Embedding High Capacity Covert Channels in Short Message Service (SMS)

M. Zubair Rafique[1], Muhammad Khurram Khan[1],
Khaled Alghatbar[1], and Muddassar Farooq[2]

[1] Center of Excellence in Information Assurance
King Saud University, Riyadh, Saudi Arabia
[2] Next Generation Intelligent Networks Research Center
nexGIN RC, NUCES, FAST-Islamabad, Pakistan
{zrafique.c,mkhurram,kalghtbar}@ksu.edu.sa,
muddassar.farooq@nexginrc.org

Abstract. Covert Channels constitute an important security threat because they are used to ex-filtrate sensitive information, to disseminate malicious code, and, more alarmingly, to transfer instructions to a criminal (or terrorist). This work presents zero day vulnerabilities and weak-nesses, that we discovered, in the Short Message Service (SMS) protocol, that allow the embedding of high capacity covert channels. We show that an intruder, by exploiting these SMS vulnerabilities, can bypass the existing security infrastructure (including firewalls, intrusion detection systems, content filters) of a sensitive organization and the primitive content filtering software at an SMS Center (SMSC). We found that the SMS itself, along with its value added services (like picture SMS, ring tone SMS), appears to be much more susceptible to security vulnerabilities than other services in IP-based networks. To demonstrate the effectiveness of covert channels in SMS, we have used our tool GeheimSMS[1] that practically embeds data bytes (not only secret, but also hidden) by composing the SMS in Protocol Description Unit (PDU) mode and transmitting it from a mobile device using a serial or Bluetooth link. The contents of the overt (benign) message are not corrupted; hence the secret communication remains unsuspicious during the 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 without raising an alarm over suspicious activity.

Keywords: Short Message Service (SMS), SMS Covert Channels, SMS PDU.

1 Introduction

Covert channels based attacks on modern distributed systems pose a serious security threat as they are widely used in leaking sensitive information, secret Botnets

[1] The demo of GeheimSMS has been presented in Black Hat USA Arsenal event.
http://www.blackhat.com/html/bh-us-10/
bh-us-10-specialevents arsenal.html#rafique

J.J. Park et al. (Eds.): STA 2011, CCIS 186, pp. 1–10, 2011.

communication, the dissemination of malicious codes and the propagation of terrorists instructions. The major component that is exploited through such channel is the privacy assurance of the modern systems. A covert storage channel involves the embedding of secret and hidden messages by the sender and the interpretation of these secret and hidden messages by the receiver [1].

The current state-of-the-art research mainly concentrates on the embedding of a covert storage channel by exploiting the vulnerabilities in common Internet protocols such as TCP/IP [2] [3] [4] [5], UDP [6], HTTP [7], VoIP [8], SSH [9] and FTP [10]. The underlying idea of embedding a covert channel is based on the theory proposed by Lampson in [11]. The National Computer Security Center, a branch of the United States' National Security Agency (NSA), has established a standard called "Trusted Computer Security Evaluation Criteria (TCSEC)," which specifically refers to secret disseminating of information from a higher classification compartment to a lower classification in secure systems [1].

While previous research on covert storage channels was based on Internet based protocols, there has been no work done (to the best of our knowledge) to analyze the covert channel susceptibility of cellular services. We therefore undertook an empirical study to analyze the susceptibility of SMS to covert channel vulnerabilities. Note that SMS has become the most used cellular service, which is substantiated by a recent report that more than 5.5 trillion text messages were sent over carrier networks world-wide in 2009 [12]. The trend appears to be increasing as a survey projects that 6.6 trillion messages will be exchanged globally during 2010 [12].

This work presents zero day vulnerabilities and weaknesses, which we discovered, in the Short Message Service (SMS) protocol, which is the most used cellular networks service that allows the embedding of high capacity covert channels. The major contribution of our work is demonstration that intruders can exploit vulnerabilities in the 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 the existing IP based security infrastructures (including firewalls, Intrusion Detection Systems (IDSs), content filters) of an enterprise; which would allow, disgruntled employees to covertly leak sensitive and secret information of an organization, and (3) to embed high capacity covert channels using different SMS value-added-services; which would make it possible for a 1 KB of file to be transferred in less than 3 minutes. To demonstrate the effectiveness of covert channels in SMS, we practically embed data bytes (not only secret, but also hidden) by composing the SMS in Protocol Description Unit (PDU) mode and transmitting it from a mobile device using a serial or Bluetooth link. To conclude, our pilot studies showed that we need to quickly fix the vulnerabilities in the SMS protocol;-before this important service can be exploited by intruders. The security analysis of an SMS covert channel is not within the scope of this paper.

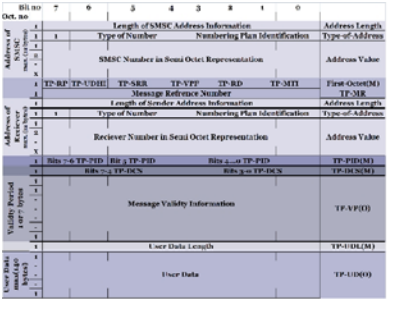
The rest of the paper is organized as follows. Section 2 provides a brief description of the SMS structure. We demonstrate the embedding of a covert channel by exploiting SMS vulnerabilities in Section 3. Our real world experiments on an active network, with more than 172 million subscribers, are discussed in Section 4. Finally, we conclude our paper in Section 5.

2 SMS Overview

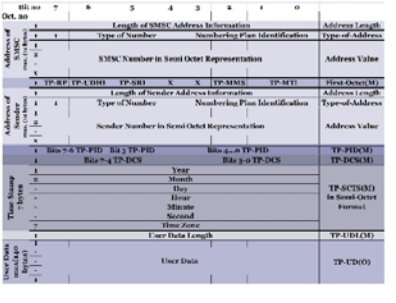
Mobile Originated (MO) messages are sent and received through SMSC, which has store and forward functionality. Once SMSC receives a message, it tries to forward it to the user; if the user is not available, it queues SMS and tries to retransmit it later. Once a mobile phone receives SMS from SMSC, it is processed by the GSM modem – the interface between GSM network and the application processor – of a mobile phone. An SMS can be sent and received in two modes: (1) PDU, and (2) text. Most of the 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 the PDU mode of SMS protocol. In the next section, we briefly describe the PDU mode. An interested reader, however, is referred to [13] for details of SMS.

2.1 SMS Encoding Modes

As mentioned before, PDU and text are two well known modes for sending and receiving SMS. In the text mode, plain text is given to the GSM modem, which selects a suitable scheme to encode the given text and puts it in the user data (payload) of an SMS. Moreover, it also attaches a default SMS header to the payload and sends it. In comparison, in the PDU mode, it is possible to manipulate the fields of an SMS header and also modify the contents of the 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 using the SMS-SUBMIT format. Similarly, an SMS is received at a mobile phone in the SMS-DELIVER format from SMSC. Figures 1(a) and 1(b) depict the formats of the SMS-SUBMIT and SMS-DELIVER PDU's. A user can send maximum of 140 bytes of user data in single SMS message. More information is transfer through CSMS.



(a) SMS-SUBMIT PDU Format



(b) SMS-DELIVER PDU Format

Fig. 1. SMS PDU Formats

2.2 Concatenated SMS (CSMS)

CSMS makes it possible to fragment a long SMS into small messages and send them separately using different SMS-SUBMIT PDUs. The application at the receiver does the reassembly and they appear as a single SMS to the end user. In order to send a CSMS, the TP-UDHI (see Figure 1) bit is set in the headers of all of the fragmented SMSs. This bit indicates that an optional User Data Header (UDH) is present in the payload, which the receiving device uses to concatenate the different fragments. The fields of a typical UDH are shown in Table 1.

Table 1. Fields of CSMS UDH

Fields	Description
1	It indicates the length of UDH.
2	(Information Element Identifier (IEI)): It tells the receiving 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 the currently received fragmented SMS.

3 SMS Covert Channel (SMSCC)

This secret communication through SMS is a modified version of the prisoner problem introduced by Simmons [14]. We here demonstrate how sender (Alice) and receiver (Bob) can exploit vulnerabilities in the SMS header and user data to covertly transfer information in an otherwise benign SMS. The objective is to embed covert information in such a way that to an independent warden (Wendy), it is a benign SMS containing characters, pictures and ring tones. Once Bob receives the benign SMS, he extracts the covert information from it. Similarly, Bob can also send covert information to Alice using the same information hiding technique (this makes the covert channel bidirectional). Moreover, the SMS covert channel has a high capacity because it is possible to transfer multiple bytes in a single SMS. Figure 2 shows the covert channel communication through SMS.

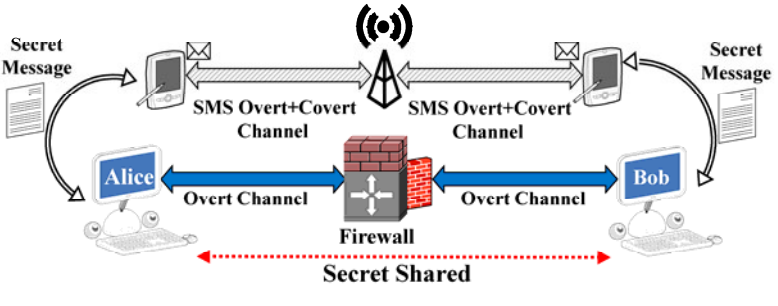


Fig. 2. SMS Covert Channel

In order to embed a convert channel, we make use of AT[2] commands. These commands are used to control – through serial or blue tooth connections – the GSM modem of a mobile phone using an external controller. The external controller allows the users to compose an SMS in PDU mode and modify different header fields in the SMS-SUBMIT and SMS-DELIVER formats. As a result, Alice can covertly transfer information in these fields. (The challenge, however, is that it should not affect the benign SMS.) Now, we will discuss six vulnerabilities that allow covert channel communication through SMS. (Our pilot studies show that, utilizing these techniques, Alice can secretly communicate with Bob on any active cellular network using a variety of mobile phones.)

3.1 Vulnerability 1: Single Text SMS

Recall from Table 2.2 that UDH is used to transmit a concatenated SMS. 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, which tricks the receiving device into thinking that the concatenated SMS consists of just one fragment and the current SMS is that fragment. We have empirically found that a mobile phone uses UDH in a single text message. (Ideally, this option should not have been allowed because the purpose of CSMS is to transmit messages having user data of more than 140 bytes.) As a result, the reference number, i.e., Field 4, has become redundant and Alice can covertly send data in it.

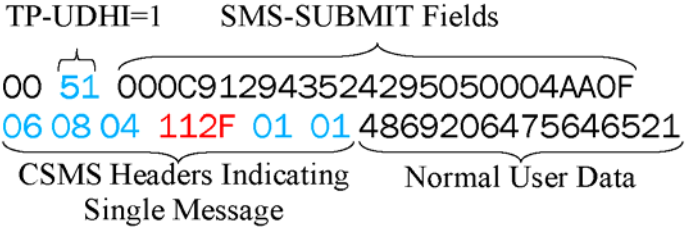


Fig. 3. (Vulnerability 1) Single Text SMS

Figure 3 shows the encoding of a single text message by exploiting the UDH of CSMS. In Figure 3, the "00" in the first octet tells the GSM modem to use the default SMSC information stored in the Subscriber Identity Module (SIM) of a mobile phone. Then, the TP-UDHI bit is set to "1" to indicate that particular SMS contains a UDH. Alice is sending the secret information "112F" in Field 4 of the UDH. With this technique, Alice has 16 Symbols S because of hexadecimal representation and 4 semi-octets (2 bytes) n available for covert communication. This leads to a channel capacity of $\log_2(S^n)$ (16 bits) per SMS.

[2] AT commands are the de' facto standard for controlling the modems.

3.2 Vulnerability 2: Misusing Reference Number

By logically extending the previous vulnerability, we can misuse the reference number for Field 4 of an actual CSMS to covertly communicate secret information. (Remember from Section 2.2, that the reference number field is mandatory for the reassembly of a fragmented SMS at the destination device.) Because a sending device can put any value in the reference number, this technique is hard to detect compared with the previous one. Again, Alice has 16 Symbols S because of hexadecimal representation and 4 semi-octets (2 bytes) n available for covert communication. This leads to a channel capacity of $\log_2(S^n)$ (16 bits) per CSMS. Figure 4 shows an example of a CSMS in which the "4F4B" reference number is chosen to secretly communicate "OK".

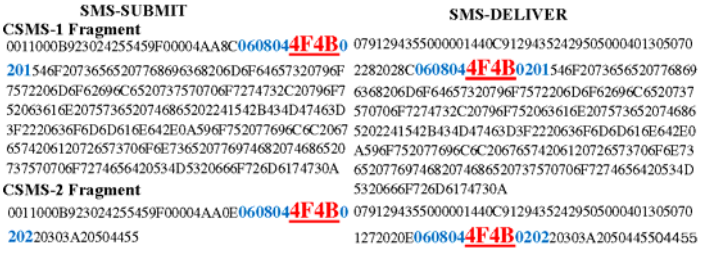


Fig. 4. CSMS Reference Number Misuse

3.3 Vulnerability 3: Misusing Originator Port in Picture SMS

With the advent of Value Added Services (VAS), which provide pictures, tones, and logos etc., the use of SMS has significantly increased. A destination port field of "158A" in the extended UDH (see Figure 5) is used to indicate to the receiving device that SMS contains a picture. (A picture SMS is sent using CSMS because of its large size.) The picture (to be sent) is first converted into the Over the Air (OTA) format (a standard size of 72x28 pixels). It is then encoded in an hexadecimal format and sent in the user data of CSMS [13]. The UDH of a picture (see Figure 5) also contains the picture display options (e.g., height, width, etc.). The originator port (we empirically determined that any value between "0000" and "FFFF" is legal in the case of a picture SMS) is not used by the receiving device. As a result, Alice can misuse it for covert communication with Bob.

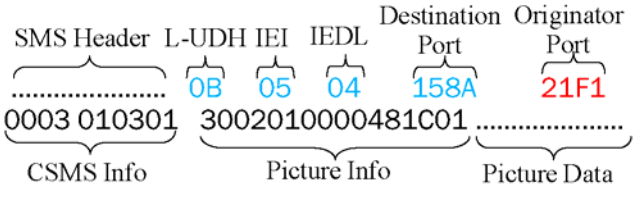


Fig. 5. Picture SMS Header

In this technique, the channel capacity is directly proportional to the number of fragments in the concatenated SMS used to transmit a picture. If Alice uses m fragments, then the channel capacity would be m * $\log_2(S^n)$ 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 Vulnerability 2, our capacity increases to (m * $\log_2(S^n)$+ 16) bits per picture SMS. Alice can use 256 pictures (each having two fragment concatenated SMSs) and encode the covert data (2 bytes) in the originator port field of each fragment of a CSMS. If we assume that the average transfer rate of a GSM modem (send/receive) is 10 SMS per minute, a 1 KB file can be transmitted secretly in 512 fragmented SMSs, which is expected to take less than 52 minutes.

3.4 Vulnerability 4: Melodious Sound

iMelody is the standard format used for creating *user defined* monophonic sounds with the basic Enhanced Messaging Service (EMS) [15]. An iMelody sound consists of the iMelody sound header, sound body and sound footer. Figure 6 shows the basic iMelody format and its encoded PDU.

Once a mobile phone receives an EMS containing the sound, the mobile set uses the Sound Data Length (as indicated in Figure 6), headers and footers of iMelody to parse the enclosed sound data. After decoding the iMelody sound body, it finally plays the tone. This methodology is vulnerable to data injection because it allows for the padding of extra data – a secret message or a malicious code – after the iMelody footer in the PDU of the user data. Our investigations validate that the extra padding has no effect on the quality of the sound and also that the padded data is not visible to a mobile user. The channel capacity in this scenario depends on the size of the sound data and iMelody structure.

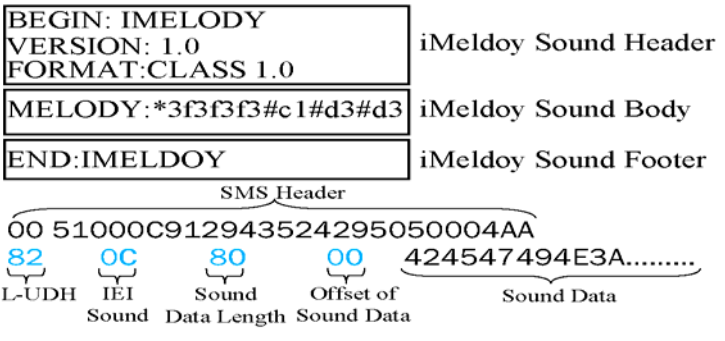


Fig. 6. iMelody Format for Sounds/Music

It is possible to use a CSMS in which all of the fragments have the same iMelody header and footer, but the melody data is portioned among these fragments [15]. (Note that SMS can carry a maximum of l = 280 semi-octets (140 bytes) of data.)Let z be the cumulative size of UDH including iMelody header, footer, and sound body of (in semi-octets). As a result, Alice has l-z semi-octets per SMS remaining at her

disposal to secretly communicate with Bob. If the melody is encoded in m fragments of CSMS, then the capacity of the covert channel is $m*\log_2(S^{kz})$ (S = 16) bits per melody (EMS) message. In Figure 6, the value of z with 32 bytes of melody data is 192 semi-octets. If we add 8 semi-octets for the sound EMS header, the total size would become 200 semi octets. As a result, Alice can now covertly send 80 semi-octets (or 40 bytes) in a single sound message. To conclude, Alice can secretly transfer 1 KB file in just 26 (1024/40) tone messages. If we assume a 10 SMS per minute send/receive rate for a GSM modem, then the time required to covertly transfer a 1 KB message should be less than 3 minutes.

3.5 Vulnerability 5: Encoding Option

Recall from Figure 1, that TP-DCS is used to indicate a 7 bit, 8 bit or 16 bit data encoding scheme for an SMS [13]. Our pilot studies show that Alice can secretly send 1 bit of data to Bob assuming that the covert data is "0" (when text is encoded in 7 bit or 16 bit) or "1" (when 8 bit text encoding is used). Figure 7 shows that Alice covertly sends 2 bits of covert data "01" in an overt text "hi" (7 bit encoding) and "Gud" (8 bit encoding). This technique has a low capacity but is very hard to detect.

Fig. 7. TP-DCS used for Covert Channel

3.6 Vulnerability 6: Status Report

The TP-SRR Status Report Request bit in the first octet of SMS-SUBMIT demands an acknowledgment from the receiving device. The field appears in the TP-SRI of the first octet of SMS-DELIVER and asks the receiving device to acknowledge a received SMS (see Figure 1). This allows Alice to embed a 1 bit capacity covert channel (see Figure 8) in SMS. Our pilot studies show that some operators do not support the status report feature to reduce the SMS load on SMSC.

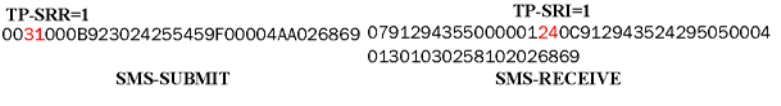


Fig. 8. TP-SRR used for Covert Channel

4 Experiments and Results

We now report the results of our experiments – on real world active cellular networks using a number of different types of mobile phones. We ran our experiments on the GSM network of one of the largest mobile operators, with more than 172 million

active subscribers world-wide. Its network core consists of an NSN infrastructure with (GSM 900/1800) specifications.

We wanted to transfer a 1 KB file – by embedding covert channels through the above-mentioned six vulnerabilities – using our GeheimSMS tool. Our goal was to understand the end-to-end delay for transferring this file by utilizing different SMS send rates per minute. We have tabulated the results of a 10 SMS per minute send rate in Table 2. It took 51 minutes to transfer the file by exploiting vulnerability 1 (single text). In the case of reference number vulnerability, it took approximately twice the time (1 hour 48 minutes) to transfer the same file. We used 256 pictures and transmitted each of them in 256 concatenated SMS CSMSs having 2 fragments each. The transfer time in this case was 57 minutes as compared with an estimated time of 52 minutes (see vulnerability 4 in Section 3). We used 26 single SMS ring tones transferring 40 bytes of secret data on the average and the same file was transferred in less than 3 minutes. As expected, the transfer time using the last two vulnerabilities was about 16 hours.

Table 2. Timing Results for Transferring 1-KB Covert Message on Active GSM Network

Vul. No.	Transfer Time	No. of SMS
1	51 minutes	512 SMS
2	1 hour 48 minutes	512 CSMS
3	57 minutes	256 Pictures
4	2 minutes 50 sec	26 Ring Tones
5 and 6	16 hour 12 minutes	8192 SMS

5 Conclusion

In this paper, we prove that intruders (or terrorists) can exploit vulnerabilities in SMS to secretly organize and execute criminal (or terrorist) activities by embedding high capacity covert channels in SMS. We empirically proved that it is possible to send multiple covert bytes within a single SMS; as a result, 1 KB of data can be transferred in less than 3 minutes. We also proved that intruders (or terrorists) can exploit the vulnerabilities in SMS to secretly organize and execute criminal (or terrorist) activities. Our pilot studies showed that the appearance of an overt SMS (with an embedded covert channel) remains unaffected, and encoded messages can be easily sent/received on active cellular networks without raising an alarm about suspicious activity. The detection of SMS covert channels will be the subject of our future research.

References

1. National Computer Security Center, US DoD: Trusted computer system evaluation criteria. Technical Report, DOD 5200.28-STD (December 1985)
2. Ahsan, K., Kundur, D.: Practical data hiding in TCP/IP. In: Proc. of the 9th Workshop on Multimedia & Security, pp. 25–34. ACM, Texas (2002)

3. Rowland, C.: Covert channels in the TCP/IP protocol suite. First Monday 2(5-5) (1997)
4. Giffin, J., Greenstadt, R., Litwack, P., Tibbetts, R.: Covert messaging through TCP. In: Dingledine, R., Syverson, P.F. (eds.) PET 2002. LNCS, vol. 2482, pp. 194–208. Springer, Heidelberg (2003)
5. Murdoch, S., Lewis, S.: Embedding covert channels into TCP/IP. In: Barni, M., Herrera-Joancomartí, J., Katzenbeisser, S., Pérez-González, F. (eds.) IH 2005. LNCS, vol. 3727, pp. 247–261. Springer, Heidelberg (2005)
6. Fisk, G., et al.: Eliminating steganography in internet traffic with active wardens. In: Petitcolas, F.A.P. (ed.) IH 2002. LNCS, vol. 2578, pp. 18–35. Springer, Heidelberg (2003)
7. Bauer, M.: New covert channels in http: adding unwitting web browsers to anonymity sets. In: Proc. of the 2003 ACM Workshop on Privacy in the Electronic Society, pp. 72–78. ACM, NY (2003)
8. Mazurczyk, W., Kotulski, Z.: New VoIP traffic security scheme with digital watermarking. Computer Safety, Reliability, and Security, 170–181 (2006)
9. Lucena, N., Pease, J., Yadollahpour, P., Chapin, S.: Syntax and semantics-preserving application-layer protocol steganography. In: Fridrich, J. (ed.) IH 2004. LNCS, vol. 3200, pp. 164–179. Springer, Heidelberg (2005)
10. Zou, X., Li, Q., Sun, S., Niu, X.: The Research on Information Hiding Based on Command Sequence of FTP Protocol. In: Khosla, R., Howlett, R.J., Jain, L.C. (eds.) KES 2005. LNCS (LNAI), vol. 3683, pp. 1079–1085. Springer, Heidelberg (2005)
11. Lampson, B.: A note on the confinement problem. Communications of the ACM 16(10), 613–615 (1973)
12. Portio-Research: Mobile Messaging Future (2010-2014), http://www.portioresearch.com/
13. GSM-ETSI: 03.40. Technical realization of the Short Message Service (SMS) (1998), http://www.3gpp.org/ftp/Specs/html-info/0340.htm
14. Simmons, G.J.: The prisoners problem and the subliminal channel. In: Proc. of Advances in Cryptology (CRYPTO), pp. 51–67 (1984)
15. Le Bodic, G.: Mobile Messaging technologies and services: SMS, EMS and MMS. John Wiley Sons Inc., Chichester (2005)
16. The Trusted System Evaluation Criteria. Fred Cohen Associates, http://all.net/books/orange/chap8.html