A Secure Digital Signature Approach for SMS Security

Neetesh Saxena
Department of Computer Sc. & Engineering
Indian Institute of Technology, Indore, India

Narendra S. Chaudhari
Department of Computer Sc. & Engineering
Indian Institute of Technology, Indore, India

ABSTRACT

Global System for Mobile (GSM) is a second generation cellular standard developed to cater voice services and data delivery using digital modulation. 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 between mobile phone devices. SMS framework allows two peers to exchange encrypted and digitally signed SMS messages. The communication between peers is secured by using public key cryptography. The identity validation of the contacts involved in the communication is implemented through ECDSA signature scheme. In the next part, there is the description of ECDSA approach and a modified approach based on ECDSA for mobile phones, which signs SMS. At the end, there is described attack on ECDSA for secured SMS and future extension of the application.

General Terms

SMS Security, Algorithms et. al.

Keywords

GSM, SMS security, ECDSA, ECDLP, public key cryptography

1. INTRODUCTION

The mobile phone is already an integral part of the lives of more than 1.8 billion people worldwide. Mobile usage is increasing in volume as well as diversity. More than 80 % of mobile users do not leave home without their phones. The Short Message Service (SMS) facility plays a leading role in this adoption. The first SMS message was sent over the Vodafone GSM network in the United Kingdom on 3 December 1992, from Neil Papworth of Sema Group using a personal computer to Richard Jarvis of Vodafone using an Orbitel 901 handset. The first commercially sold SMS service was offered to consumers, as a person-to-person text messaging service by Radiolinja in 1993. SMS as used on modern handsets was originally defined as part of the Global System for Mobile Communications (GSM) series of standards in 1985 as a means of sending messages of up to 160 characters, to and from GSM mobile handsets.

SMS messages are currently one of the most widespread forms of communication (in 2008 about six trillion SMS were sent globally [1]). We have seen many unusual or strange applications, such as devices which allow the switching on and off of house heating systems using an SMS [2]. Alternatively, through SMS, whenever the temperature of a refrigerator exceeds a certain threshold, it is possible to automatically communicate the problem [3]. Figure 1 and Figure 2 show the role of GSM in this world with different distributers of this service in 2005 [20]. Indeed, through SMS, fridges can even signal when they are running out of beer [4]. This is the case, for example, of the service provided by the Province of Rimini Mobility Agency, in Italy, which allows registered users to buy electronic tickets using a simple SMS which contain a standard fixed string of text [5]. So, services like the ones we mentioned before are prone to be attacked by malicious users [6].

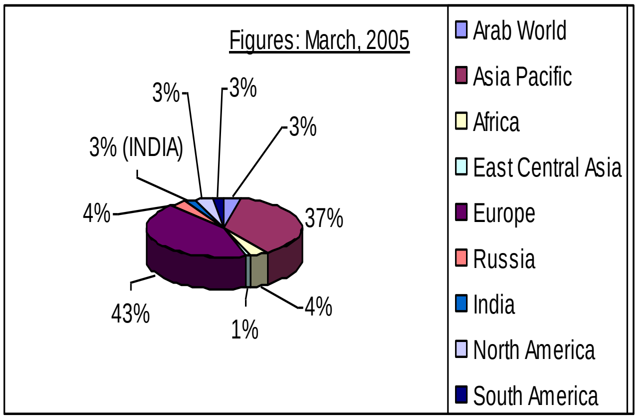

Fig 1: GSM in World: GSM across the world [20]

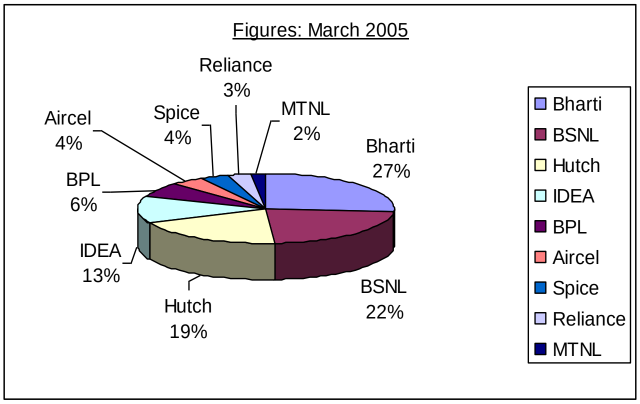

Fig 2: GSM in World: Mobile Operators distribution [20]

The mobile communications has experienced a great acceptance among the human societies. At the beginning of 2007, the worldwide number of mobile users reached to 2.83 billion people [7]. It has also been used in applications where the other party is an information system. This includes a wide variety of applications ranging from remote control of the apparatus [8] to the m-banking and m-payment. The GSM with the greatest worldwide number of users suffers from many security problems. The tapping can be realized in a mobile phone too. We can find the tapping programs in the market today. These programs re-send received and sent SMS to an attacker’s number. The program is hidden after installation. This program can be even uninstalled remotely, when the phone receives an SMS in a proper format [12]. The attacker can tap an SMS, but also send a fake SMS. Today, you can send SMS with arbitrarily phone number of the sender. It is possible to prepay this service on certain websites [13].

2. RELATED WORK

In a study by Mary Agoyi and Devrim Seral [1] large key size algorithms are not suitable for SMS encryption due to small memory and low computational power of mobile phones. Elliptic curve's ability of providing high security with smaller key size makes it very useful in resource-limited device such as mobile phone. This has put Elliptic curve at an advantage over the RSA and ELGamal in SMS encryption. In the paper of Alfredo De Santis, Aniello Castiglione and Umberto Ferraro Petrillo [2] the results seem to show that RSA and DSA cryptosystems perform generally better than ECDSA, except when using very large keys. Nassim Khozooyi, Maryam Tahajod and Peyman khozooyi [3] are discussed the security of mobile network protocol along with information security for governmental transactions. A new public key-based solution for secure SMS messaging (SSMS) is introduced by M. Toorani and A. Beheshti Shirazi [4]. It efficiently combines encryption and digital signature and uses public keys for a secure key establishment to be used for encrypting the short messages via a symmetric encryption. Since it deploys elliptic curves and a symmetric encryption algorithm, it has great computational advantages over the previously proposed public key solutions while simultaneously providing the most feasible security services. In a study of D. Lisonek and M. Drahansky [5] the application for securing of SMS has been designed and implemented, which prevents tapping and also substituting. For securing, it has been chosen the asymmetric cipher RSA. Brutal force decryption of RSA cipher with a length of 1,024 bit keys has not been successfully implemented yet. The best success is from year 2005, where J. Franke (University of Bonn) was able to factorize number with a length of 663 bits. The attacker also cannot be able of tapping and gradually building up the dictionary because it is used in the OAEP padding scheme. In the paper of C. Narendiran, S. Albert Rabara and N. Rajendran [6] an end-to-end security framework using PKI for mobile banking is proposed. The security framework solution allows us to provide strong customer authentication and non-repudiation by employing public-key cryptography for customer certificates and digital signatures. It is observed that the AES algorithm utilized less computation time and memory for encrypting the user's data. The AES model shows greater performance than the 3DES and RSA model that uses Public Key Infrastructure. Mahmoud Reza Hashemi and Elahe Soroush [7] proposed a secure m-payment protocol for mobile devices. They used a 163-bit key for ECC computations, which is proven to be equivalent to a 1024-bit key for RSA. The results show that ECDSA consumes less power than DSA. However, ECDSA and RSA digital signature algorithms have complementary power costs. RSA performs signature verification efficiently, while ECDSA imposes a smaller cost for signature generation. In the paper of Mohsen Toorani, Ali Asghar and Beheshti Shirazi [8], the security of the GSM network is evaluated, and a complete and brief review of its security problems is presented. Some practical solutions to improve the security of currently available 2G networks are also proposed.

3. SMS ARCHITECTURE

Short messages are delivered in GSM signaling channels between the Mobile Station (MS) and the Base Transceiver Station (BTS). Figure 3 shows the basic architecture for GSM-SMS [19]. The messages flow as normal calls, but they are routed from the MSC to a Short Message Service Center (SMSC). The SMSC stores the message until it can be delivered to the recipient(s) or until the message's validity time has elapsed. The recipient can be a normal MS user or a SMS gateway. The gateways are servers that are connected to one or more SMSCs to provide SMS applications for the MS users. These applications include ring tone and icon delivery, entertainment, bank services, and many other beneficial services.

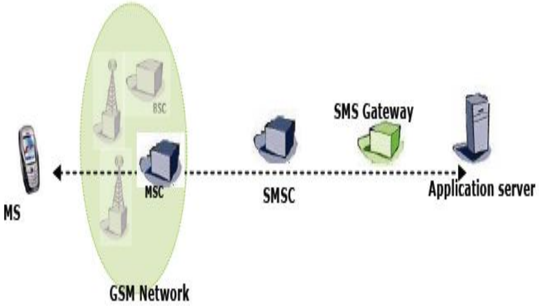

Fig 3: SMS Architecture [19]

3.1 SMS Security Issues and Vulnerabilities

Two important aspects for any entity using consumer technologies such as SMS for business purposes:

1. SMS is not a secure environment.

2. Security breaches often occur more easily by concentrating on people rather than technology.

The contents of SMS messages are visible to the network operator's systems and personnel. Therefore, SMS is not an appropriate technology for secure communications. Most users unaware of how easy it is to intercept SMS messages. It would likely be a relatively complex to hack into a telecom provider's systems to obtain the content of SMS messages, but finding staff privileged to look at SMS messages and persuading them to reveal the contents is much easier. The underlying specifications and technology for SMS transmission leave many security gaps. These gaps make SMS vulnerable to –

Snooping: - On device, at the store and forward network elements

SMS Interception:-Over the air, in wired network

Spoofing: - Using commercial tools, own SMS gateway

Modification: - Using conventional hacking techniques

Attacks on GSM, the SMS Carrier Technology: - Often the weakest link in security is the mobile phone itself. Even leaving the mobile phone unattended inadvertently could expose your private and confidential messages to snooping.

3.2 SMS Security

Data security has at least four important requirements to meet, as listed [19]:

(1) Secrecy: Also known as confidentiality. It is the effect of keeping unauthorized parties from accessing private information. Interception is the typical direct attack on secrecy.

(2) Integrity: It is preventing anybody other that authorized parties from modifying the computer system assets like writing, changing status and deleting and creating files. Among the methods of attacking integrity we found modification, replay and reordering of messages.

(3) Availability: It is the fact of being able to access information when needed and the prevention of unauthorized parties from withholding access to information. Inception and denial of service are the attacks over availability.

(4) Authenticity: Prevents that unauthorized parties can change the content of message or place random messages in the network. Fabrication, which is the unauthorized insertion of data, is an example of an attack over authenticity.

The attacks are divided into two types: Passive and Active [16] [13]. Passive Attacks imply the eavesdropping on, or monitoring of transmissions. The intruder's goal is to obtain information that has being transmitted for further traffic analysis or release of message contents. Passive Attacks are very difficult to detect because they do not involve any alteration of the data. However, it is possible to prevent the success of these attacks. Then, the emphasis in dealing with passive attacks is on prevention rather than detection. Active Attacks involve some modification(s) of the data stream or the creation of a false stream. It can he subdivided into four categories: modification of messages, masquerade, replay, and denial of service. It is clear from our discussion than active attacks have the opposite characteristics of passive attacks. Whereas passive attacks are difficult to detect measures are available to prevent their success. Active attacks are very difficult to prevent absolutely, because it would require physical protection of all communications facilities and paths at all times. The goal is to detect them and to recover from any disruption or delays caused by them [13].

4. APPROACH FOR SMS SECURITY

To protect the SMS contents, we usually do the encryption. It may be symmetric or asymmetric encryption. But our focus here is on the non-repudiation. If a SMS has sent by user 'A' to user 'B', 'A' must not deny that he has sent SMS. This feature can be achieved by imposing digital signature. The popular digital signature algorithms are DSA and elliptic curve based ECDSA. The bit size of the public key believed to be needed for ECDSA is about twice the size of the security level, in bits. By comparison, at a security level of 80 bits, meaning an attacker requires the equivalent of about 2^{80} signature generations to find the private key, the size of a DSA public key is at least 1024 bits, whereas the size of an ECDSA public key would be 160 bits. On the other hand, the signature size is the same for both DSA and ECDSA.

4.1 ECDSA algorithm

The mathematical basis for the security of elliptic curve cryptosystems is the computational intractability of the elliptic curve discrete logarithm problem (ECDLP) [13]. Select a rational point G on E(GF(2^n)) , called it base point, find n which is a prime number satisfies the formula nG = O, and select a one-way secure Hash function h(m) (such as SHA-1). For each system user, he has a private key 'd', calculate the public key P = dG. If user A wants to sign on the message m, the scheme can be described as:

1. User A selects an integer k randomly, 0 < k < n, calculate k*G = (x, y), r = x mod n; if r = 0, return to (1).

2. Calculate e = h(m) ; s = k^{-1} (e + r * d) mod n, if s = 0 , return to (1).

3. Take (r, s) as the digital signature of message m by A.

The verification of digital signature:

1. Calculate e_1 =h(m_1), u= s^{-1} * e_1 mod n, v= s^{-1} * r mod n

2. Calculate X= u*G + v*P = s^{-1} (e_1 * G + r * d * G) = s^{-1} (e_1 + r * d) * G = k * G = (x_1, y_1)

3. If X = 0, this signature is refused; else calculates $r_1 = x_1$ mod n; if $r = r_1$, the confirmer accepts this signature.

4.2 Possible Attack

The per-message secrets k used to sign two or more messages should be generated independently of each other. In particular, a different per-message secret k should be generated for each different message signed; otherwise, the private key d can be recovered [9]. Note that if a secure random or pseudorandom number generator is used, then the chance of generating a repeated k value is negligible. Suppose that the same per-message secret k is used to generate ECDSA signatures (r, s_1) and (r, s_2) on 2 different messages m_1 and m_2. Then

s_1= k^{-1} ($e_1 + d*r$) mod n, and s_2 = k^{-1} ($e_2 + d*r$) mod n

where e_1 and e_2 are the message digest of some cryptographic algorithms like SHA1. Thus,

e_1 = SHA1(m_1) and e_2 = SHA1(m_2)

then, k * s_1 = ($e_1 + d*r$) mod n, and k * s_2 = ($e_2 + d*r$) mod n,

Thus, k (s_1 - s_2) = ($e_1 - e_2$) mod n

k = $(s_1 - s_2)^{-1}$ * ($e_1 - e_2$) mod n

Thus, an adversary can determine the k, and use it to recover d.

4.3 Alternate Proposed Approach

The proposed approach is inspired and extended from the approach [10] and is similar to the ECDSA algorithm. Choose a single number based on the elliptic curve logarithm problem may not be secure enough in quantum computer's environment. There must be some more complexity in order to make the system more secure.

Parameters

The following are commonly required parameters over the elliptic curve domain.

(1) A field size q, {q = p, if p is an odd prime (the common practice), or q = 2^m if q is a prime power.}

(2) Two parameters (a, b) ∈ Fq for elliptic group Eq (a, b), to define the elliptic-curve equation E over Fq: $y^2 = x^3 + ax + b$ (mod q) in case that q>3, where $4a^3 + 27b^2 \neq 0(\bmod q)$. E should be divisible by a large prime number.

(3) A finite point B = (x_b, y_b) whose order is a large prime number in E(Fq), where B≠O (O denotes infinity) such that the order of B is n.

(4) Choose m points $B_1, B_2, ...B_m$, with order n in the group E(Fq).

Key Generation

Signer A generates the public key, as follows:

Step1: Randomly select m integers ($d_1, d_2, ...d_m$) from the interval [1, n−1] as the secret-key pair.

Step2: Compute the corresponding public key P to ($d_1, d_2, ...d_m$), as follows.

P = $d_1 * B_1 + d_2 * B_2 + ... + d_m * B_m$

Signature Generation

Signer A generates the signature for the message msg, as follows.

Step1: Randomly select 'm' numbers ($k_1, k_2, ...k_m$) from [1, n−1] to compute T.

T = $k_1 * B_1 + k_2 * B_2 + ... + k_m * B_m$

Step2: Convert the message msg and the value T into one integer e using hash-function operation.
e = h(msg, T)

Step3: Generate the signature ($s_1, s_2, ...s_m$), as follows:

$$s_1 = (k_1 + d_1 * e) \bmod n$$
$$s_2 = (k_2 + d_2 * e) \bmod n$$

…

$$s_m = (k_m + d_m * e) \bmod n$$

Step4: Send (e, $s_1, s_2, ...s_m$) to the verifier.

Signature Verification

The verifier confirms the validity of the signature for message, as follows:

Step1: Determine R following R = $s_1 * B_1 + s_2 * B_2 + ... + s_m * B_m$ – e*P

Step2: Determine e following e = h(msg, R).

Step3: If the resulting e meets with the received one, then validate the signature; otherwise, reject it.

{R = $s_1 * B_1 + s_2 * B_2 + ... + s_m * B_m$ – e*P =
[$(k_1 + d_1 * e) \bmod n$] B_1 + [$(k_2 + d_2 * e) \bmod n$] B_2 +
……+ [$(k_m + d_m * e) \bmod n$] B_m – e*P =
($k_1 * B_1 + k_2 * B_2 + ... + k_m * B_m$) +
e*($d_1 * B_1 + d_2 * B_2 + ... + d_m * B_m$) – e*P = T + e*P – e*P =
T }

5. SECURITY ANALYSIS

In the typical digital signature schemes such as ECDSA, a public key only corresponds to one secret key [10]. Given the secret key d, let the public key P be derived according to the equation P=dB, and let the signature T be derived using a random number ‘k’ following the equation T = k*B. If T equals to the public key P, then the corresponding secret key is the same as ‘k’, as shown below.

*T = k*B* and *P = d*B*

But, it’s almost impossible to find out

$k_1 * B_1 + k_2 * B_2 + ... + k_m * B_m$ =

$d_1 * B_1 + d_2 * B_2 + ... + d_m * B_m$

If an attacker attempts to derive the secret-key from the public key, he has to encounter the difficulty of solving the ECDLP.

The case when an attacker intends to forge an individual signature for a message m. To forge a valid individual signature for a message m, an attacker randomly selects a point P to determine e following e = h(msg, R). In addition to P and e, the attacker derives the signature by the public data ($B_1, B_2, ...B_m$) and P. Such solutions of unknown numbers ($s_1, s_2, ...s_m$) here also depend on the ECDLP, and it is infeasible in reasonable computational security.

6. VULNERABILITY WITH SAT

It’s necessary to check the vulnerability of the existing elliptic curve based digital signature algorithms. These algorithms are based on the hardness of elliptic discrete logarithm problem which is still a NP-Complete problem. For this purpose we prefer to do it using Boolean Satisfiability Problem (SAT), as we have polynomial time solution for 2-SAT and 3-SAT.

7. CONCLUSION & FUTURE WORK

The elliptic curve discrete logarithm problem is significantly more difficult than the integer factorization problem. For the most part, the well-known RSA system must use 1024 bit keys, only then can it attain computationally reasonable security; the ECC needs only 160 bit keys [14]. So, at the same level of security, the speed of ECC is several times faster than RSA system; it can also saves on key storage space [15]. Elliptic curve discrete logarithm problem based systems are considered as secure systems but in quantum computer’s environment these systems may not be secured. In future, it’s necessary to check the vulnerability of the system (attempt to break) and develop a secure environment. So, the focus will be on Boolean satisfiability problem (SAT). We’ll check the vulnerability of the systems based on elliptic curve discrete logarithm problem as we have a polynomial time algorithm for 2-Sat and 3-SAT. Our most focus will be on 3-SAT [17][18].

8. ACKNOWLEDGMENTS

Authors give thanks to IIT Indore for their support and co-operation for this research work.

9. REFERENCES

[1] M. Agoyi, D. Seral, “SMS Security: An Asymmetric Encryption Approach”, Sixth International Conference on Wireless and Mobile Communications, 2010, pp. 448-452.

[2] A. D. Santis, A. Castiglione and U. F. Petrillo “An Extensible Framework for Efficient Secure SMS” International Conference on Complex, Intelligent and Software Intensive Systems, 2010, pp. 843-850.

[3] N. Khozooyi, M. Tahajod, P. Khozooyi, “Security in Mobile Governmental Transactions”, Second International Conference on Computer and Electrical Engineering, 2009, pp 168-172.

[4] M. Toorani and A. B. Shirazi, “SSMS-A secure SMS messaging protocol for the m-payment systems”, IEEE Symposium on Computers and Communications, July 2008, pp. 700–705.

[5] D. Lisonek and M. Drahansky, “SMS Encryption for Mobile Communication”, International Conference on Security Technology, (SECTECH’08), Dec. 2008, pp. 198–201.

[6] C. Narendiran, S. A. Rabara, N. Rajendran, “Performance Evaluation on End-to-End Security Architecture for Mobile Banking System”, IFIP Wireless Days, 2008, pp. 1-5.

[7] M. R. Hashemi, E. Soroush, “A Secure m-Payment Protocol for Mobile Devices”, IEEE CCECE/CCGEI, 2006, pp 294-297.

[8] M. Toorani, A. Beheshti, “Solutions to the GSM Security Weaknesses”, the Second International Conference on Next Generation Mobile Applications, Services, and Technologies, 2008, pp. 576-581.

[9] A. Menezes, M. Qu, D. Stinson, Y. Wang, “Evaluation of Security Level of Cryptography: ECDSA Signature Scheme”, Certicom Research, 2001.

[10] Y. F. Chung, K. H. Huang, F. Lai, T. S. Chen, “ID-based digital signature scheme on elliptic curve cryptosystem”, Computer Standards & Interfaces, vol. 29, 2007, pp. 601-604.

[11] W. R. C. Phan “Fixing the Integrated Diffie-Hellman - DSA Key Exchange Protocol”, IEEE Communications Letters, vol. 9, no. 6, JUNE 2005, pp. 570-572.

[12] [Online]: http://www.rsa.com/rsalabs/node.asp ?id=2879

[13] W. Stallings, “Cryptography and Network Security”, 4th Ed., Prentice Hall, 2005, pp. 58-309.

[14] J. Peng, Q. Wu “Research and implementation of RSA algorithm in Java”, International Conference on Management of e-Commerce and e-Government, 2001.

[15] S. J. Aboud, M. A. AL-Fayoumi1, M. Al-Fayoumi and H. S. Jabbar “An Efficient RSA Public Key Encryption Scheme”, Fifth International Conference on Information Technology: New Generations, 2002.

[16] A. Nadeem, M. Y. Javed, "A Performance Comparison of Data Encryption Algorithms," First IEEE International Conference on Information and Communication Technologies, 2005, pp. 84- 89.

[17] N. S. Chaudhari, "Polynomial Solvability of 3-SAT -Part III: Polynomial algorithm for 3-SAT", NHSS, Udaipur, India, ISBN: 978-81-7906-266-1, Feb 2011, pp. 71-76.

[18] N. S. Chaudhari, "Polynomial Solvability of 3-SAT -Part II: Algorithmic formulations for 2-SAT", NHSS, Udaipur, India, ISBN: 978-81-7906-266-1, Feb 2011, pp. 59-64.

[19] N. Saxena and A. Payal, "Enhancing Security System of Short Message Service for M-Commerce in GSM", International Journal of Computer Science & Engineering Technology (IJCSET), ISSN: 2229-3345 vol. 2, no. 4, April 2011, pp. 126-133.

[20] [Online]: www.classle.net/sites/default/files/text/28978/RK-3_GSM_Network.ppt