# Forensics and SIM cards: an Overview

Fabio Casadei
Antonio Savoldi
Paolo Gubian
University of Brescia

## Abstract

Nowadays there are many tools for the extraction of data objects [SWGDE] from SIM cards; unfortunately, most of them are proprietary, or their use is restricted to law enforcement and this is contrary to the Daubert test for acceptability from the scientific community. In this paper, we present an open source tool for data objects extraction from SIM and USIM cards which is capable of extracting all observable memory and all the non-standard files that are found in every SIM card.

First, a description of the tool from a digital forensics perspective will be provided. Then, the technological background of the tool will be sketched. After that, the core algorithms will be described and explained. Then, motivations for the choice of an XML format for output will be given and the format described. In conclusion, the possible lines of evolution will be presented.

## Introduction

This paper will introduce the SIMbrush tool, a new tool developed for the Linux and Windows platforms, aimed at extracting the observable portion of the filesystem of a SIM card. The authors' intention has been to provide the Digital forensics community with an open source tool that could be tested, improved, and complemented with other tools to constitute the basis for creating a forensically sound platform for the digital investigation of SIM cards.

Although the last several years have seen a spike in the production of documents on the digital forensics topic, there is not much available material regarding GSM forensics in particular. Moreover, software tools aimed at this scope are difficult to examine because, even when they are not confidential, their source code is closed and no documentation on the internal functioning is provided by their publishers.

From a software point of view, only one tool for this purpose, in the authors' knowledge, is in the open source arena: TULP2G[1], a framework developed by NIS (the Netherlands Institute of Forensics) implemented in C# (the language of the .NET Microsoft suite). Its purpose is the extraction of data objects from mobile equipment and SIM cards, but only on Windows platforms. Other tools are either proprietary or their use is restricted to law enforcement personnel: for example, *Cards4Labs*[2], which is available to law enforcement agencies only, and *EnCase*[3]

[1] http://tulp2g.sourceforge.net
[2] http://www.forensischinstituut.nl
[3] http://www.guidancesoftware.com

which is an example of a commercial tool. More information about mobile forensics tools can be found at http://www.e-evidence.info/cellular.html.

SIMbrush makes use of the *pcsc* [PCSC] library to interface itself with smart card readers. This is a middleware capable of managing communications with a variety of readers from different manufacturers. Documentation and software are available at http://www.linuxnet.com

## Digital Forensics Perspective

Digital forensics is a very recent branch of information technology, which was been established as a scientific research area only in 2001. The definition of *Digital Forensic Science*, in fact, was formalized in [DFRWS 2001]:

> The use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations.

This definition highlights the main tasks, or *categories* of the digital forensics research. For each task, a set of subtasks, or *techniques* has been proposed. Categories may be viewed as main phases or steps of an investigative process, while techniques may be viewed as the actions that an investigator has to take to complete a phase of the investigative process. The investigative process itself has been defined in [DFRWS 2001] and is shown in Table 1.

According to this process, the SIMbrush tool can be placed in the imaging technologies group of techniques in relation to the preservation phase. That is, its mission is to extract from a SIM card, defined as the physical item [SWGDE], the information stored in it, to the widest possible extent, and to produce as output what is called a primary image, which can be subsequently used throughout the investigation instead of the physical item itself, which is secured as evidence and never used any more. The primary image itself is seldom used, because it acts as a master from which working copies can be created for the investigators. All the precautions taken for the physical item are used also for the primary image acting as the master.

This method of work can be applied if the primary image, from which digital evidence is derived, maintains its digital integrity throughout the entire process. Digital evidence integrity is defined as the property whereby digital data has not been altered in any manner since the time it was created, transmitted, or stored.

Besides digital integrity, an imaging tool is required to produce a forensically sound digital evidence, that is, a copy which contains, as an absolute minimum, the full operating area of information stored in all active semi-permanent storage [Bates 1999]. It is clear that such a requirement cannot be satisfied when the physical item is a SIM card, because trying to extract such a copy could harm the physical item itself, resulting in an investigation that is not forensically sound (that, is adherent to the principles and best practices of Digital Forensic Science) because digital integrity is not ensured.

That said, it is interesting to explain why SIM card investigation is valuable and what pieces of information we might expect to be extracted from a SIM. The first aspect is the fact that the subscriber of a mobile telephony system essentially wants a means to communicate: this implies an exchange of information (voice and data) potentially useful for investigations. Second, every mobile telephone system traces the position of handset terminals to exchange information between the mobile part and the fixed part of the system. Since the subscriber needs the handset to transmit and receive information, he/she will bring the handset in his/her pocket, precluding the use of it from other people. Therefore, in most cases, there is a univocal relationship between the user and his/her handset, and this is very interesting from an investigator's point of view. Note that this marks a big difference from fixed telephone systems, where a terminal identifies only a geographical location (home, business, etc.) but not the users of that terminal.

| Indentification | Preservation | Collection | Examination | Analysis | Presentation |
|---|---|---|---|---|---|
| Event/Crime Detection | Case Management | Preservation | Preservation | Preservation | Documentation |
| Resolve Signature | Imaging Technology | Approved Methods | Traceability | Traceability | Expert Testimony |
| Profile Detection | Chain of Custody | Approved Software | Validation Techniques | Statistical | Clarification |
| Anomalous Detection | Time Synchronization | Approved Hardware | Filtering Techniques | Protocols | Mission Impact Statement |
| Complaints |  | Legal Authority | Pattern Matching | Data Mining | Reccomended Countermeasure |
| System Monitoring |  | Lossless Compression | Hidden Data Discovery | Timeline | Statistical Interpretation |
| Audit Analysis |  | Sampling | Hidden Data Extraction | Link |  |
| Etc. |  | Data Reduction |  | Spatial |  |
|  |  | Recovery Techniques |  |  |  |

**Table 1 - Investigative process for Digital Forensic Science (adapted from [DFRWS 2001]).**

The second aspect is related to the kind of information the SIM stores:

- *Information about the subscriber*: the SIM stores the *International Mobile Subscriber Identity (IMSI)*, which is a unique identifier for each subscriber in the system, as specified in [ETSI EN 300 927 v5.4.1]. Information about preferred languages could be of help in determining the subscriber’s nationality. *Mobile Station ISDN (MSISDN)* could be used to retrieve the calls originated by the user towards other phone numbers.

- *Information about acquaintances of the subscriber*: subscribers can maintain a list of the numbers they call or they are called from more frequently or that are of importance to them. Furthermore, subscribers could be registered to one or more groups of subscribers if so called “multicalls” are enabled.

- *Information about SMS traffic:* it is possible to read SMS messages sent and received by the subscriber out of the SIM card, and to see for every received SMS whether it has been read or not.

- *Information about subscriber's location*: the SIM stores the last area where the subscriber has been registered by the system.

- *Information about calls*: the last numbers dialed are stored in a file in the SIM filesystem. The key used to encrypt the last call is stored there too.

- *Information about the provider*: it is possible to extract the provider name and the mobile network commonly used for communications, along with mobile networks that are forbidden to the subscriber.

- *Information about charge*: some charge information could be stored in the SIM.

- *Information about the system*: every SIM card has a unique ID stored in it. All the services to which the subscriber could be enabled, along with the actual status of abilitation, are stored in the SIM.

Many more data are stored in the SIM, but those just mentioned suffice to justify an effort to extract them. By looking carefully at Table 1, it is possible to see that, from identification to presentation, the execution of at least one technique from every category seems to be essential in every investigation. That is, some kind of identification is necessary for starting every investigation; preservation must be carried out for every piece of physical item; every data object must be examined; a timeline and chain of evidence must be built and a presentation must be scheduled. Therefore, identification, preservation, examination, analysis and presentation are "operational" categories. On the other hand, collection is a "management" category, as it involves considerations about the way to manage data objects (choosing a preservation strategy, hardware, software, methods, ways to reduce the amount of data without losses and to eventually recover it; assigning legal authority). Hence, the design of a preservation tool like SIMbrush must take into account that the real users of its output will be the people at the examination stage. Implications of these considerations will be clarified in the section titled, "Interfacing SIMbrush with other parts of the investigative process."

## Technological Background

In this section, the technology involved in the process of extraction of data objects from a SIM card is described. Mainly, two technologies are involved: mobile telephony systems and smart cards.

Mobile telephony systems

SIMbrush can be used to extract data objects from any SIM card used in a GSM system. Extraction of data from subscriber modules belonging to other systems (such as D-AMPS, CDMA and PDC) is out of its scope. This design choice was taken based on the fact that GSM, and its successor UMTS, besides being the most widespread worldwide, are rapidly increasing their penetration in the markets where other systems are prevailingly used.

The *Global System for Mobile Communications (GSM)* is a very complex system, specified by a huge corpus of standards, issued by the *3rd Generation Partnership Project (3GPP)* and adopted by the *European Telecommunications Standards Institute (ETSI)* for the European countries. It is not possible here to describe the GSM telephone system: for that purpose, standards are the best resource; instead, only a brief presentation of the SIM card will be provided here.

The GSM system can be very roughly divided into two parts: the infrastructure part and the end user part. The infrastructure part involves all standard network devices and protocols among them (with levels of abstraction ranging from the physical level to application level), along with their internal hardware and software features, which are not specified by any standard. The end user part is every component of the system that is normally used by a subscriber. Specifically, the end user part coincides with the *Mobile Station (MS)*. The MS can be further partitioned into the *Mobile Equipment (ME)*, which is the handset or mobile phone, and the *Subscriber Identity Module (SIM)*. The interface between the SIM and the ME is specified by [ETSI TS 100 977 v8.12.0] and this is where SIMbrush “acts”.

Over the years, the GSM mobile system has been upgraded with several subsystems, the most notable of which is GPRS. Today, GSM lives side by side with its successor, the *Universal Mobile Telecommunications System (UMTS)*. The main difference between GSM and UMTS resides in the increased bandwidth for data exchange, which enables a lot of services that could not be implemented before. However, organization of the network and MS is almost the same, and so are the roles of the components of the *User Equipment (UE)*. The UE consists of the ME and the *User Services Identity Module (USIM)*. Standards state that, for interoperability between the old system and the new system, every UE must operate also with a GSM SIM and every USIM must operate correctly if inserted into a GSM MS. This is good news for SIMbrush, for the tool can operate also with USIMs. The only differences will be the greater amount of data and the presence of new standard (for example *Multimedia Message Service (MMS)*) and non-standard files.

Smart cards

The previous subsection explained where the SIM resides in the GSM system, but did not clarify what exactly a SIM is; this is the purpose of this subsection. The relationship between SIMs and smart cards is an “is a” relationship. That is, every SIM is a smart card or the set of all existing SIMs is a subset of the set of all existing smart cards. Smart cards are standardized by ISO; in particular, SIMs are contact (as opposed to contactless) smart cards, which are specified in [ISO 7816]. The principal concern of smart card design is the security of the data stored in it. The term “security” can be further specified by expanding it into its four main meanings of confidentiality, authentication, non repudiation, and integrity. As a consequence of a SIM being a smart card, the aforementioned requirements are used to accomplish the following tasks:

**Confidentiality**: user privacy must be guaranteed by encrypting voice and data traveling over the air. The keys of cryptographic algorithms that implement this feature reside in the SIM.

**Authentication**: no unauthorized user should be able to access the system. The keys of the authentication algorithms reside in the SIM.

**Integrity**: no user should be able to alter the data within the SIM to implement frauds, for example by increasing the charge on a prepaid SIM or by enabling restricted services without paying for them.

**Non repudiation**: the sender can verify that a certain recipient has received a particular message, which means that the message has binding force.

A smart card can be viewed as a safe containing data. As a safe, it is very well armored against every unauthorized or unforeseen access. A very important fact that must be taken into consideration is that, just as an attempt of intrusion into a safe protected by a security system could lead to an alarm, tampering attempts with a smart card could lead to an irreversible blocking of the card; this block can only be resolved by substituting it with a new smart card issued by the same provider. From a forensics perspective, this leads to the conclusion that no sound forensic investigation can be carried out using tools that try to force anomalous behavior on the part of the SIM or which require a physical manipulation of it[4].
This is why SIMbrush does not make use of any of these “black hat” techniques; instead, it interfaces with the SIM in the standard way. As the only information that a smart card offers to the outside world is the data inside its filesystem, SIMbrush tries to extract it.

A smart card's filesystem is stored in an internal EEPROM, protected by the security features of the card. It has a hierarchical tree structure, with a root called *Master File (MF)*. As in many other filesystems, there are two classes of files: directories, called *Dedicated Files (DF)* and files, called *Elementary Files (EF)*. They could be viewed as the nodes and leaves of a tree, respectively. The MF is a DF. The main difference between a DF and an EF is that a DF contains only a *header*, whereas an EF contains a header and a *body*. The header contains all the meta-information that quantitatively relates the file to the structure of the filesystem (available space under a DF, number of direct children, length of a record, etc.) and security information, whereas the body contains information related to the application for which the smart card has been issued. Depending on the structure of the body, four types of EF are possible in a smart card’s filesystem:

[4] For example, a tool exists that can recover the authentication key, called *Ki*, from a SIM card, by cryptanalysis of a batch of responses to the RUN GSM ALGORITHM command from the SIM itself. However, there is a built-in upper limit to the number of times this command can be issued: if, during this analysis, the tool exceeds this number of attempts, the SIM becomes irreversibly blocked.

**Transparent EF**: these files are organized as a sequence of bytes. It is possible to read all or only a subset of their contents by specifying a numeric interval.

**Linear-fixed EF**: the atomic unit for these files is the record, instead of the byte. A record is a group of bytes that have a known coding: every record of the same file represents the same kind of information. In a linear-fixed EF, all the records have the same length.

**Linear-variable EF**: same as linear-fixed EF, but here a the length may vary from one record to the other.

**Cyclic EF**: these files implement a circular buffer where the atomic unit of manipulation is the record. Therefore, the concepts of first and last are substituted by those of previous and next.

SIM cards, which are a proper subset of smart cards, do not allow linear-variable EFs, implementing only transparent, linear-fixed and cyclic EFs. Every file is unambiguously identified by its *ID*, which acts as the name of the file. No two files in the whole filesystem can have the same ID. The operations allowed on the filesystem are coded into a set of commands that the *interface device (IFD)*, which is the device capable of interfacing with a smart card and setting up a communication session, issues to the smart card, and then waits for responses. The IFD acts therefore as the *master* and the smart card as the *slave*. This is different in so called *proactive smartcards*, which are capable of issuing commands to the IFD, but SIMbrush does not support them and research has still to be done to understand whether this behavior could be useful in a forensic environment in general and in an imaging technologies environment in particular. The aforementioned commands, by means of which it is possible to interact with a SIM card's filesystem, are:

**SELECT**: this command, which is fundamental to SIMbrush, selects a file for use and makes the header of that file available to the IFD;

**STATUS**: has the meaning of a SELECT with MF as argument;

**READ BINARY**: reads a string of bytes from the current EF;

**UPDATE BINARY**: updates a string of bytes in the current EF;

**READ RECORD**: reads one complete record in a record-formatted file;

**UPDATE RECORD**: updates one complete record in a record-formatted file;

**SEEK**: searches the records of a record-formatted file for the first record which starts with the given pattern;

**INCREASE**: adds the value passed as a parameter by the IFD to the last increased/updated record of the current cyclic EF and stores the result in the oldest increased/updated record. It is used for incrementing time or charge information;

**GET RESPONSE** : in SIM cards, if some data is to be communicated from the smartcard to the IFD after a command, it is the IFD itself that has to request it, using this command.
What is important to note is that there is no command to eliminate or create files. No command to quickly browse the filesystem is available, either;

Those mentioned are the most important commands of a SIM card's operating system and have been reported here for completeness. In SIMBrush's core algorithm, only the SELECT and GET RESPONSE commands are used, thus preserving all data in the filesystem; indeed, all data are extracted without modification, in read only access mode.

Smart cards can be compared with safes. Like safes, they implement many security systems to protect their content: data. One of such security system is the *access conditions*. A short introduction to access conditions in a SIM card is provided in the following. If all the aforementioned commands were executable by anyone at any time, all sensible data stored in the filesystem would be readily available to the external world. Access conditions are constraints to the execution of commands which filter every execution attempt to make only those people who are authorized served, and only for the duration of their authorization. There are 16 access conditions, shown in Table 2, and every file in the filesystem has its own specific access conditions for each command. Access conditions are organized in levels, but this organization is not hierarchical: that is, authorization for higher levels does not imply authorization for lower levels.

| Level | Access condition |
|---|---|
| 0 | ALWays |
| 1 | CHV1 |
| 2 | CHV2 |
| 3 | Reserved for GSM future use |
| 4 to 14 | ADM |
| 15 | NEVer |

**Table 2 - Access conditions and level coding for SIM cards.**

Briefly, the meaning of these access conditions is:

**ALW**: the command is always executable on the file;

**CHV1**: the command is executable on the file only if one among *Card Holder Verification 1 (CHV1)* code or *Unblock Card Holder Verification 1 (UNBLOCK CHV1)* code has been successfully provided;

**CHV2**: same as CHV1, but using *Card Holder Verification 2 (CHV2)* code or *Unblock Card Holder Verification 2 (UNBLOCK CHV2)*;

**ADM**: allocation of these levels is a responsibility of the administrative authority which has issued the card: the card provider or the telephony provider which gives the card to its subscribers.

**NEV**: the command is never executable on the file;

### Description of the Core Algorithm

As said before, the underlying problem is that no command exists to quickly browse the filesystem, such as the *dir* or *ls* commands in the DOS or Linux operating systems. The structure must therefore be deduced. Reference standards ([ETSI TS 100 977 v8.12.0]) help in the solution of this problem.

First, the standards say that no two files can have the same ID (filename) and there are a lot of files that have a standard ID; for example, 3F00 identifies the master file of a SIM card's filesystem. Second, the SELECT command may be issued with any file as argument, with no restrictions. This leads to the opportunity to "brush" the ID space by issuing a SELECT command for each valid "name", from 0000 to FFFF, obtaining either a warning from the SIM when the ID does not exist (that is, the file with that name is not present in the filesystem of the SIM under examination), or the header of the file (that is, of the file with that name present in the filesystem of the SIM under examination) when it does.

With these two pieces of information, it seems possible to obtain the header of every file present in the filesystem of the SIM with a single scan of the ID space. This is only partially true. In fact, the standards define the concepts of *current file* and *current directory*. The current file is simply the last successfully selected file. The current directory is the last successfully selected DF, or the parent DF of the current file, if the current file is an EF: it defaults to MF and may coincide with the current file. At any time, there are exactly a current file and a current directory. The current directory determines which files are selectable or not, according to the following rules:

1. MF is selectable no matter what the current directory is;
2. The current directory is always selectable;
3. The parent of the current directory is selectable;
4. Any DF which is an immediate child of the parent of the current directory is selectable;
5. Any file which is an immediate child of the current directory is selectable;

It is possible to associate a set of files and directories to each of the above mentioned groups:

1. The first set is called MF_SET. It has a single element: the MF.
2. The second set is called CURRENT_SET. It has a single element: the current directory.
3. The third set also has a single element: the parent of the current directory. It is called the PARENT_SET.
4. The fourth set has the obvious name of DF_BROTHERS_SET.
5. The fifth set is called SONS_SET.

At any time, selection must obey the rules of selection just explained: this can be formalized by introducing another set, which represent, given the current directory, all the files and directories on which issuing a SELECT command results in a successful response from the SIM if and only if the file exists:

$$\begin{aligned} \text{SELECTABLE\_SET} = \ & \text{MF\_SET} \cup \\ & \text{CURRENT\_SET} \cup \\ & \text{PARENT\_SET} \cup \\ & \text{DF BROTHERS\_SET} \cup \\ & \text{SONS\_SET} \end{aligned} \tag{1}$$

It is worth pointing out that, between the set of every possible current directory and the set of every possible SELECTABLE_SET, the relation of selection is univocal. This fact leads to the important result that, given a current directory, all its direct children are unambiguously characterized by:

$$\begin{aligned} \text{SONS\_SET} = \text{SELECTABLE\_SET} \ \backslash \ & \\ (\text{MF\_SET} \cup & \\ \text{CURRENT\_SET} \cup & \\ \text{PARENT\_SET} \cup & \\ \text{DF\_BROTHERS\_SET}) & \end{aligned} \tag{2}$$

The above relation is important because it makes it possible to reconstruct the entire filesystem tree contained in a SIM card, even without commands to explicitly explore it. More precisely, at this stage the structure of the entire filesystem has been reconstructed, and for each file the header has also been extracted. However, the interesting part of the filesystem resides in the body of EFs; extracting this information is subject to access conditions limitations. In its present stage, SIMbrush is able to extract the body of those files whose access conditions are ALW and CHV1/CHV2, the second case being possible only if the appropriate codes are provided. An attack against these codes, even if possible in some way, is not acceptable from a digital forensics point of view, as stated in the first section.

To clarify the concepts illustrated, it is useful to explain how SIMbrush reconstructs the filesystem of a SIM, by simulating its behavior with an example. The starting point is MF, because this is the default current directory of a SIM card. This initial situation is shown in step 0 of Table 3.

The key point is that, at this stage, the MF_SET is known and coincides with MF, the CURRENT_SET is also known and coincides with MF, and PARENT_SET and DF_BROTHERS SET are empty sets because MF is the root of the filesystem tree. Under these conditions, Equation (2) becomes:

$$\text{SONS\_SET} = \text{SELECTABLE\_SET} \ \backslash \ \text{MF\_SET} \tag{3}$$

Step 0 is completed. Step 1 starts with the determination of the sets of interest for the first child of MF, namely DF 7F10. MF_SET is known and coincides with MF, CURRENT_SET is also known and coincides with DF 7F10, PARENT_SET is known and coincides with MF and DF_BROTHERS_SET is also known and coincides with DF 7F4F. After the extraction of SONS_SET from

SELECTABLE_SET, step 1 is completed. No DF is present among the sons of DF 7F10 and so recursion for this branch stops at this depth level. Step 2 will proceed in the same way but on DF 7F4F, as shown in Table 3. Figure 1 shows the SIM card filesystem reconstructed. Obviously, all information necessary to reconstruct the links between nodes is indirectly obtained from direct child relationships and recursion. It is important to note that browsing the entire file ID space, while slowing the process of extraction, allows us the extraction of non standard files which otherwise would be unreachable. From a Digital forensics perspective, this is an advantage that largely overcomes the overhead in computation time.

| Step | 0 | 1 | 2 |
|---|---|---|---|
| CURRENT_SET | {3F00} | {7F10} | {7F4F} |
| MF_SET | {3F00} | {3F00} | {3F00} |
| PARENT_SET | {} | {3F00} | {3F00} |
| DF_BROTHER_SET | {} | {7F4F} | {7F10} |
| SELECTABLE_SET | {3F00,7F10, 7F4F} | {3F00,6F3A, 6F3B,…,6F4B, 7F10,7F4F} | {3F00,6F16, 6F1C,6F1E, 7F10,7F4F} |
| SONS_SET | {7F10,7F4F} | {6F3A,6F3B, …,6F4B} | {6F16,6F1C, 6F1E} |

**Table 3 - Evolution of the core algorithm in reconstructing the example filesystem.**

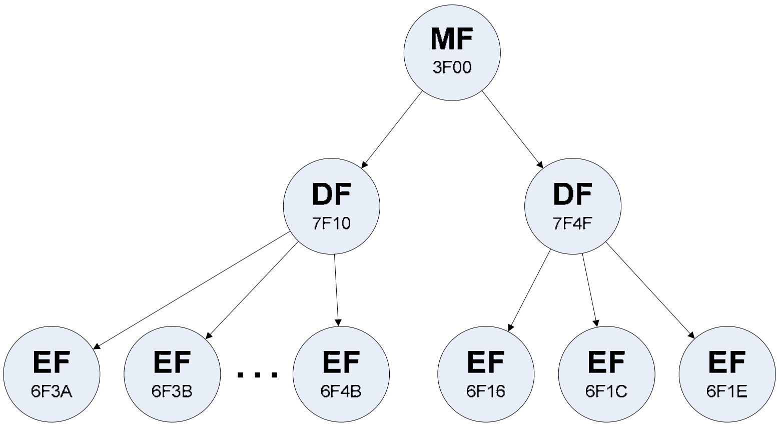

**Figure 1 – Example SIM filesystem.**

It is interesting to analyze the algorithm’s pseudo-code to understand, in more detail, the filesystem reconstruction algorithm. The main procedure will build a binary tree, which is a suitable data structure for SIM card’s data, capable of containing all filesystem data. From reference standard [ETSI TS 100 977 v8.12.0] it is known that a SIM card’s filesystem is organized as an n-ary tree structure, but considering equivalence between n-ary and binary trees, the latter has been chosen for our implementation.

```
Procedure Build_Tree
     Expand_DF( PARENT_SET = 0,
                CURRENT_SET = {MF},
                DF_BROTHERS_SET = 0 );
End
```

```
Procedure Expand_DF( PARENT_SET: NODE,
                     CURRENT_SET: NODE,
                     DF_BROTHERS_SET: NODE )
     Select(CURRENT_SET);
     SELECTABLE_SET = Brush(CURRENT_SET);
     SONS SET = SELECTABLE_SET     \
               (MF_SET             U
                CURRENT_SET        U
                PARENT_SET         U
                DF_BROTHERS_SET );
     For each node N belonging to SONS_SET,
          Place_in_tree(N);
     If N equal DF Then
          Expand_DF(
               PARENT_SET = CURRENT_SET,
               CURRENT_SET = N,
               DF_BROTHERS_SET = DF_BROTHERS_SET \ {N} );
End
```

For added clarity, each element of the pseudo-code is described below:

**Build_Tree**: is the procedure which initializes the parameters of  recursive function.

**Expand_DF**: is the recursive function that, starting from the filesystem's root, brushes the ID space, searching all existing EFs and DFs and applying the previous relation to find all sons of current node, which are placed in a binary tree data structure. For each son, if this is an EF then it is placed in the data structure; otherwise, if it is a DF then the **Expand_DF** function acts recursively, updating all interested sets.

**NODE**: is the main data structure to store all filesystem's data.

**Select**: sends a SELECT command to the SIM card.

**Place_in_tree**: updates the binary tree data structure of a SIM by adding a new DF.

**Brush** selects a Dedicated File, passed as the argument, which becomes the current DF, and brushes the entire file ID's space, obtaining the SELECTABLE set as a result related to such DF. A flowchart of the procedure is reported in figure 2.

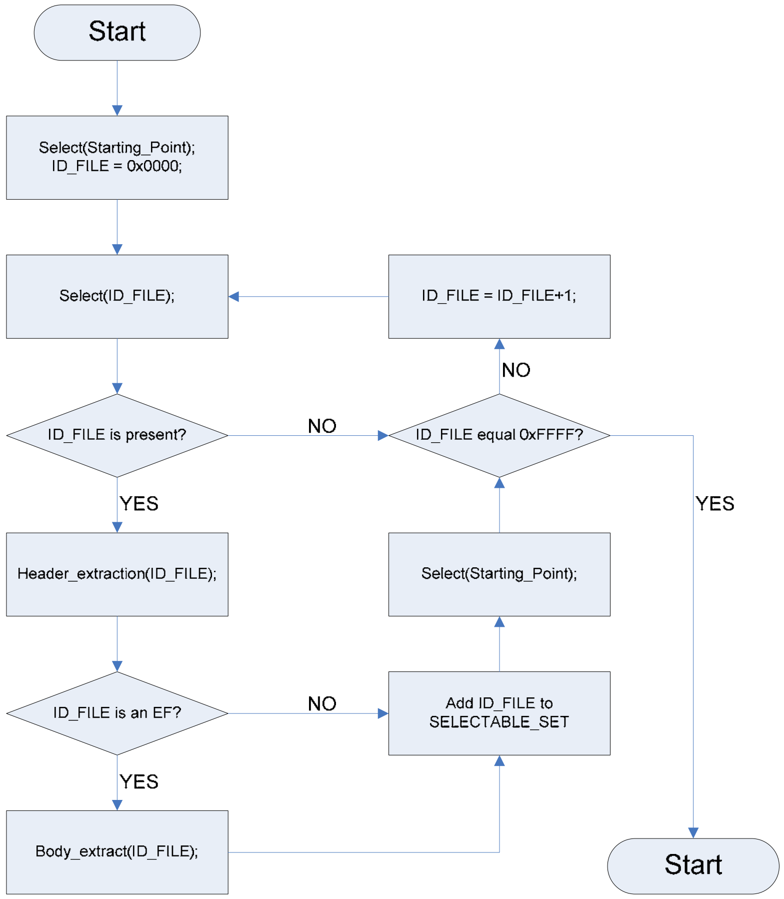

**Figure 2 – Flow chart of the Brush procedure.**

### Interfacing SIMbrush with other parts of the investigative process

As explained earlier, the SIMbrush tool pertains to a well defined area of competence, corresponding to the *imaging technologies* technique of the *preservation* category. To be useful, the tool must provide some interface to other tools of the *examination* category that can process its output. To do that, it is necessary to figure out how the characteristics of the user of such tools will change, based on the category of the tool itself.

Users of tools which belong to the preservation category will be persons whose job is to run the tool against a physical item and, respecting some standard protocol issued by the collection staff, produce a primary image master with its digital integrity preserved, which will be used for archiving and generation of sound work copies to be passed on to examination experts. Therefore, the output of a preservation tool should be designed to facilitate the production of examination tools usable by the examination experts.

The job of examination experts is to filter the contents of an uninterpreted image of the physical item, interpreting raw data to find information of interest for the investigation, and making this subset available to the analysis experts. This way, huge amounts of information (most of it of no interest to the investigation) can be reduced to an amount affordable by analysis experts, who are then able to compose these pieces of truth (often coming from different sources) in a consistent line of time, supported by an appropriate chain of evidence. Therefore, examination experts perform the task of scanning all pieces of information present on the data objects and selecting those of interest. The best paradigm to help the accomplishment of this task is that of *navigation*. Examination experts need a tool that supports searches, selections and the following of the hierarchical structures of the information. Fortunately, in the Web era, navigation is a well known paradigm that is supported by consolidated tools.

The idea, borrowed from Brian Carrier's *Autopsy Forensic Browser*[5], is to make digital evidence surfable inside a Web browser.

SIMbrush adheres to this line of thought, producing an XML formatted file as output. The goal is to arrive at the definition of a standard XML-derived language, with the definition of a proper DTD, capable of representing the information extracted from a SIM card. It is important to note that a different language is needed to represent SIM data at different levels of abstraction. In the authors' opinion, two languages would suffice: one for raw data representation and the other for interpreted data representation. Relating data present in the SIM with other data of interest (for example, subscriber name and address) is the responsibility of the analysis category and therefore out of the scope of preservation or examination tools. The following XML template shows an informal prototypal definition of a language to represent raw data extracted from a SIM card. SIMbrush uses this language to format its output.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE simcard SYSTEM "sim_dtd.dtd">
<simcard level="raw">
  <atr>...</atr>
    <mf>
      <header>...</header>
        <ef>
          <header>...</header>
          <body>
            <sw>...</sw>
            <content>...</content>
          </body>
        </ef>
        <ef>
        ...
        </ef>
        <df>
          <header>...</header>
            <ef>
              <header>...</header>
              <body>
                <sw>...</sw>
                <content>...</content>
              </body>
            </ef>
            <ef>
              <header>...</header>
              <body>
                <record>
                  <sw>...</sw>
                  <content>...</content>
                </record>
```

[5] http://www.sleuthkit.org

```
        <record>
         <sw>...</sw>
         <content>...</content>
        </record>
       </body>
      </ef>
    </df>
    <df>
    ...
    </df>
  </mf>
</simcard>
```

## Results and Future Work

SIMbrush has been implemented in ANSI C to maximize portability and to achieve platform independence. It supports two main operating modes: extraction of all standard and non standard files or extraction of standard files only. The main difference between the two modes is in the acquisition time, ranging from about one hour and twenty minutes for a complete dump to about one minute for an extraction of standard files only.

As previously mentioned, SIMBrush uses the *pcsc* library to interact, in a standard way, with any type of SIM/USIM card and, consequently, to extract all observable contents from its filesystem. For testing purposes, it has been used on a Toshiba laptop with the Suse Professional 9.3 distribution of Linux and an Athena smart card reader, but any PC/SC [PC/SC] (the international reference standard for interface devices) compliant smart card reader can be used.

SIMbrush has been tested against several SIM/USIM cards, from older 8KB EEPROM GSM SIMs to 128KB GSM/GPRS SIMs and USIMs; all tested cards, about 20, are from European telephony providers, such as TIM, Vodafone, Omnitel, Radiolinja and H3G. Table 4 reports some data about the tested SIM cards showing data such as provider, country of card issuance, EEPROM size, type of GSM phase (2, 2.5 or 3), status of card such as active, non active or blocked and, finally, types of services that can be used.

Brushing time is directly proportional to the number of DF present in the filesystem. If T is the total brushing time in seconds and N is the number of DF of the filesystem, including the MF, the following relation holds:

$$T = K * N \qquad (4)$$

where K is a constant whose value has been empirically determined to be about 1200 seconds (that is 20 minutes). This high value is mainly due to the fact that the protocol used at the SIM-IFD interface imposes a bit rate of 9600 bps. All commands and responses are transmitted over this single serial line in half-duplex mode and so, the file ID space composed of 65536 elements (from 0000 to FFFF), 131072 among commands and responses, each of which several bytes long, must travel over this slow interface.

| # | Provider | Country | EEPROM | GSM Phase | State | Services |
|---|---|---|---|---|---|---|
| 1 | TIM | Itlay | 8KB | 2 | Not active | Base services |
| 2 | TIM | Itlay | 16KB | 2+ | Active | Base services |
| 3 | OmniTel | Itlay | 16KB | 2+ | Active | Base services |
| 4 | Vodafone | Itlay | 32KB | 2+ | Blocked | Base services + GPRS |
| 5 | Wind | Itlay | 64KB | 2+ | Active | Base services + GPRS + eMLPP |
| 6 | TIM | Itlay | 128KB | 2+ | Active | Base services + GPRS |
| 7 | Radiolinja | Finlandia | 128KB | 2+ | Active + GPRS | Base services |
| 8 | H3G | Italy | 128KB | 3 | Active + UMTS | Base services + UMTS |

**Table 4 – List of some SIMs/USIMs analyzed with SIMBrush.**

| ID | File type | READ | Structure | Father |
|---|---|---|---|---|
| 0000 | EF | NEV | TRANSP | MF |
| 0002 | EF | ALW | TRANSP | MF |
| 0005 | EF | ALW | TRANSP | MF |
| 0100 | EF | NEV | TRANSP | MF |
| 0200 | EF | NEV | TRANSP | MF |
| 2F20 | EF | ADM | LINFIX | MF |
| 2F30 | EF | ADM | LINFIX | MF |
| 2F31 | EF | ADM | LINFIX | MF |
| 2F32 | EF | ADM | LINFIX | MF |
| 2F33 | EF | ADM | LINFIX | MF |
| 2F34 | EF | ADM | LINFIX | MF |
| 2FEE | EF | NEV | TRANSP | MF |
| 2FEF | EF | NEV | TRANSP/LINFIX | MF |
| EECF | EF | ALW | TRANSP | MF |
| 7F4F | DF | - | - | MF |
| 0005 | EF | ADM | TRANSP | DFGSM |
| 0006 | EF | ADM | TRANSP | DFGSM |
| 6F1B | EF | NEV | TRANSP | DFGSM |
| 6F16 | EF | NEV | TRANSP | 7F4F |
| 6F1C | EF | NEV | LINFIX | 7F4F |
| 6F1E | EF | NEV | LINFIX | 7F4F |

**Table 5 – List of all non standard files extracted from a VODAFONE SIM.**

Such a time penalty is largely compensated by the advantages of the extraction. Besides finding every file that is specified by standards (and every different batch of SIM cards issued by a mobile telephone network provider may implement a different subset of the standards), in fact, the tool is capable of finding a lot of non standard

EFs and DFs, probably containing information that complements what standards prescribe with further details.

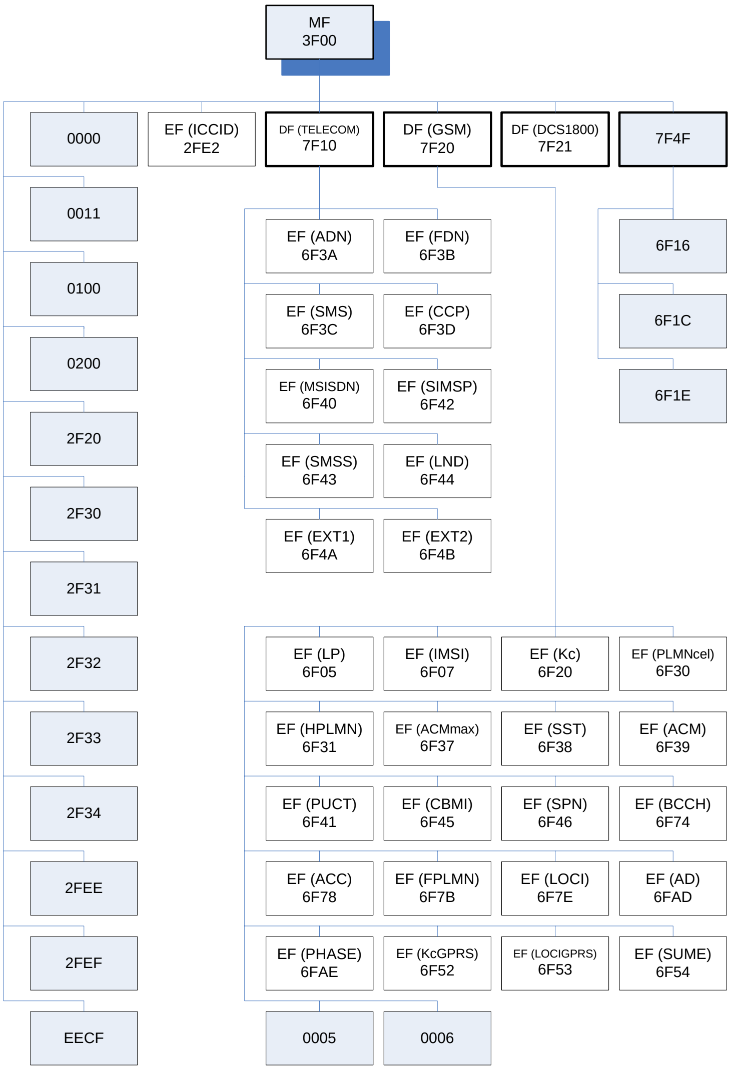

**Figure 3 – Example of filesystem extracted from a VODAFONE SIM, with 32K of EEPROM. Non standard files are highlighted.**

Figure 3 shows an example of a complete SIM card filesystem extracted from a VODAFONE SIM. It is interesting to note that non standard files are highlighted and are about 35% of total. Although there is no official interpretation of these files, the extraction process is more forensically sound in this way, because all observable memory has been dumped. Table 5 shows the non standard file features in terms of ID, file type, access privilege, structure type and, finally, parent of file. It is interesting to note that, potentially, hidden files could be used for steganographic purposes; that is, to hide sensible information in non standard locations of SIM/USIM cards. Although this is true only when access conditions permit it (ALW, CHV1/CHV2), this fact show that, potentially, there could be hidden areas where sensible information could be concealed.

Meanwhile, *pcsc*, the middleware used by SIMbrush to drive the IFD, supports the simultaneous operation of dozens IFDs. That is, an extraction does not constitute a bottleneck for a forensic lab, because many simultaneous extractions could proceed in parallel.

A lot of work could be done to improve and expand SIMbrush. First, a *Protocol Type Selection (PTS)* algorithm could be implemented to take advantage of those SIMs/USIMs that implement a faster interface with the IFD. Second, SIMbrush could be improved to communicate with *proactive SIMs*. Third, a formal definition, with the purpose of resulting in a wide agreement among the scientific community, for XML output could be created using a DTD. Fourth, research could be carried out to discover the meaning and coding of non standard files extracted by SIMbrush. Fifth, SIMbrush is not able to extract the bodies of those files that have read access conditions at the ADM or NEV level. The existence of a way to read these bodies while maintaining the  soundness of the tool has yet to be investigated.

A very useful work would be that of complementing SIMbrush with another tool at the examination stage. This tool should be able to interpret raw data extracted by SIMbrush and present them to the examination expert, providing him/her with an easy way to filter, browse and select those data. This tool currently is under development.

An important issue regards the possibility of verifying that the extracted data are really of the SIM card and not counterfeit by some procedure or library. For standard data this is quite simple and the test procedure adopted uses, for example, phone book entries or SMS messages in the cellular phone prior to extraction and, subsequently, these data are compared with the translated raw data, thus verifying equality.

The following XML code is a portion of SIMBrush output which shows an ICCID code that is reported on the SIM card and is easy to verify.

```
<ef>
   <header>
   00 00 00 0A 2F E2 04 00 04 FF 44 01 01 00 00 90 00
   </header>
   <body >
      <sw>
      90 0
      </sw>
      <content>
      98 93 10 00 00 00 45 49 65 25
      </content>
   </body>
</ef>
```

A possible XML template to represent the result of the translation is reported next:

```
<ef>
   <id>
   2F E2
   </id>
   <size>
   10
   </size>
   <acREAD>
   ALW
   </acREAD>
   <acUPDATE>
   ADM
   </acUPDATE>
   <acINCREASE>
   NEV
   </acINCREASE>
   <acREHABILITATE>
   ADM
   </acREHABILITATE>
   <acINVALIDATE>
   ADM
   </acINVALIDATE>
   <status>
   not invalidated
   </status>
   <structure>
   transparent
   </structure>
   <content>
      <imsi>
      89 39 01 00 00 00 54 94 56 52
      </imsi>
   </content>
</ef>
```

This portion of XML output contains all data necessary to lead a forensics investigation. We briefly report an explanation of various XML tags.

- **ID** is the file identification and is unique; *2F E2* stands for ICCID.
- **Size** refers to the file size.
- **acREAD**, **acUPDATE**, **acINCREASE**, **acREHABILITATE** and **acINVALIDATE** refer to access conditions of the file.
- **Status** is related to the validity and readability of the file.
- **Structure** defines the typology of file (transparent, linear-fixed, cyclic).
- **Content** is the really informative content of file, and in this case reports the IMSI number decoded from raw representation according to standard ETSI 100 977.

An important issue relates to whether the SIM card's data are modified during the extraction process. As previously mentioned there are only two commands of the SIM card operating system that are used, namely SELECT and GET RESPONSE. In no way, with these commands, it is possible to modify the filesystem content or, in other words, change any value of any file. SIM card access, in every phase of the extraction process, is in read only mode as can be verified by analyzing the reference documentation for such commands [ETSI TS 100 977 v8.12.0].

## Summary and Conclusions

SIMbrush is a new forensic imaging tool for SIM/USIM cards. Its open source nature fits the needs of digital forensic science. It has a number of advantages:

- It can extract the entire filesystem of a SIM/USIM card (both standard and non standard files), without constraints about the manufacturer, the issuer or the provider of the card, as it interfaces with it in a standard way;
- It discovers a lot of non standard files, usually used to store information considered somewhat hidden;
- It could be executed simultaneously in a number of instances on the same machine, without overloading the system;
- It presents its output in a standard textual XML representation, making it suitable for archiving purposes, for the use by different examination tools, for compression and for Web integration.

It also has a couple of disadvantages:

- The time for brushing a SIM/USIM, in full extraction mode, exceeds one hour for most cards;
- It cannot extract the body of those files with ADM or NEV access conditions.

Starting from the consideration that a bit for bit image of a SIM card is impossible if digital integrity is a constraint, SIMbrush tries to extract all possible data, or, in other words, all observable memory, from the SIM card in a standard way. While, at present, it is at an experimental stage, if it will succeed in awakening the interest of the scientific community, this tool could become very useful in real investigative processes.

## About the Authors

**Fabio Casadei** graduated from the University of Brescia in Electronics in March 2005 with a Dr. Ing. thesis on GSM/UMTs forensics. E-mail: fabio.casadei@mercurio-ws.com.

**Antonio Savoldi** is a Ph.D. student at the University of Brescia. His areas of research include security of embedded systems, steganography, steganalysis, digital watermarking, digital forensics and software testing. E-mail: antonio.savoldi@ing.unibs.it
More information can be obtained at http://www.ing.unibs.it/~antonio.savoldi

**Paolo Gubian** is an associate professor of Electrical Engineering at the University of Brescia. E-mail: gubian@ing.unibs.it.

## References

[Bates 1999] BATES J.: "Fundamentals of Computer Forensics", *Information Security Technical Report*, vol. 4, Supplement 1, pp.16-17, 1999.

[Carrier 2002] CARRIER B.: "Open Source Digital forensics Tools: The Legal Argument", *@stake*, Research Report, Oct. 2002.

[SWGDE] Scientific Working Group on Digital Evidence: "Proposed Standards for the Exchange of Digital Evidence" http://www.fbi.gov/hq/lab/fsc/backissu/april2000/swgde.htm

[PC/SC] Personal Computer/Smart Card: http://www.gemplus.com/techno/pcsc/

[DFRWS 2001] DFRWS: "A roadmap for digital forensic research", *DFRWS TECHNICAL REPORT DTR-T001-01 FINAL*, 2001.

[ETSI TS 100 977 v8.12.0] ETSI: "Digital cellular telecommunications system (Phase 2+); Specification of the Subscriber Identity Module - Mobile Equipment (SIM - ME) Interface (3GPP TS 11.11 version 8.12.0 Release 1999)", *Technical Specification*, 2004.

[ETSI EN 300 927 v5.4.1] ETSI: "Digital cellular telecommunications system (Phase 2+); Numbering, addressing and identification (GSM 03.03 version 5.4.1 Release 1996)", *European Standard (Telecommunications series)*, 2000.

[ISO 7816] ISO: "Identification Cards - Integrated circuit cards with contacts", *International Standard*, Parts 1-15.