Chapter 17
SIM Cards Forensics

Learning Objectives
The objectives of this chapter are to:

- Understand SIM Card Architecture
- Know about common forensic evidence that can be found on a SIM card
- Understand SIM card forensic analysis process
- Become familiar with forensic SIM tools

Information is everywhere in term of any digital device, but today the mobile phones are more attractive place for the data because of usage every day. In addition, mobile phones are one of the key points that investigator must concern about. Therefore, Digital forensic expert must look after each possible evidence that could be extracted from mobile phones. In this chapter, we mainly focus on the data that can be found in a mobile phone SIM card.

A subscriber identity module or subscriber identification module (SIM), also known as an SIM card contains distinctive data that is different from the data, which is captured by forensic acquisition of the device itself [1]. In this chapter, we are going to introduce SIM cards forensics. However, this kind of forensics regularly is an essential stage in mobile forensics where important evidences can be extracted. As a result, we will have different section for each as follows.

17.1 The Subscriber Identification Module (SIM)

SIM stands for "Subscriber Identity Module", which is a small electronic card that conserves the identity of the subscriber who is an authorized mobile phone user for the GSM cellular networks. In addition, it stores the encryption keys used in the

X. Lin, *Introductory Computer Forensics*,
https://doi.org/10.1007/978-3-030-00581-8_17

Fig. 17.1 SIM Card

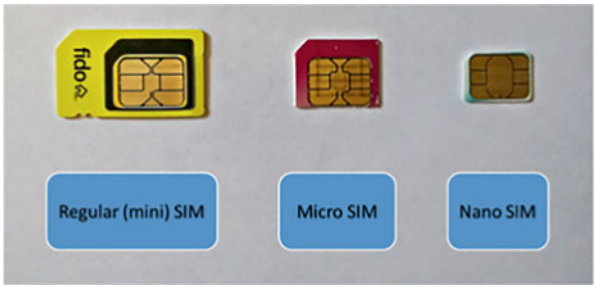


Fig. 17.2 Different sizes of SIM cards

authentication of users on the cell network. Universal Integrated Circuit Card (UICC) is the technology, which is used for this kind of smart cards where GSM SIM card is an example of this technology (An example of SIM card shown in Fig. 17.1). In simple words, network service provider uses smart cards to authenticate the users to its network which is SIM cards. Thus, the main role of SIM cards is to provide and prove the identity. Additionally, SIM cards have another function; it provides small memory for the user to store contacts and keep logs of calls and SMS.

These cards come in varied sizes such as Mini, Micro, and Nano SIM cards as shown in Fig. 17.2. In fact, the first Sim card has the credit card size. After that, mini SIM card is used, which is also called Regular or Standard SIM card with dimension size of 15 mm of width, 25 mm of height, and 0.76 mm of thickness. As the phone devices become smaller, the size of SIM cards shrinks to lower size level without losing its functionality where nowadays the Nano SIM card is the smallest one. The Micro SIM dimension size is 12 mm of width, 15 mm of height, and 0.76 mm of thickness while the Nano SIM card has 8.8 mm of width, 12.3 mm of height, and 0.67 mm of thickness. Furthermore, embedded SIM card (eSIM) is currently used which the SIM card is combined with the device circuit board with dimension size of 5 mm of width, 6 mm of height, and less than 1 mm of thickness.

Nowadays SIM cards are shipped in a one size fitting all manner, such that the Nano SIM card can be taken from the Micro SIM card, which may also be removed from the Regular SIM card. It is known as multi-SIM card. In other words, you can pop out the SIM card you need for your phone, shown in Fig. 17.3.

All SIM cards consist of two parts of data storage. One part is used for system information and the other is used for user information which is locked by user PIN code. From the user part, contacts, call logs, and SMS could be recovered even these deleted entities. Three essential information is stored in the system part namely

Fig. 17.3 Multi-SIM card

ICCID, IMSI, and MSISDN numbers and location area data. ICCID (Integrated Circuit Card ID) is the serial number of SIM card. While, IMSI stands for "International Mobile Subscriber Identity" which is used to identify the subscriber on the network. Finally, MSISDN stands for "Mobile Subscriber Integrated Services Digital Network Number". The MSISDN is actually the phone number of the device. In addition, Location Area Identity (LAI) information can be extracted from SIM cards. The network uses LAI information to track the zone area of the device that holds the SIM card to provide the available service zones. This information is very important to spot location history of the device as well as device holder. Later, we will give more details about these information in the system part and how we can extract it from the SIM model.

17.2 SIM Architecture

SIM card as any smart system consists of a processor, memory, and operating system. The microprocessor and OS could be Java card that uses Java programming language platform for embedded devices or exclusive to the issuer.

The actual SIM chip is covered by a plastic frame. However, only the connection surface (shown in Fig. 17.4) is exposed to the outside world which holds communication between the device and the SIM card through a serial interface. Furthermore, the SIM slot of devices is usually not reachable to keep SIM card from external access. The slot normally is located under the device battery or on one side of the device which can be accessed using a pin hole as shown in Fig. 17.5.

SIM card consists of both volatile and non-volatile memory. First, RAM (Random Access Memory) which is the volatile memory is used to execute programs in SIM card. On the other hand, the non-volatile memory types are ROM (Read Only Memory) and EEPROM (Electrically Erasable Programmable Read Only Memory).

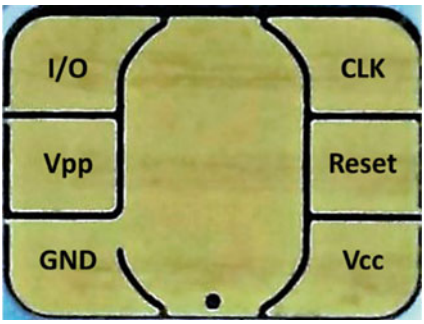


Fig. 17.4 SIM card pinout

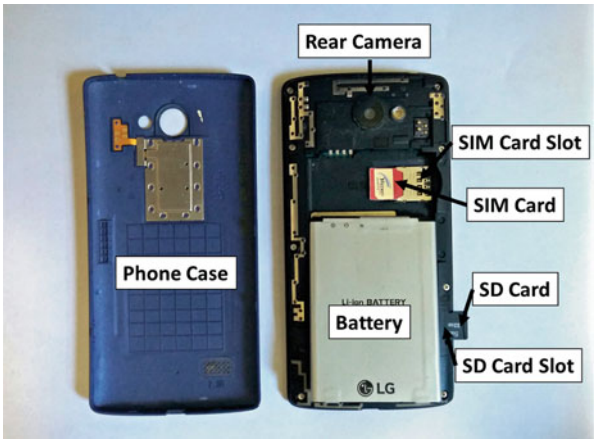


Fig. 17.5 SIM card slot

In ROM, the operating system is stored as well as the built-in security data such as encryption scheme and user authentication. However, EEPROM contains the main data and have hierarchical based file system structure.

The file system tree of SIM cards has three classes of identifiers, namely, main files, dedicated files, and elementary files. Each file system identifier has a header and body. The header contains information about the files such as file type, file metadata, and permissions. Main Files (MF) is the root file directory of SIM file system which starts from 0x3F00 memory address. Thus, in any SIM file system, must be one MF or more. The Dedicated Files (DF) are the underlying level directories of MF which are used by SIM card services. Their contents and functions had been defined by the GSM 11.11 Specifications. All DF directories identifiers starts with 0x7FXX address. Elementary Files (EF) is exactly where user data are stored thus recovering those files are the main goal of a forensics investigator. In fact, the EF file can exist under MF (memory address = 0x2FXX) or DF (memory address = 0x6FXX). Two types of data can be stored in EF storage which are plain data and listed data. This file system tree is shown in Fig. 17.6.

The operating system controls all access operations of SIM card files. It performs the normal operation set such as read, write, create, etc.. However, EF data can be

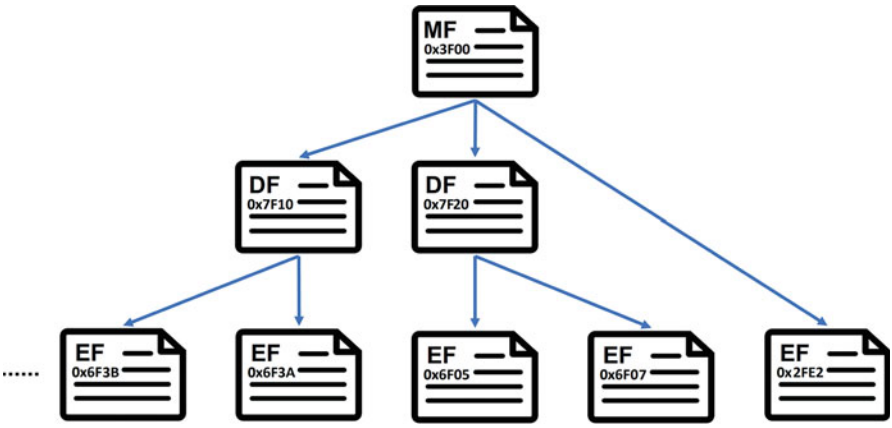


Fig. 17.6 File system tree of SIM cards

stored in three formats, namely linear, cyclic, and transparent. Fixed linear files are files which the data are stored sequentially in records that have fixed length format. To reach these sequences of records, a record number is used and by applying read next and read previous operations. Cyclic files are a sequence of records when the operating system tries to access the next record after the last record in the sequence then it returns to the first one. The last type is the most used format which is the transparent. There is no standard format because it is just sequences of bytes. Any application can use its own format to store its data. However, any file of transparent format must be linked with control information that manages accessing the file.

17.3 Security

Based on application, the access operations over files are controlled by permissions that associated with files. As mentioned before, all MF, DF, and EF files are associated with security permissions with various levels. Personal Identification Number (PIN) is used to protect the data of subscriber where this code (4–8 digits) is used for verification. However, the PIN code is used over the subscriber data and the system data. Personal Unlocking Key (PUK) also known as PIN Unblocking Key is used to reset the forgotten PIN code.

As an investigator, firstly, the SIM card must be removed from the device especially if the PIN code is locking the device. Therefore, the SIM and the device are handled alone. Most of commercial tools such as Cellebrite [2] and XRY [3] show a warning message when the SIM locking code is enabled. There are a fixed number of tries to enter the valid PIN code usually three attempts. Therefore, counting the remaining number of guessing tries is very important because the failure of providing the valid PIN code in three attempts will block the SIM card.

Fig. 17.7 PIN and PUK codes

On the other hand, PUK is used to reset the PIN code in this case. However, also entering PUK code has certain number of attempts (usually ten attempts) where forensic tools don't provide limitless attempts of guessing the PUK number. Failing to provide the correct PUK code in the limited number of attempts will block the SIM card endlessly.

Moreover, unlocking PIN code through the device is very dangerous because the failure will lead to block the device. That is why we suggest removing the SIM card at the beginning of the forensic process. Unfortunately, without unlocking the PIN code, the investigator cannot extract the user data from the SIM filesystem. Moreover, the PUK code can be obtained by the network service provider using ICCID number which can be found in the filesystem or sometimes maybe found printed in the SIM itself as shown in Fig. 17.7. But each service provider has its policy of keeping the PUK code of each subscriber where there is time to live limit for all PUK codes. Therefore, in some cases, the investigator cannot recover the PUK code so the PIN code. In this case, the user data inside the SIM card will not be extracted anymore and only system data can help in the investigation.

The best case for SIM card forensic is when the investigator has the valid PIN code. As mentioned before, there are a limited number of tries for SIM unlocking in contrast of smartphones which investigator have an unlimited number of tries using the forensic tool during the physical acquisition. However, the investigator can acquire the default PIN code using the service provider documentation. Usually, the default PIN code is 0000.

In addition, the investigator may be able to clone the SIM cards that locked by PIN code. Then using some commercial tools such as Cellebrite and XRY, the PIN code is removed from the copied SIM. Applying the unoriginal to the device may unlock the device especially non-smartphone devices.

17.4 Evidence Extraction

For SIM cards, contacts, call logs, and SMS messages are stored in the EF filesystem. To find these evidences, the extracted data must be decoded to help the investigator to read the extracted evidences [4, 5]. Why is it important to look at evidences inside the SIM memory while contacts, call logs, and SMS could be extracted from the device itself? The answer is simple. For instance, the contacts saved in the device may differ from the contacts saved in the SIM directly. Also, some deleted SMSs could be recovered from the SIM card memory. Further, in some cases, the SIM card is the only object that the investigator has on his hand because the device is missing or broken.

17.4.1 Contacts

As mentioned before, SIM cards may contain some particular contacts that differ from contacts found in the device. This is because SIM cards allow users to save contacts directly on it. Also, phone devices permit users to choose between the device, SIM card, and cloud to save their contacts. The number of contacts that the SIM card can hold differs from one to another. The old versions of 32 K SIM card can store up to 250 contacts in their memories while in the newer versions increased up to 500 contacts in 64 K SIM cards and 600 contacts or more in 128 K SIM cards. However, it also may differ due to manufacturer and service provider specifications. Contacts data saved in EFs of SIM cards is known as Abbreviated Dialing Numbers (ADN). However, ADN is stored by the device user and can not be accessed by the network service provider. Therefore, it is very helpful for the investigator to make suspects connection available as an example.

17.4.2 Calls

In a mobile phone device, outgoing, incoming, and missed call logs are stored while only the outgoing calls are saved in SIM cards. Device configuration determines if the outgoing calls log will be stored to SIM card storage or not. However, outgoing calls log on the SIM card could be different and not the same as device log. Therefore, calls logs must be extracted from the SIM card by a forensic investigate which could ensure useful evidences. Outgoing calls List is known as Last Dialed Numbers (LDN) or Last Numbers Dialed (LND).

17.4.3 SMS

Short Message Service (SMS) is one of the important evidence that could be extracted from SIM card. Users can send and receive text messages containing up to 160 English characters or 70 other language alphabets. The SMS body is encoded using special 7-bit encoding called GSM 03.38 encoding or Unicode for non-English characters. Indeed, large messages that exceed the upper limit are divided into several SMSs. The sender device disassembled the large message while the receiver reassembled after receiving all parts of the message. Note that there exist many third party instant messaging applications such as Google Hangouts discussed in previous chapter, and the messages sent by these applications are not stored in the SIM storage.

17.5 Case Studies

In this section, we will provide case studies to show how and where evidences can be extracted from SIM card. As mentioned before SIM data forensics is a significant stage in mobile device investigation. In the previous section, we have indicated that contacts, outgoing calls, instant messages even deleted ones, and some system data can be extracted from SIM Card. Next, we will show how to do data acquisition of SIM model. Then a data analysis of recovered SIM data will be provided for all types of evidences.

17.5.1 Experiment Setup

SIM:

- Mini SIM Model

Tools:

- Cellbrite UFED Touch
- HxD 2.0—Hex editor
- PDU Converter—SMS Server Tools 3 (http://smstools3.kekekasvi.com/topic.php?id=288)
- Number analysis tools—International Numbering Plans (https://www.numberingplans.com)

17.5.2 Data Acquisition

In a digital investigation, data acquisition is a critical step, which refers to the process by which data in any digital devices or network is extracted and stored in a forensically sound way. Data is then processed for use in forensic analysis. In this

case study, we use Cellebrite UFED Touch to extract SIM Data. The detailed data acquisition process is described below:

1. After inserting the Sim card into Cellebrite UFED Touch, you will get the below figure to determine the "Extract from" what device, and then select Sim card.

2. Select the extraction type, which will be for our case is file system extraction.

3. Select the Extraction Location, or storage device for extracted SIM Data, which will be the removable device.

4. At this step, you must check that you have Inserted the SIM card into the SIM card reader slot located in the middle of the front panel, then; select continue.

5. After clicking Continue, if the SIM card is partitioned, a prompt appears to select an appropriate partition to read such as SIM (GSM) or USIM (3GPP). In our case, you will select SIM (GSM).

6. Then, the Extraction will be in progress.

7. Once the extraction process is complete, the SIM data extraction summary screen appears, displaying a summary of the extraction process. By this step, the image will be ready in the removable device if you click finish.

17.5.3 Data Analysis

After we extract the data from the SIM card, we must be able to locate the needed data such as contacts, deleted messages, etc. Also some information can be found in the system part namely ICCID, IMSI, and MSISDN. However, locating the needed evidence from the extracted data is not straightforward due to special file system of a SIM card as mentioned before. Each type of information could be found in certain file identifier (EF). Therefore, we are going to show steps to extract those evidence by knowing in which EF are stored and how to read such a data.

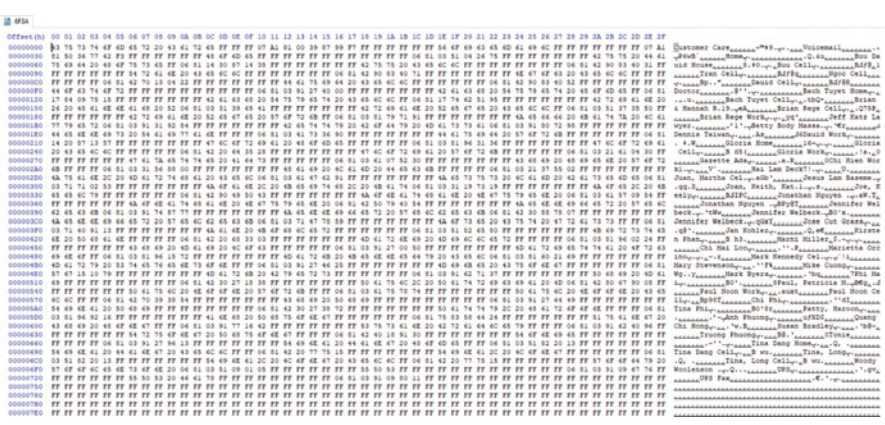
17.5.3.1 Contacts

As mentioned before, Abbreviated Dialing Numbers (ADNs) is the contacts data which is saved in EFs of SIM cards. ADNs are usually stored in EF 6F3A which is EF underlie under DF header. Due to limited memory resource, some of SIM cards may use EXT1 (0x6F4A) EF records to save additional data of ADN. ADN data is saved in fixed linear format. In memory, the contact name is stored in ASCII code format as a plain text. The contact number associated with the contact name comes on the next record. The contact number record has a special format where each group of bytes refers to certain information will be discussed next. The first byte indicates the number of bytes are used for the number in the record. The next byte will have either 0x81 or 0x91 hex values. If the number is an unknown number then the value will be 0x81. However, this byte gets 0x91 value when the contact number is an international number. The rest of bytes which the length of it specified by the first byte are for the contact number itself. The format used for storing contact number is the reverse nibble which means the least significant 4-bits must be read first in a byte. For instance, if the contact name record has the hex value equal to (0x 4C 69 6E FF FF . . . FF) and the contact number record value is (0x 06 81 86 57 28 93 70 FF . . . FF) then decoding the ASCII code of the first record will be (Lin) which is the contact name. From the next record, the first byte (0x06) indicates that only the next six bytes are used for the contact number. The number is set as unknown number due to (0x81) value. Finally, the contact number can be extracted from (0x 86 57 28 93 70) value. As we mentioned, the number is stored in reverse nibble order thus the decoded number associated with the contact name "Lin" is (687-582 3907).

Let's give some practical examples here, contacts (ADN) are stored in EF 6F3A which located under MF 7F10 as shown in figure below:

caseSIM\FileDump_SIM\SIM_2G_3G SIM\SimDump\3F00\7F10\6F3A]

We use a Hex editor to open ADN file as shown below:



We know that all contacts are saved in this certain EF but now we need to decode each record of contacts to obtain the contact name and associated number. Each contact has to have two records, where the first record is for the contact name while the other record is for the contact number. Thus, it is useful to present the hex file in 15 bytes per row representation as shown below:

6F3A

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	
00000000	43	75	73	74	6F	6D	65	72	20	43	61	72	65	FF	FF	Customer Care
0000000F	FF	07	A1	81	00	39	87	99	F7	FF	FF	FF	FF	FF	FF	
0000001E	56	6F	69	63	65	6D	61	69	6C	FF	FF	FF	FF	FF	FF	Voicemail
0000002D	FF	07	A1	81	50	36	77	42	F3	FF	FF	FF	FF	FF	FF	
0000003C	48	6F	6D	65	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	Home
0000004B	FF	06	81	03	51	04	26	75	FF	FF	FF	FF	FF	FF	FF	
0000005A	42	75	75	20	44	61	75	69	64	20	48	6F	75	73	65	Buu Dauid House
00000069	FF	06	81	14	30	87	14	38	FF	FF	FF	FF	FF	FF	FF	
00000078	42	75	75	20	43	65	6C	6C	FF	FF	FF	FF	FF	FF	FF	Buu Cell
00000087	FF	06	81	42	90	83	40	31	FF	FF	FF	FF	FF	FF	FF	
00000096	54	72	61	6E	20	43	65	6C	6C	FF	FF	FF	FF	FF	FF	Tran Cell
000000A5	FF	06	81	42	90	83	40	71	FF	FF	FF	FF	FF	FF	FF	
000000B4	4E	67	6F	63	20	43	65	6C	6C	FF	FF	FF	FF	FF	FF	Ngoc Cell
000000C3	FF	06	81	42	70	13	04	22	FF	FF	FF	FF	FF	FF	FF	
000000D2	44	61	75	69	64	20	43	65	6C	6C	FF	FF	FF	FF	FF	Dauid Cell
000000E1	FF	06	81	42	90	83	40	52	FF	FF	FF	FF	FF	FF	FF	
000000F0	44	6F	63	74	6F	72	FF	FF	FF	FF	FF	FF	FF	FF	FF	Doctor
000000FF	FF	06	81	03	91	27	40	00	FF	FF	FF	FF	FF	FF	FF	
0000010E	42	61	63	68	20	54	75	79	65	74	20	48	6F	6D	65	Bach Tuyet Home
0000011D	FF	06	81	17	84	09	75	15	FF	FF	FF	FF	FF	FF	FF	
0000012C	42	61	63	68	20	54	75	79	65	74	20	43	65	6C	6C	Bach Tuyet Cell
0000013B	FF	06	81	17	74	62	51	95	FF	FF	FF	FF	FF	FF	FF	
0000014A	42	72	69	61	6E	20	26	20	48	61	6E	6E	61	68	20	Brian & Hannah
00000159	52	06	81	03	31	39	65	41	FF	FF	FF	FF	FF	FF	FF	
00000168	42	72	69	61	6E	20	52	65	67	65	20	43	65	6C	6C	Brian Rege Cell
00000177	FF	06	81	03	51	37	35	50	FF	FF	FF	FF	FF	FF	FF	
00000186	42	72	69	61	6E	20	52	65	67	65	20	57	6F	72	6B	Brian Rege Work
00000195	FF	06	81	03	81	79	71	91	FF	FF	FF	FF	FF	FF	FF	

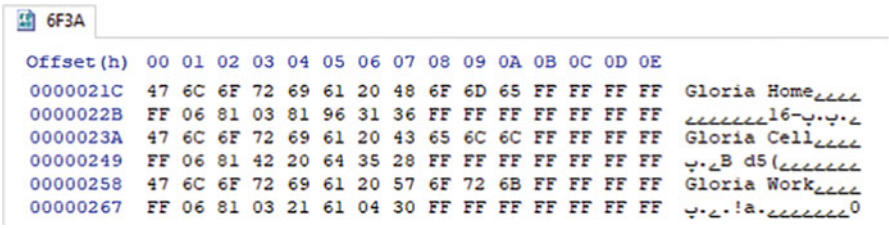
Let's take an example here,

6F3A

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	
000000F0	44	6F	63	74	6F	72	FF	FF	FF	FF	FF	FF	FF	FF	FF	Doctor
000000FF	FF	06	81	03	91	27	40	00	FF	FF	FF	FF	FF	FF	FF	

The contact name in this example is "Doctor" which can be decoded using ASCII code encoder. The next record is for the associated phone number of "Doctor". The first byte has the value of "06" which indicates the number of bytes are used for the contact number is 6 bytes. The next byte has the value of 0x81 which means this phone number either local or unknown. The rest used bytes are "03 91 27 40 00". To decode the number, we should remember that the number is stored in reverse nibble order thus the decoded number is "301-972 0400".

Another Example is given below:



We can observe that these three numbers are belong to the same person named "Gloria" where the first number is her "Home" number while the second one is her "Cell" phone number and finally the last one is her "Work" number. After decoding, her home, cell, and work phone numbers are "301-869 1363", "240-246 5382", and "301-216 4003", respectively.

17.5.3.2 Calls

Outgoing calls (LDN) is usually stored as EF 6F44. Also, some of SIM cards could store additional LDN data in EXT1 EF records. Same as ADN EF, each recent outgoing call data is saved in two records; the name and the associated number. However, some dialed number may not be associated with a name when the contact name is not saved.

17.5.3.3 SMS

Same as contacts and call logs data, SMS data could be stored in the phone and SIM card and has memory limit. SMS data is saved in EFs which usually stored in EF 6F3C. Also, SMS data is saved in fixed linear format of records or in transparent format. SIM card saves incoming text messages associated with its timestamps and phone number. Furthermore, the deleted messages could be found in SMS EF as long as there is no newer message which overwrites the deleted message because it is considered as free space. This is similar to hard disk dealing with deleted files.

In fact, one SMS message is saved through a set of data where each piece of data contains specific information of SMS such as service center information, contact number, timestamp, and message body. The first record in SMS set is the Short Messaging Service Center (SMSC) record. From this record, the investigator can extract different information such as if the SMS is read or not and service center number. The first byte of the record is used to indicate if the SMS was read (0x01) or deleted (0x00). The second byte indicates the number of bytes are used for the service center number. While, the next one is used to identify the number (0x81—unknown or 0x91—international number). The rest of bytes give the SMSC number in reverse nibble format. For example, if the extracted hex value of SMSC record is

equal to (0x 01 06 91 71 94 54 33 65) then we can conclude that this message was read and SMSC number is an international number (+1 749-453356).

Sender number data is directly followed the SMSC data. The first byte indicates the number length of sender number in decimal. The next byte is used to identify the number (0x81 - unknown or 0x91—international number). The following bytes of specific length give the contact number who sent the message.

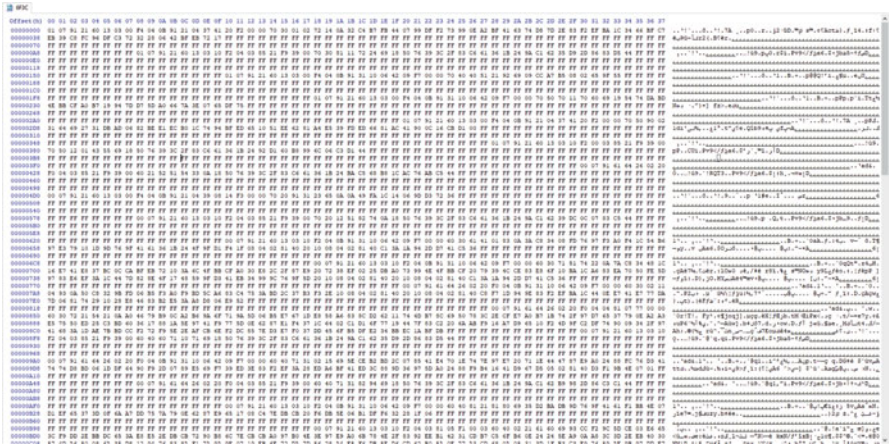
SMS timestamp data follows the phone number data after 2 bytes. However, the byte proceeds the timestamp data is used to indicate the encoding type of message body (0x00 means the default 7-bit encoding is used). The timestamp data (6 bytes) contains the date (Year, Month, Day) and the time (Hour, Minute, Second) when the SMS was sent, one byte per field. Also, the date (Year, Month, Day) and time (Hour, Minute, Second) are stored in reverse nibble format. For example, if the SMS timestamp data is equal to (0x 71 80 82 71 24 70) then it means the timestamp associated with the message is (August 28, 2017 05:42:07 PM).

Finally, the last record which stored after the timestamp data directly is used for the message content and must be decoded to extract and read the SMS body. However, the first byte indicates the length of the message body.

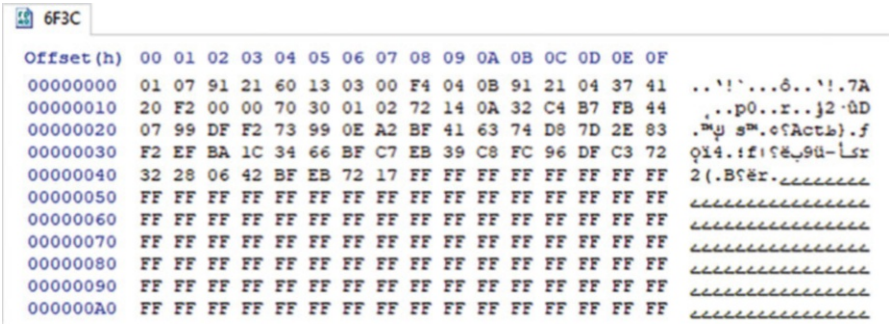
Now will give an example to decode SMS messages from the SIM model we have in our case study. SMS data are stored in EF 6F3C which located under MF 7F10 as shown in figure below:

caseSIM\FileDump_SIM\SIM_2G_3G SIM\SimDump\3F00\7F10\6F3C]

We use a Hex editor to open the file where all message are located as shown below:



Let's decode one message and its metadata to give an example here. The message data is given below:



- The first byte has the value "01" which indicates that this SMS was read.
- The second byte indicates the number of bytes are used for the service center number. In this case, it has the value of 7, which means the next 7 bytes "91 21 60 13 03 00 F4" determine the service center number.
- The next one "91" is used to identify the type of number which is international.
- The rest of 6 bytes "21 60 13 03 00 F4" give the SMS center number in reverse nibble format. Thus, the number of SMSC is "1 206-313 0004".
- The next bytes identify the sender number where "91" is used to identify the number (0x81 - unknown or 0x91 - international number).
- The following bytes "21 04 37 41 20 F2" of specific length give the contact number who send the message which is "1 240-731 4022".
- The byte "00" is used to indicate the encoding type of message body (0x00 means the default 7-bit encoding is used).
- The timestamp data is "70 30 01 02 72 14" (6 bytes) which contains the date and the time when the SMS was sent. As mentioned before, the date (Year, Month, Day) and time (Hour, Minute, Second) are stored in reverse nibble format. Thus, "70" means "2007" year, "30" means "March" month, and "01" means "10th" day. Also for the time, "02 72 14" is decoded to "20:27:41" in the form of (hh: mm:ss). Finally, we can observe that the message was sent in (March 10, 2007 08:27:41 PM).
- The last record is used for the message content (7-bit encoding). In our example, the SMS content data is "C4 B7 FB 44 07 99 DF F2 73 99 0E A2 BF 41 63 74 D8 7D 2E 83 F2 EF BA 1C 34 66 BF C7 EB 39 C8 FC 96 DF C3 72 32 28 06 42 BF EB 72 17". This data can be decoded using any simple 7-bit encoder. We use the encoder provided at the link (http://smstools3.kekekasvi.com/topic.php?id=288). The message content is "Don't forget to change your clocks forward 1 hour", as shown below:

USSD Entry/Display ◉ GSM 7bit packed ○ UCS2 ☐ Cell Broadcast (whole PDU)

```
C4 B7 FB 44 07 99 DF F2 73 99 0E A2 BF 41 63 74 D8 7D 2E 83 F2 EF BA 1C 34 66 BF C7
EB 39 C8 FC 96 DF C3 72 32 28 06 42 BF EB 72 17
```

(Padding as defined on GSM 03.38 version 5.6.1 (ETS 300 900) page 17) Convert >

Result

```
USSD/User Data without length information
Alphabet: GSM 7bit

Don't forget to change your clocks forward 1 hour.
Length: 50
```

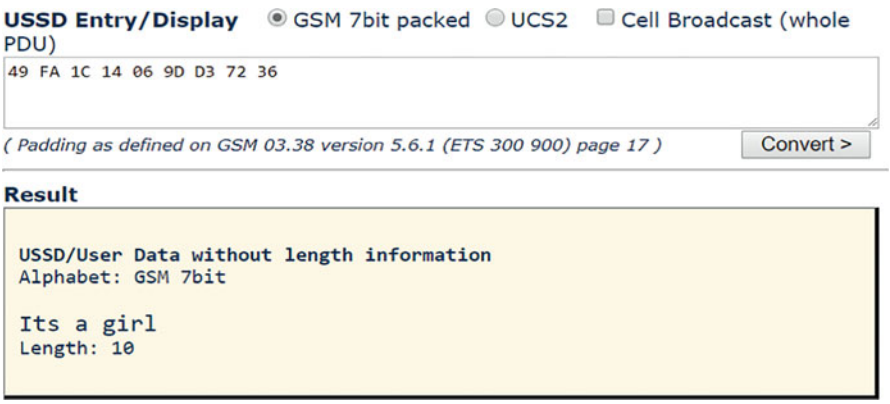
Another Example is shown below:

6F3C

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
000004D0	00	07	91	21	60	13	03	00	F4	04	0B	91	21	04	39	08	..'!`...ô..'!.9.
000004E0	14	F3	00	00	70	20	91	31	23	65	0A	0A	49	FA	1C	14	.ó..p '1#e..Iú..
000004F0	06	9D	D3	72	36	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	. Ór6ÿÿÿÿÿÿÿÿÿÿ
00000500	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000510	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000520	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000530	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000540	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ

From the first byte, we can know that this message is a deleted message but fortunately we can still recover it from the SIM card memory. Therefore, the capability of SIM card forensics to recover the deleted SMS is an advantage for investigation and collecting the evidences. Using the same step in previous example, we can obtain the followings:

- The SMS center number is international and the number is "1 206-313 0004".
- The sender number is also international and the number is "1 240-938 0413".
- The Timestamp data is "70 20 91 31 23 65". After decoding, we can know that the deleted message was sent in (February 19, 2007 01:32:56 PM).
- Using the 7-bit encoder, we can decode the content data "49 FA 1C 14 06 9D D3 72 36" and know the deleted message which is "Its a girl" as shown below:



17.5.3.4 System Data

As mentioned before, the system and network data are stored in the EF files. Four significant information can be extracted such as ICCID, IMSI, MSISDN. The identifiers for this information are stored in 0x2FE2 DF for ICCID, 0x6F07 EF for IMSI, and 0x6F40 EF for MSISDN. However, the ICCID, IMSI, and MSISDN numbers are stored in reverse nibble order. For instance, if the ICCID is equal to "89 31 04 10 10 16 01 87 10 55" then the actual value will be (0x 98 13 40 01 01 61 10 78 01 55).

ICCID which is the serial number of SIM card consists of 19–20 digits number including single check digit as a checksum. For above example, (89 310 410 10 160187105 5) is 20 digits ICCID where 89 refers to industry identifier of telecommunication; 310 refers to country code "US"; 410 refers to network code "AT&T"; 160187105 refers to the unique SIM serial number; 5 is the check digit.

IMSI is used to identify the subscriber on the network where network service provider uses this unique 15 digits number for each device benefits from the network. For example, 310 410 123456789 is 15 digits IMSI where 310 refers to country code "US"; 410 refers to network code "AT&T"; 123456789 refers to the unique subscriber ID number (Fig. 17.8).

In our case study, the identifiers for this information are stored in 3F00 0x2FE2 DF for ICCID, 0x6F07 EF for IMSI, and 0x6F40 EF for MSISDN as shown below:

ICCID data is stored in reverse nibble format in 19 digit number, where the extracted value is "98 10 62 20 02 10 52 95 95 F7" and can be read as "89 01 26 02 20 01 25 59 59 7". The first byte "89" refers to industry identifier of

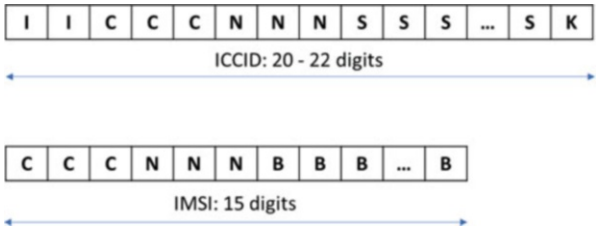


Fig. 17.8 ICCID and IMSI formats

telecommunication; “01” refers to country code “US”; “260” refers to network code “T-Mobile”; “0220012559597”, this remaining values refer to some account information such as account ID and checksum. We can check the extracted ICCID number using the SIM number analysis tool in this link which is provided by the International numbering plans (https://www.numberingplans.com/?page=analysis&sub=simnr).

caseSIM\FileDump_SIM\SIM_2G_3G SIM\SimDump\3F00\2FE2]

2FE2

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00000000	98	10	62	20	02	10	52	95	95	F7						

MSISDN number has the same encoding format as the contacts. So the first record is for the name “MSISDN1” and the next record stored the phone number of this SIM model. In our case, the MSISDN data is “07 81 21 04 39 08 24 F1” and this data can be decoded as following: The first byte “07” refers to number of bytes that are used for the phone number; “81” refers to type of number which is local; in reverse nibble format, the given number of SIM card is “1 240-938 0421”.

caseSIM\FileDump_SIM\SIM_2G_3G SIM\SimDump\3F00\7F10\6F40]

6F40

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	4D	73	69	73	64	6E	31	FF	FF	FF	FF	FF	FF	FF	FF	FF	Msisdn1
00000010	07	81	21	04	39	08	24	F1	FF	FF	FF	FF	FF	FF	FF	FF	.!.9.$ñ
00000020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
00000030	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
00000040	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
00000050	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
00000060	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
00000070	FF	FF	FF	FF	FF	FF	FF	FF									

Review Questions

1. Why SIM card forensics is important?
2. What uniquely identifies a subscriber on GSM cellular network?
3. What are PIN and PUK numbers?
4. List the types of file system identifiers associated with its memory addresses?
5. Mention at least five kinds of EF files (with its memory addresses and type of stored data)?
6. How to decode reverse nibble format?
7. Where and Why 7-bit encoding are used?
8. What are the four parts of SMS data that could be recovered from the SIM card memory?

9. Why some deleted SMSs could be found in the SIM card memory and not all?
10. What kind of information could be found in the system part of SIM card?

17.6 Practice Exercise

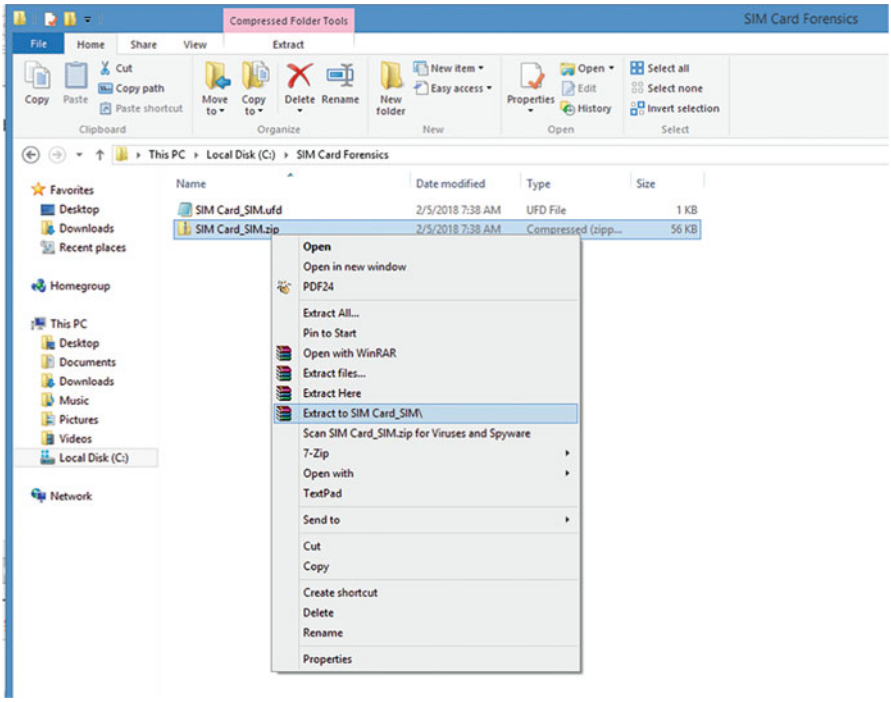
The objective of this exercise is to gain experience in using a hex viewer to interpret SIM Card and parse out data from the SIM card.

In the zipped file which contains all the data files used in the book, you will find a ch17 subfolder that contains all the files for this exercise. Make sure you copy and paste these files into a particular folder on your hard disk.

17.6.1 Setting Up the Exercise Environment

In order to begin this exercise, you need to prepare the following SIM card image:

1. Go to the folder that you put the files needed for Chap. 17.
2. Unzip the image file "SIM Card_SIM.zip" by selecting "Extract here by the same name" from the drop-down menu.

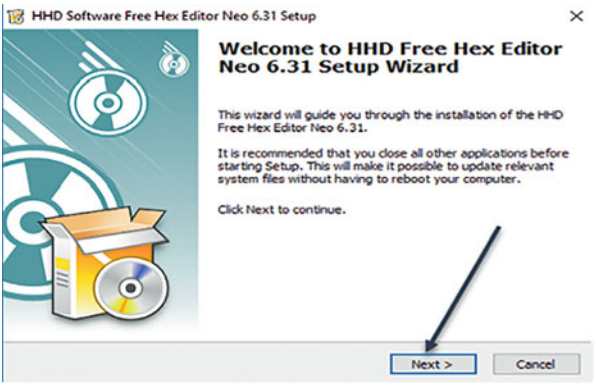


Also, you need to install a hex viewer. Note that there are many free hex editor tools available online. Here, we introduce two, one of them is WinHex and another one is Hex Editor Neo. You can use any one of them or any other Hex Editor.

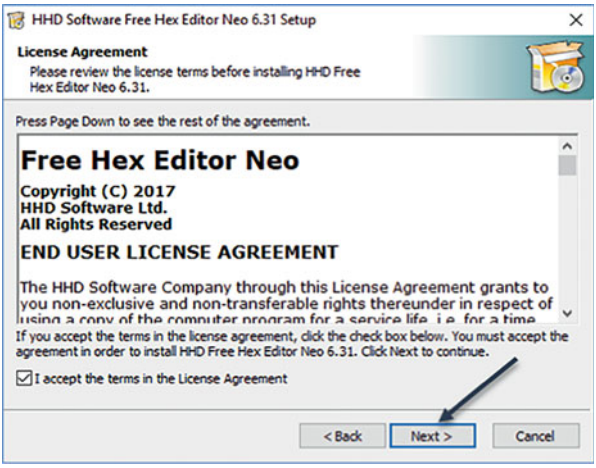
To show you how to do setup for example Hex Editor "Neo.exe" as follows:

Download this tool, go to https://www.hhdsoftware.com/free-hex-editor.

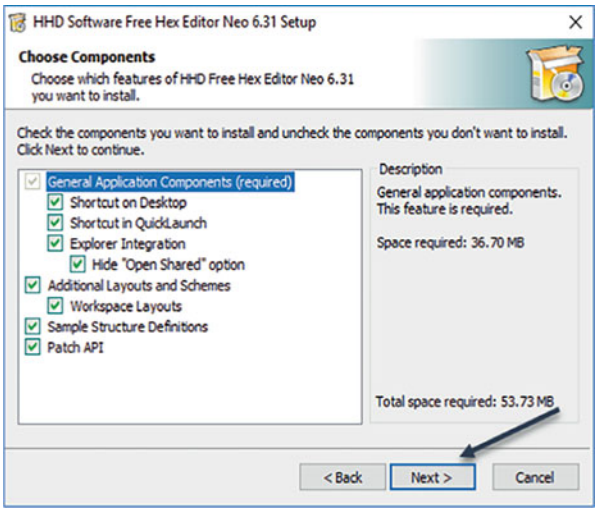
Install Hex Editor by running downloaded installer.



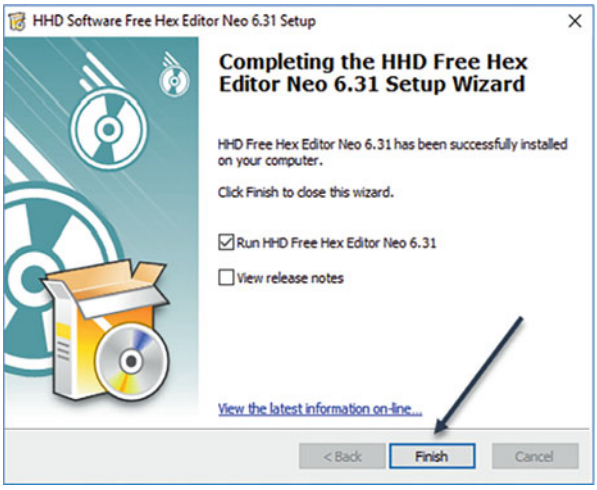
Read and accept the terms in the license agreement to continue installation.



Accept default options during the installation.



Click Finish to complete installation of Hex Editor Neo.

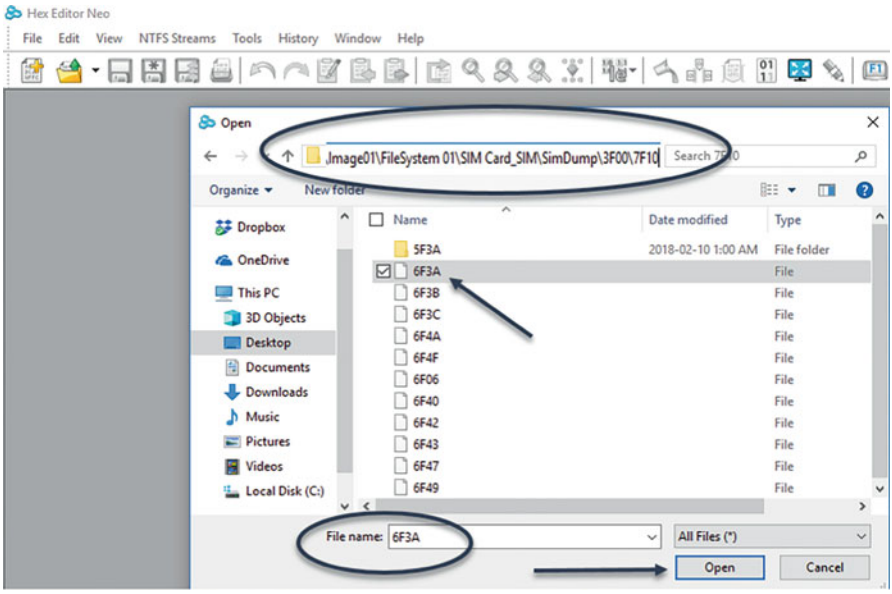


17.6.2 Exercises

Answer the following questions:

Part A: Contacts

Q1. Based on what you have learned so far, manually find the last contact and its associate number in the given image. Hint: All contacts are stored in EF 6F3A, shown below:



Q2. Find the number that belongs to "Hanco Car".

Part B: SMS

Q3. How many SMS messages in the file system dump.

Q4. Convert the components of the last SMS message in the file system dump. Make sure to decode each record within the message. HINT: The SMS messages are located in the 3F00/7F10/6F3C folder.

Q5. Determine whether there are some deleted SMS messages in the file system dump or not. HINT: After locating the SMS messages in the 3F00/7F10/6F3C folder, the initial or the first byte marks the status flag of SMS if its "active" will be (0x01), and "deleted" if it's (0x00). Note that many commercial tools cannot parse both messages at same time, but manually you can.

Part C: System Data

Q6. What are the equipment identifiers for this SIM card, ICCID and IMSI? Where and how did you recover this data in the file system dump?

References

1. SIM Card Forensics – Complete Forensic Analysis of SIM Cards Explained. http://www.dataforensics.org/sim-card-forensics/
2. https://www.cellebrite.com/
3. https://www.msab.com/
4. M. T. Abdelazim, N. AbdelBaki, A. F. Shosha. Digital Forensic Analysis of SIM Cards. 2016 International Conference on Security and Management (SAM'16).
5. Swenson C., Manes G., Shenoi S. (2006) Imaging and Analysis of GSM SIM Cards. In: Pollitt M., Shenoi S. (eds) IFIP International Conference on Digital Forensics 2005 - Advances in Digital Forensics.