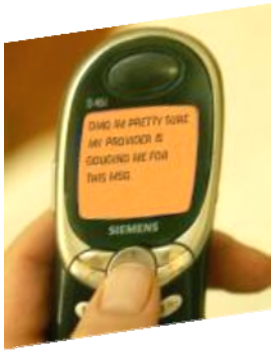
Exposing the CCN (Criminal Cellular Network): SMS Vulnerabilities to Embed High Capacity Covert Channels

M. Zubair Rafique
Research Engineer
Muddassar Farooq
Director nexGIN RC



It is not the only message you have received!

Abstract

For security organizations whose responsibility it is to pledge security, any mode of communication without inspection between entities to evade a 'security evaluation criteria' highlights a serious risk. Covert Channels constitute an important security threat since they are used to ex-filtrate sensitive information, to disseminate malicious code and more alarmingly to transfer the criminal (or terrorist) instructions ["Bin Ladens Messages Could Be Hiding In Plain Sight"].

This work presents '0' day vulnerabilities and weaknesses, that we discovered, in Short Message Service (SMS) protocol – the most used service of Cellular networks – that allow embedding of high capacity covert channels. We show that an intruder, by exploiting SMS vulnerabilities, can bypass existing security infrastructure (including firewalls, intrusion detection systems, content filters) of a sensitive organization and primitive content filtering software at an SMS Center (SMSC). We've found that the SMS in itself and along with its value added services (like picture SMS, ring tone SMS) appears to be much more susceptible to security vulnerabilities as compared to the other services in IP based networks.

To demonstrate the effectiveness of covert channels in SMS, we have developed a new tool – GeheimSMS – that embeds data bytes (not only secret, but also hidden) by composing the SMS in Protocol Description Unit (PDU) mode and transmits it from a mobile device using serial or Bluetooth link. The contents of overt (benign) message are not corrupted; hence the secret communication remains unsuspicious in transmission and reception of SMS. Our experiments on active cellular networks show that 1 KB of a secret message can be transmitted in less than 3 minutes by sending 26 SMS messages without raising alarm for a suspicious activity.

By illustrating these vulnerabilities and loopholes in SMS, we will recommend methodologies that will help sensitive organizations in particular and network operators in general in revamping their confidentiality, privacy, and security infrastructure. Moreover, we believe that our work will force the security community to issue a Request For Comments (RFC) for SMS security before criminals (or terrorists) start exploiting it (if they are not already doing it).

Introduction

Most organizations/enterprises deploy a "trusted communication model" [2] that guarantees security and privacy of confidential data. A trusted communication model is usually based on three requirements: (1) Only legitimate users are allowed access to sensitive data based on their security clearance level, (2) transfer of sensitive information to unauthorized users is restricted by any mode of communication, and (3) an information exchange is only possible through allowed procedures. Recently, intruders and terrorists have started using covert channels [1] in well known protocols to secretly communicate; therefore, they easily bypass requirements (2) and (3). The detection of covert channels1 (even in conventional systems) is a challenge; as a result, they are becoming serious threat in network communications. But a group of security researchers refute these claims by sticking to the opinion that covert channels are not a major security threat in conventional networking computing systems because of their low capacity [4].

In this paper, we show that the above-mentioned trusted communications model and thesis about covert channels have become irrelevant in the scenario of ubiquitous availability of mobile networks. Remember SMS has become the mostly widely used GSM service and this claim is substantiated by a recent report that more than 5.5 trillion text messages were sent over carrier networks worldwide in the year 2009 [5]. The trend appears to be increasing as a survey projects that 6.6 trillion SMS messages would be exchanged globally during the year 2010 [5].

The major contribution of our work is to demonstrate that intruders can exploit vulnerabilities2 in SMS protocol: (1) to secretly communicate or transfer sensitive data (from inside or outside an organization) by embedding high capacity covert channels in legitimate (overt) SMS, (2) to bypass existing IP based security infrastructure (including firewalls, Intrusion Detection System (IDS), content filters) of an enterprize; as a result, disgruntled employees can covertly leak sensitive and secret information of an organization, (3) to embed high capacity covert channels using different SMS formats; as a consequence, 1 KB of file can be transferred in less than 3 minutes, and (4) to evade detection in real-time because of large SMS traffic on network, which makes it a challenge to deploy effective and efficient covert channel detection tools at SMSC. To conclude, our pilot studies show that we need to quickly fix the vulnerabilities in SMS protocol; before this important service of GSM network can be exploited by intruders (or terrorists).

SMS Technical Overview

Usually the SMS received by the Short Message Service Center (SMSC) on the mobile phone is handled through (Global System for Mobile) GSM modem. (Figure 1 shows the logical architecture of mobile phones.) The GSM modem is controlled through standardized AT (AT commands are the de facto standard language for controlling the modems.) commands. Moreover, the modem also provides the interface with GSM network and the application processor of a mobile phone. The SMS received from the modem is delivered to the operating system of the mobile phone through the telephony stack, which has a multiplexing layer that allows multiple applications to access the modem at the same time. Furthermore, telephony stack decodes the (Application Program Interface) API-calls into corresponding AT commands and AT result codes to different category messages.

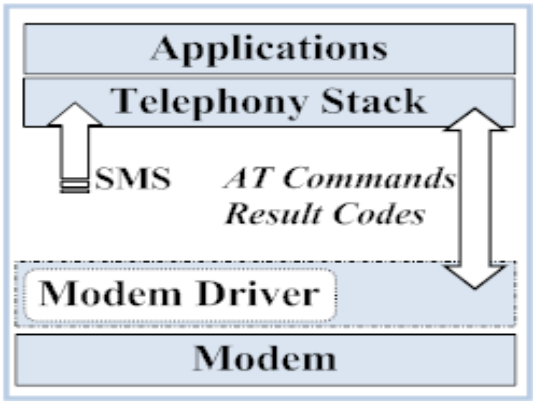


Figure 1: Logical Architecture of Mobile Phone

Through AT commands an SMS can be sent and received in two modes: (1) (Protocol Description Unit) PDU, and (2) text. Most well known services – Wireless Application Protocol (WAP), voice mail notifications, information retrieval, Multimedia Messaging Service (MMS), secure transaction services (mobile banking), and Over-the-Air (OTA) – use PDU mode of SMS protocol. We will now describe the PDU mode. PDU mode is used to encode the SMS header and user data (payload) in hexadecimal or decimal semi-octet format. In PDU it is possible to manipulate the fields of an SMS header and also modify the contents of user data. (The complete SMS is encoded in hexadecimal octets or decimal semi-octets.) In the PDU mode, an SMS is transferred from a mobile phone to SMSC by composing it with SMS-SUBMIT format.

Similarly, an SMS is received at a mobile phone in SMS-DELIVER format from SMSC. Figures 2 and 3 depict the format of SMS-SUBMIT and SMS-DELIVER PDU's. The different fields are briefly described in different Table 1, 2, 3 and 4.

07912943050010F011000C912943524295050000AA08C834885C27974

Figure 3: SMS-SUBMITT Format

07912943550000010 40C91294352429505000001205271536502 08C834885C279743

Figure 2: SMS-DELIVER Format

Table 1: SMS-SUBMITT Fields

SMS-OCTETS	Field Name	Length(bytes)	Description
07	Address Length	1	It indicates the cumulative size of address value and type of address fields.
91	Type of Address	1	It shows whether a phone number is represented in international or national format.
2943050010F0	Address Value	Variable	It contains the address of an SMSC.
11	First-Octet of SMS-SUBMITT	1	Define in Table
00	TP-MR	1	It is an ID assigned to a SMS in SMS-SUBMIT format by a local GSM modem.
0C	Address Length	1	It indicates the cumulative size of address value and type of address fields.
91	Type of Address	1	It shows whether a phone number is represented in international or national format.
294352429505	Address Value	Variable	It contains the address of receiver phone numbers.
00	TP-PID	1	It informs about the networking protocol and nature of SMS data (e.g. data download SIM).
00	TP-DCS	1	It specifies encoding scheme (7/8/16bit, compressed / uncompressed, message class) of UD.
AA	TP-VP	1 to 7	It informs SMSC that after how much time it should discard an undelivered SMS.
08	TP-UDL	1	It represents the size of the payload.
C834885C279743	TP-UD	Max. of 140	It contains the payload of an SMS and its maximum size is 140 bytes.

Table 2: First Octet of SMS-SUBMITT

Fields	Length (bits)	Description
TP-RP (Reply Path)	1	It tells SMSC to route the reply of a SMS on the same path followed by the received SMS.
TP-UDHI (User Data Header Indicator)	1	It indicates whether the user data contains optional headers.
TP-SRR (Status Report Request)	1	It demands an acknowledgment from the receiving device.
TP-VPF (Validity Period Format)	2	It specifies the format of TP-VP field.
TP-RD (Reject Duplicate)	1	It informs SMSC to reject duplicate messages.
TP-MTI (Message Type Indicator)	2	It informs SMSC that the SMS is in SMS-SUBMIT or SMS-DELIVER PDU format.

Table 3: SMS-DELIVER Fields

	Field Name	Length(Bytes)	Description
07	Address Length	1	It indicates the cumulative size of address value and type of address fields.
91	Type of Address	1	It shows whether a phone number is represented in international or national format.
294355000001	Address Value	Variable	It contains the address of an SMSC.
04	First-Octet of SMS-DELIVER	1	Define in Table
00	TP-MR	1	It is an ID assigned to a SMS in SMS-SUBMIT format by a local GSM modem.
0C	Address Length	1	It indicates the cumulative size of address value and type of address fields.
91	Type of Address	1	It shows whether a phone number is represented in international or national format.
294352429505	Address Value	Variable	It contains the address of receiver phone numbers.
00	TP-PID	1	It informs about the networking protocol and nature of SMS data (e.g. data download SIM).
00	TP-DCS	1	It specifies encoding scheme (7/8/16bit, compressed / uncompressed, message class) of UD.
01 20 52 71 53 65 02	TP-SCTS	7	It indicates the receiving time (local) of an SMS in SMS-DELIVER format only.
08	TP-UDL	1	It represents the size of the payload.
C834885C279743	TP-UD	Max. of 140	It contains the payload of an SMS and its maximum size is 140 bytes.

Table 4: First Octet of SMS-DELIVER

Fields	Length (bits)	Description
TP-RP (Reply Path)	1	It tells SMSC to route the reply of a SMS on the same path followed by the received SMS.
TP-UDHI (User Data Header Indicator)	1	It indicates whether the user data contains optional headers.
TP-SRI (Status report Indicator)	1	It demands an acknowledgment from the receiving device.
Bit 3 and 4 not used		
TP-MMS (More Message to Send)	1	It informs SMSC to reject duplicate messages.
TP-MTI (Message Type Indicator)	2	It informs SMSC that the SMS is in SMS-SUBMIT or SMS-DELIVER PDU format.

Concatenated SMS

CSMS allows fragmenting a long SMS into small messages and send them separately using different SMSSUBMIT-PDU. The application at the receiver does the reassembly and it appears a single SMS to the end user. In order to send a CSMS, TP-UDHI (see Table 2) bit is set in the headers of all fragmented SMS. The bit indicates that an optional User Data Header (UDH) is

present in the payload, which the receiving device uses to concatenate different fragments. The fields of typical UDH for CSMS are shown in Table 5.

Table 5: Fields of CSMS UDH

Fields	Description
1	It indicates the length of UDH.
2	(Information Element Identifier (IEI)): It tells the receiving the device about the objective of using UDH.
3	(Information Element Data Length (IEDL)): It indicates the number of fields in UDH.
4	(Information Element Data (IED)): It contains a CSMS reference number to identify different fragments of the same CSMS.
5	It indicates the total number of fragments of a CSMS.
6	It indicates the sequence number of currently received fragmented SMS.

We use AT commands to control – through serial or blue tooth connections – GSM modem of a mobile phone by an external controller. The external controller allows the users to compose an SMS in PDU mode and modify different header fields in SMS-SUBMIT and SMS-DELIVER format; as a result, an advisory can covertly transfer information in these fields. (The challenge, however, is that it should not affect the benign SMS.) Figure 4 shows the simple procedure of sending and receiving SMS in the PDU mode through AT commands. Now we will discuss six vulnerabilities that allow covert channel communication through SMS.

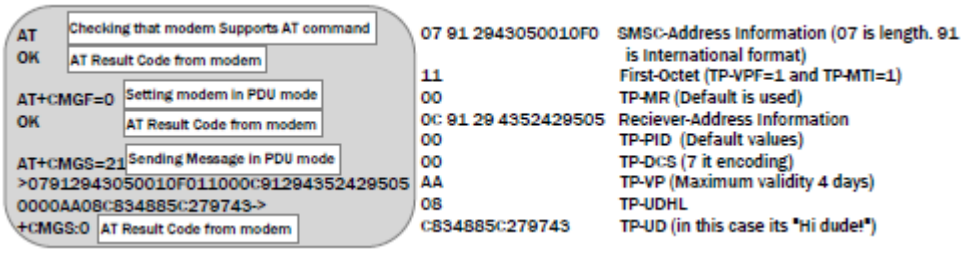


Figure 4(a): AT Commands to Send SMS

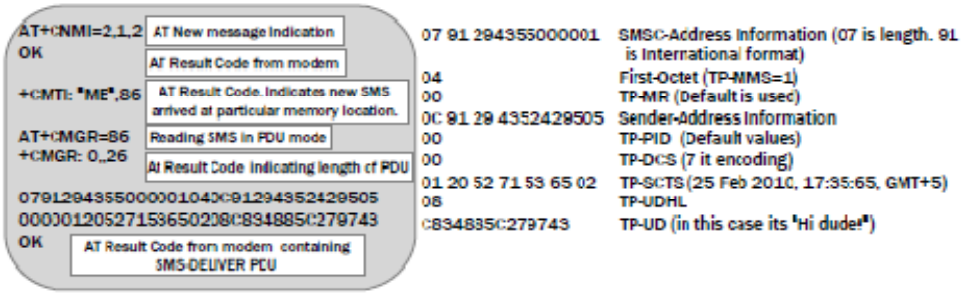


Figure 4(b): AT Commands to Receive SMS

Now we will discuss six vulnerabilities that allow covert channel communication through SMS. We will show that, utilizing these techniques, an advisory can secretly communicate with Bob on any active GSM network by using a variety of mobile phones.

Vulnerability 1: Single Text SMS

Recall from Table 5 that UDH is used to transmit a concatenated SMS; and in this case Field 5 and Field 6 indicate the total number of fragments and the sequence number of the current fragment respectively. We exploit a vulnerability by setting both fields to 01 that tricks the receiving device that the concatenated SMS consists of just one fragment and the current SMS is that fragment. We have empirically evaluated that mobile phone uses UDH in a single text message. (Ideally, this option should have not been allowed because the purpose of CSMS is to transmit message having user data more than 140 bytes.) As a result, the reference number i.e. Field 4 has become redundant and an advisory can covertly send data in it. Figure 5 shows the encoding of a single text message by exploiting UDH of CSMS. In Figure 5, "00" in the first octet tells GSM modem to use default SMSC information stored in the Subscriber Identity Module (SIM) of a mobile phone. Then TP-UDHI bit is set to 1 to indicate that particular SMS contains a UDH. An imposter is sending secret information "112F" in Field 4 of UDH.

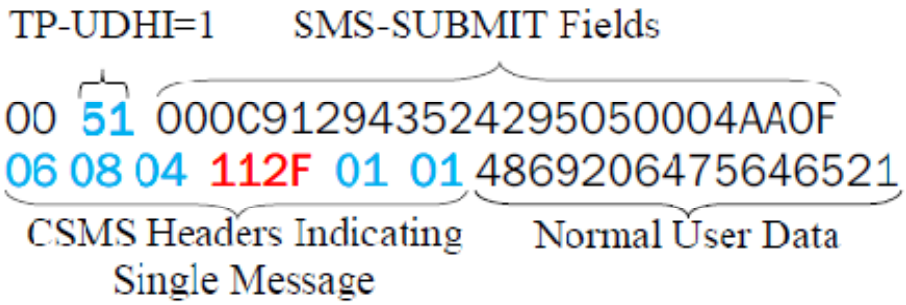


Figure 5: Single Text SMS Vulnerability

In this technique, an imposter has 16 Symbols *S* because of hexadecimal representation and 4 semi-octets (2 bytes) *n* available for covert communication. This leads to the channel capacity of (16 bits) per SMS. Figure 6 shows that a secret message "ATTACK OK" is covertly sent in four SMS messages containing overt text "Hi", "Gud", "Safe" and "Bye". Since information is transferred in Field 4, hence content filter on user data would not be able to raise any alarm.

Figure 6: Secret Message "ATTACK OK" by Exploiting Vulnerability 1

Vulnerability 2: Misusing Reference Number

If we logically extend the previous vulnerability, we can misuse the reference number – to covertly communicate secret information – Field 4 of an actual CSMS. (Remember that the reference number field is mandatory to do reassembly of fragmented SMS at the destination device.) Since a sending device can put any value in the reference number, this technique is hard-to-detect as compared with the previous one. Again, an advisory has 16 Symbols *S* because of hexadecimal representation and 4 semi-octets (2 bytes) *n* available for covert communication. This leads to the channel capacity of (16 bits) per CSMS. Figure 7 shows the example of a CSMS in which "4F4B" reference number is chosen to secretly communicate "OK".

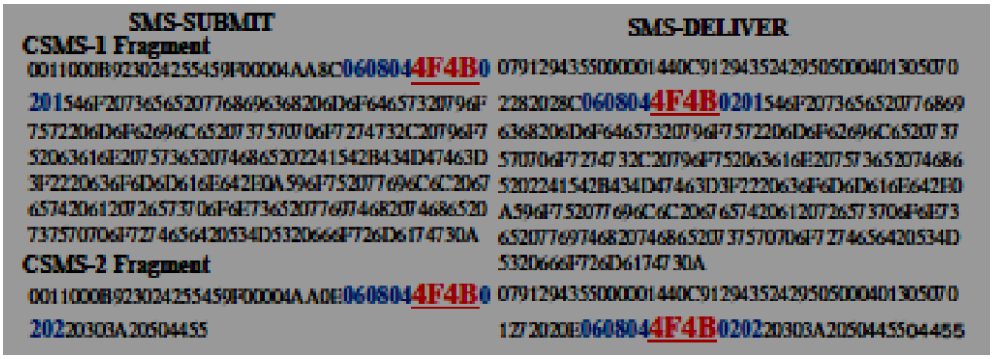


Figure 7: CSMS Reference Number Misuse

Vulnerability 3: Misusing Originator Port in Picture SMS

With the advent of Value Added Services (VAS), the usage for pictures, tones, and logos etc. SMS has significantly increased. A destination port field of "158A" in the extended UDH (see figure 8) is used to indicate the receiving device that SMS contains a picture. (Picture SMS is sent using CSMS because of its large size.) The picture (to be sent) is first converted into Over the Air (OTA) format (a standard size of 72x28 pixels) and is encoded in a hexadecimal format and then sent in the user data of CSMS [6]. The UDH of a picture (see Figure 8) also contains the picture display options (e.g. height, width etc.). The originator port (we empirically determined any value between "0000" to "FFFF" is legal in case of a picture SMS) is not used by the receiving device. As a result, it can be misused it covert communication.

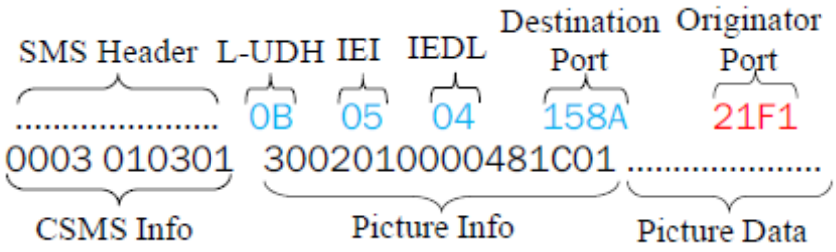


Figure 8: Picture SMS Header

It is interesting to note that an advisory can chose different originator ports in different fragments of the same CSMS (albeit compromising detection). This fact is depicted in Figure 9 in which advisory covertly sends "Hi", "No", and "OK" in a SMS-SUBMIT of picture SMS by misusing originator port field. (For brevity, we show only part of UDH.) The intercepted message will be

the picture shown in Figure 9. In this technique, the channel capacity is directly proportional to the number of fragments of a concatenated SMS used to transmit a picture. If Alice use m fragments, then the channel capacity would be bits per picture SMS, where we have 16 Symbols *S* because of hexadecimal representation and 4 semi-octets (2 bytes) *n* (of originator port) available for covert communication. If we combine it with the Vulnerability 2, the capacity increases to (+ 16) bits per picture SMS. An advisory can use 256 pictures (each having two fragments concatenated SMS) and encode the covert file data (2 bytes) in the originator port field of each fragment of a CSMS. If we assume the average transfer rate of a GSM modem (send/receive) is 10 SMS per minute, the 1 KB file can be transmitted secretly in 512 fragmented SMS that is expected to take less than 52 minutes.

Figure 9: Picture SMS Originator Port Misused

Vulnerability 4: Melodious Sound

iMelody is the standard format used for creating user defined monophonic sounds/melodies with the basic Enhanced Messaging Service (EMS) [7]. The iMelody sound consists of iMelody sound header, sound body and sound footer. Figure 10 shows the format of the basic iMelody format and its encoded PDU. Once a mobile phone receives an EMS containing the sound, mobile set uses the Sound Data Length (as indicated in Figure 10), headers and footers of iMelody to parse the enclosed sound data. After decoding sound body of iMelody, it finally plays the tone. This methodology is vulnerable to data injection because it allows for padding extra data – a secret message or a malicious code – after iMeldoy footer in the PDU of user data.

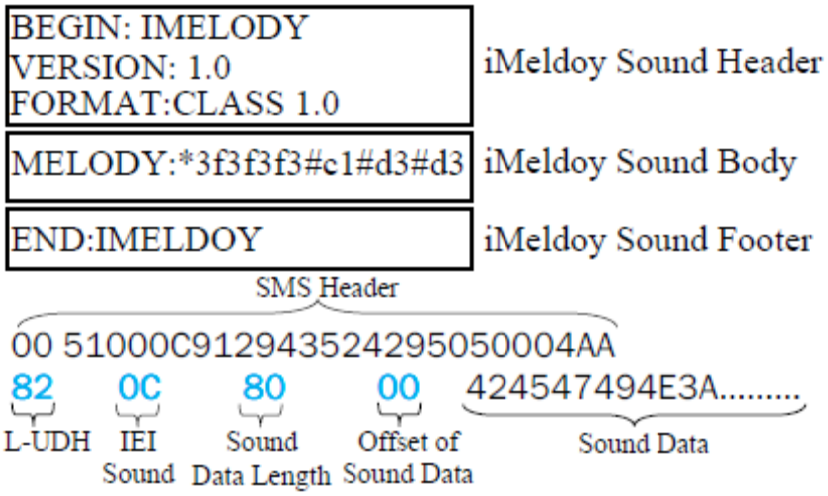


Figure 10: iMelody Format for Sounds/Melodies

Figure 11 shows SMS-SUBMIT and SMS-RECEIVED PDU of a sound SMS. Alice has covertly sent the word "dangerous" to Bob by padding it after the footer of iMelody (bold octets in the red color). Our investigations validate that the extra padding has no effect on the quality of sound and the padded data is also not visible to a mobile user. The capacity of channel in this scenario depends on the size of sound data and iMelody structure. It is possible to use CSMS in which all fragments have the same iMelody header and footer, but the melody data is portioned among these fragments [7]. (Note SMS can carry maximum of l = 280 semi-octets (140 bytes) of data.)

Figure 11: Melodious Sound Vulnerability

Let z be the cumulative size of UDH – including header, footer, and sound body of iMelody (in semi-octets). As a result, Alice has remaining $l - z$ semi-octets per SMS at her disposal to secretly communicate with Bob. If the melody is encoded in m fragments of CSMS, then the capacity of covert channel is (S = 16) bits per melody (EMS) message. In Figure

10, the value of *z* with 32 bytes of melody data is 192 semi-octets. If we add 8 semi-octets for sound EMS header to it than the total size would become 200 semi octets. As a result, an advisory can now covertly send 80 semi-octets (or 40 bytes) in a single sound message. To conclude, an advisory can secretly transfer 1 KB of file in just 26 (1024/40) tone messages. If we assume 10 SMS per minute send/receive rate of a GSM modem, then the time required to covertly transfer 1 KB message should be less than 3 minutes.

We have proven our argument that it is very important to redefine the structure and protocol of SMS on an urgent basis to effectively block its misuse by mafias or terrorist organizations. We now briefly describe our tool – GeheimSMS – that exploits above-mentioned vulnerabilities in SMS to enable Alice and Bob to secretly communicate or share sensitive information.

ARCHITECTURE OF GEHEIMSMS

We have developed a customized software tool in C# .NET – GeheimSMS – to empirically demonstrate, embedding of covert channels, on active GSM networks. Out tool consists of three main modules: (1) PC-modem interface module, (2) covert message embedder/extractor module, and (3) SMS PDU encoder/decoder module. GeheimSMS runs on a PC4 and communicates with a mobile phone through serial port, USB or blue tooth. The snap shot of our tool with the architecture is shown in Figure 14 and 15.

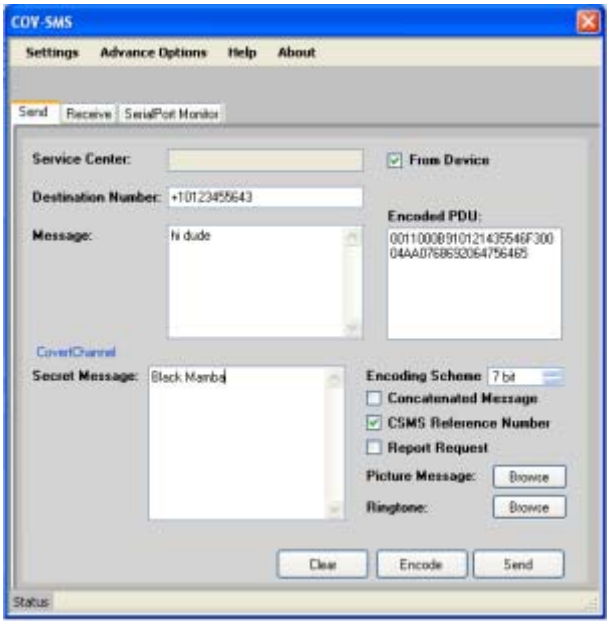


Figure 12: GeheimSMS

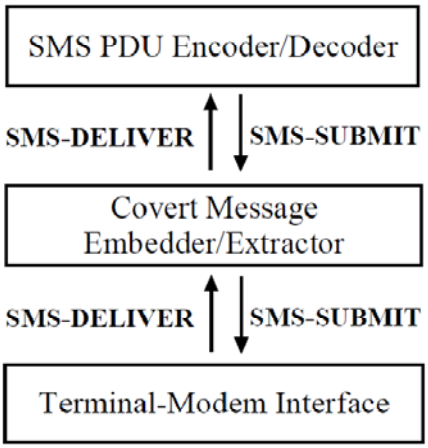


Figure 13: Architecture of GeheimSMS

PC-Modem Interface Module

This module has two assigned tasks: (1) configuration of a mobile device by using a terminal PC, and (2) to communicate with GSM modem for sending and receiving SMS through solicited AT commands and AT result codes. The configuration of the mobile device includes the setting of relevant parameters: COM port number, baud rate, number of data, parity and stop bits and flow control specification. The modem is configured in the PDU mode using "AT+CMGF" command. Furthermore, the routing of a new message to the PC terminal is done by using "AT+CNMI" command. On receiving new SMS, the module reads the SMS from a mobile and sends a SMS-DELIVER to covert message embedder/extractor module.

Covert Message Embedder/Extractor

Once it receives the SMS-DELIVER PDU from a PCmodem interface module, the covert message within the received SMS is decoded. Similarly, for encoding the message it gets the SMS-SUBMIT from the SMS PDU encoder/decoder module and encodes the covert messagein it and forwards it to the PC-modem interface.

SMS PDU Encoder/Decoder Module

This module takes input from the user (receiver number and SMSC number) along with a normal message and encodes it in the SMS-SUBMIT PDU. Moreover, it gets SMS-DELIVER PDU from the covert message embedder/ extractor module and displays the original contents of the received message.

Table 6: Mobile Phones Specification

Company	Model	Operating System
Sony Ericsson	w200i	Sony Ericsson
Nokia	6110	Symbian OS v9.2
Samsung	e900	Samsung Proprietary
Sony Ericsson	w800i	Sony Ericsson

EXPERIMENTS AND RESULTS

We now report the results of our experiments – on real world active GSM networks – using a number of different types of mobile phones. The list of mobile phones is shown in Table 6. We ran our experiments on the GSM network of one of the largest mobile operators having more

than 172 million active subscribers world wide. Its network core consists of NSN infrastructure with (GSM 900/1800) specifications. The operator has deployed Acision SMSC for the SMS traffic. We now want to transfer 1 KB file – by embedding covert channels through above-mentioned four vulnerabilities – using our GeheimSMS tool. Our interest is to understand the end-to-end delay of transferring this file by utilizing different SMS send rate per minute. We have tabulated the results for 10 SMS per minute send rate in Table 7.

Table 7: Timing Results for Transferring 1KB File on Active GSM Networks

Vulnerability No.	Transfer Time	No. of SMS
1	51 minutes	512 CSMS
2	1 hour 48 minutes	512 CSMS
3	57 minutes	256 Pictures
4	2 minutes 50 sec	26 Ring Tones

It takes 51 minutes to transfer the file by exploiting vulnerability 1 (single text). In case of the reference number vulnerability, it takes approximately twice the time (1 hour 48 minutes) to transfer the same file. We use 256 pictures and transmit each of them in 256 concatenated SMS CSMS having 2 fragments each. The transfer time in this case is 57 minutes as compared with an estimated time of 52 minutes (see vulnerability 3). We use 26 single SMS ring tones – transferring 40 bytes of secret data on the average – and the same file is now transferred in less than 3 minutes.

SECURITY ANALYSIS

The understanding of majority of security researchers is that covert channels are not a major security threat in conventional networked computing systems because of their low capacity [4]. An important contribution of our work is that we show that this notion is no more relevant in next generation mobile networks because we prove that it is possible to send multiple bytes within a single SMS; as a result, 1 KB of data can be transferred in less than 3 minutes. Moreover, the detection of our proposed covert channels is also difficult because they utilize different fields of a legitimate SMS. If we are willing to compromise capacity (in case of vulnerability 5 and 6), it makes detection a significant challenge. Since we are using storage channel; therefore, a sender and a receiver need not to worry about the synchronization issues. Last but not least, our pilot studies show that the appearance of an overt SMS (with an embedded

covert channel) remains unaffected, and encoded messages can be easily sent/recieved on active cellular networks without raising an alarm of a suspicious activity.

To conclude, covert channels in SMS are real and significant threat and now it is a real challenge for us to mitigate it. This challenge must be addressed – as suggested in [8] – that "A Trusted Computing Base should provide, wherever possible, the capability to audit the use of covert channel mechanisms with bandwidths that may exceed a rate of 1 bit in 10 seconds".

CONCLUSION

In this talk, we prove that intruders (or terrorists) can exploit vulnerabilities in SMS to secretly organize and execute criminal (or terrorist) activities by embed high capacity covert channels in SMS. As a result, 1 KB of data can be transferred in less than 3 minutes. The worst case scenario could be that imposters transmit malware in the covert channel. Therefore, we propose that it is imperative that SMS protocol must be thoroughly reviewed with a security perspective to ensure its misuse by mafia (or terrorist). Note, if intruders complement current vulnerabilities with well known encryption techniques, then it is nearly impossible to decipher the message (even if covert channel is detected) in realtime. Moreover, if they use intelligent stegnographic techniques – especially in case of picture and sound SMS – the detection of the covert channel will in itself become a significant. Therefore, security researchers need to focus on developing effective and efficient covert channel detection tools for SMS.

REFERENCES

[1] "Bin Ladens Messages Could Be Hiding In Plain Sight." USA Today, 2001, http://www.usatoday.com/life/cyber/ccarch/2001/12/19/maney.htm.

[2] National Computer Security Center, US DoD, "Trusted computer system evaluation criteria." Technical Report, DOD 5200.28-STD, Dec. 1985.

[3] "Documents obtained by the Electronic Privacy Information Center (EPIC) released through the Freedom Of Information Act can be reviewed at:," http://www.epic.org/open gov/foia/secrets.html.

[4] G. Shah et al., "Keyboards and covert channels," in Proc. of the 15th conference on USENIX

Security Symposium. USENIX Association Berkeley, CA, USA, 2006, pp. 59–75.

[5] Portio-Research, Mobile Messaging Future 2010-2014, http://www.portioresearch.com.

[6] GSM-ETSI, "03.40," Technical realization of the Short Message Service (SMS), 1998, http://www.3gpp.org/ftp/Specs/html-info/0340.htm.

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

[8] "The Trusted System Evaluation Criteria," Fred Cohen Associates, http://all.net/books/orange/chap8.html.