Journal of Information Assurance and Security.
ISSN 1554-1010 Volume 9 (2014) pp. 157-166

An HTTPS approach to resist Man In The Middle attack in secure SMS

Muhammad Murad Khan, Majid Bakhtiari, Saeid Bakhtiari

Faculty of Computing
Universiti Teknologi Malaysia
Skudai, Malaysia
muradtariq.tk, bakhtiari.majid}@gmail.com, saeid_bakhtiarii@yahoo.com

Abstract: **Short Messaging Service (SMS) has succeeded in teleporting expressions on click of a button using textual medium. Because of guaranteed delivery, people use it on daily basis for connectivity. Not only humans but Information Systems have also utilized it to connect, known as automated messaging. Although automated messaging holds prominent SMS market share but full potential was not reached because of the fact that SMS has no security at all. Any SMS sent or received can be intercepted and manipulated by Man In The Middle (MITM). To resist MITM in next generation smartphones we have proposed a new framework which can be used to secure both human and automated messaging. This new framework use HyperText Transfer Protocol Secure (HTTPS) for secure key exchange, ECC, RSA as encryption algorithm and GSM network to send and receive encrypted messages.**
Keywords: SMS, HTTPS, ANDROID, ECC, RSA.

I. Introduction

Most of us are familiar with use of SMS in daily life but formally SMS can be defined as a wireless service which is globally accepted for transmission of alpha numeric data between two mobile subscribers [1]. Mobile number can today be considered as customers secondary identity and service providers use it to exchange vital information. Although SMS market consists of human and automated messaging, full SMS potential was not reached because it has security weaknesses in transport as well as storage layer which can be exploited by Man In The Middle to hijack communication [2].
In this paper GSM Network architecture along with related security issues are discussed in Section II and Section III. Existing security solutions are discussed in Section IV. Main security threat i.e. MITM Attack in discussed in Section V followed by SSL and HTTPS as solution to MITM Attack are discussed in Section VI. Framework to secure SMS is proposed in Section VII and ECC, RSA integration into proposed framework is discussed in Section VIII. Experiments are illustrated in Section IX, conclusion is made in Section X and finally support for this project is Acknowledged. This research was done as a Masters project [3] and extends paper published in ISDA 2013 conference [4].

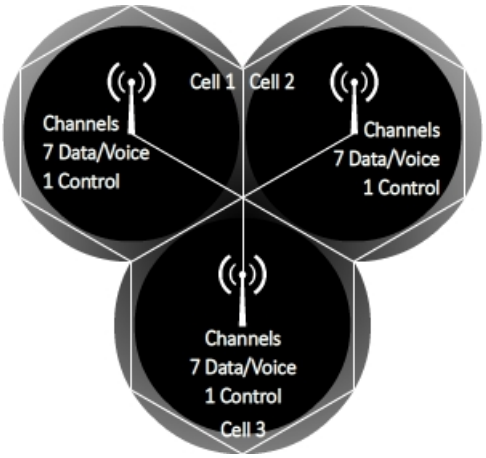

Figure. 1: Cell channel division

II. GSM Network Architecture and Wrokflow

In this section, first underlying hardware e.g. GSM Network Components and Architecture is defined followed by SMS operations over network [5].

A. GSM Structural Components

SMS has guaranteed delivery and parallel availability. To get deeper insight, components involved in SMS communication are described below.
Mobile Station (MS): Mobile station is an end to end terminal device used to send, receive and store short message. Term ”mobile station” is used for mobile phones and programmable GSM modems.
Short Message Service Centre (SMSC): SMSC is SMS server residing inside operator network responsible for reception, storage, forwarding and delivery of SMS between end users.
Base Station System (BSS): Base station system is part of mobile network which is responsible for traffic handling and signaling between network switching system and mobile station. Base Station usually assigns seven channels to Data/Voice and one channel for control signals inside a cell, where cell covers a specific geographical area [6] as shown in Fig. 1.

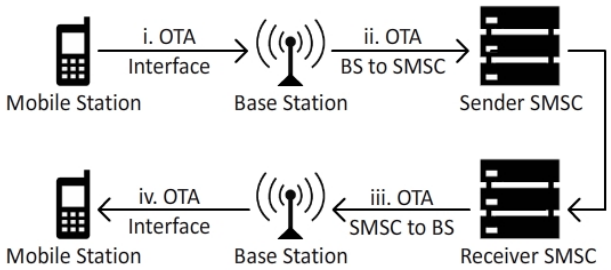

Figure. 2: SMS exchange over network

B. SMS Transmission

Now as we know operations executed by MS, SMSC, BS, a simple illustration in Fig. 2 will show how SMS is communicated between two parties [7] .

i) Mobile device sends SMS to the closest base station using standard transmission and reception channel (control channel) called On-The-Air interface.

ii) From base station SMS is forwarded to SMSC over SS7.

iii) Source SMSC sends SMS to destination SMSC.

iv) After querying destination location, Short Messaging Service Center forwards message over SS7 to the closest base station near destination.

Through On-The-Air Interface, SMS reach its destination and then a delivery report is initiated back to the sender.

C. SMS Formats

SMS messages can be constructed using three formats block mode, text mode and PDU mode. Each mode has its benefits and flaws [8] [9].
Block mode: In block mode SMS is divided into block and transmitted over network. Block mode has an advantage that it can perform error correction based on check sum operation on blocks of data but cannot allow Voice or Data communication until Block Mode is terminated.
Text Mode: This mode of operation is most common. Messages in this mode are sent and received in subset of ACSII characters with 7 bit representation, making one SMS size equal to 160 characters. This mode requires setting extra perimeter like SMSC number before sending and receiving message.
PDU Mode: This mode is most flexible and is becoming common in smart phones. It supports different characters set that include Arabic, Chinese etc, compression, tunes and binary images. This mode also includes SMSC number and extra parameters inside the message payload therefore making SMS self-identifying.

D. Flaw

SMS was designed without any security consideration as compared to voice communication which shares secret keys at start. Security might not be a major requirement in past but today SMS carry confidential information e.g. email passwords reset codes, ATM Pin codes etc. with confidential data moving around on network.SMS can be read inside mobile network, monitored by organizations and even modified before delivery. To address this issue, many researchers have suggested security frameworks to establish confidential link for communication but still all of them are likely to have man in the middle attack. [10]

III. Security Issues

Security has three main pillars which are confidentiality, integrity and authenticity (CIA). These pillars are defined below [11].
Confidentiality: It is a process of hiding data so that it can be seen only by data generator and person for whom data is generated. Usually this hiding is done by encryption algorithms.
Integrity: Integrity ensures that data was not modified and is integrated until it reaches recipient, also providing tools to recipient to verify message integrity. Different algorithms to check data integrity are SHA series, MD5, CRC etc.
Authenticity: Authenticity is a process which guarantees that data was actually generated by message sender and prevents impersonation attack.
With no secure infrastructure, SMS faces following transport and storage threats [12].

A. Transport Layer Threats

Man in the Middle Attack: in this attack, eavesdropper intercepts communication between two parties and transparently observes communication. MITM attacks can be passive where no actual change is done be MITM but in active attack, MITM can modify message or even generate itself on someone else behalf.
Replay Attack: Attacker can use previously sent message and retransmit it to cause replay attack. This type of attack is dangerous especially in bank transactions.
Spamming Attack: Spamming Attack happens when companies advertise their product over SMS without recipient permission.
Denial of Service attack: DoS attacks have two types. One exist on sender side, where sender can send bulk SMS in very short period of time resulting in control channel congestion therefore disabling call reception and initiation in sender cell. Second DoS attack exists on recipient side where bulks of queued SMS are received exhausting mobile and control channel resources on recipient side. At night of New Year this phenomenon is observed globally where everyone sends New Year greetings to all contacts at same time.
SMS Spoofing: In this Attack, sender sends message on behalf of someone else mobile number. This happens by using international SMSC which allow sender to edit sender number, allowing sender to impersonate any number.

B. Storage Threats

Message Disclosure: SMS is stored as plaintext in mobile or SIM memory therefore it can be disclosed to anyone who can get hold. This can be avoided by encryption but encryp-

tion keys should not be stored at same place where SMS are stored.

IV. Existing Security Frameworks

Security threat can exist on either transport or storage layer as identifies, to address these security issues following solutions were proposed.

A. Proposed Transport Layer solution

GSM protocols secure voice communication using A5 algorithm which encrypts voice before transmitting over the network, Md. Asif Hossain and Shah Newaz proposed to use existing encryption algorithm for voice and upgrade GSM protocols to encrypt SMS with existing A5 encryption implementation. Although A5 algorithm has been compromised [13] but this concept was not implementable because it suggested upgradation of existing GSM infrastructure throughout the world.

B. Proposed Application Layer Solutions

Security is implemented in applications which perform security operations on SMS before is it transmitted over GSM network. Proposed frameworks for securing SMS on application layer are as follow.

- First sender and receiver application share key exchange SMS followed by encrypted SMS with shared key. This approach is discussed by Songyang Wu in High Security Communication Protocol for SMS [10].
- Both sender and receiver register with a third party to get public and private key for encrypting and decrypting SMS. This third party is known as Public Key Infrastructure. This approach is discussed by Alfredo De Santis et al. in An Extensible Framework for Efficient Secure SMS [14].

Although both approaches can guarantee CIA but critical part is how does framework propose key exchange. If key exchange channel is intercept-able then security cannot be guaranteed as MITM can easily and transparently integrate itself between communication and intercept all secure messages. Further explanation is given in following section.

V. Man In The Middle Attack

Man in the middle is a person who can intercept communication between two users and can manipulate information obtained [15]. Many solutions were proposed to address this problem among which verification of a third party is most commonly used. Third party gives guarantee for authentic users before communication starts.

A. Avoiding MITM Attack

Any secure communication start with exchange of secret key either by direct sharing or by distribution from third party and therefore is most critical part for securing a communication. Ideal solution to resist MITM attack is to exchange secret keys over a separate secure channel rather sharing them over same channel used for communication. In case of secure SMS although many solutions exist, but none can guarantee secure key exchange. This is because most use same GSM network (SMS) to exchange encryption keys or other use HTTP communication protocol. Underlying problem is that both channels are accessible to MITM and therefore can be intercepted.

VI. SSL and HTTS

A. Secure Socket Layer (SSL)

Secure socket layer can be defined as cryptographic protocol that can provide secure communication over internet. For authentication and key exchange it uses asymmetric, symmetric encryption for confidentiality and for integrity it relies on message authentication code.
SSL is used to provide secure communication between two users and is implemented over TCP layer. SSL consists of two protocols as illustrated in Fig. 3.
SSL record protocol is used to provide handshake, key exchange and algorithm alignment services to different top layer protocols such as HTTP to produce HTTPS service. SSL uses certificates to prove end-to-end authenticity by certification authority.

B. Hypertext Transfer Protocol Secure (HTTPS)

HTTPS is an application specific implementation of HTTP over SSL. It is used to secure communication between client and server over internet, with guarantee by a third party known as certification authority (CA). HTTPS provides safety against MITM attack by enabling two way encryption which discourages tampering, eavesdropping and information forgery. To enable HTTPS service for a website, hosting company needs to purchase certificate from certification authority for required website. Browser requests HTTP data over port 80, whereas HTTPS data is communicated over port 443. A simple workflow of HTTPS is illustrated in Fig. 4.
As HTTPS websites are usually accessed by browsers, browser first receives and verifies certificate, once verified it displays it to the client in different ways, usually changing color of the address bar. Firefox browser puts a lock icon at address bar when certificate is verified by the issuing authority as demonstrated in Fig. 5.

C. MITM Attack on SSL

Only SSL attacks existing till today are SSL stripping attack and SSL sniffing Attack [16].
SSL Sniffing Attack: In SSL sniffing attack man-in-the-middle sniffs identification packets of SSL communication, server replies SSL certificate to client which is intercepted by man-in-the-middle and a self-signed certificate is forwarded to client. Client browser intimates client about this forged certificate, but still many people were found ignoring this alert and continued their routine operation, without knowing the fact that their private information is visible to an eavesdropper. Fig. 6 illustrates attack execution step by step
SSL Stripping Attack: In SSL stripping attack, eavesdropper setup redirection service between client and server. When

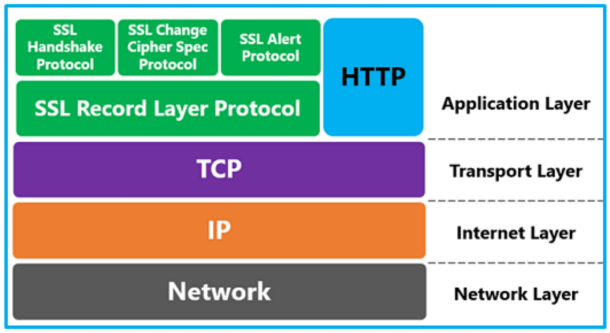

Figure. 3: SSL layers

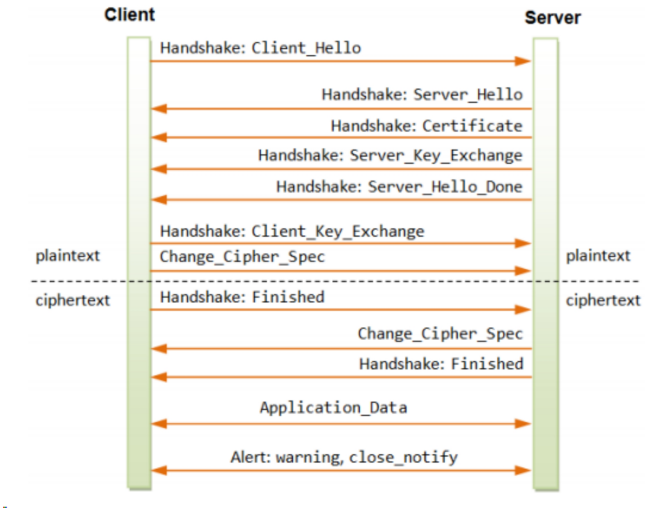

Figure. 4: Client Server Information Exchange

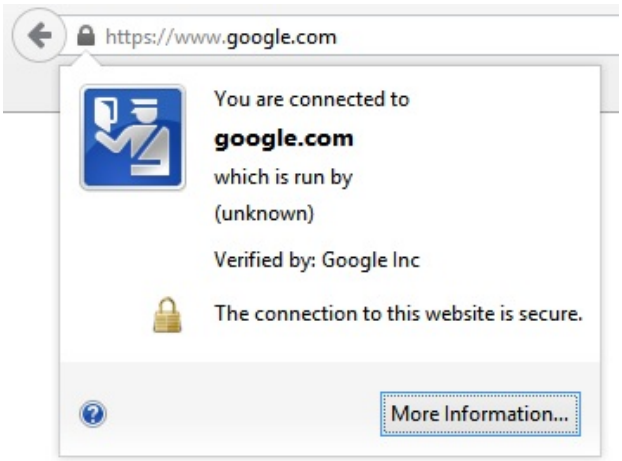

Figure. 5: Firefox Certificate Verification Exchange

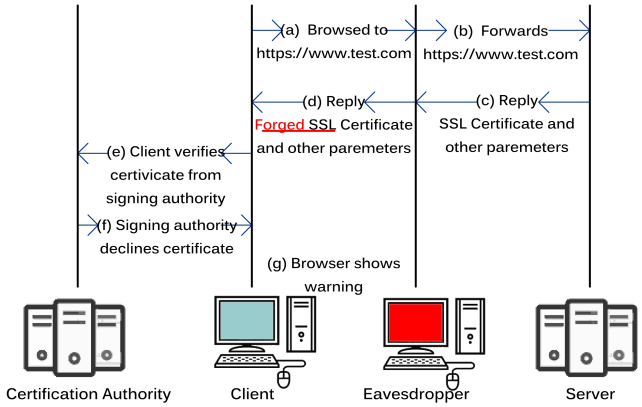

Figure. 6: SSL Sniffing Attack Verification Exchange

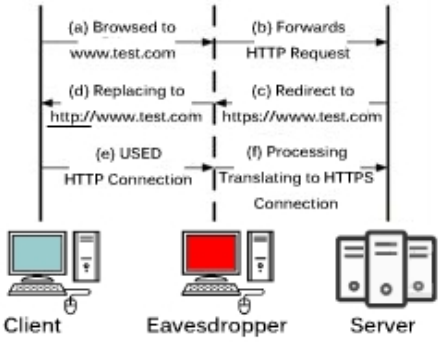

Figure. 7: SSL Stripping Attack Verification Exchange

client requests any website using simple HTTP request, server sends redirect command to clients browser. MITM intercept this request and redirects to HTTPS on its machine, whatever content from server is received over HTTPS, is converted to HTTP data and sent to client by MITM, client cannot detect this attack unless he is sure that he should receive a secure page. Work flow is shown in Fig. 7.

D. Solution

Both attacks can be easily prevented, SSL stripping attack can be prevented by accessing website directly using HTTPS prefix whereas SSL sniffing attack can be prevented by not accepting certificates that cannot be verified.

VII. Proposed Framework

Before we continue to describe proposed Framework, main components of design are needed to be explained.

Secure web server: Key management is done by a web application which maintains a public key directory over HTTPS. This web system is known as Key Distribution System (KDS).

Smart Phone: Smartphone is responsible of downloading, installing and registering application. Once application is installed and registered with KDS, user can send encrypted messages using smart phone application.

Communication Channel: Two different communication channels exist, first is internet connection, e.g. EDGE, 3G, LTE, WiFi etc. second connection is SS7 based connection. Both application users connect to Secure Webserver using

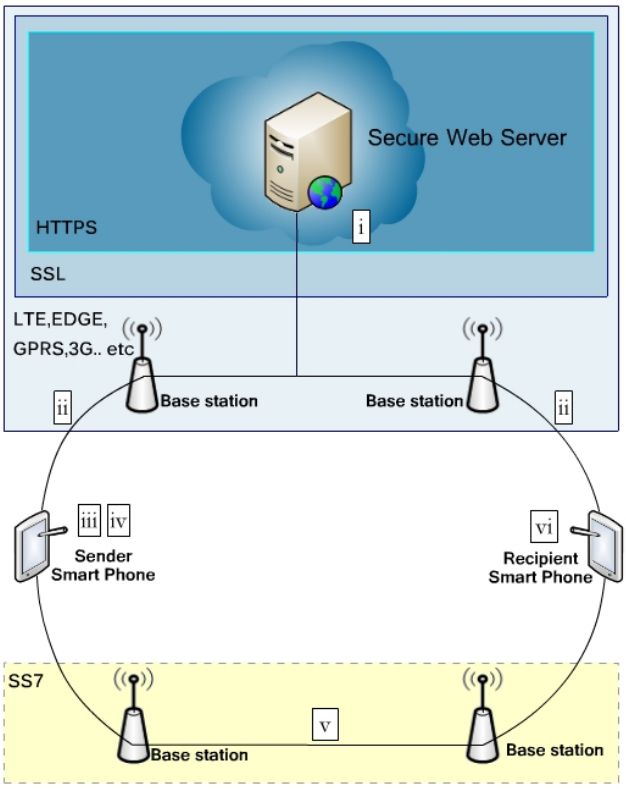

Figure. 8: Proposed Framework

HTTPS connection for key exchange and use SS7 channel to exchange encrypted SMS.
As a solution of current MITM problem in secure SMS communication, following framework was proposed. In Fig. 8 secure communication takes place between a web server and a smart phone application.
Following steps illustrate workflow of the proposed system.

i) User will download and install application into mobile.

ii) On launching application, application will connect to KDS over HTTPS and once connected, it will automatically register the application using user mobile number, RSA/ECC public key and expiry time.

iii) For sending encrypted SMS, sender will first download public key of recipient from PKDS.

iv) Sender will encrypt SMS with recipient public key and send over GSM network.

v) GSM network cannot read message content and also cannot perform MITM attack as key were never exchanged over GSM network.

vi) Recipient will decrypt message with private key.

Sharing keys over HTTPS can guarantee that MITM attack cannot happen because MITM can intercept HTTPS encrypted data but cannot decrypt it if HTTPS is properly configured between smart phone applications and secure web server.
This framework is limited to IOS, Windows, Android and J2ME based recent smart-phones because these operating systems allow developer to encrypt data and send as SMS. For identification of SMS by application, SMS structure

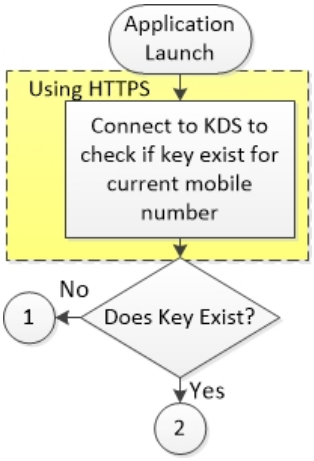

Figure. 9: Application Start

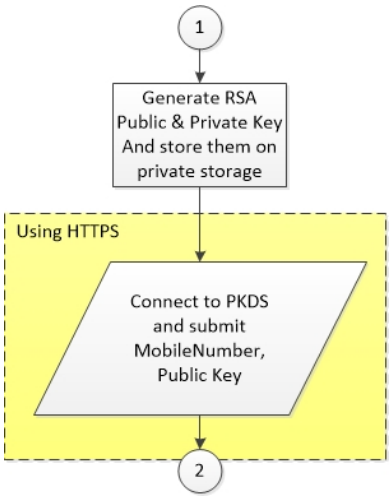

Figure. 10: Keys Generation and Registration

Table 1: ECC, RSA PUBLIC AND PRIVATE ATTRIBUTES

Algorithm	Public Attributes	Private Attributes
RSA	*e,n*	*d,n*
ECC	$E(a,b), P_{public}, Pt_{(x,y)}$	*d*

needs to be defined which should be composed of header and body. Header will be used to recognize encrypted messages whereas body of message will be encrypted with shared keys between users

VIII. ECC And RSA Integration

In this section public and private attributes of ECC and RSA are identified and framework flow of these attributes is illustrated. Public and private attributes of ECC and RSA are written in Table. 1.

For RSA, e is exponent used to encrypt data publically whereas n is modulus prime. Encryption equation will be $c = x^{e} \ mod \ n$ Where x is data to be encrypted and c is cipher text. d is decryption exponent and n is modulus prime therefore decryption equation is $x = c^{d} \ mod \ n$, whereas for ECC, $E(a, b)$ represents ECC equation used for encryption with parameters a,b. P_{public} is a public key generated by multiplying private random number d with $Pt_{(x,y)}$ which is point on elliptic curve. Encryption equation is $C_1 = r.Pt_{(x,y)}$, $C_2 = Plaintext + r.P_{public}$ Where r is a random number selected by sender for encryption, both C_1, C_2 are sent to recipient. On recipient side d is private random number which is used as to decrypt encrypted data. Decryption equation is $Plaintext = C_2 - (d.C_1)$. These equations need to be implemented inside mobile application so that mobile application can generate encryption keys, encrypt and decrypt SMS. Detailed operational flowchart of mobile application is illustrated in Fig. 9, Fig. 10 and Fig. 11. Generated asymmetric keys can be used to encrypt messages for long duration as certificate used in HTTPS communication.

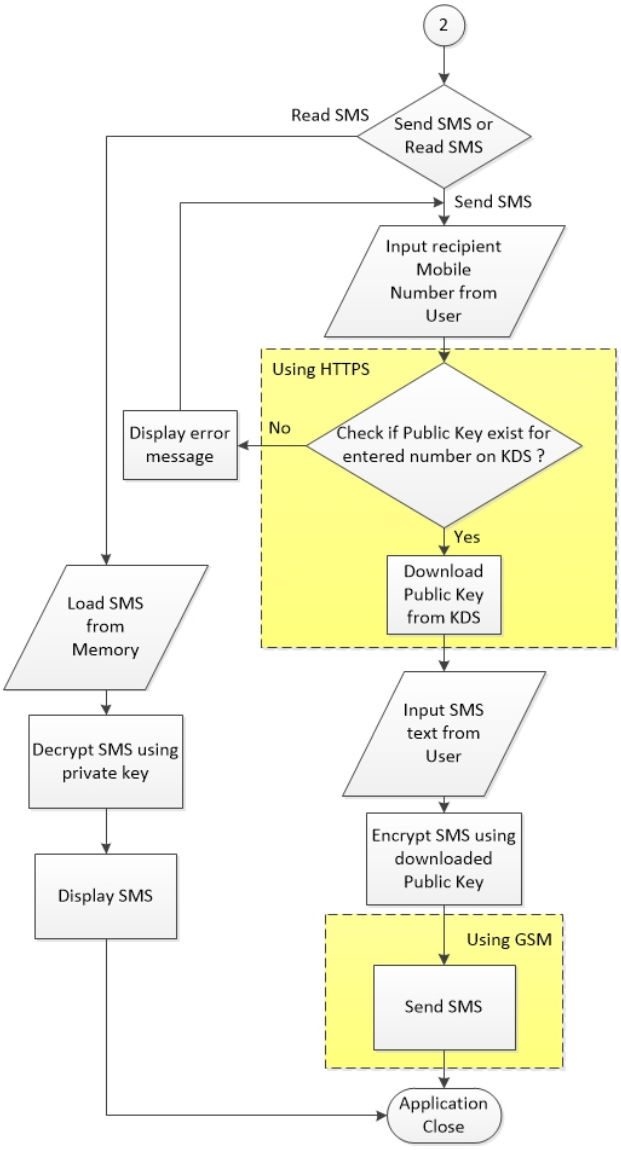

Figure. 11: Send and Read SMS

IX. Experiment

Based on proposed framework, an experimental setup was designed which consisted of three main components i.e. SMS structure, web based secure key distribution system and Android application. RSA encryption algorithm was used throughout experiment because it does not require extra effort to map plaintext into curve point coordinates which is required in ECC encryption. Also RSA encryption and decryption libraries are already present in Android operating system.

A. SMS Structure

A simple SMS structure was implemented to differentiate encrypted SMS from normal SMS. Structure consisted of a header to identify encrypted message i.e. ASCII ”01”, a random number from 0 to 9 to identify group of split messages and encrypted characters. Total SMS character length was 160 characters. Structure is presented in Fig. 12

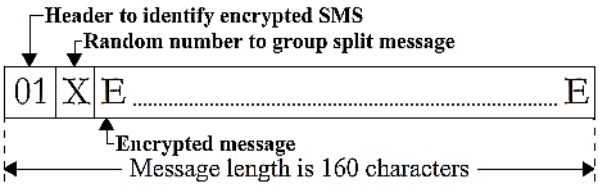

Figure. 12: SMS Structure

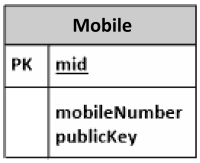

Figure. 13: KDS database entity

Table 2: PHP API functionality

Page/API call	HTML Return	Conditions
chcknumber.php ?number=[val]	does-not-exist	if number doesn't exist already
chcknumber.php ?number=[val]	Exists	if number already exists
regnumber.php ?number=[val] &key=[val]	registeration-done	if number and key are successfully registered to web application
regnumber.php ?key=[val] &key=[val]	number-exists	if number already exist new submitted key will not be registered
getpublickey.php ?requestednumber=[val]	[Base 64 encoded Public Key]	if number exist public key will be returned
getpublickey.php ?requestednumber=[val]	does-not-exist	if number does not exist

B. Secure key distribution system

Key distribution system consisted of a database to store public keys along with other essential attributes and PHP API pages were programmed to helped mobile application to read and write to database. Wampserver was used for implementation, as it provided Phpmyadmin database environment for MySQL and PHP as web scripting language. To make communication secure between webserver and mobile API requests, custom self signed SSL certificate was generated using openSSL and integrated into wampserver. Database table schema to store information is shown in Fig. 13 and Table. 2 highlights PHP APIs and their functionality.

C. Android application

Android application performs three main tasks which are KDS registration, send SMS and read SMS. KDS registration is required on both sender and receiver to exchange encrypted messages. One important thing in this framework is that application generates public and private key and submits public key to KDS rather than KDS generating key pair and distribute to applications. This approach is more resistant to attack as private key is never exchanged over network.

KDS registration: ANDROID suport RSA key generation, encryption and decryption. When application starts for the first time, it checks for public.key and private.key files in private resources. If these files doesn't exist already, application generates RSA 1024 bit key pair and store as public and private key file into private resources, Fig 9, 10 illustrate described process. Sample private and public key content are listed in Table 3 and Table 4 respectively.

Application checks if it is registered or not each time it starts by requesting following API call.

HTTPS://localhost/registernumber.php?number=[number]

Table 3: Resource public.key

File Attributes	values
Name	public.key
Content Type	Base 64 Encoded
Content	MIICdQIBADANBgkqhkiG9w0BAQEFAASCAl8wggJbAgEAAoGBALiHjAq37GPhSUZ4mT1J/8qGFeQAdvI1pifTjEhR4sPrm/7O9kFPSZO0Lqv/ojRO9Ffio9rtcepTBMDwEdHmbyNnExkY0EB4wDjs2QwCX5wU5rWOdo9oaK1lpneOoOcVe8t31nSHFe69vVZ4upIpT1HIxkMr0LnqYOC1WPScu76PAgMBAAECgYBAi8lEhok9drQ23kBb2+KJaNWHZ1Iz7ugrzientFzpwyRRwPsKZT2LWN9qcoS+X0/A3hoKnjvWhyuGgXL6ROxikO0UNNP/COB/WOM0sjIksuvo6MGpAyX/Kcw3gdztKsBXfi9Tp4cAXeOIXLJ3omqXC4SH9B0tiX3IADUVdIflgQJBAPFNGEtx6Xg4h4M44jUdl5dXGXy+g/sIjoYzRElq+oM7ij7SyiDaTITBtbCcL8Jk KsxBAtR5kLAMmAeoc6wJ/v8CQQDDxSRLBWeb/A3c2BJVsOYxWSjcCteDL2e7GJilg01flEC0AJAaFjOzYMdrCnVJi8RVPRJTFZFpX6htZe6V3dBxAkBNV5an5/UbUZMt3jezKq8CFSrYLwCCn/V+0kPa/G0bTyQ6irm62f2Yt7YSy50LQ6vsGBJeJDDaHM9F9RpOVE3lAkAjIdDWibzDOGOAe2Im8J0OxBJaErQEPdCeudZC9aJAAcnuw58Yn11d1oNTCUhtD2EHbvNH57n1sA34u+8qA+lxAkA+SPm1rZDhgDeor+WJI0gVqyyTQOavyYfAW5QFJc9orjrkDFdaYAN4cb14Gy1lUWDgnCg5FMl7fzZmm67qnDGQ

Table 4: Resource private.key

File Attributes	values
Name	private.key
Content Type	Base 64 Encoded
Content	MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC4h4wKt+xj4UlGeJk9Sf/KhhXkAHbyNaYn04xIUeLD65v+zvZBT0mTtC6r/6I0TvRX4qPa7XHqUwTA8BHR5m8jZxMZGNBAeMA47NkMAl+cFOa1jnaPaGitZaZ3jqDnFXvLd9Z0hxXuvb1WeLqSKU9RyMZDK9C56mDgtVj0nLu+jwIDAQAB

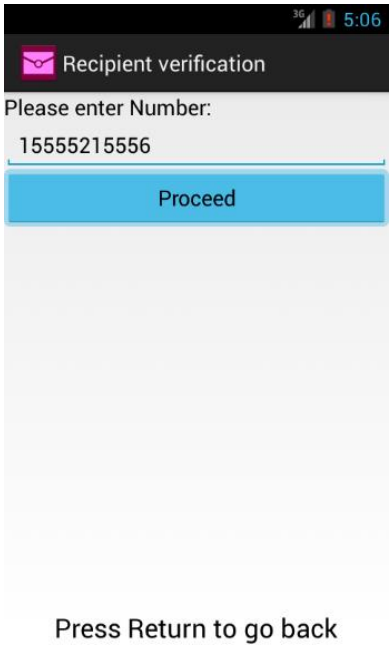

Figure. 14: KDS verification

&key=[publickey]
webservice responds to application accordingly. If Application is not already registered, it registers its public key along with mobile number. This communication is done over HTTPS and therefore its is not possible to for MITM to intercept public key.

Send SMS: In order to send encrypted SMS, two requirements should be fulfilled. First recipient mobile number should be known and second recipient should be registered on KDS. To begin with,sender should enter recipient number into application and click proceed as illustrated in Fig 14.

On pressing ”proceed” button, two tasks are executed, first it is checked if requested number is registered to KDS using checknumber.php page and if number exist public key of recipient is downloaded using getpublickey.php. If user does not exist on KDS, an error message is displayed to user. Once public key is downloaded from KDS, application moves to Send SMS screen where user can type message and send, see Fig 15. This SMS is encrpted with RSA public key before it is sent over GSM network.

Read SMS: On reciever side, group of messages are recieved as shown in Fig. 16. On observing these messages, it can be deduced that frist three character are similar for grouped messages, in case of given example it is 014. 01 is identification header for the application where as 4 is a random number to identify set of encrypted message. Following characters in both messages are encrypted data that can be decrypted by recipient private key. For decrypting message, application is launched and read SMS option is selected, Fig 11 displays flowchart of described operations.

Once application identifies encrypted messages starting with ”01” character sequence, it groups split messages based on grouping number which appear at 3rd character position of encrypted messages. Encrypted payload is extracted from same group messages and concatenated to form one message. Fig. 17 shows how application lists encrypted and normal messages. Once encrypted message is selected, encrypted

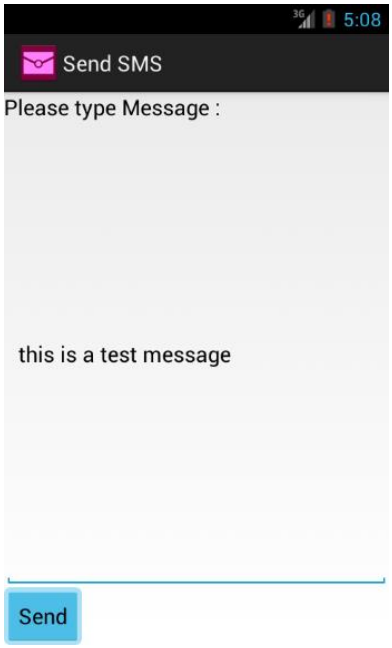

Figure. 15: Input SMS, encrypt and send

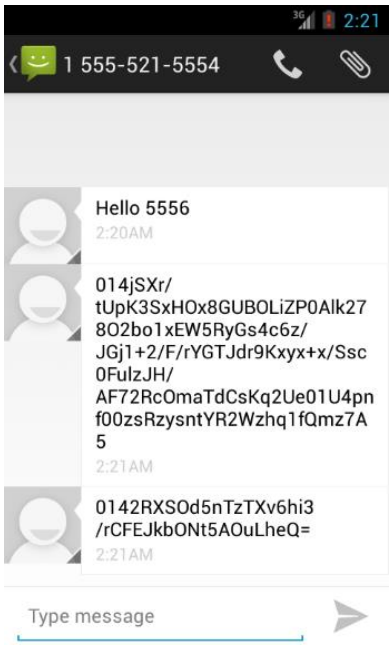

Figure. 16: Recipient SMS inbox

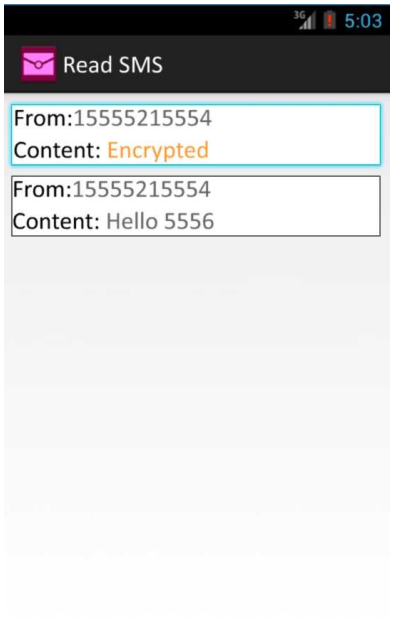

Figure. 17: Application listing received SMS

Figure. 18: Application decrypting SMS

text is decrypted using recipient private key and displayed to user as shown in Fig. 18.

X. Conclusion

In this paper we identified serious security issues related to SMS and proposed a new approach to safeguard SMS from MITM attack. Framework provided confidentiality and authenticity to SMS but this work can be extended to cover message integrity and replay resistance by introducing message digest and nonce inside message payload.

Acknowledgments

The authors thank the anonymous reviewers for their valuable comments and to Universiti Teknologi Malaysia (UTM) for the ERGS Grant Vote number 4L127 that is sponsored by Ministry of Higher Education (MOHE) and Research Management Centre, Universiti Teknologi Malaysia, Skudai, Johor.

References

[1] I. E. Consortium, *The Basics of Telecommunications (2002)*. International Engineering Consortium.

[2] M. W. Khan, "Sms security in mobile devices: A survey," *Int. J. Advanced Networking and Applications*, vol. 5, no. 02, pp. 1873–1882, 2013.

[3] M. M. Khan, "Https approach to resist mitm attack in secure sms," Master thesis, Universiti Teknologi Malaysia, Faculty of Computing, 2013.

[4] M. B. Muhammad Murad and S. Bakhtiari, "An https approach to resist man in the middle attack in secure sms using ecc and rsa," in *2013 International Conference on Intelligent Systems Design and Applications (ISDA 2013)*.

[5] G. Le Bodic, *Mobile Messaging technologies and services: SMS, EMS and MMS*. John Wiley & Sons, 2005.

[6] C. Peersman, S. Cvetkovic, P. Griffiths, and H. Spear, "The global system for mobile communications short message service," vol. 7, pp. 15–23.

[7] A. Medani, A. Gani, O. Zakaria, A. A. Zaidan, and B. B. Zaidan, "Review of mobile short message service security issues and techniques towards the solution," vol. 6, pp. 1147–1165.

[8] X. Wang and Y. Yang, "Method and implementation of sending and receiving mobile phone messages," in *International Forum on Computer Science-Technology and Applications, 2009. IFCSTA '09*, vol. 1, pp. 173–175.

[9] G. Peersman, P. Griffiths, H. Spear, S. Cvetkovic, and C. Smythe, "A tutorial overview of the short message service within gsm," *Computing & Control Engineering Journal*, vol. 11, no. 2, pp. 79–89, 2000.

[10] S. Wu and C. Tan, "High security communication protocol for SMS," in *International Conference on Multimedia Information Networking and Security, 2009. MINES '09*, vol. 2, pp. 53–56.

[11] W. Shin, J.-L. Lee, D.-H. Park, and C.-H. Chang, "Design of authenticity evaluation metric for android applications," in *Digital Information and Communication Technology and it's Applications (DICTAP), 2014 Fourth International Conference on*. IEEE, 2014, pp. 275–278.

[12] N. Saxena and N. S. Chaudhari, "Securesms: A secure sms protocol for vas and other applications," *Journal of Systems and Software*, vol. 90, pp. 138–150, 2014.

[13] A. Hossain, S. Jahan, M. Hussain, M. R. Amin, and S. Newaz, "A proposal for enhancing the security system of short message service in GSM," in *2nd International Conference on Anti-counterfeiting, Security and Identification, 2008. ASID 2008*, pp. 235–240.

[14] A. De Santis, A. Castiglione, G. Cattaneo, M. Cembalo, F. Petagna, and U. Petrillo, "An extensible framework for efficient secure SMS," in *2010 International Conference on Complex, Intelligent and Software Intensive Systems (CISIS)*, pp. 843–850.

[15] Y. Desmedt, "Man-in-the-middle attack," in *Encyclopedia of Cryptography and Security*. Springer, 2011, pp. 759–759.

[16] S. Puangpronpitag and N. Sriwiboon, "Simple and lightweight HTTPS enforcement to protect against SSL striping attack," in *2012 Fourth International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN)*, pp. 229–234.

Author Biographies

Majid Bakhtiari is Senior Lecturer in the Faculty of Computing, University Technology Malaysia (UTM). His research interests centered around the area of cryptography, cryptanalysis, Information Security and Cloud computing. His experiences in field of cryptography and cryptanalysis is more than 30 years. He has designed, educated and installed different data crypto-system from 1985 to 2006. Also he has experiences of crypto-algorithm breaking in field of voice encryption and stream cipher systems. His current research works concentrate on Cryptography, cryptanalysis, security in cloud computing, Steganography, Watermarking and etc. He is an expert in the fields of security system designing in big organizations. He holds a BSc in electronic, MSc security information, PhD in computer science (cryptography).

Saeid Bakhtiari is a PhD student at UTM University since autumn, 2012. He received his B.Sc. degree in Software Engineering from Shomal University in 2010, and his M.Sc degree in computer science from UTM University in 2012. His research interests include software development, development of elliptic curve cryptosystems, cryptography, network security, steganography, image processing and watermarking.