Secured Messaging Using Hybrid Compression Technique

Dhamak Nikhil, Patil Jyotsna, Gajare Sonali
Pune University Dept. of Comp. Engg. India
Pune University Dept. of Comp. Engg. India
Pune University Dept. of Comp. Engg. India

Abstract: *In today's era SMS message is very fast communication way. Communication done through SMS are more faster and simple. So many people are using SMS communication in their day to day life. Also confidential information is exchanged using SMS. So there is need to secure SMS from different threats. The threats are like DoS attack, Message Disclosure, SMS viruses, Phone crashes. Also need to ensure message is send by authorized sender. In order to achieve all these needs this paper describes the solution for SMS security. We describe hybrid compression encryption technique to secure data. This technique firstly encrypt SMS by Elliptic curve encryption technique and after that compress the encrypted SMS using lossless compression technique.*

Keywords: *Compression; Decryption; Encryption; Security; SMS*

I. Introduction

SMS stands for Short Message Service. SMS is technology which enables sending and receiving of messages between mobile phones. This uses wireless technologies like CDMA and TDMA. The GSM and SMS standard were originally developed by ETSI. ETSI is abbreviation for European Telecommunication Standard Institute[1]. SMS become widespread tool for business and social messaging due to rapid development in mobile communication. Personal and official messages are getting shared between people cost effectively with the help of SMS.SMS has limit for alphanumeric messages is up to 1120 bits.SMS text supports languages internationally. Also it supports languages like Arebic,Chinese,Japanes and Korean.SMS uses communication protocol such as Short Message Peer-to-Peer (SMPP).

II. Traditional Problem

In our regular messaging system when SMS sends from A to B, at that time until it receives to B it stored at SMS center, so it is visible to operator due to that message disclosure problem occurs. Again there is problem arises like memory wastage which hangs mobile phone. Our today's, SMS technique do not verify whether messages comes from trusted sender or not. The user does not authenticate network so that same mobile network get misused. Sometimes corrupted messages affects mobile phone badly.

III. Security

1. **SMS security need:**
   - **1.1** Authentication: Confirm true identifies between sender and receiver.
   - **1.2** Confidentiality: Only authorized senders and receivers can access decrypted messages.
   - **1.3** Integrity: Prevent tampered also ensure that receiver can check whether message has been modified.

2. **SMS security threats**

2.1 Message Disclosure: In short message service, encryption is not applied so message could be intercepted and snooped during transmission. Also SMS when stored at SMSC, they are in plain text format so easily understandable to operator.

2.2 Denial of Service (DoS) Attacks: By sending repeated messages to target mobile phone, making it inaccessible made DoS attack.

2.3 SMS phone Crashes: If mobile phone receives a particular type of malformed short message then phone get infected and becomes inoperable.

2.4 SMS Viruses: No reports of viruses being attached to short messages, but as mobile phone are getting more powerful and programmable, so potential of viruses being increased.

IV. Securing Sms Techniques

We describe two technique used to secure SMS. The two main steps of these techniques are encryption and compression process.

1.Encryption

First step to encrypt message. Encryption can be classified into two categories symmetric and asymmetric. In symmetric encryption technique single key is used for encryption and decryption while in asymmetric technique different keys are used for encryption and decryption . one of which is public key and other is private key. The major disadvantage of symmetric encryption is that key distribution done through a third party. So we are using asymmetric encryption technique for securing SMS. There are following asymmetric encryption technique. Rivest Shamir And Adleman (RSA), ELGamal and Elliptic curve.

1.1 RSA cryptosystem

```
Algorithm:
Step1: Choose two large primary numbers P & Q.
Step2: Calculate N=P*Q
Step3: Select public key  E, such
         that it is not factor of
         (p-l) and (Q-l)
         i.e. 8(n)=(P-l)*(Q-l)
         E < 8(n)
         gcd(8 (n),E)=1
Step4: Select the private key D. where D=Ki
         mod8(n)
Step5: For encrypt message M. C=ME mod n
Step6: Send cipher text as the message to the
          receiver.
Step7: To decrypt the cipher text M=CD mod n
```

1.2 ELGamal Cryptosystem

```
Algorithm:
Stepl: Choose a prime number P and generator g.
Step2: Choose an integer K such that 2::; K 2: P-2.
Step3: Calculate Y=gK mod P
Step4: The public key is {p, g, y} and Private key is
          K.
Step5: To encrypt message M ,
          Choose an integer 1  i.e. 1≤ 1 ≥P-2.
          Therefore, the ciphertext {a ,b} is
          a= $g^1$ mod  p
          b= M $Y^1$ mod P
Step6: Send ciphertext {a ,b} to the receiver
Ste 7: To decrypt
        M=b / ($a^K$ mod p)
```

1.3 Elliptic Curve Cryptosystem

Algorithm:

Step1: Consider a message 'Pm' sent from A to B. 'A' chooses a random positive integer 'K', a private key 'nA' and generates the public key P A=nA *G and produces the cipher text 'Cm' consisting of pair of points Cm={kG,Pm+kPB}

where G is the base point selected on the Elliptic Curve, PB=nB*G is the public key of B with private key 'nB'

Step2: To decrypt the ciphertext, B multiplies 1st point in the pair by B's secret & Subtract the result from the 2nd point Pm + kPB - nB(kG) = Pm + k(nB G) - nB(kG)=Pm[2]

2.. Compression

Compression is the art of representing the information in a compact form rather than its original or uncompressed form. Compression can be done by using lossy or lossless technique. In lossy technique data may be loss. So here we are using lossless compression algorithm. Some of the main technique use are the Huffman Coding, Run Length Encoding, Arithmetic coding and Dictionary Based Encoding. For better performance Shannon Fano get used.

2.1 Shannon Fano:

Algorithm:

1. Create table providing frequencies.

2. Sort symbol according to frequency in descending order.
3. Start with the entire table.

I. Division
II. Seek pointer to the first and last symbol to the segment.
III. Divide the segment into two parts, which are equal in sum of frequencies.
IV. Add a binary 0 to the code words of the upper part and 1 to the lower part.
V. Search for the next segment containing more than symbol and repeat division.
VI. Coding of the origination data according to the code word in the table.

V. Security Analysis

1. Security Strength Comparison

Asymmetric cryptosystem security depend on hard mathematical equation such as integer factorization, finite field discrete logarithm and Elliptic curve discrete logarithm. RSA security based on integer factorization. ELGamal security depend on computational difficulty discrete logarithms in finite field. Elliptic curve security depend on computational difficulty of discrete logarithm. National Institute of Standard and Technology (NIST) guideline on security strength comparison. Though Elliptic curve is of smaller key size, offers equivalent security compare to RSA and ELGmal which are of large key sizes. Table 1 shows NIST security strength comparison.

Table 1:Algorithm security strength comparison

Security bits	RSA	ELGamal	Elliptic
80	1024	1024	160
112	2048	2048	224
128	3072	3072	256
192	7680	7680	384
256	15360	15360	512

Elliptic curve of 160 and 224 bits sizes offer equivalent security as compared with 1024 and 2048 bits sizes for RSA and ELGamal. In Fig. 1 shows that RSA and ELGamal of equal key size offers same security strength.

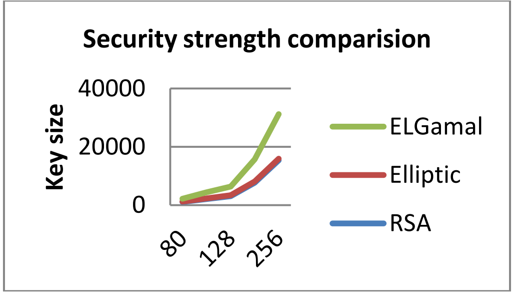

Fig. 1 Security strength comparision

2. Encryption comparison

2.1 Key generation time

Time taken to generate a key pair, called key generation time. Five test runs for each key size of same algorithm were performed and the average was the calculated using equation(1):

$$K_t \ = 1/n \sum_{i=1}^{n=4} \mathrm{t_i}$$

$\mathrm{T_i}$ is the consecutive key generation time and $\mathrm{K_t}$, is the average key generation time.

Table 2.1Key generation time in milliseconds

Key size	RSA	ELGamal	Elliptic
160	951	406	2437
224	1237	830	4449
256	1957	895	6451
512	16863	15946	40317

In table 2.1 key generation time and key sizes are related to each other. Rise in key size, leads to increase in key generation time. We can say that Elliptic curve have a lower key generation time compared to RSA and ELGamal of equivalent security key strength.

2.2 Encryption time

The time taken to process SMS plaintext into cipher text is called encryption time. The average of the encryption time is calculated using following equation(2):

$$\mathrm{E_t} = 1/n \sum_{i=1}^{n=3} e_i$$

Where , e_i is the consecutive encryption time and E_t is average encryption time.

Table 2.2. Encryption time in millisecond

Key size	RSA	ELGamal	Elliptic
160	18	2816	2696
224	28	5842	6268
256	37	7098	8242

Table 2.2 shows that ELGamal have a high encryption time compared to Elliptic curve and RSA which are of equivalent security strength.

2.3 Decryption time

Time taken to process cipher text back to the plain text called Decryption time. It is calculated using following equation(3):

$$\mathrm{D_t} = 1/n \sum_{i=1}^{n=3} d_i$$

Where , d_i is the consecutive decryption time and D_t is average decryption time.

Table 2.3. Encryption time in millisecond

Key size	RSA	ELGamal	Elliptic
160	10	18	1292
224	28	27	2907
256	37	37	3932

Table 2.3 do not show significant difference in decryption time of 1024 bits key size for RSA and ELGamal as compared to 160 bits key size for Elliptic curve.

V. Implimentation

Key generation, encryption, decryption time analyzed as above, specifies that time increases with an increase in key size. Large key size algorithm not suitable for SMS encryption due to small memory & low computational power of mobile phone. Elliptic curve provide high security with smaller key size, so we are implementing security threats based on this algorithm. And encrypted message is then compressed using lossless algorithm.

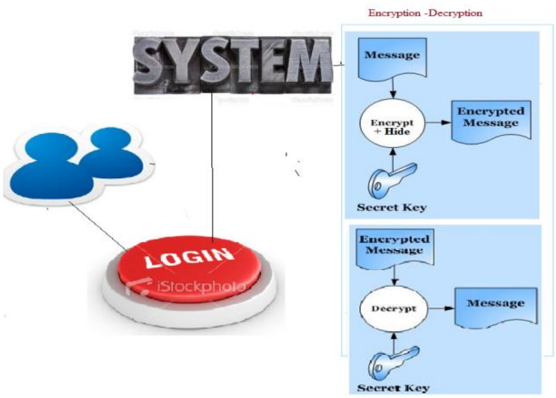

Fig 2: Hybrid Compression Technique

The steps involved are:

Step1: Take SMS.
Step2: Determine SMS receiver.
Step3: Encrypt the received SMS using Elliptic curve algorithm.
Step4: Compress encrypted SMS.
Step5: Send compressed SMS.

VI. Conclusion

In our paper we combines encryption & compression techniques for securing SMS. For encryption we have used Elliptic curve algorithm while for compression lossless algorithm in that Shannon Fano algorithm used. Our proposed system provides additional security related to authentication, confidentiality and integrity. Also we eliminates threats like DoS attack, phone crashes, message disclosure etc. Due to the use of compression algorithm message length also get decrease.

Acknowledgement

We have made this opportunity to thank those selected few who have extended their kind cooperation and guidance and have made this paper success. With the deep sense of humbleness, our sincere heartfelt gratitude to our, Guide Prof. S. D. Jondhale, Department of Computer Engineering whose valuable guidance and keen interest inspired our to complete the seminar in a successful manner. He offered us all the independence and at the same time kept an eye to proceed on the right track. We would like to thank him for his moral as well as technical support for obtaining information from other institute and establishments. We also thankful to all Teaching and non Teaching members of Systems, Pernambuco, Brazil, 2009, pp. I 65- 170. Computer Engineering Department who helped us throughout this task.

Reference

[1] A Medani, A Gani, O.Zakaria, A.A. Zaidan, "Review of mobile short message service security issues and techniques towords the solution", Scientific Reaserch and Essays voI.6(6),pp. I 147-1 165,18 March, 201 IJ. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2.Oxford: Clarendon, 1892, pp.68-73.

[2] R. Ghosal and P. H. Cole, "Elliptic curve cryptography", University of Adelaide Auto-ID Labs, Technical Report.

[3] J. P. A1buja and E. V. Carrera, 'Trusted SMS communication on mobile devices", I Ith Brazilian Workshop on Real-Time and Embedded

[4] R. R. chavan and M. sabnees, “Secuerd mobile messaging”, International Conference On Computing, Electronics and Electrical Technologies [ICCEET] 2012.