Proceedings of the 4th WSEAS Int. Conf. on Information Security, Communications and Computers, Tenerife, Spain, December 16-18, 2005 (pp213-217)

Secure Communication System over a GSM Network

RAFAEL ÁLVAREZ[1], JESUS-ALBERTO OLIVER[2], JOSE-FRANCISCO VICENT[3]
and ANTONIO ZAMORA[4]
Departamento de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante
Campus de Sant Vicent del Raspeig, Ap. Correos 99, E-03080, Alicante
SPAIN

This work was partially supported by Generalitat Valenciana grant number GV04B-462

Abstract: - As standard GSM protection methods are now well-known to be insecure, we have designed a new secure communication system over GSM network which is developed using the Symbian OS platform. It uses standard cryptographic methods such as key generators, key exchange protocols and public and private key cipher algorithms. This system allows communicating text, voice and any kind of data securely over GSM as well as over other communication channels.

Keywords: - security, mobile communications, GSM, Symbian, cryptography, Blum Blum Shub, Diffie-Hellman, AES.

1 Introduction

There has been a massive spread of communications over different transport systems over the last few years. Since most communication channels and protocols are not secure, there is a need for protecting the information transmitted in that communication. A lot of products trying to guarantee the privacy, authenticity and integrity of the data sent have been developed together with public key and private key cryptosystems.

Global System for Mobile Communication (GSM) was developed having in mind communication security. It uses the A3 algorithm for authenticity, A8 for session key generation and A5 for ciphering. The latter has four versions: A5/0, without encryption; A5/1, used in Europe; A5/2, a weaker version created in order to export GSM to United States and A5/3 used by the new 3G networks.

Over the last few years, it has been proved that security systems designed for GSM are insecure. Several cryptanalysis have been carried out successfully breaking all the algorithms involved in GSM security. With regards to A3 and A8, we should emphasize the work done at the University of California, at the Swiss Federal Institute of Technology and at IBM [10]. As for A5, we could emphasize the work carried out at the Israel Institute of Technology [2] and by Biryukov, Shamir and Wagner [8], who demonstrated that it was possible to break these algorithms in a few seconds.

There are some products that try to solve these shortcomings, for example the company Cryptophone [5], which suggests the alternative of proprietary hardware. There are some different models including mobile phones, PDAs and land lines. Connections can only be established among products manufactured by the above company. The downside is they are not compatible with other manufacturers; as mentioned before, and their high price.

Our proposal creates a secure communication over GSM, which allows us to use any other communication channel, since the system has been designed regardless of the physical layer.

Furthermore, while the system is aimed at becoming detached from the type of data to be sent, our main priority consists of real time audio transmission. The independent structure of the system allows text messages or any other kind of data to be sent as required.

The system described in this paper is implemented on the Symbian platform in its version 7.0, being the most widespread nowadays in mobile phones. It uses both symmetric and asymmetric cryptography, thus guaranteeing data privacy.

2 Preliminaries

Symbian OS is an advanced, open operating system licensed by the world's leading mobile phone manufacturers including Nokia, Sony-Ericcson and Alcatel. It is designed for the specific requirements

of advanced 2G, 2.5G and 3G mobile phones. Symbian combines the power of an integrated applications environment with mobile telephony, bringing advanced data services to the mass market.

Symbian's graphical interface has two different aspects, SeriesX0 (Series60, Series80 y Series90) and UIQ. SeriesX0 is used in traditional button-based devices, while UIQ is aimed for tactile screens (for example Sony-Ericcson's P800 and P900).

Symbian applications are developed by using C++ with the EPOC 5 (Symbian predecessor) API, as well as every new version add-ons and the SeriesX0 or UIQ interface components, depending on the destination device.

Symbian system services are accessed following the client-server pattern through IPC (inter-process communication) by sending asynchronous requests based on message passing. Each asynchronous request placed with a system service is done by using an active object, which will be activated once the request has been completed. The execution of an active object is controlled by an active scheduler.

The Symbian Multimedia Framework has been used. It provides the necessary methods to access the multimedia functionalities of the device by sending requests to the Media Server (in this particular case, the Audio Streaming API is used).

The audio streaming API is the interface to streaming sampled audio data to and from the low level audio controller part of the Multimedia Framework (MMF).

The client is also notified (for audio playback and recording) when the stream is opened and available for use (opening takes place asynchronously), and when the stream is closed.

This API can only be used to stream audio data, with the data being stored or sourced from a descriptor. Client applications must ensure that the data is in 16 bit PCM format as this is the only format supported. The API does not support mixing. A priority mechanism is used to control access to the sound device by more than one client.

A system mechanism called polymorphic dynamic link library (DLL) is used in this system. A polymorphic DLL contains an interface defined by a gate function and an abstract class with at least one virtual function.

The polymorphic DLL exports the gate function at ordinal 1, which typically constructs a concrete class derived from the abstract interface. The virtual function is then called, and the functions of the class are available.

Examples of polymorphic DLLs in Symbian OS include device drivers, GUI application programs, and many more.

Standard cryptographic methods are used, Blum Blum Shub [4] for key generation, Diffie-Hellman [7] for key exchange and AES [1] for ciphering data.

3 System Design

The system has been designed with three modules that are clearly differentiated: ciphering methods, encryption server and user application (see Fig. 1).

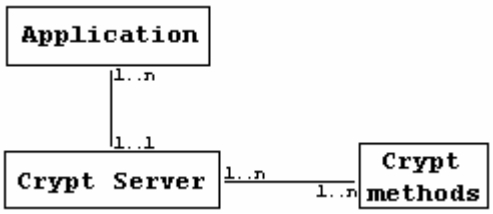

Fig. 1. Module diagram

Cryptographic methods are encapsulated inside DLLs that are used by the encryption server. These libraries have been implemented using the polymorphic DLL API, which allows to use each one randomly and to load them in execution time without having to link against them. The Blum Blum Shub, Diffie-Hellman and AES methods have been implemented, thus being able to extend the system by adding any other key generation, asymmetric cryptography or symmetric cryptography methods.

As for the encryption server, it is a Symbian system service that manages the cipher methods used by loading each method as needed. It also manages the encoding, decoding, key assignation or generation asynchronous requests independently. This allows for several simultaneous requests from different applications, which means that each one of them is starting a session on the server, thus avoiding delays. It also manages priorities of the DLLs installed on the device.

The user application manages the connection establishment and the communication protocol as well as data traffic and its input and output. This incorporates an intuitive and user-friendly graphical user interface, developed by using the SeriesX0 API, in this particular case Series60 (see Fig. 2).

With regards to audio management, the media server is used. It is a Symbian system service that, in this case, allows accessing the audio input and output devices by the means of the IPC system mechanisms.

A streaming process is done for the audio input, involving five buffers of 512 Bytes of capacity each, which are used cyclically for sending requests to the

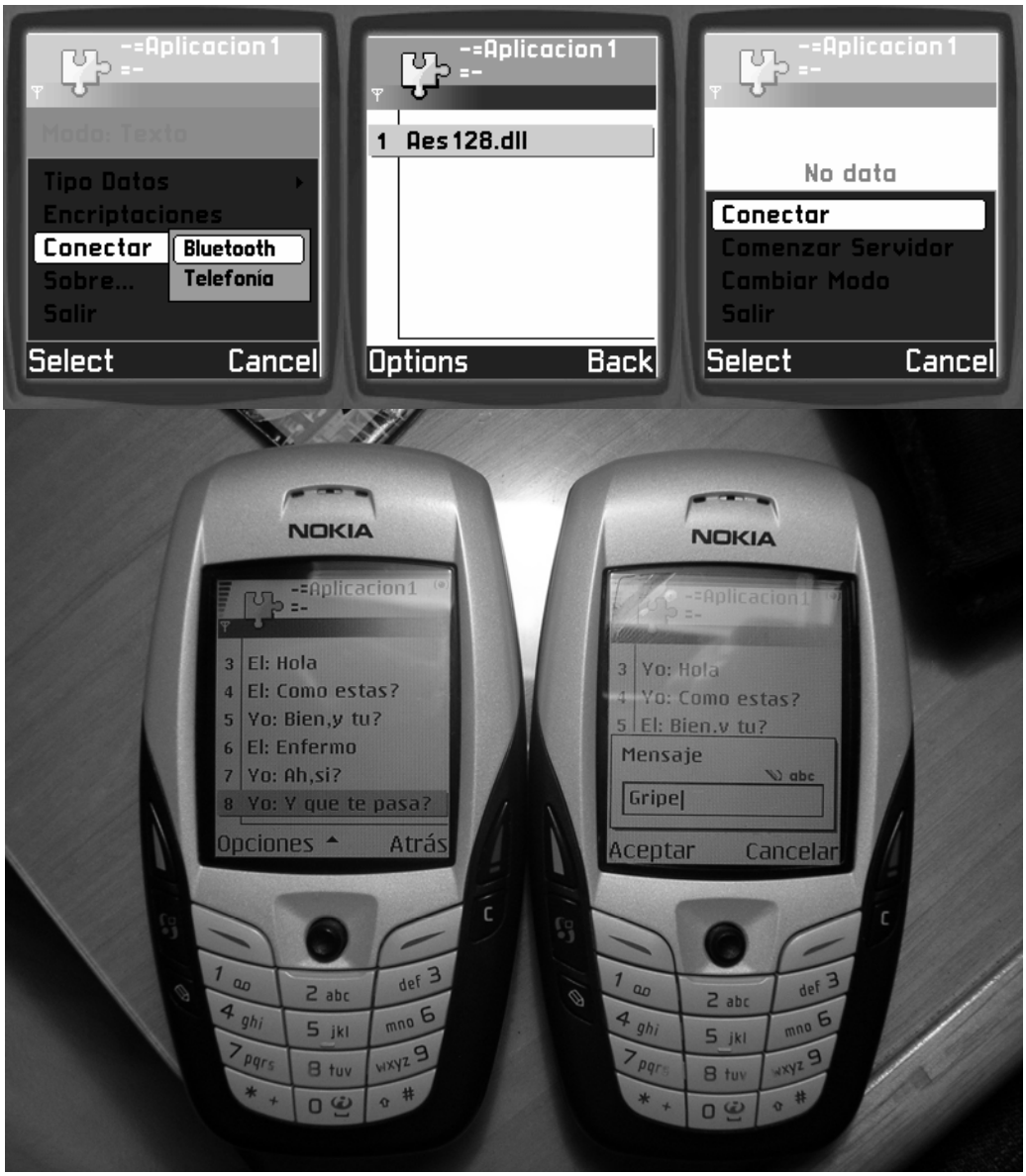

Fig. 2. Application Screenshots

Symbian media server. The latter stores the audio input data in the consecutive buffers and conveys that the process is finished by means of asynchronous signals.

As for the output, the buffers received are accumulated until having reached a data block of approximately a second tenth (1536 Bytes). A request is subsequently done to the media server, which will play it and inform when it is done in order to send the next audio buffer and so on (see Fig. 4).

It can also be highlighted that a connection API has been developed, whereby the system becomes detached from the communication channel to be used: GSM, Bluetooth, TCP/IP are some examples. Nowadays, the development has been focused on Bluetooth, although the tests done over GSM have been successful.

4 Ciphering and Communication

Communication protocol (see Fig. 3):

1 **Negotiating the cipher method to be used**.

Since it is possible to have several DLLs installed on the system, in the beginning of the communication it will be negotiated which of these methods will be used to encode the data, depending on their priority. The client will send to the server the prioritized list of the cipher methods installed, and the server will choose one that both of them have installed

with the greatest priority, returning the relevant method to the client.

2 **Key generation**.

The well-known pseudorandom generator Blum Blum Shub, frequently used to generate session keys, has been used.

3 **Session key exchange**.

The Diffie-Hellman method is used in order to share the session key

4 **Ciphered communication**.

From now on, the data sent will be ciphered.

a Data sending.

As previously stated, a streaming process is done by using 512 Bytes blocks that are encoded using AES by means of requests to the encryption server that will be subsequently sent through the communication channel.

b Receiving and playing back.

As the information is received, it is sent on to the encryption server in order to decipher it and, later, accumulate it until three buffers (1536 Bytes) are ready to be played.

5 Conclusions

We can emphasize the improvement that this system implies as opposed to the basic GSM protection methods, since it provides a secure way of protecting communications by using standard cryptographic methods. It is also worth mentioning its high level of compatibility, which has been achieved as a result of Symbian being such a widespread system in mobile phones, against other commercial alternatives that are based on proprietary hardware.

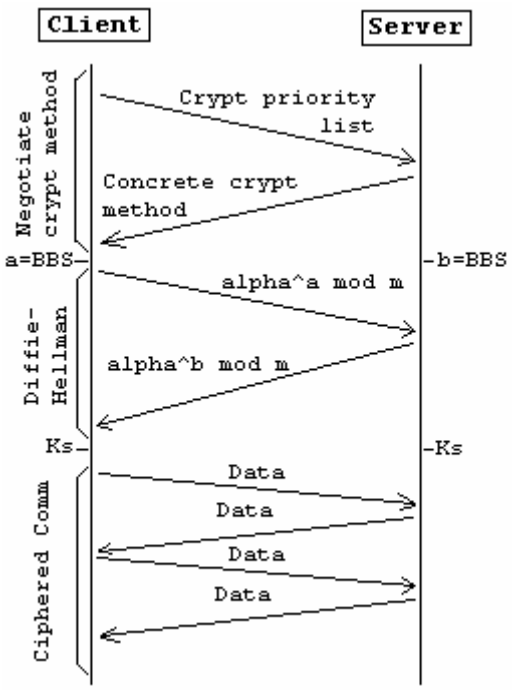

Fig. 3. Protocol definition

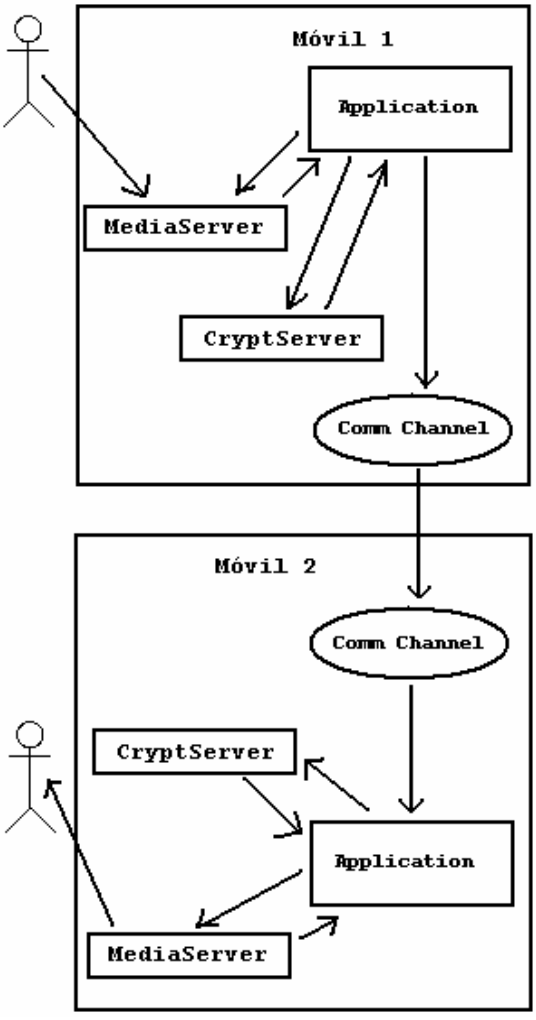

Fig. 4. Data path.

References:

[1] Advanced Encryption Standard (AES). Federal Information Processing Standard Publication, 197. November 2001.

[2] Barkan E., Biham E., Keller N., Instant Cyphertext-Only Cryptanalysis of GSM Encrypted Communication, Advances of Cryptology, CRYPTO 2003, D. Boneh (ed), Santa Barbara, California, USA: Springer-Verlag, LNCS, vol. 2729, 2003, pp 600-616.

[3] Biryukov A., Shamir A., Wagner D., Real Time Cryptanalysis of A5/1 on a PC, Fast Software Encryption Workshop, 2000, pp. 1-18.

[4] Blum, L., Blum, M., Shub, M., A Simple Unpredictable Pseudorandom Number Generator. SIAM J. Comput. vol. 15, 1986, pp. 364–383.

[5] Cryptophone http://www.cryptophone.de

[6] Daem J. and Rijmen V., The Rijndael Algorithm. Springer-Verlag, 2002.

[7] Diffie W.D., Hellman M.E.. New Directions in Cryptography. IEEE Transactions on Information Theory, Vol. 22, 1976 pp. 644-654.

[8] Edwards L., Barker R., Developing Series60 Applications, Addison-Wesley, 2004.

[9] Golic J.D. Cryptanalysis of Alleged A5 Stream Cipher, in Advances in Cryptology-EUROCRYPT'97, W. Fumy (ed.), Berlin: Springer-Verlag,. LNCS, Vol. 1233, 1997, pp 239-255.

[10] GSM Security http://www.gsm-security.net

[11] Harrison R., Symbian OS C++ for Mobile Phones, Wiley, 2003.

[12] Jipping M. J., Symbian OS Communications Programming, Wiley, 2003.

[13] Symbian http://www.symbian.com