How to Turn a GSM SIM into a Web Server

Projecting Mobile Trust onto the World Wide Web

Scott Guthery, Roger Kehr, Joachim Posegga

Scott Guthery, Mobile-Mind, sguthery@mobile-mind.com
Roger Kehr, Deutsche Telekom Research , Roger.Kehr@Telekom.de
Joachim Posegga, Deutsche Telekom Research, Joachim.Posegga@Telekom.de

Key words: Smart Cards, GSM SIMs, Web Servers, Security Infrastructure

Abstract: We describe the WebSIM, an approach that integrates GSM SIMs into the Internet. The underlying idea is to implement a Web Server inside a SIM, and to allow for transparent access to it from the Internet.
The contribution of our approach is that a SIM, which is currently a security module (smart card) fitted in a GSM mobile phone, becomes also a personal security server in the Internet. Like any other server in the Internet, it speaks TCP/IP and processes HTTP requests, e.g. for accessing certain SIM services (e.g. authentication) via CGI scripts.
The Internet connectivity of a SIM inside a mobile phone can be achieved by having a proxy host tunnel IP packets to the SIM over SMS.

> If we couldn't predict the Web, what good are we?
> Bob Lucky, Vice President Bellcore, 1995

1. INTRODUCTION

Much of today's wireless Internet excitement is focused on the opportunities created by pushing the World Wide Web out onto mobile telephone networks. Micro-browsers and WAP handsets seek to turn the mobile telephone into a downscale laptop computer. Relatively little thought has gone into wondering what the mobile telephone network can bring to the Web.

Secure, reliable authentication, which is a basic prerequisite for billing customers for services on a large scale, still has no globally-accepted solution. Various attempts have been made to provide the required security technology for the Internet, but none of them has widely succeeded so far. All approaches have in practice either been considered as too insecure, or too hard to establish at the user's side. With its strong similarity to the ubiquitous credit card, a smart card is a compelling component but the required infrastructure for smart card-based solutions has been found to be hard and costly to set up.

GSM, on the other hand, provides a widely used security infrastructure, in the form of symmetric keys distributed in subscriber identity modules (SIMs). More than 250 million GSM subscribers carry around these reduced size smart cards in their mobile phones. Mobile phones can thus be seen as "wireless card readers", with the add-on of providing an I/O channel to a user for applications running inside the SIM.

The theme of the current work is that while the Web is bringing its content to the mobile phone, the mobile phone can bring its trust to the Web. The idea is to provide the authentication and authorisation capabilities from the GSM SIM to Web-based applications in a Web-friendly way; viz. as a Web server. Such a WebSIM, like any other server in the Internet, speaks TCP/IP and is transparently accessible from Internet hosts via HTTP. Specific services offered by server-enabled SIMs, for example authentication, can be implemented on the SIM using CGI scripts.

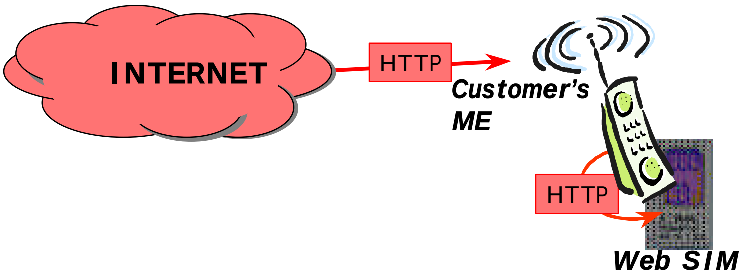

Figure 1. HTTP Requests to a SIM

Technically, this is achieved by implementing a Web server inside the GSM SIM and allowing for HTTP requests to this SIM and HTTP responses from it. Seen from the GSM perspective, this HTTP server provides selected parts of the existing application programming interface of a GSM SIM to the Internet. This makes communicating with a SIM residing in a mobile phone identical to communicating with any Web server running in the Internet (cf. *Figure 1*), and the SIM can be transparently accessed from the Internet via HTTP, e.g. for authentication purposes.

2. THE WEBSIM: TECHNICAL DETAILS

A GSM SIM is an operator-trusted security server in GSM, performing computations on behalf of the GSM subscriber. The idea behind our approach, the WebSIM, is to extend the GSM SIMs into the Internet, by having them to understand a widely-used Internet protocol, HTTP, and to allow for transparent access over HTTP from the Internet. This means a GSM SIM residing in a mobile phone is not only a security server in the GSM network, but also becomes available as a security server in the World Wide Web, acting again on behalf of the GSM subscriber. We refer to such a SIM as a WebSIM.

A WebSIM, like any other server in the Internet, speaks TCP/IP and processes HTTP requests. Technically, this can be achieved by implementing a small, stripped-down Web server in a GSM SIM and making the SIMs accessible from the Internet. In this way, communicating with a SIM in a mobile phone, e.g. for authenticating a customer in the Internet, is the same as communicating with any Web server running in the Internet.

Seen from the GSM perspective, the idea behind the WebSIM is to make the interface of today's GSM SIMs (ETSI GSM 11.11 and GSM 11.14) partially available on the Internet.

2.1 The SIM's Web Server

Running a Web server in a SIM is less of a problem than one might think, in fact such servers for standard (non-GSM) smart cards were introduced in [Rees 99]. Clearly, a Web server in a SIM is not expected to host large amounts of information or HTML documents, but to provide a convenient interface to services of the SIM. These services, most of which will probably be security-related, can then be accessed via the standard protocol of the Web, HTTP, and implemented as server-side scripts on the SIM.

It is explicitly not an objective to implement versions of standard Internet services and protocols that are in full compliance with the specifications that define them. While fully compliant implementations of existing standards are certainly desirable, we are willing to give a little on full compliance, implementing only a strict subset of the specification, in order to realise an efficient yet useful smart card implementation. This design philosophy could be summarised as “It’s not how well the dog sings but that the dog sings at all.”

A stripped-down version of the HTTP 1.0 protocol, which just covers the absolutely necessary part and only allow for one connection at a time, can be easily implemented with an application of less than 10K bytes inside the SIM. In particular, we can very elegantly implement this functionality as an applet on top of a GSM SIM Toolkit platform (ETSI GSM 02.19, 03.19) and then use the Toolkit’s interpreter for server-side scripting. This also allows for interacting with the user of the SIM’s mobile phone, since SIM Toolkit provides also an appropriate API for I/O (GSM 11.14).

2.2 Networking

Once we have an HTTP-server running in the SIM we need to connect it to the Internet. An elegant approach would be to see the mobile phone as a gateway router that passes IP packets to the SIM. If we do not want to assign a separate IP address to the SIM, we could also configure it as a process listening to port 80 on the mobile phone.

This approach, while technically elegant and easy to integrate into technologies like GPRS, requires modification of the handset and the creation of a new ETSI standard. Even if such a standard could be agreed to in a timely fashion, significant market penetration by compliant equipment would take at least two to three years. We therefore propose another approach, which can be implemented using today’s protocols and mobiles.

The innovation barrier of the ME can be circumvented by a solution very common in the Internet; viz. a proxy server. We set up a proxy for the SIM on the Web and have this proxy tunnel HTTP packets through SMS to the SIM (cf. *Figure 2*). SMS messages arrive directly in the SIM and can be processed as required, e.g. by having Toolkit Applets register for such SMs. Thus, we circumvent the handset by using existing protocols and standards.

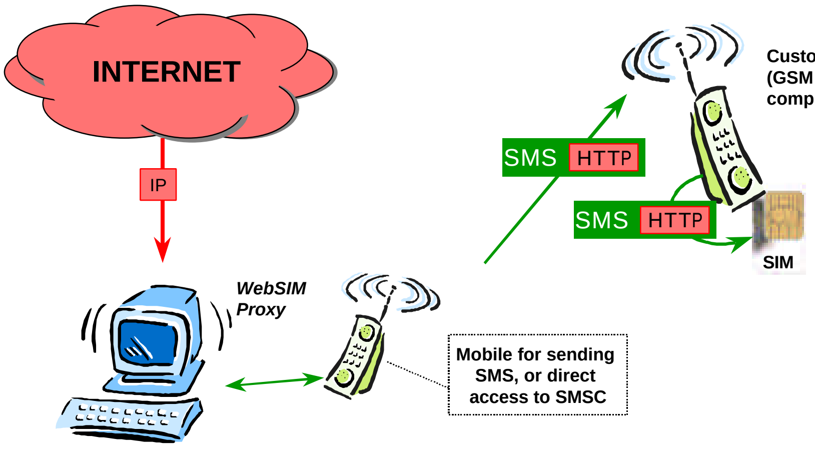
INTERNET
IP
WebSIM Proxy
SMS HTTP
SMS HTTP
Customer's ME (GSM 11.14 compliant)
SIM
Mobile for sending SMS, or direct access to SMSC

Figure 2. HTTP Tunnelling over SMS

The procedure for proxy-based IP-communication with the SIM over SMS is as follows:

1. An Internet host sends an HTTP request to the SIM's proxy.
2. The proxy embeds the request in a specially tagged SMS and sends it to the SIM.
3. The SIM passes the incoming SMS-encapsulated HTTP packet to the SIM that has registered to handle such tagged SMS.
4. The HTTP packet is extracted processed by the Web server in the SIM.
5. The HTTP response is embedded in SMS again and sent back to the proxy.
6. The proxy extracts HTTP response from the SMS and sends it back to the client in the Internet that sent the request.

As a result, the SIM can be transparently accessible by TCP/IP and HTTP from any Internet host. TCP/IP de-capsulation is handled by the proxy and the HTTP payload passed to the SIM in an SMS message. The response from the SIM is re-encapsulated by the proxy and returned to the Internet.

The proxy server approach also has some additional advantages since it can:

– implement a firewall between the Internet and the GSM network.
– guard against denial of service attacks.
– perform address translation (NAT) between the Internet address of a SIM card and the GSM address of the handset which holds the SIM.

- perform accounting and billing for WebSIM services
- eliminate the need for implementing a TCP/IP stack in the SIM

When used in practice, the WebSIM processes HTTP-bearing IP packets, for example, an URL request such as

```
http://websim.dtrd.de/+49000000000/sign=(2A49C01...)
```

This HTTP request can initiate the signing of the data in brackets in the SIM of the named GSM phone. After processing the request, which might consist of running other SIM-internal applications or commands, the result is sent back to the originating Internet host. Thus, integrating the capabilities of the GSM SIM into Internet applications is just like communication with any other Web server on the Internet.

2.3 Implementation

We have implemented a prototype of a WebSIM and its proxy that allows access a few SIM services over HTTP. The proxy's name is websim.dtrd.de[1], the SIM is identified by its phone number. The current implementation provides access to the following services:

a) `http://websim.dtrd.de/+49000000000/info`
Returns information about LAI and LAC of the SIM (GSM 11.14 PROVIDE LOCAL INFO)

b) `http://websim.dtrd.de/+49000000000/si=(item1,item2,item3,...)`
Prompts the user of the phone with a GSM 11.14 SELECT ITEM command offering the choices listed in brackets, separated by ",". Each item is interpreted as a string, the overall length of all strings must not exceed 120 bytes. The user's choice is returned.

c) `http://websim.dtrd.de/+49000000000/gi=(prompt)`
Runs the GSM11.14 GET INPUT command and returns the text that has been entered.

d) `http://websim.dtrd.de/+49000000000/dt=(text)`
Runs the GSM11.14 DISPLAY TEXT with the argument supplied.

[1] Access to the server is restricted.

e) `http://websim.dtrd.de/+49000000000/sign=(abcdef0123456789)`
   Encrypts the argument (interpreted as a string of hexadecimal characters) and returns the result.

The length of the arguments (given in brackets) is, for the sake of simplicity, restricted to fit into one SMS.

2.3.1 Proxy Implementation

The proxy is a Linux laptop running an Apache Web server with a couple of CGI scripts. These CGI scripts (implemented in Perl) take an incoming HTTP request, embed it in an SMS message, and send it off to the specified phone number.

Sending of SMS is done through a GSM mobile that is attached to the laptop with a PCMCIA modem card, which is used for sending and receiving SMS[2]. Short messages are sent by turning the modem into TPDU mode (GSM 07.05), using a tag that causes the message to go directly to the Web Server application in the destination SIM (cf. GSM 03.48 and GSM 11.14).

Receiving SMS messages is detected by a separate process looping on the laptop that continuously polls the attached mobile for incoming short messages which are responses to pending HTTP requests. If an incoming short message is detected, it is fetched, the HTTP response is extracted and TCP/IP encapsulated and returned to the Internet client that sent the corresponding request.

2.3.2 Web Server Applet

The Web server runs as an applet on the SIM Toolkit platform (GSM 03.19) in a Schlumberger Simera SIM. The applet is written in Java, its size is currently about 7K bytes of Java byte codes. For the sake of simplicity the applet makes the following restrictions:

a) an HTTP-Request must not exceed one SMS, and one SMS can only contain one request.
b) the card handles only one request at a time, i.e. there is no session management inside the card.

Both restrictions can be easily overcome if needed.

We did not space-optimise the applet code at all; and we believe that it can be stripped down to a size of about 5K bytes. Noteworthy, adding new commands to the server Applet does not significantly increase its size: HTTP

[2] A more efficient variant would be to connect the proxy directly to the network's short message service centre (SMSC) which is the store-and-forward point for all SMS messages.

can be seen as a general-purpose application launching protocol, and once the basic HTTP-handling functionality is implemented, adding an extra command increases the applet only slightly: the difference between an applet providing only the SELECT ITEM command, and the version handling the five commands above is only a few hundred bytes.

2.3.3 Example Request

The HTTP-request `http://websim.dtrd.de/+49000000000/info` results in the following response (without HTTP headers):

```
LAI: 262 01
LAC: 730C
```

262 is the country code (Germany), 01 denotes the network (D1-Telekom), and 730C is the Local Area Code (Karlsruhe, Germany).

Overall processing time depends largely on SMS transport time, which is usually between 5 and 20 seconds one way. The proxy needs 3-4 seconds to send and receive the short messages, the handset and SIM-internal processing takes roughly another 5-6 seconds. So, the complete processing of such an HTTP request takes typically about 30 seconds.

3. APPLICATIONS OF THE WEBSIM

The WebSIM is not an application per se: it opens up the SIM to the Internet and provides an Internet-compliant interface to SIM services. Thus, it is a horizontal technology (more precisely a middleware) that supports "dot com" style applications.

We sketch a few of these applications for different domains below. Much more is possible, and in fact the most promising aspect is that the WebSIM a very convenient middleware for integrating it into Internet applications. Rather than having to deal with a different interface to a SIM each time, Internet applications can access and activate these applets in their own language, HTTP.

3.1 Provision of a secure I/O channel for Electronic commerce

Assume a customer of an Internet bookshop ordered a book for US$ 20. When ordering it, the phone number of a WebSIM phone was provided, and the Internet bookshop can now launch a simple HTTP-request such as http://www.../+49000.../si=(***Bookshop:,Confirm%20USD20",Cancel,...)

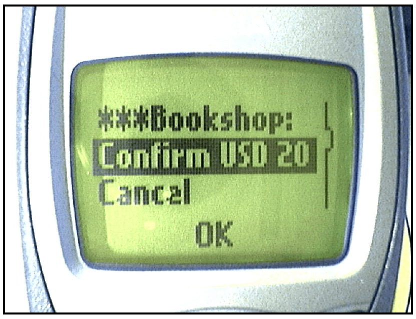

Figure 3. Screen Shot

This would result in a menu being displayed on the mobile phone of the customer (cf. *Figure 3*), and the Internet merchant would have established a relatively secure I/O channel to its customer (or whoever is using the customer's phone).

The security of this I/O channel can easily be enhanced by connecting over HTTPS to the proxy, or by cryptographic means as we will see next.

3.2 Authentication on the Internet

Assume Alice wants to authenticate Bob over the Internet, Oscar is Bob's GSM operator that issued a WebSIM SIM_B to Bob. Consider the following basic skeleton of a protocol:

1. B → A: +490000000 *[Bob's telephone number]*
2. A → $\mathrm{SIM}_{B:}$ http://www.oscar.com/+490000000/sign(RAND)
3. SIM_B → A: hash(Ki; RAND, IMSI)
4. A → O: https://www.oscar.com/
verify(RAND, hash(Ki; RAND, IMSI), +490000000)
5. O → A: yes/no

In step 1 Bob gives Alice the phone number of his SIM. Alice then sends a random number (challenge) to the SIM_B in step 2. The SIM_B returns a hash of a secret key Ki, the random number RAND, and the SIMs IMSI[3] to Alice in step 3.

[3] IMSI = International Mobile Subscriber Identity number.

Alice can now send the response she got in step 3 to Oscar the operator, who can verify the result. Oscar knows the hash algorithm, the secret key Ki, and he can associate the phone number with the IMSI that was used in the encryption. Note that nobody besides Oscar needs to know Ki and hash. Note also that the messages of step 4 and 5 should be sent over a secure channel, e.g. by using https://... between Alice and Oscar.

Such a protocol can be easily refined to meet various needs one has for authentication, like including an explicit conformation from Bob, or a time stamp (for instance taken from the SMS), etc. It is a classical challenge/response authentication which can be applied to many scenarios (home banking, access control, etc.), or it can be easily adapted to provide, for instance, a session key for other purposes. For security reasons, the scenario can also be used with keys other than Ki, or with a key derived from Ki.

Essentially, this scenario is based on the principle hat the mobile phone can be used as a “wireless” card reader containing a authentication token, and the security infrastructure of GSM can be easily accessed from the Internet.

3.3 Physical Access

Another nice example for using a WebSIM is the following one. Assume pushing your doorbell at home results in a request such as:

```
http://websim.dtrd.de/+49000000/si=(Open,Call Intercom,Cancel)
```

If you are standing in front of your door and have pushed the doorbell button, you will of course select “Open” on the menu appearing on your phone. If not, you can select to be connected with your home's internal intercom or simply cancel the request if you don’t want to be disturbed.

In the case that you are connected to the intercom (which is in turn bridged to your mobile handset) you can converse with your visitor and if the situation warrants it, open the door remotely even if you aren't at home.

3.4 Handset Configuration

There are a number of SIM services local to the mobile handset that would be much easier to handle with Web interfaces. For example, the management of the SIM-internal phone book, which can be very conveniently, updated using a Web browser if the WebSIM understands HTTP POST-methods.

4. CONCLUSION

We have presented the WebSIM, an approach integrates the GSM security module (SIM) into the Internet. The underlying idea is to integrate an HTTP server into a GSM SIM, allowing Internet connections to be made to it. This turns the SIM, which is security server for the GSM subscriber, into a WebSIM, which is a general-purpose security server for the subscriber on the Internet.

Such a WebSIM, like any other server in the Internet is transparently accessible from Internet hosts via TCP/IP and HTTP. Specific services offered by SIMs, e.g. authentication, can be accessed using CGI scripts from Internet hosts.

Seen from the GSM perspective, this HTTP server extends parts of the existing external interface of today’s GSM SIMs into the Internet and is seamlessly reachable from the Internet, acting as a security server for a GSM subscriber.

The main contribution of our approach is to provide the function of SIMs in Internet-compliant protocols anyone can use. This means that the barriers for smart card applications today, 1) the lack of integration of smart cards into information technology architectures and the World Wide Web and 2) the complex interaction with cards using APDUs, is overcome by providing a simple, standard protocol, i.e. HTTP, for accessing SIM services.

A WebSIM can be accessed from anywhere on the Web using familiar Web protocols, and application programmers do not need to cope with smart card-specific interfaces any more. As with other servers in the Internet, the WebSIM processes HTTP-bearing IP packets, For example, an URL request such as

```
http://websim.dtrd.de/+491710000000/sign=2A49C01
```

would initiate the identity application running in the WebSIM of the named GSM phone.

After processing the request, which might consist of running other SIM-internal applications or commands, the result is sent back to the originating Internet host. Thus, integrating SIM-based security into Internet applications involves the same programming techniques as communicating with any other Web server on the Internet.

Sales of mobile handsets are forecast to soon exceed sales of personal computers. While desktop and laptop computers are great for viewing art at the Louvre, when it comes to taking action, the mobile handset with its trust-bearing SIM not to mention its portability may prove to be the primary Web surfing device. WebSIM enables the mobile to add value to the Web as opposed to simply receiving content from it.

5. REFERENCES

[Barber 99] J. Barber, "The Smart Card URL Programming Interface," Gemplus Developer Conference, June, 21 - 22, CNIT Conference Center, Paris-La Defense, France. 1999.

[Becker 99] C.B. Becker, B. Patil and E. Qaddoura, IP Mobility Architecture Framework, IETF-Draft, October, 1999.

[Campbell 99] A. Campbell, J. Gomez, C-Y. Wan, Z. Turanyi, and A. Valko, Cellular IP, IETF-Draft, October, 1999.

[Comer 95] D. Comer, Internetworking with TCP/IP, Prentice Hall, 1995.

[Di Giorgio 99] Rinaldo Di Giorgio, An introduction to the URL programming interface, Java World, September 1999.

[GSM 02.19] Digital cellular telecommunications system (Phase 2+, Release 98): Subscriber Identity Module Application Programming Interface (SIM API); Service description; Stage 2. European Telecommunications Standards Institute, Sophia Antipolis, France. Unpublished Draft. 1999.

[GSM 03.19] Digital cellular telecommunications system (Phase 2+); Subscriber Identity Module Application Programming Interface (SIM API); SIM API for Java Card™ ; Stage 2. European Telecommunications Standards Institute, Sophia Antipolis, France. Unpublished Draft. 1999

[GSM 03.60] Digital cellular telecommunications system (Phase 2+); Genheral Packet Radio Service (GPRS); Service description; Stage 2. European Telecommunications Standards Institute, Sophia Antipolis, France. Available from http://www.etsi.org/. 1999.

[GSM 11.11] European digital cellular telecommunications system (Phase 2); Specification of the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface (GSM 11.11). European Telecommunications Standards Institute, Sophia Antipolis, France. Available from http://www.etsi.org/. 1998.

[GSM 11.14] European Digital cellular telecommunications system (Phase 2+): Specification of the SIM application toolkit for the Subscriber Identity Module-Mobile Equipment (SIM-ME) interface (GSM 11.14). European Telecommunications Standards Institute, Sophia Antipolis, France. Available from http://www.etsi.org/. 1998.

[Gustafsson 99] E. Gustafsson, A. Jonsson, E. Hubbard, J. Malmkvist, Requirements on Mobile IP from a Cellular Perspective, IETF-Draft, June, 1999.

[Guthery 00] S. Guthery, J. Posegga, Y. Baudoin, J. Rees, "IP and ARP over ISO 7816-3", IETF Internet-Draft, February, 1, 2000.

[Rees 99] Jim Rees and Peter Honeyman: Webcard: a Java Card web server. CITI Technical Report 99-3, Center for Information Technology Integration, University of Michigan. October 1999. http://www.citi.umich.edu/projects/sinciti/smart card/webcard/citi-tr-99-3.html

[Reichenbach 98] Martin Reichenbach, Herbert Damker, Hannes Federrath, Kai Rannenberg: Individual Management of Personal Reachability in Mobile Communication; pp. 163-174 in Louise Yngström, Jan Carlsen: Information Security in Research and Business; Proceedings of the IFIP TC11 13th International Information Security Conference (SEC '97): 14-16 May 1997, Copenhagen, Denmark; Chapman& Hall, London, 1998Herbert Damker, Ulrich Pordesch, Kai Rannenberg, Michael Schneider: Aushandlung mehrseitiger Sicherheit: der Erreichbarkeits- und Sicher-heitsmanager; In: Günter Müller, Kurt-Hermann Stapf: Mehrseitige Sicherheit in der Kommunikationstechnik - Erwartung, Akzeptanz, Nutzung; Addison-Wesley-Long-man, 1998.

[RFC 1945] T. Berners-Lee, R. Fielding, and H. Frystyk. Hypertext Transfer Protocol -- HTTP/1.0, IETF RFC 1945, May, 1966.

[RFC 2396] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform Resource Identifiers (URI): Generic Syntax, IETF RFC 2396, August, 1998.

[RFC 2616] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, Leach, and T. Berners-Lee: Hypertext Transfer Protocol -- HTTP/1.1, June, 1999

[Vaha-Sipila 99] A. Vaha-Sipila, URLs for GSM Short Message Service, IETF-Draft, May 19, 1999.