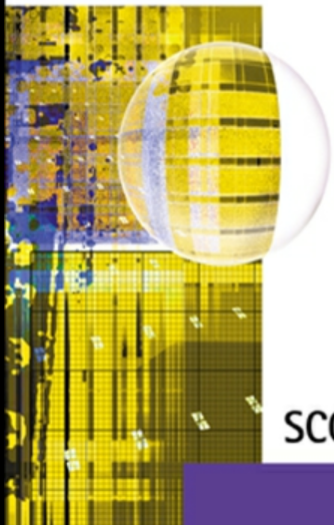


McGraw-Hill TELECOM

PROFESSIONAL

MOBILE APPLICATION DEVELOPMENT

Using SMS and the SIM Toolkit



*Building
Smart
Phone
Applications*

SCOTT GUTHERY • MARY CRONIN

Mobile Application Development

with SMS and the SIM Toolkit

Scott B. Guthery

Mary J. Cronin

McGraw-Hill

New York • Chicago • San Francisco • Lisbon
London • Madrid • Mexico City • Milan • New Delhi
San Juan • Seoul • Singapore • Sydney • Toronto

Copyright © 2002 by McGraw-Hill Companies, Inc. All rights reserved.
Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

ISBN: 978-0-07-139447-5
MHID: 0-07-139447-8

The material in this eBook also appears in the print version of this title: ISBN 978-0-07-1-37540-5, MHID 0-07-137540-6.

All trademarks are trademarks of their respective owners. Rather than put a trademark symbol after every occurrence of a trademarked name, we use names in an editorial fashion only, and to the benefit of the trademark owner, with no intention of infringement of the trademark. Where such designations appear in this book, they have been printed with initial caps.

McGraw-Hill eBooks are available at special quantity discounts to use as premiums and sales promotions, or for use in corporate training programs. To contact a representative please e-mail us at bulksales@mcgraw-hill.com.

1 2 3 4 5 6 7 8 9 0 DOC/DOC 0 9 8 7 6 5 4 3 2 1

The sponsoring editor for this book was Marjorie Spencer, the editing supervisor was Steven Melvin, and the production supervisor was Sherri Souffrance. It was set in Vendome by Patricia Wallenburg.

TERMS OF USE

This is a copyrighted work and The McGraw-Hill Companies, Inc. ("McGraw-Hill") and its licensors reserve all rights in and to the work. Use of this work is subject to these terms. Except as permitted under the Copyright Act of 1976 and the right to store and retrieve one copy of the work, you may not decompile, disassemble, reverse engineer, reproduce, modify, create derivative works based upon, transmit, distribute, disseminate, sell, publish or sublicense the work or any part of it without McGraw-Hill's prior consent. You may use the work for your own noncommercial and personal use; any other use of the work is strictly prohibited. Your right to use the work may be terminated if you fail to comply with these terms.

THE WORK IS PROVIDED "AS IS." MCGRAW-HILL AND ITS LICENSORS MAKE NO GUARANTEES OR WARRANTIES AS TO THE ACCURACY, ADEQUACY OR COMPLETENESS OF OR RESULTS TO BE OBTAINED FROM USING THE WORK, INCLUDING ANY INFORMATION THAT CAN BE ACCESSED THROUGH THE WORK VIA HYPERLINK OR OTHERWISE, AND EXPRESSLY DISCLAIM ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. McGraw-Hill and its licensors do not warrant or guarantee that the functions contained in the work will meet your requirements or that its operation will be uninterrupted or error free. Neither McGraw-Hill nor its licensors shall be liable to you or anyone else for any inaccuracy, error or omission, regardless of cause, in the work or for any damages resulting therefrom. McGraw-Hill has no responsibility for the content of any information accessed through the work. Under no circumstances shall McGraw-Hill and/or its licensors be liable for any indirect, incidental, special, punitive, consequential or similar damages that result from the use of or inability to use the work, even if any of them has been advised of the possibility of such damages. This limitation of liability shall apply to any claim or cause whatsoever whether such claim or cause arises in contract, tort or otherwise.

Introducing SMS and the SIM

Wireless devices have overtaken every other technology—including the Internet—in global adoption. By 2003 more than a billion people will be using a wireless phone or personal digital assistant (PDA) for voice and data communications. Three factors that have helped to drive this phenomenal growth have also inspired this book:

1. The worldwide availability and popularity of an inexpensive Short Message Service (SMS);
2. The evolution of the Subscriber Identity Module (SIM) inside GSM phones into a standardized and secure application platform for GSM and next-generation networks; and
3. The demand for applications that let people use their mobile phones for more than just talking.

Let's take a quick look at how SMS and the SIM have contributed to the growth of wireless applications and then discuss what you can expect to learn from this book.

The number of SMS messages sent every month has risen from about 1 billion messages in July 1999 to more than 20 billion in July 2001, with projections that the total number of SMS messages exchanged in 2001 will top 200 billion. These SMS exchanges range from simple text greetings or questions sent between individual subscribers (sometimes called "texting") to news and information services offered by the wireless carriers, to more advanced applications offered by third parties such as retrieving data from a corporate sales database or mobile banking. One result of all

this texting and other SMS activity is that wireless carriers now view SMS as an important source of revenues. Another outcome is that hundreds of millions of subscribers are ready and eager to try out interesting new services based on SMS. But to move beyond the basic text message delivery and create applications that can be customized and trusted, developers need a standardized and secure application platform. That's where the SIM comes in.

The SIM is a smart chip that was designed as a secure, tamper-resistant environment for the cryptographic keys that GSM carriers use to authenticate individual subscribers to the network connection and track those subscribers' activities once they are on the air. The SIM maintains a constant connection to the network as long as the mobile device remains on. This location-aware, authenticated connection is what allows subscribers to "roam" from network to network around the world and, very importantly from the viewpoint of the carrier, the SIM keeps track of and reports on the subscriber's network usage and roaming activity so that the carrier can bill customers accurately.

The only way to ensure that the SIM can accomplish its handoff of subscribers from one network to another without interrupting communication is to base all of its functions on very detailed international standards. Every GSM equipment manufacturer and carrier adheres to these standards, which cover everything from the physical size and characteristics of the chip to the way it handles and stores incoming information. Anyone developing applications that interact with the SIM also has to become familiar with the relevant standards and keep up with changes. This book describes the most important standards in detail and points readers to online sources of complete standard documentation and updates.

The SIM is also an essential part of the move to higher speed and more capable "next-generation" wireless networks, discussed later in this chapter. Because the 2001 digital network is referred to as the second generation (analog wireless was the first generation), these upgraded networks have been dubbed 2.5G (a significant notch up from the current speed and performance) and 3G. Although the timetable and technology for rolling out next-generation networks differs around the world, carriers everywhere recognize the importance of keeping today's SIM and SMS applications working during and after the upgrade. Therefore, the SIM will manage the roaming of traffic between generations of networks and between geographic locations. In addition, applications that work with today's SIM standards will be in a good position to take advantage of

the higher speed and multimedia capabilities of the 3G networks as they emerge.

Carriers, mobile equipment makers, and other service providers agree that applications are the most important driver for continued growth of wireless data exchange. The providers are searching for new killer applications to generate additional revenues from their networks and increase subscriber use and loyalty. They see that individual subscribers are looking for applications that will allow them to get more from their mobile phones or wireless PDAs. Businesses need applications that make mobile employees more productive and enable them to reach their mobile customers. There are different ideas about who should develop such applications. Some carriers prefer to do their own development work, whereas others contract with third-party developers or look to the SIM and mobile equipment vendors to provide the applications. One way or another, the demand for applications continues to increase.

Wireless Application Protocol (WAP), which many people thought of as the fastest route to mobile applications, was something of a wake-up call for network operators. When wireless communications were all about voice, the operators controlled every aspect of the mobile phone. The emergence of WAP allowed well-known Web-based services like yahoo.com and literally hundreds of start-up WAP sites to download programs to the mobile handset and take control of the screen and the keypad. The wireless operators looked around and discovered that all they still really controlled was the SIM, a tiny computer deep in the guts of the mobile phone that was designed to protect security, not support applications. We'll discuss how this computer sprouted an application programming interface called the SIM Application Toolkit (SAT) and other development tools like the SIM Micro-Browser in [Chapter 10](#), but you should know that today's SIMs are an underappreciated platform for a rich variety of mobile applications.

At the same time, application developers, especially developers who are expert in creating SMS and SIM-based applications are in short supply. It is hard to find all the information needed to start using SMS and SAT, and even harder to find clear examples of how to program specific applications. This book provides a step-by-step explanation of the commands, standards, and programming techniques that will take you from basic SMS applications to advanced SAT functionality. If you want to learn more about SMS and SIM development, this is the place to start.

Foundations and Definitions

SMS is the abbreviation for Short Message Service. SMS is a way of sending short messages to mobile telephones and receiving short messages from mobile telephones. "Short" means a maximum of 160 bytes. According to the GSM Association, "Each short message is up to 160 characters in length when Latin alphabets are used, and 70 characters in length when non-Latin alphabets such as Arabic and Chinese are used."²

The messages can consist of text characters, in which case the messages can be read and written by human beings. SMS text messages have become a staple of wireless communications in Europe and Asia/Pacific and are gradually gaining popularity in North America. The messages also can consist of sequences of arbitrary 8-bit bytes, in which case the message probably is created by a computer on one end and intended to be handled by a computer program on the other.

SIM is the abbreviation for Subscriber Identity Module. As its name implies, its original purpose (and continuing role) was to identify a particular mobile user to the network in a secure and consistent manner. To accomplish this, the SIM stores a private digital key that is unique to each subscriber and known only to the wireless carrier. The key is used to encrypt the traffic to and from the handset. It is essential to keep this key out of the hands of mischief makers who might get hold of a SIM and try to steal the subscriber's identity. Because smart cards were designed to be extremely difficult to crack under a variety of attacks, the smart card's core electronics and design architecture were adopted as the base of the SIM. Building applications for the SIM has a lot in common with designing smart card applications and, as we will see later, the standards that guide the evolution of smart cards and the SIM have started to converge in the international standard-setting bodies.

One of the most important standards for SIM application developers is the SIM Application Toolkit (SAT). As the name implies, the SAT standardizes the way in which applications besides the subscriber's private keys can be developed for and loaded onto the SIM. Wireless carriers are understandably sensitive about guarding the security of the SIM and preserving its primary function of subscriber identity and encryption. Because the carrier controls what code is loaded directly onto the SIM, adhering to SAT standards in building your application doesn't mean that it will run on any given network. Typically, there is a testing and certification process required for any application that is not developed di-

rectly by the network providers or SIM vendors.

On the one hand, such a process can make it difficult to get your applications on the SIM because, if any Tom, Dick, or Sally can download programs to the SIM it wouldn't be a trusted computer. On the other hand, when you do get your applications on the SIM, you will be in good company. Or, if your applications don't require the full-blown trust and security apparatus built into the SIM, you can work with SMS and a tool called the USAT Interpreter to interact with Web-based information via the SIM. As more SIMs capable of running virtual machines such as Java come to market, you can also develop applications that can be downloaded over the air—as long as the application is acceptable to the wireless carrier. This book explains the range of possibilities and illustrates the steps involved in developing those possibilities.

The SIM is the smaller of two computer chips inside a GSM mobile handset. Early SIMs typically were $\frac{1}{3}$ million instruction per second (MIP) with 3K memory, and most SIMs in use today are $\frac{1}{2}$ MIP with 16K memory. To handle virtual machines and larger applications, the current high-end SIM provides 32K of memory, with 64K SIMs anticipated within the next year. The computer chip that runs the handset is much larger, typically with a couple of megabytes of memory and a couple of MIPs of computer power. The larger chip controls the keypad and the display, encodes and decodes voice conversations, and runs the protocols that enable the handset to connect to the telephone network. The SIM may be a small computer compared with the handset computer and a tiny one compared with PDA and notebook processors, but its size doesn't have to be a gating factor for innovative applications. In fact, the SIM has about the same computing power as the first IBM PC and that computer opened the eyes of corporations and individuals to the potential of word processing, spreadsheets, and other applications to change the way we do our work and live our lives.

Bear in mind that there are other ways of exchanging data with a mobile telephone that are not covered in the following chapters. General Packet Radio Services (GPRS) is one example. There are also other ways to build mobile applications. WAP is one of the best known and has a large following. Nevertheless, SMS and the SIM have some characteristics that make them attractive for many types of application.

SMS is cheap, always on, gets through when other messages don't, is a store-and-forward system and is quite easy to build with. The SIM is portable so you can move it from one mobile device to another; it is tamper resistant, so it can be used to hold sensitive data; and it provides access to the

full range of capabilities of the handset. One sweet spot for applications using SMS and the SIM is trusted transactions. Although this includes mobile commerce and financial transactions, the trust inherent in the SIM can be leveraged to a much broader group of applications where privacy and performance are important. The case-study chapters describe how companies and carriers are using this trust in real-world situations.

An SMS message nearly always gets through. If the mobile phone isn't on when you send a message, the system holds it until the phone is turned on and then delivers it. The system also can generate a return receipt that tells you that the message has been delivered. SMS messages are encrypted, so there is no fear that your message will be snatched out of the air and read. You can even add your own encryption to an SMS message so that not even the phone company can read what you are sending. There are many standards, software packages, and service providers that make building industrial-strength SMS applications easy, quick, and even fun (if you have a somewhat distorted sense of fun).

SMS and SIM in the Network Context

Before we plunge into the details of development, it is important to understand the network context in which SMS communicates with the SIM and the mobile device. The dynamic duo of SMS and SIM works as follows. The part of your application on your desktop computer or corporate server creates an SMS message to be sent to the part of your application on the mobile. This message is handed off to the short message center of your local telephone company with the telephone number of the mobile you want it sent to. The telephone company finds the mobile and passes the SMS message to it. The message has a flag set in it that tells the handset to pass the message to the SIM. The message also has a flag that says which application on the SIM should receive the message. When the SIM receives the message from the handset, it checks to see which application to give it to and hands it off to the mobile side of your application.

[Figure 1-1](#) illustrates the flow of traffic.

Figure 1-1

Message flow from server to screen.

Another example is the use of the computational capabilities of the SIM is to generate a temporary key that is used to encrypt the communication between the handset and the network. This is the air link and the link that is most subject to freeform eavesdropping. After you have been successfully authenticated, the SIM uses the random number to create a key that is used to encrypt the traffic between the phone and the visiting network. This key also is generated by your home network and given to the visiting network over landlines so that the key used for protecting the phone and the network never travels over the air.

There are many books and papers on the security and cryptography used in the GSM network, so if this aspect of GSM strikes your fancy, there is much to learn. We've included some references to this body of literature in the bibliography if you want to pursue this path. For the purpose of application development, what is important is that the SIM is a secure computer that is underpinned with perhaps the biggest key infrastructure outside the government. The SIM and this key infrastructure can be harnessed to inject trust into your applications and the transactions they generate.

SIM-based applications aren't about a daily horoscope or a late-breaking football score. We'll leave those to untrustworthy handset applications based on WAP or whatever replaces it. SIM-based applications are about taking action with the assurance that the data on which the action is based is from who is says it is from (authenticated and nonreputable), that it hasn't been tampered with (integrity), and that nobody else is privy to what is happening (confidentiality).

I like to think of my GSM phone as the ultimate remote control from which I can issue commands about every aspect of my life from anywhere in the world to anywhere in the world. This makes the SIM a very, very attractive application development platform.

Smart Cards 101

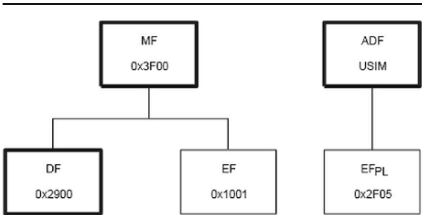
Just as there is a large body of information on GSM cryptography, there are many sources of information about smart cards. The cryptography we can assume because it really happens under the covers. To use the SIM, we have to understand a little about smart cards because, at the most fundamental technical level, a SIM is a just a smart card that happens to have a mobile phone as its reader.

A smart card contains a hierarchical file system just like your desktop computer or laptop computer. Rather than having human readable names like "My Documents" or

"Chapter6.doc," directives and files in a smart card consist of two binary bytes. Thus we speak of the directory 0×7F01 or the file 0×45C6. The root of the hierarchy is called the Master File (MF). Its name is always 0×3F00.

There are different kinds of files, just like on your PC, but not nearly so many different kinds. There are directories, of course, and files that contain an unstructured block of data such as a name or an account number. There are also record files that contain a series of fixed-size records, each of which may have well-defined, fixed-size fields. Directories are referred to as DFs (unless they contain all the information pertinent to a specific application, in which case they are called ADFs) and files are generically referred to as EFs (Figure 7-2).

Figure 7-2
Examples of directories and files.



There are about 80 files on a SIM card. Because not everybody can remember that 0×2F05 is the file that contains the subscriber's preferred language, the convention has been adopted to use a descriptive subscript on DF or EF when talking or writing about these files. The language preference file, for example, would be referred to as EF_{PL} (for Preferred Language). Nevertheless, when you talk to the SIM about this file, you have to use its two-byte name, 0×2F05.

Talking to a smart card such as a SIM means sending it commands to perform actions. For example, if we wanted to retrieve from the SIM the subscriber's language preference, we would use the SELECT FILE command to focus the SIM's attention on EF_{PL} and then the READ BINARY command to instruct the SIM to return to us the contents of the file.

If the SIM is just a smart card, why can't we use it in a plain old smart card reader? Why do we have to get at it through the handset or even through the entire GSM network? The fact is you can communicate with a SIM through a plain old smart card reader, but you can do only those things that you the subscriber are allowed to do. However, this way of using the SIM can be an

advantage. In fact, it can produce some very interesting applications.

An even more exciting possibility is using the SIM as a way to inject trust into transactions coming from your laptop or desktop computer. After all, it is still you, whether it is you standing at a bus stop with a mobile phone in your hand or sitting at your desk wanting to buy a book on Amazon.com. At this point however, many network operators are so busy trying to control all the applications on the mobile network that they don't seem to understand that they can offer additional trusted services and make money in other networks such as the very nonmobile World Wide Web.

Let's get down and dirty and take a look at the actual bytes that are sent to the SIM to read the subscriber's language preference. First, we have to focus the SIM's attention on the file from which we want to read data, namely good old 0×2F05. To do this we send the SELECT FILE command to the SIM. SELECT FILE for 0×2F05 looks like this:

CLA	INS	P1	P2	Lc	File ID	
0xA0	0xA4	0x00	0x60	0x02	0x2F	0x05

It's seven bytes long and consists of six fields. The CLA ("class") field provides some information about the format of the command. Value 0×A0 means this is a GSM command.

The INS ("instruction") value 0×A4 says that this is the SELECT FILE command. P1 ("parameter 1") isn't used in this command and the P2 (not surprisingly, "parameter 2") value 0×60 says don't send back any information about the file, just focus all of your attention on it. The Lc value indicates that the next file name is 2 bytes long. The File ID bytes are the name of the file. Lots of other options are available by putting codes in the other fields, but we'll concentrate on the simple case for now.

Even though we said don't send back any information about the file, the SIM always sends back a 2-byte status word saying how things went. If the 2 bytes, 0×90 and 0×00, come back, then everything went OK and the SIM is ready for the next command. If something went wrong, then the SIM would send back an error code. For example, if it couldn't find the file you asked for, it would send back 0×6A 0×82, which translates freely as "File not found."

OK, we got back a 0×9000 so the SIM is concentrating on 0×2F05, the preferred language file, EF_{PL}. The subscriber's languages are listed in this file in order of preference, with the most preferred language being first. Each language is coded on two bytes as two alpha-numeric characters.

The coding of languages as two characters is given in ISO 639. Let's read the first 2 bytes using the READ BINARY command:

CLA	INS	P1	P2	Le
0x00	0xB0	0x00	0x00	0x02

The INS value of 0xB0 says that this is the READ BINARY command. P1 and P2 together are offset into the file where the read should begin. In our case, because we want to read the first 2 bytes, the offset is 0. The Le value indicates how many bytes we want to read.

In this case, the SIM sends back 4 bytes:

0x65 0x0F 0x90 0x00

The first 2 bytes are the encoding of the subscriber's most preferred language and the second 2 bytes are our old friend 0x9000, which from the SIM's point of view indicates that everything went OK. The first 2 bytes are ASCII for "eo," which means the person prefers to speak Esperanto.

The commands sent to the SIM to ask it to do something are called APDUs. APDU stands for Application Protocol Data Unit. The number of commands that the SIM understands has grown over the years. Today, there are 22 APDUs that the SIM understands. All interaction with the SIM boils down to the handset sending one or another APDU to the SIM and catching what comes back. Even when you put your own application on the SIM, as we discuss in [Chapter 12](#), it will have to define the APDUs that must be sent to the SIM to talk to it.

The Evolution of the SIM

Before we go into details, we should say something about the SIM hardware and software architecture and about how these are evolving.

From its first use in a GSM phone until very recently, a SIM was an 8-bit microcontroller, usually an Intel 8051 or a Motorola 6805. The amount of memory in the SIM has grown. Initially containing only 256 bytes of RAM and 3 kilobytes of EEPROM, it has grown to 1,024 bytes of RAM and 32 kilobytes of EEPROM. All the software for the SIM—the operating system and the code that handles the file system and the APDUs—is burned into 32 to 64 kilobytes of ROM. The 80 odd data files are kept in the EEPROM, and the RAM is used as an I/O buffer for communication with the handset. This is about all that is needed for the SIM to perform its cryptographic duties and provide some relatively trivial telephony services such as storing a phone book of

frequently dialed numbers. Beginning in the mid-1990s, as new applications started to find their way onto the mobile telephone, the SIM started to morph into a full-fledged application platform. Needless to say, this put all kinds of evolutionary pressure on the SIM operating software and the SIM hardware. The software started to sprout virtual machines, principally the Java virtual machine, to host nontelephony applications, and the hardware started to grow additional memory and computational abilities.

Roughly speaking, there have been three generations of SIMs. In the first generation, SIM was just a SIM. The only thing the SIM smart card could do was be a SIM. The smart card hardware was committed solely to SIM functionality. In the second generation, additional applications could be added to the SIM. The SIM was still the uber-application, but it could call on other applicationettes to create custom services for the subscriber. In the third and current generation, the SIM itself becomes an application and all applications have equal (well, almost equal) standing on the platform.

Generation	Name	Characteristic
1	Purpose-built SIM (Figure 7-3)	SIM performs network authentication and universal telephony services such as phone book and SMS message storage
2	SIM with applications (Figure 7-4)	Additional customer-specific services can be added as SIM Toolkit (STK) applications after the SIM is in the field
3	SIM as an application (Figure 7-5)	SIM is one of the multiple telephony applications, specifically the authentication application, on the platform

Figure 7-3 Architecture of a first generation SIM.

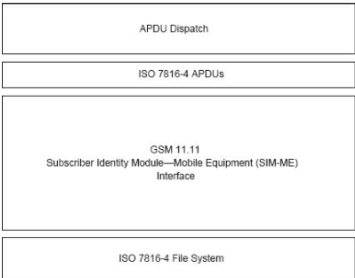


Figure 7-4 Architecture of a second generation SIM.

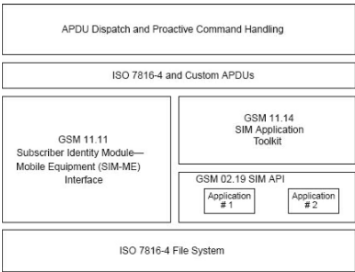
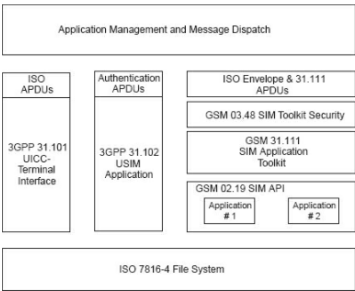


Figure 7-5 Architecture of a third generation SIM.



There is a fourth generation on the horizon that we will discuss in [Chapter 12](#).

Who Are You?

Throughout its evolution, the SIM has never lost sight of its primary function—to authenticate the subscriber to the network. At its heart, it is an identity token; as a network identity, it has become more, not less, important. Its unique ability to perform this authentication function is what has attracted applications to the SIM platform.

Initially, the SIM could recognize three entities:

- **ADM**—This is the network operator who performs Administration duties on the SIM. ADM might be thought of as the UNIX superuser.
- **PIN**—This is the subscriber, the person toting the phone and the person who enters the PIN to activate the SIM on the phone so that it can get the phone on the network.
- **PIN2**—This typically is the entity paying the bills for use of the mobile. PIN2 is stronger than PIN and can, for example, reset PIN, but it is not as strong as ADM.

Associated with every file on the SIM smart card is a set of