ISSN:0975-9646

Sri Rangarajan et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 4 (2) , 2013, 285 - 288

Securing SMS using Cryptography

Sri Rangarajan, N. Sai Ram, N. Vamshi Krishna

Department of CSE,
Vignana Bharathi Institute of Technology
Ghatkesar,
Hyderabad, India.

***Abstract*— Short message service (SMS) is established as a widely used and wide spread approach for text messaging in present day's immensely mobile reliant world. Apart from electronic chatting, SMS today has become an accustomed source for communicating confidential or proprietary information. When this is the situation the imperative factor is "Security". One commonly used technique to provide security is Encryption. It plays a vital role when confidential data is proceeding in the network by serving as a fortification for the original raw data avoiding intrusion. There are many conventional and symmetric encryption algorithms available to bestow this, each having its own level of security and performance. The most important aspect needed to be considered while using cryptography to provide SMS security is the storage and processing capabilities of the mobile phone which is the main source of the SMS. Considering all aspects this paper proposes a means of providing high authentication and security to the messages shared which can be efficiently used in small devices like mobile phones.**

***Keywords*— SMS, Elliptic Curve Cryptography, Security, Key Exchange, Encryption, Decryption.**

I. INTRODUCTION

During the last few years the use of mobile devices has been increasing exponentially. SMS (Short Message Services) is part of the services of the GSM and other cellular networks that provides a mechanism to transmit short messages to, and from mobile devices. The large base of mobile devices and the SMS acceptance have promoted its utilization for some unconventional applications rather than sending short and brief conversations.

SMS is quite simple; the maximum message's length is 160 characters. SMS uses only a set of characters as data type. It requires very low bandwidth and it is a low cost service compared with others which makes it suitable for quick communication. Nevertheless its versatility, SMS has some limitations that would be important for some unconventional applications. For some applications, like transactions, payments and monitoring, it would be helpful to incorporate some services that can provide confidentiality, integrity, authentication and non-repudiation services which are standard for network security.

The idea of using Elliptic Curves in cryptography was introduced by Victor Miller and N. Koblitz as an alternative to established public-key systems such as DSA and RSA. The Elliptic curve Discrete Log Problem makes it difficult to break an ECC as compared to other conventional cryptosystems where the problems of factorization or the discrete logarithm problem can be solved in sub-exponential time. This means that significantly smaller parameters can be used in ECC than in other competitive systems. This helps in having smaller key size hence faster computations and thus proves efficient in small devices like mobile phones.

This paper is based on the end to end secure transmission of SMS using cryptography. The algorithm is based on the combination of Elliptic Curve Cryptography which provides high authentication with small key size and small computing time.

II. EXISTING SYSTEM

The necessity of providing security to SMS has been imperative since a long time and many algorithms and techniques have been implemented in various platforms to try and provide security to the messages. At present there are many algorithms based on symmetric cryptography that provides security to the messages transferred based on a shared secret key. The main disadvantage of a symmetric-key cryptosystem is related to the exchange of keys. There exists the problem of key distribution in them. Private-key systems need to use keys that are at least as long as the message to be encrypted. Symmetric encryption requires that a secure channel be used to exchange the key, which seriously diminishes the usefulness of this kind of encryption system when we talk about SMS [1].

Thus, the use of conventional cryptosystems is considered in this paper. One such conventional cryptosystem presently being used is the RSA. The RSA is a public key cryptography based on factorizing large integers. In the RSA algorithm the key generation is based on two distinct prime numbers. The public key generated is shared over the network. The biggest drawback of the RSA algorithm is that anyone having the public key can find various ways to decrypt the message. The attacks against RSA can be due the choosing of weak plain text, weak parameters selection, or inappropriate implementation. There are various attacks possible on RSA few of them are:

- Factorization Attack
- Chosen-Ciphertext Attack
- Broadcast Attack

Other than the attacks on RSA there are many other problems related to using this algorithm for SMS security. SMSs are generated from mobile phones which generally have very low processing capacities and also low memory and battery capacity. Algorithms like RSA are conventional algorithms with large key sizes which require higher memory capacities and high processing powers. Due to these reasons the usage of RSA for providing SMS security will reduce the performance of the device and also the processing time will be high [1].

The proposed scheme is based on Elliptic Curve Cryptography which is a more efficient conventional cryptosystem. The bit size of ECC is very less compared to that of RSA and also due to the discrete logarithmic problem it is very difficult to decrypt the messages by just knowing the public key. ECC works on less key size and low processing power thus can be highly efficient in small devices like mobiles.

III. ELLIPTIC CURVE CRYPTOGRAPHY

Elliptic Curve Cryptography (ECC) is a public key cryptography. In public key cryptography each user or the device taking part in the communication generally have a pair of keys, a public key and a private key, and a set of operations associated

with the keys to do the cryptographic operations. Only the particular user knows the private key where as the public key is distributed to all users taking part in the communication [2].

Every public key cryptosystem requires a set of predefined constants to be known by all the devices taking part in the communication. In the case of elliptic curve cryptography "Domain parameters" are the constants. Public key cryptography, unlike private key cryptography, does not require any shared secret between the communicating parties but it is much slower than the private key cryptography [3].

The domain parameters of elliptic curve are a sextuple:

$$T = (P, a, b, G, n, h)$$

An elliptic curve over a field K is a curve defined by an equation of the form

$$y^2 = x^3 + ax + b$$

where a, b Є K and $4a^3 + 27b^2 \neq 0$.

A. Discrete Logarithm Problem

The security of ECC depends on the difficulty of Elliptic Curve Discrete Logarithm Problem. Let P and Q be two points on an elliptic curve such that kP = Q, where k is a scalar. Given P and Q, it is computationally infeasible to obtain k, if k is sufficiently large.

k is the discrete logarithm of Q to the base P.

B. ECC Public Key Cryptosystem

In the public key elliptic curve cryptosystems, assume that entity A wants to send a message 'm' to entity B securely. Order of a point on the curve can be defined as a value n such that,

nP = P+P+..+P…. n times = O (infinity).

C. Generation of Public and Private Key

Both the entities in the cryptosystem agree upon the Domain Parameters (a, b, P, G, n). G is called generator point and n is the order of G.

Now A generates a random number $n_A < n$ as his private key and calculates his public key set $P_A = G.n_A$

B generates a random number $n_B < n$ as his private key and calculates his public key set $P_B = G.n_B$

D. Generation of common key

After exchange of the public key between the two parties

Entity A computes his Common Key by Computing $K= n_A.P_B$

Entity B computes his Common Key by Computing $K= n_B.P_A$

The two above keys have same value because:

$$n_A.P_B = n_A.\ (n_B.G) = n_B\ (n_A.G) = n_B.\ P_A.$$

E. Encryption

Consider a message 'Pm' sent from A to B. 'A' chooses a random positive integer 'k', a private key 'n_A' and generates the public key $P_A = n_A \times G$ and produces the cipher text 'Cm' consisting of pair of points $Cm = \{kG, Pm + kP_B\}$ where G is the base point selected on the Elliptic Curve, $P_B= n_B.G$ is the public key of B with private key 'n_B'.

F. Decryption

To decrypt the cipher text, B multiplies the 1st point in the pair by B's secret & subtracts the result from the 2nd point

$$Pm + kP_B - n_B(kG) = Pm + k(n_B\,G) - n_B(kG) = Pm.$$

IV. PROPOSED WORK

The proposed system uses the above described concept of elliptic curve cryptography to encrypt the message and send it over a common channel. The sender writes a message and gives the recipient's number, when he sends the message the algorithm is triggered on both the devices. The keys are generated and shared among the devices and the encryption takes place at the senders end. After encryption, the message is sent to the receiver and he decrypts it using his key to read it.

The Encryption and Decryption methods in ECC are designed to encode and decode a point on the curve and not the entire message. During encryption, each character in the message has to be converted into bytes then the bytes into points of the form (x, y) and then the points have to be encoded by mapping each of them with each point on the elliptic curve and then the entire encoded points have to be converted back to bytes and then to strings as SMS can carry only string values.

Once the message reaches the receiver, during the process of decryption, the string has to be converted to bytes; these bytes should be decoded to points again using the mapping technique and then the points to bytes and finally to characters that form the message and only then the decrypted plain text can be viewed by the receiver. The Fig 1 describes the entire process.

V. IMPLEMENTATION

The following pseudo code shows the implementation of the proposed system.

G. Generation of Public and Private Key

```
Algorithm eccGen (a, b, P, G, n) {
        Generate nS (nS<n)
        Calculate PS= G.nS
        Generate nR (nR<n)
        Calculate PR= G.nR
    }
```

H. Generation of common key

```
Algorithm eccKey (nS, PS, nR, PR) {
        Exchange PS and PR
        Sender S computes his Secret Key by
         Computing k = ns.PR
        Receiver R computes his Secret Key by
         Computing k = nr.PS
      }
```

I. Encryption

```
Algorithm eccEncrypt()
    {
        Receive the plaintext
        Use the Secret Key
        Encrypt using ECC algorithm
    }
```

J. Decryption

```
Algorithm eccDecrypt()
    {
        Receive the ciphertext
        Use Secret Key
        Decrypt using ECC algorithm
    }
```

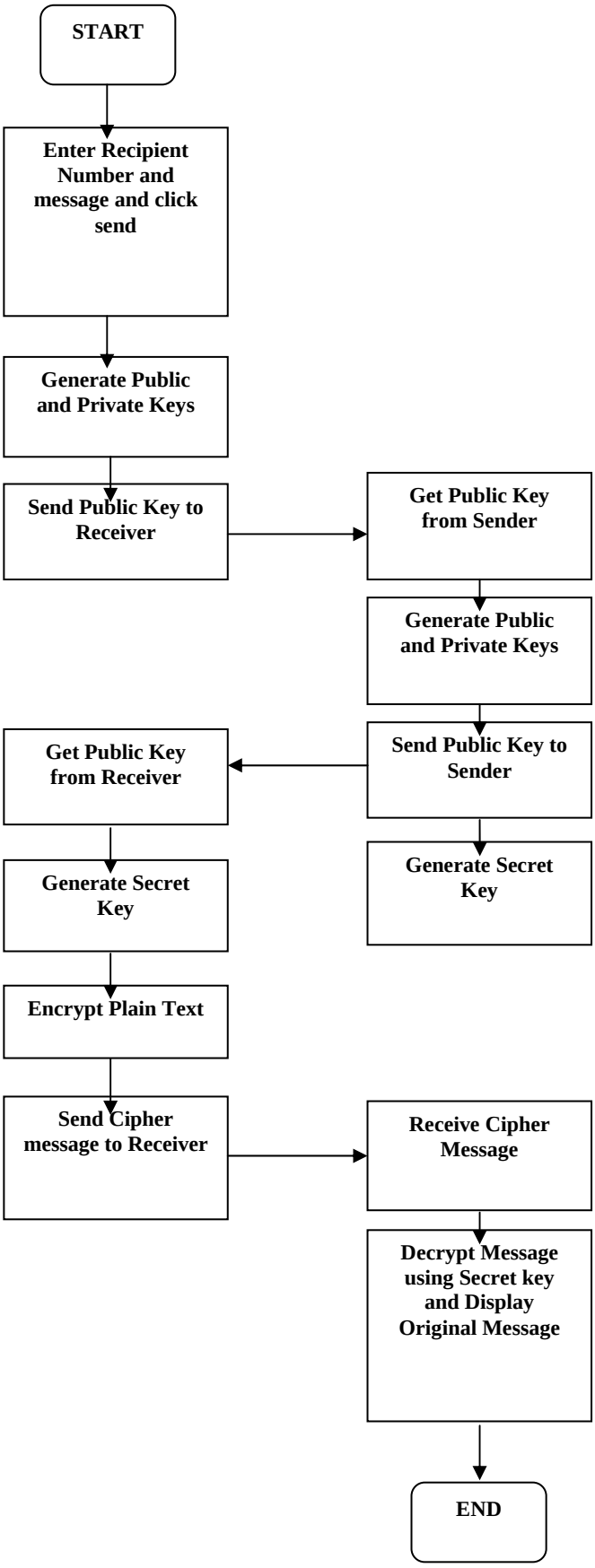

Fig. 1 Step-wise flow of the proposed System

VI. WORKING OF THE PROPOSED SCHEME

1) The sender communicates with the receiver through any normal means about the confidential discussion to be followed.
2) Once both the sender and receiver are ready with their application, the sender types in the recipient's number and the body of the message and clicks Send.
3) On this command, the ECC algorithm is triggered at the sender side and the keys are generated. The sender's Public Key is then sent to the receiver.
4) The receiver acknowledges this by clicking the Read button where the received key is read by the receiver application and the ECC algorithm is triggered at receiver's end. The receiver's Secret Key is generated and its Public Key is sent to the sender.
5) On receiving the Public Key from the receiver, sender's Secret Key is generated at the sender side and the message is encrypted using ECC algorithm and sent to the receiver.
6) The receiver receives the message and decrypts it using his Secret Key with the ECC algorithm to obtain the original message.

VII. ANALYSIS AND CONCLUSION

As mobile devices have less memory and processing power, ECC can be used for message security on mobile. Symmetric key algorithms, can be used on such device, but the authentication of the message is not guaranteed, there is a requirement of secure channel for the transfer of the message along with the key devoid which there could be possibilities of intruder attack on the messages. Considering the present use of the conventional RSA cryptosystem, there is a lot of problem with the key size and the processing speed. When implementing RSA on these devices, smaller keys must be used to meet the memory capacity but this makes the encryption weak.

ECC is useful not only in resource constrained environment like mobile, pager or smart card devices which have limited memory, limited processing capability and limited backup but also on powerful computers because it provides strong security with smaller key sizes. The key between the two parties can be shared in a common network and will not affect the security of the encrypted message due to its discrete logarithmic problem. ECC also provides authenticated transfer of the message as there is an end-to-end secure data transfer.

The most important point of using this means of secure communication is that both the sender and the receiver should be using the same application and should be active at the same time. This ensures high authentication and the fact that the SMS can be decrypted only when it was sent and thus protected from others who try to decrypt it at a later point of time.

Thus, ECC can be efficiently used in securing and authenticating the communication between the two active users. SMS can be sent from one mobile to another without intrusion. Most importantly, the messages containing delicate information are stored securely and remain undisclosed even when the device is accessed by an adversary. The message can be decrypted by the receiver as soon as it is received and never again can be decrypted by any intruder. High Confidentiality can be maintained and thus protect the message information from misuse. The most unique and vital point to be considered is the security of the encrypted data against various attacks such as Brute Force attack, pattern attack etc., due to discrete logarithm problem. This guarantees secure end to end transfer of data without any corrupt data segments.

REFERENCES

[1] Cryptography and Network Security by Behrouz A. Forouzan

[2] Certicom, Standards for Efficient Cryptography, SEC 1: Elliptic Curve Cryptography, Version 1.0, September 2000.

[3] Certicom, Standards for Efficient Cryptography, SEC 2: Recommended Elliptic Curve Domain Parameters, Version 1.0, September 2000.

[4] Implementation of elliptic curve cryptography on text and image International Journal of Enterprise Computing and Business Systems, ISSN (Online): 2230-8849, Vol. 1 Issue 2 July 2011.

[5] Securing MMS with High Performance Elliptic Curve Cryptography, International Journal of Computer Applications (0975 – 8887), Volume 8– No.7, October 2010.

[6] SMS Encryption for Mobile Communication, IEEE 2008 International Conference on Security Technology.

[7] SMS Encryption using AES Algorithm on Android, International Journal of Computer Applications (0975 – 8887) Volume 50– No.19, July 2012.

[8] State of Security for SMS on Mobile Devices, IEEE Electronics, Robotics and Automotive Mechanics Conference 2008.