Enhancing the Integrity of Short Message Service (SMS) in New Generation Mobile Devices

Ramesh Yegireddi[1], Surya Pavan Kumar Gudla[2] Kiran Kumar Reddi[3] and Naresh Tangudu[4]

**[1] CSE Department, Aditya Institute of Technology and Management,
Tekkali, Srikakulam, Andhra Pradesh 532201, India**

**[2] CSE Department, Aditya Institute of Technology and Management,
Tekkali, Srikakulam, Andhra Pradesh 532201, India**

**[3] CS Department, Krishna University,
Machilipatnam, Krishna Dist,, Andhra Pradesh 521001, India**

**[4] MCA Department, Sree Vidyanikethan Institute of Management,
Titupathi, Chittur Dist,, Andhra Pradesh 517501, India**

Abstract

In the present world SMS is a cost effective communication. It has become a massive money-making industry as most of the active mobile user's activity is messaging. SMS is treated as digital evidence in court of law at the time of trial. Most of the mobile forensic examiners are asked whether the SMS transmitted through mobile devices is fabricated or not. Messaging activity of active users on android mobile device is increasing but they are liable to do illegitimate activities. A technically sound user can modify an SMS in android mobile phone and profess it as genuine.

In the first part of this work we ascertain the truth that it is feasible to falsify an SMS in android mobile phone. We breakdown the integrity of new generation mobile devices like android based phones by modifying the SMS (changing timestamps, text and finally sender identity). We need to defy this damage. The elucidation should be adequate to manufacturers, service providers, users and mobile forensic examiners. In the second part of our work we discuss mechanism and its implementation for preventing such attempts.

Keywords: *Integrity, debugging, collision resistant, hashing, and whirlpool.*

1. Introduction

Now-a-days Mobile phones become an integral part of human life. Though these handheld devices are pocket-sized they contain call history, text messages, calendar items, e-mails, digital photographs, videos, memos, address books, passwords, and credit card numbers etc. The speedy growth of mobile devices produces openings for illegitimate activities. The information on mobile devices provide assistance to address the vital interrogations in an investigation, disclosing whom the specific individual has been in contact with, what they have been communicating about, and where they have been.

In a paper authored by Mr.Al-Zarouni [2] discusses about flasher boxes and their uses in the mobile phone forensics. He mentions how flasher boxes are used for mobile phone servicing and illegal use of flasher boxes. He discusses flasher box components and varieties and how flasher boxes are connected to mobile phone and computer. In their papers Mr. Kevin Jonkers, Someshekar Akkaladevi and others [3,5] discusses forensic use of flasher boxes . They explain three different kinds of data acquisitions namely 1) physical acquisition 2) logical acquisition and 3) flasher box acquisition. They mainly focused on flasher box testing and validation. They explained test cases for flasher box validation and test cases for data integrity validation in mobile phones. Some research work has been done on mobile message verification by Mr. Rakesh Verma, Deepak Singh Tomar and Shashi Kanth Rathore [4].

A paper on Falsifying SMS messages by Mr. Thomas Marryat and John Corcoran [1] is our base paper. In their paper they explained how they falsified an SMS in Nokia 6021 mobile phone using UFS3 flasher box. They modified the visible parameters of SMS successfully but their attempts to inject an entirely new SMS text into the handset memory were unsuccessful.

Our paper is an extension to Mr. Thomas Marryat and John Corcoran as we start our research on breaking the integrity of SMS where they [1] stopped and failed. In this paper we break the integrity of SAMSUNG i9300 (Galaxy S3) android mobile device by modify the SMS visible parameters and even text without using any flasher box.

The Organization of paper as in the section-3 choosing right cable and USB Debugging on mobile device, section-4 Altering SMS visible parameters and we discuss about modifying SMS text in SAMSUNG i9300 (Galaxy S3). Finally we discuss the counter mechanism for these illegitimate activities.

2. Breaking Integrity

2.1 Choosing Right Cable Connector

First we should connect the mobile to computer with help of suitable connector. The connector depends on type of mobile. As the first step to our experiment we connect the SAMSUNG i9300 (Galaxy S3) to the computer. Fig. 1 shows the connection between mobile device and computer.

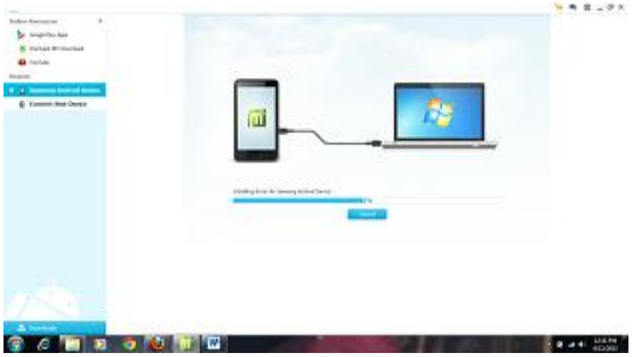
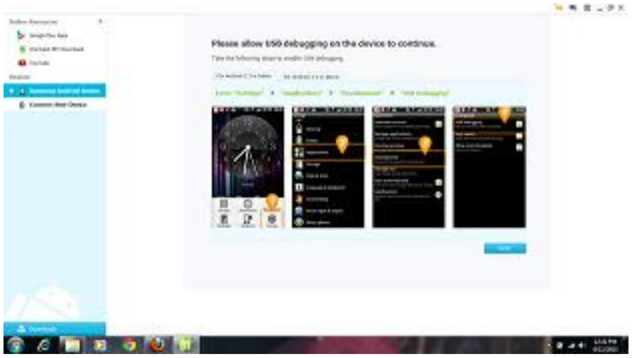


Fig. 1 Connecting Mobile Device to Computer

2.2 Enable USB Debugging in Mobile Device

To import the contents of the mobile to a PC, the PC must be installed with suitable software which enables the PC to be connected with the mobile. After connecting the targeted android mobile device (Samsung i9300 galaxy s3) to a PC, we need to dig out contents from the android mobile device. To dig out the content from Mobile device, we should enable USB debugging option in corresponding mobile. Fig. 2 shows series of steps user has to go through to enable USB debugging option in SAMSUNG i9300 (Galaxy S3).



Please allow USB debugging on the device to continue.

Fig 2 Eenabling USB debugging option in SAMSUNG i9300

Table 1 shows the USB Debugging option in different android mobiles like Micromax, Sony Xperia and SAMSUNG.

Table 1 USB debugging options in Different Mobiles.

Mobile Type	Path
Micromax	Settings → Developer Options → USB Debugging
Sony Xperia, SAMSUNG i9300 (Galaxy S3)	Settings → Application → development → USB Debugging

Table 2 shows the path to enable the USB debugging in different android version like 2.0, 3.0, 4.0 and 4.2

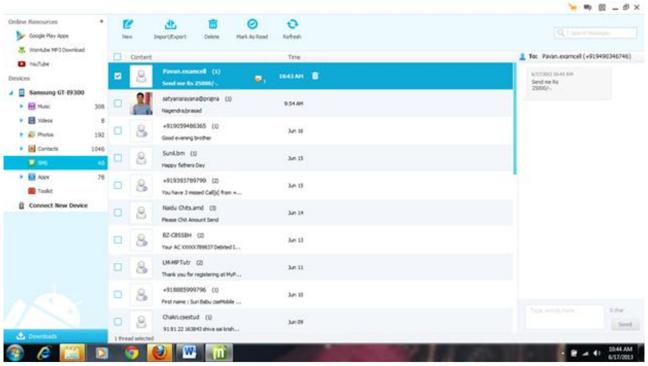
Table 2 USB debugging options in different android versions

Version	Path
Android 2.0	Settings → Applications → Development → USB Debugging
Android 3.0	Settings → Applications → Debugging → USB debugging
Android 4.0	Menu → Settings → Developer options → USB Debugging
Android 4.2	Settings → About Phone → Build Number → Tap on it 7 Times → Developer option → USB debugging

3 Modify SMS visible parameters

Once connection is established all the contents (call history, text messages, e-mails, digital photographs, videos, calendar items, memos, address books, passwords, and credit card numbers etc) of mobile are extracted to the computer. Fig 3 and Fig 4 shows the contents of SAMSUNG i9300 (Galaxy S3), extracted to the computer.

Fig 3 Contents of SAMSUNG i9300 (Galaxy S3)



Pavan.examcell (1) Send me Rs 25000/-. 10:43 AM
satyanarayana@pragna (1) Nagendraprasad 9:34 AM
+919059486365 (1) Good evening brother Jun 16
Sunil.bm (1) Happy fathers Day Jun 15
+919393789799 (2) You have 3 missed Call(s) from +... Jun 15
Naidu Chits.amd (3) Please Chit Amount Send Jun 14
BZ-CBSSBH (2) Your AC XXXXX789837 Debited I... Jun 13
LM-MPTutr (2) Thank you for registering at MyP... Jun 11
+918885999796 (1) First name : Sun Babu cseMobile ... Jun 10
Chakri.csestud (1) 91 81 22 163843 shiva sai krish... Jun 09
1 thread selected

To: Pavan.examcell (+919490346746)
6/17/2013 10:43 AM
Send me Rs 25000/-.

10:44 AM 6/17/2013

Fig 4 Inbox of SAMSUNG i9300 (Galaxy S3)

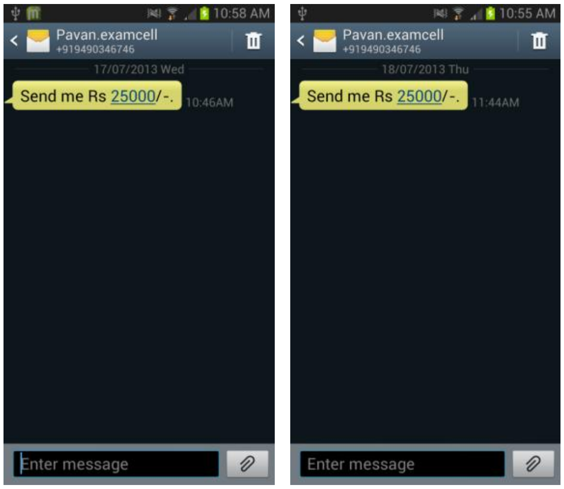
Most android based mobiles store messages in SQLite database. These messages are transmitted and stored in xml format. First, extract wanted SMS then save it on computer. This SMS will be saved in .xml format. Now open the file, change the required visible parameters of SMS then save it. Now delete original SMS in mobile then export modified xml file into mobile. Figure 4.3 shows the SMS before modification and after modification in circles which is part of xml format.

Code Before modification of timestamp

```
<sms>
 <Id>0</Id>
 <Numbers>+919490346746</Numbers>
<Body>Send me Rs 25000/-.</Body>
 <SmsType>0</SmsType>
<Time>2013-07-17T10:46:35.608+05:30</Time>
<ThreadId>679</ThreadId>
 <Status>2</Status>
 <ChatType>0</ChatType>
</sms>
```

Code after Modification of timestamp

```
<sms>
 <Id>0</Id>
 <Numbers>+919490346746</Numbers>
 <Body>Send me Rs 25000/-.</Body>
 <SmsType>0</SmsType>
 <Time>2013-07-17T11:44:55.608+05:30</Time>
 <ThreadId>679</ThreadId>
 <Status>2</Status>
 <ChatType>0</ChatType>
</sms>
```



Send me Rs 25000/-. 10:46AM

Send me Rs 25000/-. 11:44AM

Fig 5 SMS Time Stamp modification before and after

3.1 Altering SMS text

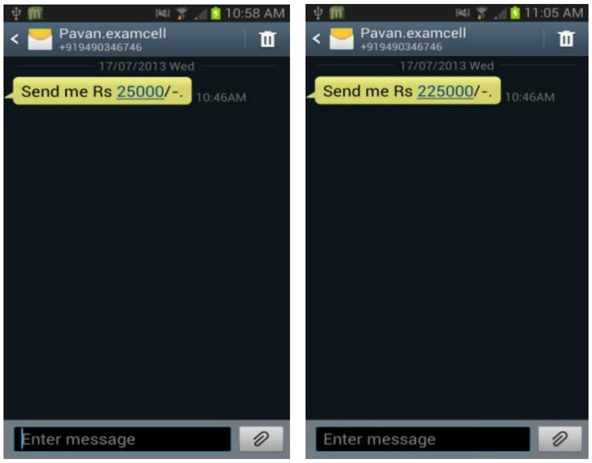
The process of altering SMS text is done as we did in altering SMS visible parameters. This SMS shall be saved in .xml format. Now open the file, change the required text of SMS then save it. Now delete original SMS in mobile and export the modified xml file into mobile.

Code Before modification of text

```
 <Sms>
 <Id>0</Id>
 <Numbers>+919490346746</Numbers>
 <Body>Send me Rs 25000/-.</Body>
 <SmsType>0</SmsType>
 <Time>2013-07-17T10:46:35.608+05:30</Time>
 <ThreadId>679</ThreadId>
 <Status>2</Status>
 <ChatType>0</ChatType>
 </Sms>
```

Code after modification of text

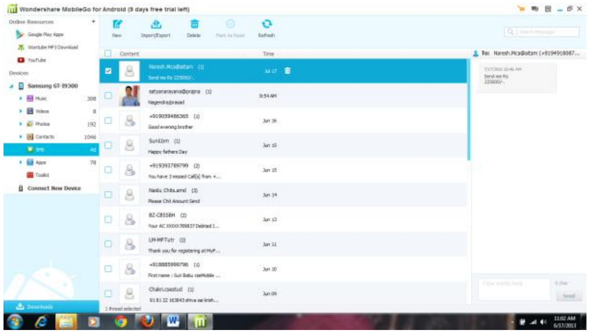
```
<Sms>
 <Id>0</Id>
<Numbers>+919490346746</Numbers>
 <Body>Send me Rs 225000/-.</Body>
 <SmsType>0</SmsType>
 <Time>2013-07-17T10:46:35.608+05:30</Time>
  <ThreadId>679</ThreadId>
 <Status>2</Status>
 <ChatType>0</ChatType>
 </Sms>
```


Pavan.examcell
+919490346746
17/07/2013 Wed
Send me Rs 25000/-. 10:46AM
Enter message

Pavan.examcell
+919490346746
17/07/2013 Wed
Send me Rs 225000/-. 10:46AM
Enter message

Fig 6 SMS Text modification before and after

Fig 7 shows the modification of SMS text and sender number in computer.


Wondershare MobileGo for Android (9 days free trial left)

Content	Time
Naresh.Mca@atam (1) Send me Rs 225000/-.	Jul 17
satyanarayana@prajna (1) Nagendraprasad	9:54 AM
+919059486365 (1) Good evening brother	Jun 16
Sunil.bm (1) Happy fathers Day	Jun 15
+919393789799 (2) You have 3 missed Call(s) from +...	Jun 15
Naidu Chits.amd (3) Please Chit Amount Send	Jun 14
BZ-CBSSBH (2) Your AC XXXXX789837 Debited I...	Jun 13
LM-MPTutr (1) Thank you for registering at MyP...	Jun 11
+918885999796 (1) First name : Suri Babu cseMobile ...	Jun 10
Chakri.csestud (1) 91 81 22 163843 shiva sai krish...	Jun 09

To: Naresh.Mca@atam (+9194918087...
7/17/2013 10:46 AM
Send me Rs 225000/-.

11:02 AM
6/17/2013

Fig 7 Modifications of SMS Text and Sender Number

4. Method for the SMS not being fabricated

4.1 Whirlpool Algorithm

To provide integrity for SMS in Android Mobile, we have different kinds of mechanisms such as “Encoding of Message” “Linked Hashing” and “Identity Based Encryption” [12]. We need a mechanism to ensure data integrity as a whole in mobile phone, not just during forensic examination. In my opinion, to provide integrity for an SMS Hashing method is preferable. In the section of this paper we propose a 512 bit collision resistance hash function Whirlpool [6, 7] through which simply SMS can then be hashed.

4.2 Whirlpool hash Function

Whirlpool is an iterated cryptographic hash function, based on the Miyaguchi-Preneel scheme [8, 9, 10, 11], that uses a symmetric-key block cipher in place of the compression function. Whirlpool is a collision-resistant 512-bit hash function designed by Baretto and Rijmen and operates on messages less than 2^{256} bits in length. Whirlpool is based on an underlying dedicated block cipher W, with the block length of 512-bit. The block cipher W employs a SPN type round function. It is claimed that the diffusion layer of W uses an 8*8 matrix with the branch number $\beta = 9$. Due to the Square pattern propagation theorem, it is guaranteed that the number of S-boxes with a different input value in four consecutive rounds is at least $\beta^2 = 81$.

By combining S-boxes with the maximum differential probability equal to 2^{-5}, it was shown that no differential characteristic over four rounds of W has probability larger than $(2^{-5})^{81} = 2^{-405}$. Accordingly, it is estimated that the previous is enough to prevent differential attack on the full hash function.

However, we found that if certain type of vector is input to the matrix, the sum of hamming weight of input and output vectors is 8, implying that $\beta < 9$. On the other hand, we have found that actually $\beta = 8$. Accordingly, the number of S-boxes with a different input value in four consecutive rounds is estimated to be at least $\beta^2 = 64$, which is lower than the estimation. This implies a gap between potential security strength and previously estimated security strength of Whirlpool. The characteristics’ of Whirlpool Hash function is listed in the table 3.

Table 3 Characteristics of Whirlpool

Characteristic	Value
Block Size	512 Bits
Derived from	Square, AES
Digest Size	512 Bits
Structure	Miyaguchi-Preneel
Number of Rounds	10
Key Expansion	Using the cipher itself with round constants as round keys

Substitution	Sub-Bytes Transform
Permutation	Shift-Column transformation
Mixing	Mixing Rows Transformation
Round Constant	cubic roots of first eighty prime numbers

4.3 Whirlpool Block Cipher

The block cipher W is

- with security and efficiency of AES
- But with 512-bit block size and hence hash.
  - similar structure & functions as AES but input is mapped row wise
  - has 10 rounds
  - a different primitive polynomial for GF(2^8)
  - uses different S-box design & values

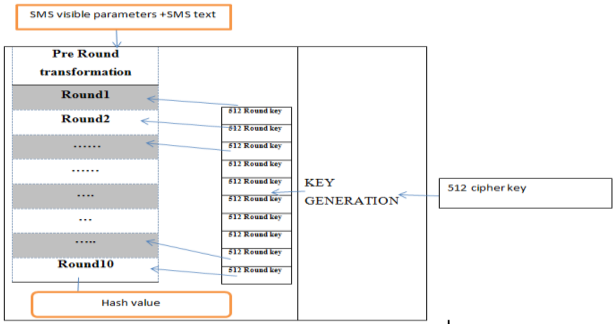


Fig 8 General idea of Whirlpool Cipher

4.4 Whirlpool Encryption and Pre Round Transformations

The SMS (Visible parameters + SMS text) is padded with bits to make it as odd multiple of 256 bits. Padding is always added, even if the message is already of the desired length. For example, if the message is 256 * 5 = 1280 bits long, it is padded by 512 bits to a length of 256 * 7 = 1792 bits. Thus, the number of padding bits is in the range of 1 to 512.The padding consists of a single 1-bit followed by the necessary number of 0-bits. i.e. ; 1000…….00000000.

A block of 256 bits is appended to the message to make SMS message as the sequence of 512-bit blocks M1, M2, …, Mx, so that the total length of the expanded message is *X** 512 bits. This appended bit string is unsigned integer tells about number of bits in original message.

Initialize the hash matrix 8*8 with zeroes which used to store intermediate hash values of whirlpool hash function which is denoted as H0. Then process SMS message 64 byte 8*8 matrix using whirlpool block cipher.Given SMS 512 blocks M1, M2, M3….Mx processed in using whirlpool logic. H0 is initial hash matrix, Intermediate hash values calculated as follows

$$H_i \;\; = E(H_{i-1}, M_i)\; Xor\; H_{i-1}\; Xor\; M_I = Intermediate\; Hash$$

Hx is hash code value where x is the number of 512 blocks.

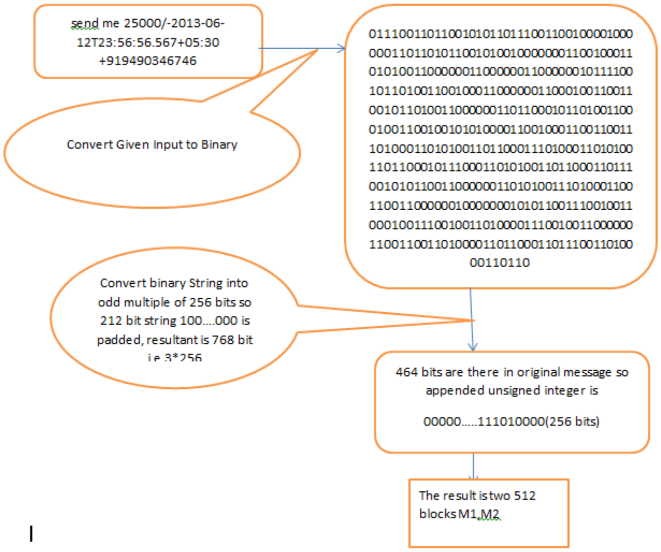


Fig 9 Dividing the message in to Blocks

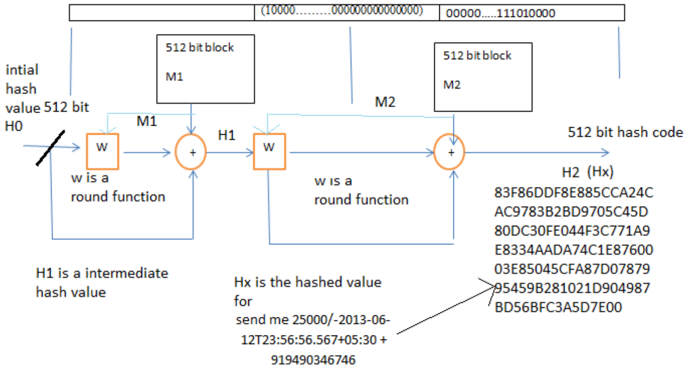


Fig 10 Generating the Hashed Message Digest

4.5 Enhancing Integrity

There are some possibilities to enhance the integrity of SMS in Android Mobile. We propose two possibilities in this regard which require a lot of work to be done on these two alternatives. According to our proposal, (1) to enhance SMS integrity, we need Whirlpool algorithm. When the mobile is equipped with Whirlpool algorithm, any modified SMS can't be restored in the inbox of the mobile. This Whirlpool Algorithm can be applied to all contents of the mobile or any specific content of the mobile. As we confine to SMS fabrication and restoration, we propose to apply the Whirlpool Algorithm specifically to the contents of inbox in the android mobile. Immediately after selecting the USB debugging option, a unique Hash Value for every inbox SMS in the android mobile is generated. Those Hash values must be stored in permanent memory of the Android Mobile. As we all know that these Hash values cannot be changed and these values will be in the permanent memory until the USB Debugging is disabled. We propose that the Hash value of the original SMS should not go along with the copy of the targeted SMS, when it is exported in Xml format to the PC.

While USB debugging is enabled, the inbox only accepts the incoming contents like calls, SMS, MMS etc. from other mobiles as they are sent with Encoding techniques and through MAP Protocol [13]. On other hand, the fabricated SMS is sent through Data transfer Protocols, and it has a new Hash value which is generated while it is being imported to the inbox. The Hash value of the fabricated SMS must be stored in other than permanent memory of the Android mobile. Now the new Hash value must be compared to the Hash values which are in the permanent memory of the Android mobile. When the Hash value of the fabricated SMS in not matched with any one of the Hash values, the inbox does not accept it. In this process, we may prevent the SMS from being fabricated.

We wish to propose another alternative which is very simple. According to our proposal (2), the exported SMS which is usually sent through Xml format to the PC. The targeted SMS is opened and fabricated. To make the targeted SMS unchangeable, the (exported) Xml file must be made as only readable when it is being exported onto the PC.

5. Conclusion

We proposed a novel and efficient way of storing SMS in Android Mobile devices such that if any SMS is taken as an evidence in the court of law. Whirlpool is not patented. It may be used free of charge for any purpose .With the help of a collision resistant 512 - bit Whirlpool Algorithm we can ensure the integrity of SMS and its parameters and we can also give assurance of integrity in case of files, because whirlpool Algorithm gives us complicated recursive hash function which offers more security i.e.; yet a little alteration in the evidence will (with an extremely high likelihood of 1-10^{-154}) outcome in a different hash.

References

[1] Thomas marryat and Corcocan John. Falsifying SMS messages.Small Scale digital Device Forensics Journal,4(1),Septemebr 2010.

[2] Marwan Al-Zarouni.Introduction to mobile phone flasher devices and considerations for their use in mobile phone forensics.Australian Digital Forensiocs conference,decemebr 2007

[3] Kevin Jonkers.The forensic use of mobile flasher boxes Digital investigation 6,2010

[4] Rakesh Verma, Deepak Singh Tomar, and Shashi Kant Rathore. Extraction
and verification of mobile message integrity. 2011.

[5] Somasheker Akkaladevi, Himabindu Keesara, and Xin Luo. Efficient forensic tools for handheld devices: A comprehensive perspective. pages 349-359.

[6] Cryptography and Network Security, Fifth Edition,William Stallings Prentice Hall 2010,ISBN-10: 0136097049

[7] P.S.L.M. Barreto and V. Rijmen, \The Whirlpool hashing function," Primitive submitted to NESSIE, https://www.cosic.esat.kuleuven.ac.be/nessie/tweaks.html, Sept. 2000

[8] Wang X, Feng D, Lai X, Yu H (2004) Collisions for hash functions: MD4, MD5, HAVAL-128 and RIPEMD. http://eprint.iacr.org/2004/199.pdf. Accessed August 2004

[9] Wang X, Yu H, Yin YL (2005) Efficient collision search attacks on SHA-0. In: Advances in
Cryptology – CRYPTO'05, vol 3621, pp 1–16

[10] Preneel B, Govaerts R, Vandewalle J (1989) Cryptographically secure hash functions: an
overview. In: ESAT Internal Report, K. U. Leuven

[11] Miyaguchi S, Iwata M, Ohta K (1989) New 128-bit hash function. In: Proceedings 4th
International Joint Workshop on Computer Communications, pp 279–288

[12] Mr. Jogu Amarendar and MR. Vinod Pathari. Ensuring Message Integrity in mobile Phones.2012.

[13] Signaling System No. 7 (SS7/C7): Protocol, Architecture, and Services by Lee Dryburgh; Jeff Hewett.

Mr. YEGIREDDI RAMESH is MCA (Computer Applications) from Osmania University and M.Tech(CSE) from JNTUH Hyderabad, Andhra Pradesh, India. He is working as Associate Professor in Computer Science & Engineering department in Aditya Institute of Technology and Management, Tekkali, Andhra Pradesh, India. He has 13 years of experience in teaching Computer Science and Engineering related subjects. He is a research scholar in JNTU Kakinada and his area of interest and research include Computer Networks, Wireless LANs & Ad-Hoc Networks, Mobile Computing and Cloud Computing. He has published several Research papers in national and international journals/conferences. He has guided more than 80 students of Bachelor degree, 25 Students of Master degree in Computer Science and Engineering in their major projects. He is a life member of ISTE and CSI.

Surya Pavan Kumar Gudla is an M .Tech Scholar (CSE) in Aditya Institute of Technology and Management, Tekkali, Andhra Pradesh, India. Completed MCA (Computer Applications) from Aditya Institute of Technology and Management, Tekkali, Andhra Pradesh, India.

Dr. Kiran Kumar Reddi is MCA (Computer Applications) from Andhra University and M.Tech(CSE) from JNTUK Kakinada, Andhra Pradesh, India and Ph D from Nagarjuna University, Guntur, Andhra Pradesh, India. He is working as Assistant Professor in Computer Science department in Krishna University, Machilipatnam, Andhra Pradesh, India. He has 15 years of experience in teaching Computer Science and Engineering related subjects. His area of interest and research include Bio-Informatics, Computer Networks, Databases, Mobile Computing and Cloud Computing. He has published several Research papers in national and international journals/conferences. He has guided more than 100 students of MCA, 15 Students of Master degree in Computer Science and Engineering in their major projects. He is a life member of ISTE and CSI.

T.NARESH has 4 years and 6 months of teaching experience in reputed colleges. Presently he working as Assistant Professor in Department of MCA, Sree Vidyanikethan Institute of Management. He completed M.Tech in 2012 at Pydah College of Engineering and completed MCA in 2009 at Aditya Institute of technology and Management. His area of Interests are Computer organization, operating system, Unix programming, and programming languages (C,C++,java), DBMS, Multimedia application development and Software Engineering .Currently he was published 4 papers in areas like text mining and network security in issues of IJERA-2012, IJCTT-2013 . He was member of IAENG.