Available online at www.pelagiaresearchlibrary.com

Pelagia Research Library

Advances in Applied Science Research, 2011, 2 (3): 373-379

ISSN: 0976-8610
CODEN (USA): AASRFC

Secured Communication Protocol for Wireless Peer to Peer Messaging

Kashinath Singh Sardar*and Kauser Ahmed

School of Computing Science and Engineering, VIT University, Vellore, India

---

ABSTRACT

Wireless peer to peer is the method by which individual users connect to each other directly, without need for a central point of management or system. It facilitate message passing between the two nodes which are directly connected. There arises a security issues for an unauthorized access to the message, hence a secured communication protocol is needed which can provide an authentication and integrity protection, which has to be establishes an end-to-end secure channel between server-side and client-side. An encryption and Decryption algorithm is used for the protocol to ensure confidentiality, integrity and non-repudiation of messages.

KEYWORDS: SMS, blowfish algorithm, cipher-block chaining (CBC) mode, J2ME.

INTRODUCTION

1.1 EncryptedSms: peer to peer encryption and decryption of SMS

Short Message Service (SMS) is the text communication service component of mobile communication systems, using standardized communications protocols that allow the exchange of short text messages up to 160 characters to mobile phone devices.

As Short Message Service (SMS) is now widely use all over the world hence it security has become a major concern So there arise security issues for an unauthorized access to the message .A secure message system requires solving the following three problems at least.
(1) Authentication -Confirm true identities between sender and receiver, and prevent impersonation attack from illegal intruders.
(2) Confidentiality - Ensure that decrypted messages are accessible only to those authorized senders and receivers.
(3) Integrity - Ensure that receivers can check out whether the message has been modified, and prevent tampered message.

1.2 Encryption and Decryption

Encryption is the process of transforming information (referred to as plaintext) using an algorithm (called cipher) to make it unreadable to anyone except those possessing special

knowledge, usually referred to as a key. The result of the process is encrypted information (in cryptography, referred to as cipher text). In many contexts, the word encryption also implicitly refers to the reverse process, decryption (e.g. "software for encryption" can typically also perform decryption), to make the encrypted information readable again (i.e. to make it unencrypted).

1.3 Peer to Peer

Peer-to-peer (P2P) or Point to point or End to End computing or networking is a distributed application architecture that partitions tasks or workloads between peers. Peers are equally privileged, equipotent participants in the application. They are said to form a peer-to-peer network of nodes.

Peers make a portion of their resources, such as processing power, disk storage or network bandwidth, directly available to other network participants, without the need for central coordination by servers or stable hosts. Peers are both suppliers and consumers of resources, in contrast to the traditional client–server model where only servers supply, and clients consumes.

MATERIALS AND METHODS

To overcome the security issues a **blowfish algorithm** is to be implemented. Reasons behind implementing **blowfish algorithm** is its ability of providing high security with smaller key size which makes it very useful in resource-limited device such as mobile phone, compare to others it is faster than other algorithm.

2.1 Analysis and Design:

2.1.1 Description of the algorithm

Blowfish is a variable-length key, 64-bit block cipher. Symmetric Key such as blow fish uses the same key for both encryption and decryption where as Public key algorithm use two keys, one for encryption and another for decryption. Public key used for encryption and private key used for decryption in case of public key encryption algorithms.

Blowfish is a key symmetric block cipher; it has 64 bit block cipher and takes a variable length key from 32 to 448 bits and the block of the blowfish algorithm is 64 bits. It has sub keys in 18 entry P Array and four S boxes of 256 entries. This algorithm had 16 rounds and each round consists of a key-dependent permutation, and a key- and data-dependent substitution. All operations are XORs and additions on 32-bit words. The only additional operations are four indexed array data lookups per round.

The input is 64 bit data element A. This is divided two halves 32 bits each.A1 and A2.

As mentioned above this algorithm has 16 rounds and each round goes through the following steps (For I = 1 to 16)

1. A1 = A1 XOR Pi
2. A2 = F(A1) XOR A2
3. Swap A1 and A2

After 16^{th} round swap A1 and A2 to undo the swap. Then, A1 = A1 XOR P17 and A2 = A2 XOR P18. Finally, recombine A1 and A2 to get the cipher text. Function F looks like this:

Divide all into four eight-bit quarters: a, b, c, and d. Then, F (A1) = ((S1, a + S2, b mod 232) XOR S3, c) + S4, d mod 232.

Decryption is exactly the same as encryption, except that P1, P2... P18 are used in the reverse order.

2.1.2 Cipher-block chaining (CBC)

In the cipher-block chaining (CBC) mode, each block of plaintext is XORed with the previous cipher text block before being encrypted. This way, each cipher text block is dependent on all plaintext blocks processed up to that point. Also, to make each message unique, an initialization vector must be used in the first block.

2.1.3 Reason for selecting CBC along with Blowfish algorithm

- CBC has been the most commonly used mode of operation.
- Its main drawbacks are that encryption is sequential (i.e., it cannot be parallelized), and that the message must be padded to a multiple of the cipher block size.
- Note that a one-bit change in a plaintext affects all following cipher text blocks, and a plaintext can be recovered from just two adjacent blocks of cipher text.
- As a consequence, decryption can be parallelized, and a one-bit change to the cipher text causes complete corruption of the corresponding block of plaintext, and inverts the corresponding bit in the following block of plaintext.

2.2 Design

The design of the SecuredSms is divided into two parts:

- **The external architecture** describes the overall design of the system how it works and how the components communicates with each other using the messages with respect to time.
- **The internal architecture** represents how the system internally works.

2.2.1 The external architecture

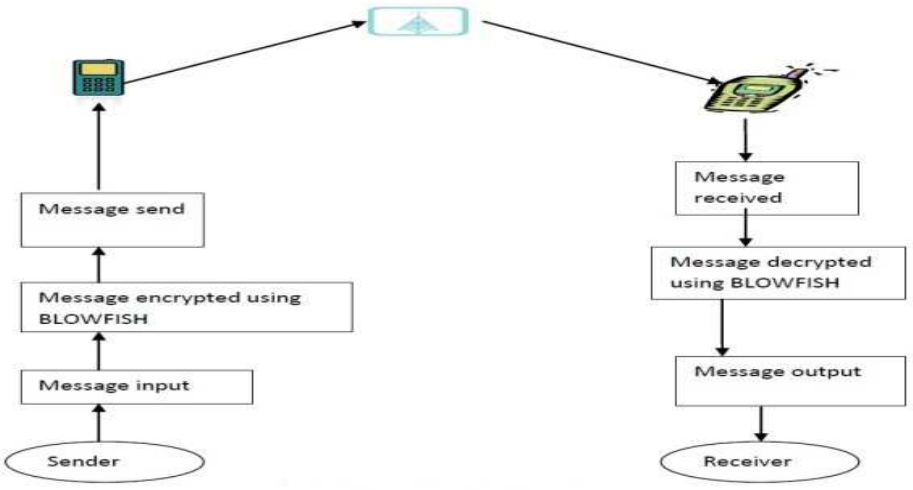

Figure1

2.2.2 The internal architecture

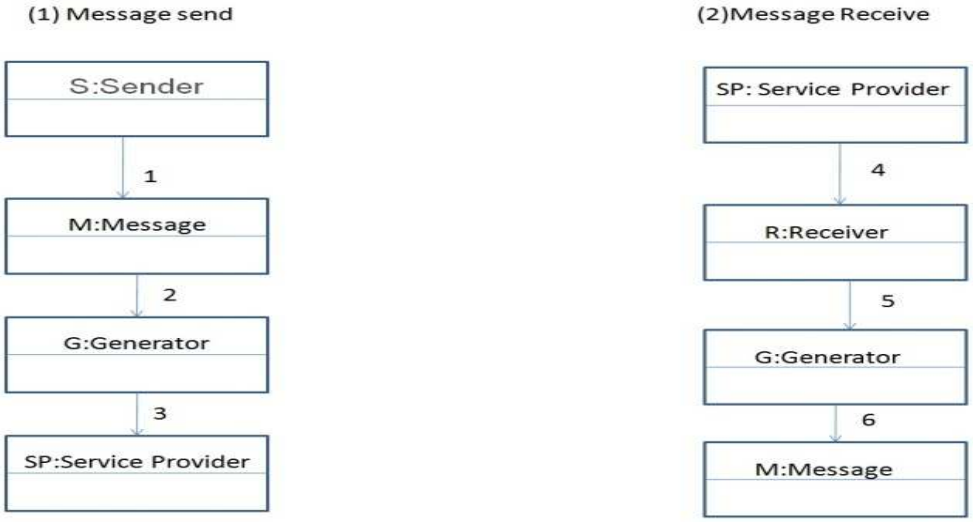

Figure 2

2.3 Implementation

2.3.1 Requirement Specification:

Software Requirements:

- OPERATING SYSTEM: - WINDOWS (XP) OR ABOVE
- PLATFORM: J2ME, JAVA
- SOFTWARE: NETBEANS IDE 6.9.

Hardware Requirements:

- PROCESSOR: A P4 at 1.5 GHz or faster.
- DISPLAY DEVICE: EGA, VGA, SVGA or other display compatible with windows.
- RAM: 512 MB or Above.
- HARDDISK: Minimum 1 GB.

2.4 EncryptedSMS MIDlet

The EncryptedSms MIDlet will encrypt a message using a password-based symmetric encryption algorithm and send it using a binary SMS. It is expected that the sender and the receiver have previously agreed on a common password, so there are no provisions for key exchange. The encryption algorithm used is **BLOWFISH** and the digest is calculated using **SHA1.** Different algorithms could be used that offer higher strength.

The SMS is simply a byte array containing a header, the ciphered text, and an optional message's digest. The header includes two bytes of metadata and two bytes containing the size of the cipher text. The first two bytes can be used to indicate properties of the message.

The MIDlet's user interface has two screens, one for sending a message and one for receiving. The sending message form contains fields for the destination number, text, password, and a choice item to indicate whether to append a digest. It also has a Send command to do the encryption and send the message.

On the receiving end, the application listens for incoming messages and upon their arrival it prompts for the password to be used in the decryption process. If a message has a digest, it will also verify the integrity of the message.

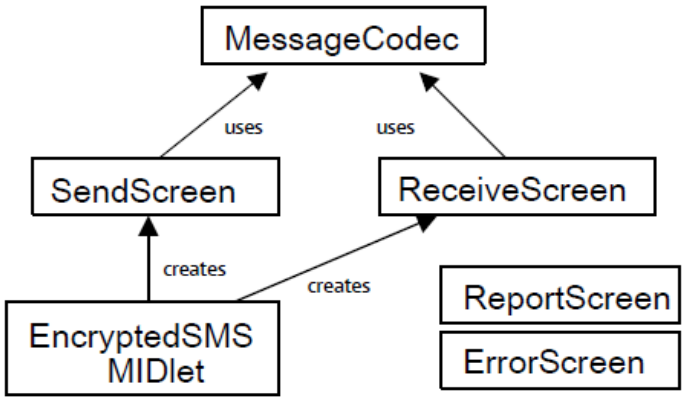

Figure3

EncryptedSMS contain following modules

2.4.1 EncryptedSMSMIDlet.java

This is the main class, and it controls the state changes between screens. It also owns the message connection and registers itself as a message listener waiting for incoming messages. When sending a message, the send Message method will launch a new thread that will take care of sending the message. The SMS port used for the connection is read from the JAD file under the port property.

2.4.2 MessageCodec.java

This utility class takes care of bundling and unbundling the messages. It creates the Block Cipher and the Digest engines and packs the headers and contents of messages. This class sees only bytes arrays and does not care whether they arrived by SMS or some other delivery method. The create Engine and create Digest method will construct a Blowfish and SHA-1 objects

2.4.3 SendScreen.java

This screen contains fields for sending the encrypted message. When the Send command is invoked it delegates the sending of the message to the main class.

2.4.4 ReceiveScreen.java

This screen is activated when a new message arrives. The SMS has been already read by the main class and passed to Receive Screen. The screen will request the message's password, delegate the decryption process to Message Codec, and display the decoded content of the message.

This utility class takes care of bundling and unbundling the messages. It creates the Block Cipher and the Digest engines and packs the headers and contents of messages. This class sees only bytes arrays and does not care whether they arrived by SMS or some other delivery method.

The create Engine and create Digest method will construct a BLOWFISH and SHA1 objects in this example. Using a different engine is sufficient for modifying those create methods.

2.4.5 ErrorScreen.java

This is a simple utility screen that displays some message information.

2.4.6 ReportScreen.java

This class shows a text report. Additionally it can display a HEX view of a byte array.

RESULTS

Message is created along with Key and send.

Message is received and key is entered at receiver side.

Message is decoded and viewed at receiver side.

HEX view of a byte array in report.

CONCLUSION AND FUTURE EXTENSION

The application for securing of SMS has been designed and implemented, which secure messages for securing, it has been chosen the BLOWFISH Algorithm. The application is running in the mobile phone and does not require any additional encryption devices. In the future, the application could also provide MMS securing. The application could be ported to

other programming platform for greater distribution among users. The application could use a less computationally demanding algorithm. Such an algorithm could be based on the elliptic curves. Applications could also automatically communicate with the certification authority, upload, download and verify the validity of keys. There could be implemented an SMS verification – for warning against suspicious SMS with false number of the sender.

REFERENCES

[1] Marko Hassinen, "SafeSMS - End-to-end encryption for SMS messages", Department of Computer Science, University of Kuopio, *8th International Conference on Telecommunications - ConTEL 2005*. ISBN: 953-184-081-4, June 15-17, **2005**
[2] David Lisonek, Martin Drahanský," *International Conference on Security Technology*, **2008** *IEEE*.
[3] Songyang Wu, Chengxiang Tan Department of Computer Science, TongJi University Shanghai, P.R. China."High Security Communication Protocol for SMS", Tan Department of Computer Science, TongJi University Shanghai, P.R. China, *2009 International Conference on Multimedia Information Networking and Security* **2009** IEEE.
[4] Shushan Zhao, Akshai Aggarwal, Shuping Liu, "Building Secure User-to-user messaging in Mobile Telecommunication Networks", School of Computer Science, Department of Electrical Engineering, University of Windsor, Canada and University of Southern California, USA,**2008** IEEE.
[5] Zhenqi Wang, Ziyan Guo, Yue Wang, "Design and Implementation of Short Message Query System for Academic Office Based-on J2ME", Network Center, North China Electric Power University, *2008 ISECS International Colloquium on Computing, Communication, Control, and Management*, **2008** IEEE.
[6] Alfredo De Santis, Aniello Castiglioned Giuseppe Cattaneo, Maurizio Cembalo, Fabio Petagnaa and Umberto Ferraro Petrillo, "An Extensible Framework for Efficient Secure SMS" ,Dept. of Computer Science and Applications "R.M. Capocelli "University of Salerno,Department of Statistics, Applied Probability and Statistics University of Rome *"Sapienza"2010 International Conference on Complex, Intelligent and Software Intensive Systems*,**2010** IEEE.
[7] www.wikipedia.org
[8] www.nokia.co.in