**CHAPTER THREE**

**DESIGN METHODOLOGY**

**3.1 System Overview**

The overall structure of the secure SMS messaging system is broken down into four subsystems. Encryption/Decryption subsystem, Combiner/Extractor subsystem, 7-bit Encoder/Decoder and Sender/Receiver subsystem. This chapter describes the design, architecture and implementation of secure SMS messaging system. The architecture includes a number of components and each component implemented as a separate entity.

**3.1.1 Encryption/Decryption Subsystem**

This subsystem contains two type of encryption, the first one is symmetric (Blowfish) cryptography and the second is asymmetric (ECC) cryptography. Blowfish has a better performance than other common encryption algorithms in different aspects such as speed, memory usage and throughput. Also Blowfish has no any known security weak points until now hence, it is used in this system for SMS content encryption. ECC is an excellent choice of asymmetric cryptography for handheld device. It grants high level of security with small key size. ECC reduces battery usage as compared with other asymmetric key cryptography. Also it provides better security with less processing overhead. In general, ECC is more appropriate asymmetric cryptography for resource constraint devices such as mobile phones.

In this mobile application solution, Blowfish is used for encrypting the original SMS messages and Elliptic Curve Cryptography (ECC) is used in this system as a secure channel to transfer the onetime key which is used to encrypt the message. The onetime key is encrypted by ECC and sent to the receiver together with the message at a time.

**3.1.1.1 Key Encryption/Decryption Subsystem**

The main task of Key Encryption/Decryption subsystem is to encrypt one time password (OTP) which is used as secret key for symmetric cryptography. This encrypted onetime password is generated by using ECC algorithm. ECC with 160-bits key size is applied. ECC with 160-key size have equivalent security level with RSA 1024-bit key size. ECC uses the receiver’s public key to encrypt the secret key. As shown in Algorithm 3.1, private-public key pair is generated for each user. The public key of a given user can be generated from its private key easily and private-public key pair is related each other. Elliptic curve key pair generation algorithm (Algorithm 3.1) is given as follows:

Input: Elliptic curve domain parameters (p, E, P, n).

Output: Public key Q and private key d.

1. Select d R [1, n-1].
2. Compute Q= d\*P.
3. Return (Q, d).

ECC is used as secure channel to transmit generated onetime password to the receiver's phone. As shown in Algorithm 3.2, to encrypt the generated secrete key using ECC, the first task is to encode the generated secrete key (in this case plain text (m)) as x-y point Pm. Then, the point Pm can encrypt as a cipher text at the sender side and decrypt as plain text (in this case the secrete key that is used to decrypt the received secure SMS) at the receiver side. Algorithm 3.3 shows the algorithm that takes the encrypted onetime password as input and it produces the secrete key.

Algorithm

3

**.**

2

**:**

A

lgorithm Used to Encrypt the Secret Key

Elliptic curve encryption algorithm

**Input:**

Public key Q, Plain text m.

**Output:**

Cipher text (C1, C2).

1

. Initialize Elliptic curve domain parameters (p, E, P, n

).

(

. Represent the plain text m as a point Pm in E

2

Fp).

. Select k

3

∈

R [1, n

-

1]

.

4

. Compute C1= K\*p.

5

. Compute C2= Pm + k\*Q.

6

2).

. Return (C1, C

This mobile app solution use Blowfish symmetric cryptography to secure (to encrypt) the message. Hence, Blowfish needs the same key to encrypt the message at the sender side and to decrypt at the receiver side. The system generates onetime password at the sender side, encrypt this key by using ECC and send to the receiver phone in order to use the same secret key.

**3.1.1.2 Message Encryption/Decryption**

Message Encryption/Decryption is part of Encryption/Decryption subsystem and it uses Blowfish symmetric cryptography technique to secure the SMS message. Blowfish is a symmetric block cipher that supports a 64 bits block size input. As shown in Algorithm 3.4, the original SMS message is divided into 64 bit block size to encrypt and decrypt it using Blowfish algorithm. The actual key is not used directly in Blowfish; instead the key is used to generate 18 32-bits sub keys called P-array. The main task of this part is generating one time password automatically and encrypts the message by using the generated secret key.

.

Algorithm

3

**.**

3

**:**

Algorithm Used to Decrypt the Secret Key

Ellipticcurvedecryption

algorithm

**Input:**

Private key d, Cipher text

2)

C1, C

(

.

**Output:**

Plain text m.

. Initialize Ellip

1

tic curve domain parameters (p, E, P, n).

2

2

. Compute Pm = C

-

dC1.

3

. Return (

m).

The following is the Blowfish Encryption Algorithm (Algorithm 3.4, default 16 rounds):

Input= 64-bit data: x. (plain text)

Output = Combined xL and xR (64-bit Cipher text)

Generate 64-bits random key (Sk).

Initialize 18 32-bits P-array from the key (Sk).

Divide x into two 32-bit halves: xL, xR.

for i = 1 to 16:

xL = xL XOR Pi xR = F(xL) XOR xR

Swap xL and xR

If i> 16

Swap xL and xR (to undo the last swap)

xR = xR XOR P17

xL = xL XOR P18

**3.1.2 Combiner/Extractor Subsystem**

The main task of combiner/extractor sub system in this system is concatenate the encrypted onetime password and encrypted SMS message at the sender side in order to produce secure SMS payload. Also at the receiver side it used for extraction the two parts of the secure SMS payload.

Combiner: the proposed Secure SMS payload is the combination of encrypted one time password and encrypted SMS message. The main function of this component is concatenating the output of Key Encryption subsystem (encrypted onetime password) and the output of Message Encryption subsystem (encrypted SMS message).

The extractor works in such a manner that the onetime password is transmitted together with the SMS message to the receiver. Before decrypting the received secure SMS, the recipient need the secrete key which is used to decrypt the secure SMS payload. As mentioned, this onetime password sent together with SMS message. Then, at the receiver side, this component extracts the onetime password from secure SMS payload and this extracted onetime password is decrypted by using receiver’s private key.

**3.1.3 Certificate Authority**

Certificate Authority is a trusted third party organization or company that provides digital certificates and public-private key pairs. The main tasks of CA in this system are generating a unique key pair for users and associate relevant user information with generated key pair. To generate the key pair it uses ECC key pair generator algorithm. Any trusted organization such as bank, that uses this Secure SMS messaging system or SMS message service provider, can be used as CA. In this system phone number of the user is used as a public key in order to remember it easily.

**3.2 Architecture of Secure SMS Messaging System**

This Secure SMS System whose architecture is has two major parts, which are sender and receiver. However, these parts have subcomponents.

**3.2.1 Generating and Sending Secure SMS Message**

The user of the system provides the port, public key of the receiver, actually this is the phone number of the receiver in order to remember it easily, and the SMS message. Before the message is sent, the content must be processed as follow:

* Calculate the message digest to ensure message integrity.
* Encrypting the message content using automatically generated one time password for Blowfish algorithm;
* Encrypting onetime password using ECC algorithm. The sender uses receiver’s public key to encrypt onetime password;
* Concatenate encrypted SMS and encrypted onetime password to generate secure SMS payload and;
* Convert both encrypted password and encrypted message to 7-bit encoding. SMS only supports 7-bit characters. Hence, 7-bit encoding subsystem converts the cipher text to GSM default characters set as shown in the use cases (Figures 3.3 and 3.4).

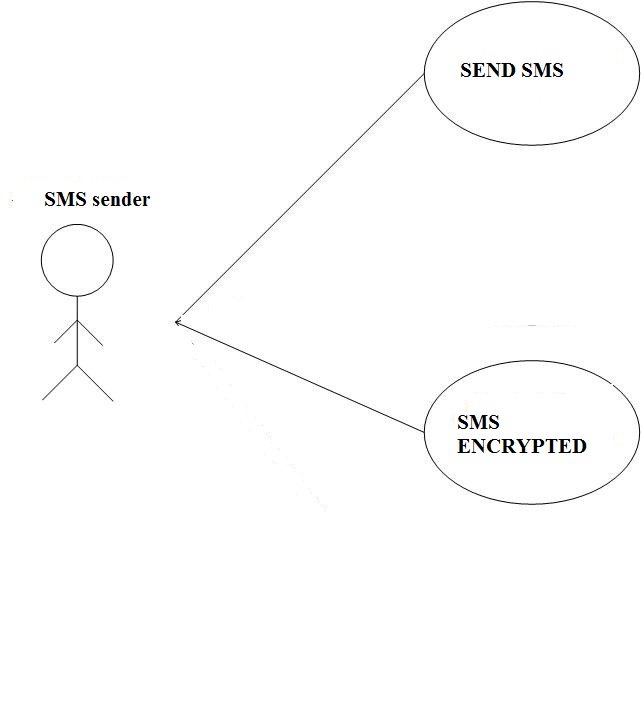
****

Figure 3.3 Use Case for SMS sender

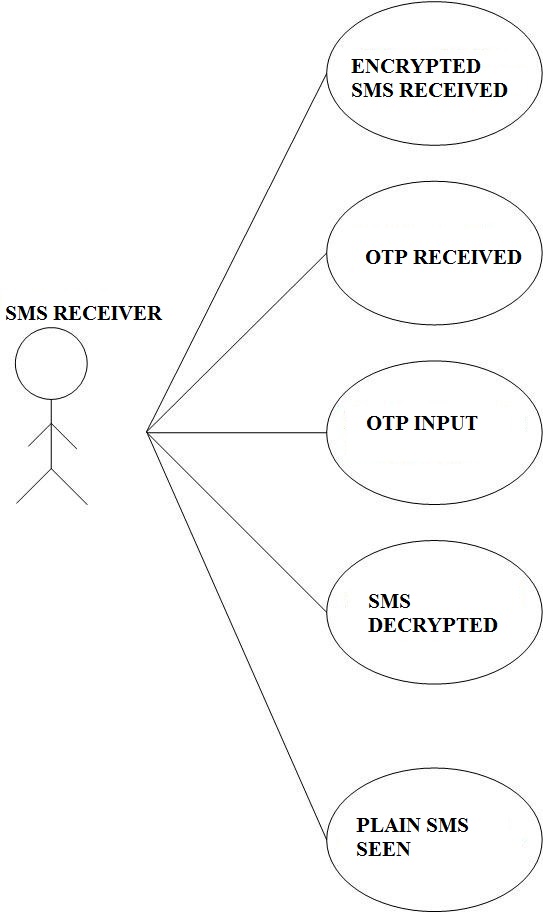


Figure 3.4 Use Case for SMS receiver

**3.2.2 Receiving and Decoding Secure SMS Message**

The receiver provides private key and port number to the application as input, when the recipient phone receives the message from the GSM network, it decodes the. After the message is received, the receiver application performs the following to get the original SMS message as shown by the dataflow diagram and flowchart in figures 3.1 and 3.2:

* First from received Secure SMS, separate the concatenated cipher text and encrypted onetime password;
* Use 7-bit decoder to recover the original characters (cipher text);
* After separating the encrypted onetime password, decrypt it by using ECC algorithm. The receiver provides receiver’s private key in order to decrypt the secrete key;
* Decrypt the cipher text by using secrete key. Blowfish algorithm is used for decrypt the cipher text; and
* Calculating the message digests and compare with the received one to ensure message integrity. After the above steps, the received message is displayed on the mobile screen if the message digest is correct.

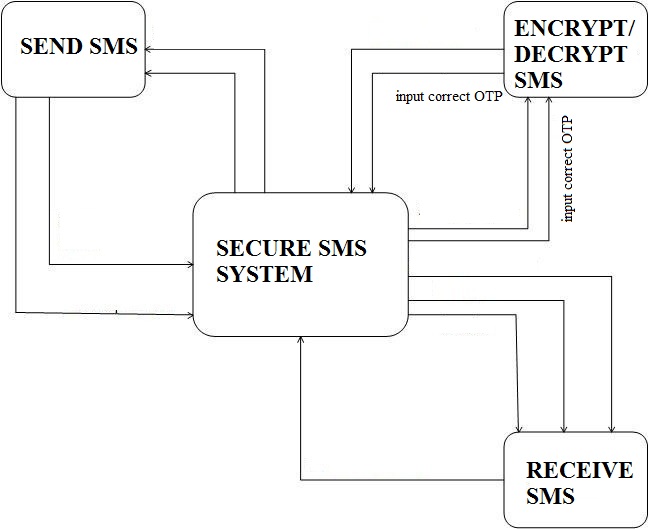
****

Figure 3.1 Dataflow diagram for the system

**Send SMS**

Is SMS Encrypted

Is OTP generated and used?

Decrypt the SMS

Display’s Decrypted SMS

No

Yes

No

Yes

Kindly use the OTP sent with the SMS and use it

**Figure 3.2**

**3.3 Programming Language Used**

It was decided to implement the android messaging system using Java programming language on android studio. Among the reasons are:

* Familiarity;
* Object Orientation;
* Level of abstraction: Java provides a good level of abstraction and it includes rich set of APIs.
* APIs: Java provides SMS messaging APIs in the WMA (Wireless Messaging APIs) set. It also provides a set of APIs to build useful user interfaces.
* Availability: Most mobile phones being sold on the market includes built in Java virtual machine.
* Portability: Java ME makes the application portable on to any java enabled phone. For these reasons we have chosen java as the programming language for the implementations.

In order to develop the secure SMS messaging application Java Wireless Toolkit development kit was integrated with android studio, android studio was used as development environment. The message sender and receiver application uses the NetBeans Mobility Pack 3.0 to simulate mobile phone environment. Since the mobile application is developed using Java, it should be able to run on any mobile phone that has Java Virtual Machine (JVM).

To enable the application to send and receive SMS message on the mobile phone, the J2ME Wireless Message API (WMA) library was used. The WMA library has various classes and methods to construct an SMS message. In order to develop the security of SMS, the Bouncy Castle Crypto package was used. Bouncy Castle offers lightweight API for many encryption algorithms. These API simplify the task in building the security solution in Java mobile application. The package contains a lightweight API suitable for use in many environments including J2ME. Bouncy Castle Crypto libraries that are only important for Blowfish and ECC algorithms were added. The selection of adding only necessary libraries of Bouncy castle package is made in order to avoiding memory overhead of the devices. The following subsections describe the implementation of each part of the secure SMS messaging application.

**3.4 Encryption / Decryption**

For this SMS application, Blowfish algorithm was used to cipher the actual SMS messages and ECC algorithm used to encrypt the onetime password, which is used to encrypt the actual message. Both Blowfish and ECC algorithms are provided by the Bouncy Castle library under the crypto package.

The message encryption and decryption method take in either a string value or a byte array as the input message and encrypt the input message using the password given in the Blowfish parameter. If a string value is passed as the message into the method, the method converts the string into a byte array and encrypts it into a ciphered byte array. This password is generated automatically by using random key generator. In the other hand the key encryption and decryption method uses ECC algorithm as secure transmission of one time password used in message encryption and decryption part. ECC take one time password automatically as input message and encrypt the input message using the receiver’s public key given as a parameter.

**3.5 7-Bit Encoding**

GSM network was used in order to send and receive SMS message. It supports only 128 characters. In order to transmit characters that are not supported by GSM, there would be need to convert the SMS to 7-bit encoding methods. To achieve this, Base64 encoding algorithm was used. It converts the data bytes of the message into 7-bits characters. The term Base64 originates from a specific MIME (Multipurpose Internet Mail Extensions) content transfer encoding. There is a large overlap between values for characters in GSM and ASCII; however, not every ASCII character is supported in GSM network and some characters have not the same ASCII value. For example, for the hex value of 24, which represents the dollar sign ($) in ASCII, but in GSM, it represents the generic currency symbol (¤) and for the hex value of 40, the character in ASCII is the commercial “at” symbol (@), but the character in GSM is an inverted exclamation mark (¡).

Base64 encoding takes three 8-bits characters and represents them as four printable characters in the ASCII standard. First, it finds the 8-bit ASCII representation of each input characters. Then, it converts the input three characters in to four parts. The next example describes how input “Man” is converted as “TWFu”. Base64 encoding conversion table is included in Appendix V [65].

|  |  |  |  |
| --- | --- | --- | --- |
| Text content: | M | A | n |
| ASCII: | 77 | 97 | 110 |

Bit pattern: 0 1 0 0 1 1 0 1 0 1 1 0 0 0 0 1 0 1 1 0 1 1 1 0

Base64 Index: 19 22 5 46

Base64 encoded: T W F U

After encoding the message using Base64 algorithm, the message size is increased by some characters. Therefore the numbers of characters send as a single SMS needs to be reduced.

**3.6 Message Digest**

The algorithm chosen to calculate the message digest is the Secure Hash Algorithm (SHA-1). The SHA-1 algorithm is also provided by the Bouncy Castle library under the crypto package. The class called ‘integrity’ is used for checking the integrity of the message. It consists of two public methods, the first method is to calculate the message digest and the second method is to check the message integrity. The method first calculates the digest of the input message from the input parameter and then it checks if the size of the digests is equal, if the size is not equal, the method throws an exception and indicate the error.

**3.7 Sending SMS Message**

To send SMS messages with Java, it requires using the WMA library. The WMA library provides classes to handle SMS messages. It uses the MessageConnection class to initialize a connection to send SMS messages. The format is “sms://<reciver phone number>:<port>”, where <port> is the port number which the receiver is listening (that is the 11 digit phone number in this case).

**3.8 Receiving SMS Message**

To implement SMS receiver application, the application must listen to a specified port. As described in Internet Assigned Numbers Authority (IANA), port numbers are divided into three ranges as well-known Ports, registered Ports, and dynamic (Private) Ports. The port number range is from 1 to 65535. The Well Known Ports are those from 0 to 1023 and it can only be used by systems that executed by privileged users. Registered Ports are those from 1024 to 49151, and Dynamic (Private) Ports are those from 49152 to 65535. Both Well Known ports and Registered ports cannot be used without IANA registration. Dynamic (Private) Ports can be used in any private applications without registration.

For this application, “50000” have been selected as listening port number which is in the range of Dynamic (Private) Ports. If the port is already used by other application installed on the receiver’s phone the application may not be able to function properly. To receive incoming messages the mobile application must implement the MessageListener class that is standard with J2ME development.

**3.9 Key Pair Generator**

The key pair generator application is only used by the certificate authority (CA) or authorized organizations that needs secure SMS messaging such as banks. This application is used for generating key pair (private key and public key) for each authorized user of the system. The proposed application uses ECC with 160-bits key size, to generate the key pair.

**3.10 Onetime Passwords (OTP) Generator Application**

The main task of this application is generating a onetime password which is used as a secret key for Blowfish algorithm in order to encrypt and decrypt the message. Since, Blowfish algorithm was used with 64-bits key size, it generates 8 characters randomly. As the name implies, this key can only be used for one communication.