, where $\boldsymbol{erfc}$ is the complementary error function

$$P_{value} = erfc(\frac{S_{obs}}{\sqrt{2}})$$

$$P_{value} = erfc( \frac{1}{\sqrt{2}})$$

$$P_{value} = erfc( 0.7071067812)$$

$$P_{value} = 0.31731050$$

By the decision rule, this sequence is randomness (ie., $P_{value} \geq 0.01$ ).

**Test Purpose of the Run Test**

The objective of this test is the total number of runs on the sequence, where a run is an uninterrupted sequence of identical bits. A run of length k consists of exactly k identical bits and is bounded before and after with a bit of the opposite value. The purpose of the test is to determine whether the number of runs of ones and zeros of various lengths is an expected for a random sequence. In particular, this test defines whether the oscillation between such zero and one is too fast or too slow [3].

**Test Description**

Example: The calculation of this sequence 01001101 11001001 then n= 16, π = 10/16= 5/8

Compute the test statistic

$$V_{n(obs)} = \sum_{k=1}^{n-1} r(k) + 1$$

V16(obs) = 1+1+0+1+0+1+1+0+0+1+0+1+0+0+1 +1=9

Compute the $P_{value}$ is assumed by

$$P_{value} = erfc( \frac{V_{n(obs)-2n\pi(1-\pi)}}{\sqrt[2]{2n}\,\pi(1-\pi)})$$

$$P_{value} = erfc\left(\frac{\left(9-2*16*\frac{5}{8}*\left(1-\frac{5}{8}\right)\right)}{\sqrt[2]{2*16}\,*\frac{5}{8}*(1-\frac{5}{8})}\right)$$

$$P_{value} = erfc(1.13137)$$

$$P_{value} = 0.109714$$

By the decision rule, this sequence is randomness (ie., $P_{value} \geq 0.01$ ).

### 4.3.2 Performance Evaluation of Testing

The evaluation results of some plain tests calculated as follow. By the decision rules, $P_{value}$ are not smaller than 0.01. Therefore, the proposed system is strong for SMS data security that using RC4-2S encryption algorithm.

**Experiment 1**

Plain text: Master of Computer Science (M.C.Sc)

Secret Key: “Ca@3S1&J4bQ@@#SONJYE”

Random Key-stream (byte): IaÞýÐ;¥º@©üã©[aNÅÆ¥x8y'O6…

Random Key-stream (bit):

011000001000101100001000110010100110101001000010001010010010001000110
111110101000000011111111100101110110110110110001111101000101100000011
111001011100001011000111000011110000010111110010000101000100110001110
101100011100100110101000001110101011011101100110001101110100101010011
001110100011

Cipher Text: `ÊjB)"7Ôþ]¶ÇÑ`ùp±ÃÁ|…c“PunÌn•3£

Frequency test: 0.6826894772086507

Run test: 0.9999999999987955

**Experiment 2**

Plain text: master of computer science (M.C.Sc)

Secret Key: “fKJILqnVS7Osdz0YIgr!”

Random Key-stream (byte): ;°LÒÂyL.þâä.ê®-B½Á ²=÷zSÉx×P³¨)c§f¤H

Random Key-stream (bit):

000110011001111001001011111010000000101101001001100000101011000010010 1
100111010100100110111001100001010111011000000111000111010000110110010
010011110110111100110001010001110100001001101010100110111100101001110
000101101110000010000011011110101000111101101010001001010001000001110
000100001101

Cipher Text: ž Kè“a,êMÌ+°8èl“ÛÌQÐš¦òœ-Áõ-ÔJ á

Frequency test: 0.9332469895575373

Run test: 0.9999999999995479

**Table 4. 2 Performance Evaluation for NIST Test and Diffusion Property**

| | **Experiment 1** | **Experiment 2** |
|---|---|---|
| Plain Text | Master of Computer Science (M.C.Sc) | Master of Computer Science (M.C.sc) |
| Initial Key | ZhW%iwlVNEAwlST8 | WZOmtwr#C5copZlc |
| Cipher Text | N•M™$Zå®3‹ÖÏ%ÅU¬Iô9<br>• ¼}ÆW«2ÉQì$ | ž 5¦à-ðLïA¶;ë{_v,EÞ<br>Û?ñ+□9¹<mw‰>c„’ |
| Frequency test | 0.9803693443109074 | 0.866385596081871 |
| Run test | 0.9799999999904022 | 0.991087297390982 |

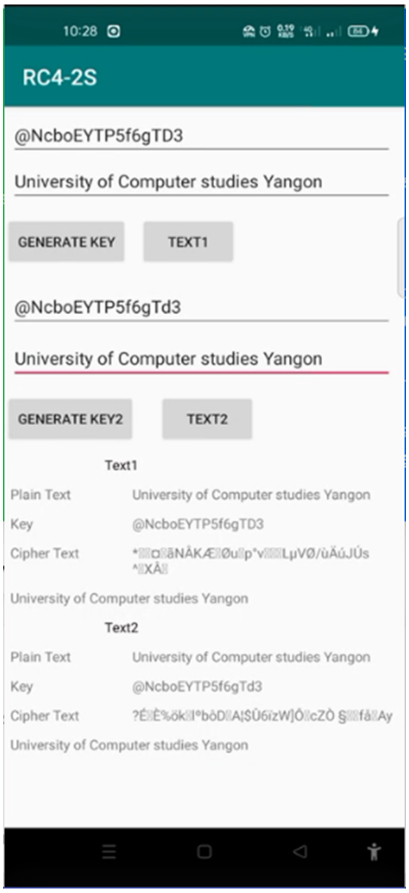
Table 4.2 describes the comparison of these two experimental results. In this used the plain text with a little change (last "s" is change to small letter). Although one bit of the plain text changes the generated cipher texts are not related (that has many changes). Therefore, this provides the diffusion properties. And this proposed system is used different initial secret key value for one message. The generated cipher text and the key are not related; therefore, this proposed system is also provided the confusion properties.

**Table 4.3 Performance Evaluation for NIST Test and Confusion Property**

| | **Experiment 1** | **Experiment 2** |
|---|---|---|
| Plain Text | Confusion Property | Diffusion Property |
| Initial Key | YJiWZpUYMxXRgSDf | YJiWZpUYMxXRgSDF |
| Cipher Text | U¶ZáÿµÉÒ\|©B£l•  #—& | ý��Ë`º+Œ=ò$Ý” X´]®w |
| Frequency test | 0.8202874962326072 | 0.1769366921873763 |
| Run test | 0.9999997163130911 | 0.9999998691714724 |

Figure 4.9 and Table4.3 show the comparison between two tests. These two tests are used the same key with a little change. The test2 is use the small letter D instead of the capital latter D in the key of test2. It is also a little change but the cipher texts of

these two tests are not related (very difference). The cipher texts are not correlative with the secret key. Therefore, this system has key sensitivity property and provide the confusion property of the cryptography.



@NcboEYTP5f6gTD3

University of Computer studies Yangon

@NcboEYTP5f6gTd3

University of Computer studies Yangon

Text1

Plain Text University of Computer studies Yangon

Key @NcboEYTP5f6gTD3

Cipher Text *□□¤□ãNÅKÆ□Øu□p°v□□□LµVØ/ùĂúJÚs ^□XÂ□

University of Computer studies Yangon

Text2

Plain Text University of Computer studies Yangon

Key @NcboEYTP5f6gTd3

Cipher Text ?É□È%ök□I°bôD□A¦$Û6ïzW]Ô□cZÒ §□□fâ□Ay

University of Computer studies Yangon

**Figure 4. 9 Comparing the Test Result**

## 4.4 Chapter Summary

In this section mention about the 15 NIST test suits and by manually calculate the frequency test and run test with example key-stream. And also test the random key-stream of the proposed system and show this proposed system is provided the confusion and diffusion properties with two experimental results. That is clearly pass both frequency test and run test.

# CHAPTER 5

# CONCLUSION

Nowadays, new technology trends are well-developed. However, SMS has become absolutely ubiquitous. SMS technology is very simple and does not encrypt the data during transmission between the sender and the receiver. Therefore, it can be intercepted by an unauthorized organization. The security of SMS data has become important. Access control plays an important role. To solve this insecurity, SMS should be used as an encrypted messaging system. The proposed system uses cryptography to encrypt the data by using one of the encryption algorithms, RC4-2S.

SMS messages are not encrypted over the network, so these messages could be intercepted and snooped during transmission. SMS theft is one type of cybercrime. Therefore, the proposed system uses a cryptographic system for data SMS to cover this cybercrime. Cryptosystems can understand the integrity of data, user authentication, and other security concerns. This proposed system uses the RC4-2S stream cipher algorithm to enhance the randomness of the key stream and to provide the security of the SMS. The key generation time is faster than the previous RC4 algorithm. The encryption and decryption times are directly related to the data file size if the data file is large enough.

In this thesis, a pseudo-random number key-stream is generated by using RC4-2S. It also, computes the randomness rate with a frequency (monobit) test and run NIST test suits. The RC4-2S key-stream is resoundingly passed by the testing result. Therefore, it is recommended that the user use the proposed system using the RC4-2S encryption algorithm that is implemented to ensure data security.

## 5.1 Advantages and Limitations of Proposed System

This system is implemented using the Java programming language on the Android Studio for Android Smart Phones. This application is online chat application and can launch this on the smart phone that have over the android version 5 (Lollipop). Users can use this application everywhere and anytime to access the internet. The user does not need to pay the SMS fees for a simple SMS. All the previous applications were developed in an offline mood, but this application was developed in an online mood. This system can use both the Myanmar language and the English language. The greatest point of this application is that the user can use it easily without having to follow confusing steps. This proposed system is implemented for Android smartphones; therefore, the user must have an Android OS phone. The user can send only text messages and no other types such as videos, photos, and audio files. This system is used by the Firebase real-time database as a storage system for the data. Therefore, this proposed system is secure until this firebase database platform is not broken.

## 5.2 Future Work

The technologies for many chat applications are developing rapidly, but not all apps use cryptosystems. Thus, these apps are not trusted for data security. This proposed system is just providing the text file only. This proposed system is saving the key at the firebase real-time database; therefore, this system is secure until the firebase is not breakdown. The secret key distribution has not been performed in this system. Therefore, the key distribution will conduct in future work. The further extension by using other technology can also be applied with other file types and can launch on IOS devices.

# AUTHOR’S PIUBLICATION

[1] Hnin Hsu Hlaing, Cho Cho San, “Secure Messaging System using RC4-2S, The Proceedings of the Conference on Parallel & Soft Computing (PSC 2022), University of Computer Studies, Yangon, Myanmar, 2022

# REFERENCES

[1] A.F.Doni, O.A.H.Maria and S.Hanif, 2020,"Implementation of RC4 Cryptography Algorithm for Data File Security"

[2] A.Kahate,2013. "Cryptography and network security" Tata McGraw-Hill Education.

[3] A.M.Sagheer, A.A.Abdulhameed and M.A.AbdulJabbar, 2013."SMS Security for Smartphone" In 2013 Sixth International Conference on Developments in eSystems Engineering (pp. 281-285). IEEE.

[4] A.Mousa and A.Hamad, 2006, "Evaluation of the RC4 algorithm for data encryption"

[5] A.Rukhin, J.Soto, J.Nechvatal, M.Smid and E.Barker,2001." A statistical test suite for random and pseudorandom number generators for cryptographic applications " Booz-allen and hamilton inc mclean va.

[6] B.A.Forouzan and D.Mukhopadhyay, 2015. "Cryptography and network security (Vol. 12)" New York, NY, USA:: Mc Graw Hill Education (India) Private Limited.

[7] B.Kiruthika, R.Ezhilarasie, and A.Umamakeswari,2015."Implementation of modified rc4 algorithm for wireless sensor networks on cc2431" Indian Journal of Science and Technology, 8(S9), pp.198-206.

[8] D.J.Bennett, 2009."Randomness" Harvard University Press.

[9] E.N.Ekwonwune and V.C.Enyinnaya, 2020, "Design and Implementation of End to End Encrypted Short Message Service (SMS) Using Hybrid Cipher Algorithm"

[10] J.Brown, B.Shipman and R.Vetter, 2007. "SMS: The short message service" Computer, 40(12), pp.106-110.

[11] J.Zhang, H.Liu,and L.Ni,2020."A secure energy-saving communication and encrypted storage model based on RC4 for EHR."Ieee Access, 8, pp.38995-39012.

[12] K.H.Myint, 2019. “A Data Confidentiality Approach to Short Message Service (SMS) on Android”

[13] M.Essaid, I.Akharraz,A.Saaidi and A.Mouhib,2018." A new image encryption scheme based on confusion-diffusion using an enhanced skew tent map"

[14] M.M.Hammood, K.Yoshigoe and A.M.Sagheer,2013."RC4-2S: RC4 stream cipher with two state tables" In Information Technology Convergence (pp. 13-20). Springer, Dordrecht.

[15] M.N.Riaz, and A.Ikram,2018. "Development of a secure SMS application using advanced encryption standard (AES) on android platform." Int. J. Math. Sci. Comput.(IJMSC), 4(2), pp.34-48.

[16] M.S.Novelan, A.M.Husein, M.Harahap and S.Aisyah, 2018. "Sms security system on mobile devices using tiny encryption algorithm". In journal of physics: conference series (Vol. 1007, No. 1, p. 012037). IOP Publishing.

[17] N.D.S.Morthty,2012."Text Messaging Encryption System" (Doctoral dissertation, UMP).

[18] O.S.Sitompul, N.H.Pasaribu and E.B.Nababan, 2018."Hybrid RC4 and Affine Ciphers to Secure Short Message Service on Android."

[19] R.Rifki, A.Septiarini and H.R.Hatta, 2018,"Cryptography using random Rc4 stream cipher on SMS for android-based smartphones" IJACSA) International Journal of Advanced Computer Science and Applications.

[20] R.S.Villafuerte, A.M.Sison, A.A. Hernandez, R.P. Medina,2020." Randomness Evaluation of the Improved 3D-Playfair (i3D) Cipher Algorithm" In 2020 12th International Conference on Communication Software and Networks (ICCSN) (pp. 240-245). IEEE.

[21] R.Upadhyay,S.Singh,V.Trivedi, and A.Soni, 2018. "Randomness test for wireless physical layer key generation" In 2018 International Conference on Advanced Computation and Telecommunication (ICACAT) (pp. 1-6). IEEE.

[22] S.A.Carr and M.Payer,2017. "Datashield: Configurable data confidentiality and integrity" ACM on Asia Conference on Computer and Communications Security.

[23] S.Das, H.Dey and R.Ghosh, 2014. "RC4 stream cipher with a modified random KSA" In Emerging Trends in Computing and Communication (pp. 169-179). Springer, New Delhi.

[24] S.Samanta, S.Dutta and G.Sanyal,2015."An Image Steganography-based Novel Approach to develop 8-Share Integrated Security Toolkit (ISTI-8)" image, 4(5), p.10.

[25] Xiao-Jun Tong,Miao Zhang,Zhu Wang,Yang Liu,2014."A image encryption scheme based on dynamical perturbation and linear feedback shift register".