

FRAMESEC: a FRAMEwork for the application development with end-to-end SECurity provision in the mobile computing environment

Bringel Filho^{1,2}, Reinaldo Braga², Windson Viana¹, Rossana Andrade^{1,2}

¹Federal University of Ceará (UFC), Computer Science Department (MDCC)
Fortaleza, Ceará, Brazil
{bringel, rossana, windson}@lia.ufc.br
<http://great.lia.ufc.br>

²National Center of High-Processing in the Brazilian Northeast (CENAPAD-NE)
Fortaleza, Ceará, Brazil
reinaldo@cenapadne.br
<http://www.cenapadne.br>

Abstract. Mobile devices offer network connectivity through wireless system communication that makes possible for applications to get access to information at any time and any place. However, the wireless link is more vulnerable to passive and active attacks than the fixed medium in the traditional networks. Then, this kind of connectivity brings security problems to the applications that deal with confidential data, such as mobile commerce. Although wireless communication systems, such as IEEE 802.11, implement protection mechanisms for data transmission, they have reported vulnerabilities and do not guarantee end-to-end security, which is desirable for these applications. Thus, this work proposes an application framework that provides a reusable structure with security mechanisms, which enable end-to-end security in the mobile computing environment.

1 Introduction

An enormous diversity of mobile devices has appeared in the market with possibility of IP connection using wireless links [1] (e.g., cellular with WAP, GPRS and i-Mode support as well as Palms and Pocket PCs with Bluetooth and IEEE 802.11).

The possibility of change from fixed access terminals to mobile devices imposes new challenges in the application development [2], such as low processing power and support for the wireless communication systems. However, the use of these systems and the increasing facility for making information available brings serious risks to security. Unfortunately, these devices offer less sophisticated security mechanisms than the ones offered in the fixed networks.

Generally, mobile devices use one of the existing wireless communication systems (e.g., IEEE 802.11, GSM/GPRS, UMTS, Bluetooth) to transmit data to the application servers, which are located in fixed networks. The transactions that are embedded in these applications and operate with confidential data must keep these data in se-

crecy. This feature is obtained through the establishment of an end-to-end security channel between the mobile device and the application server.

However, the native security mechanisms in the wireless communication systems present vulnerabilities that allow the occurrence of active and passive attacks [3]. Moreover, these mechanisms are limited to protect only the wireless link, without providing the desired end-to-end security for the applications. Therefore, it is also necessary to add security mechanisms to the superior layers (e.g., network, transport and application layers).

When the existing security protocols that operate in the transport and network layers (e.g., S/MIME, SSL/TLS) do not present vulnerabilities, they are impracticable for mobile devices that have resource limitations. Thus, it is necessary the addition of security mechanisms to the application layer. On the other hand, these mechanisms must be built in agreement with the following security requirements: authentication, confidentiality, non-repudiation and integrity. There are some cryptography algorithms (e.g., DES) that are created to deal with such requirements, which can be implemented or just reused, if they are available in the application development platforms (e.g., Personal Java, J2ME, SuperWaba and C) supported by the mobile device.

The development of these mechanisms is a difficult task, mainly, when it is performed by developers who do not have enough knowledge about security. The difficulty is increased because the security requirements generally do not occur in an isolated way. Thus, it is necessary to develop these mechanisms aiming to attend the combinations of those requirements (e.g., confidentiality and integrity). Therefore, the motivation of this work is the need for providing to the application developers a structure that enables to develop and to reuse a mechanism for the end-to-end security provision for applications in the mobile computing environment.

The remaining of this paper is divided as follows: the next section presents related works. Section 3 introduces the FRAMESEC proposal. Section 4 shows a case study that demonstrates the use of FRAMESEC, and Section 5 presents our final remarks and future work.

2 Related Works

[13] presents a solution that was developed in J2ME for end-to-end security provision in the mobile computing environment. It provides authenticity and confidentiality in the transmission between a mobile device and an application server. The confidentiality is supplied through the AES Rijndael algorithm. Therefore, the authentication is performed by the key exchange mechanism used in the solution. However, such solution does not provide services with integrity and non-repudiation.

In [12], the authors present a solution for the security application development for mobile devices, called MobiS (Mobility and Security). MobiS has 3 (three) versions: Symmetrical MobiS, Anti-symmetrical MobiS and Hybrid MobiS. [12] describes Symmetrical MobiS version that provides confidentiality and data integrity. This solution allows you to configure the cryptographic algorithms that will be used to guarantee security. Thus, its robustness is directly related to the combination of the used

algorithms as well as the size of cryptographic key. However, Symmetrical MobiS does not offer authenticity and non-repudiation services.

Therefore, the end-to-end security solutions found in the literature, which can be applied to the mobile computing environment considered in this paper, do not offer support to all security requirements. Our work then proposes FRAMESEC to provide end-to-end security to the development of applications for mobile devices.

3 FRAMESEC

FRAMESEC is a framework that offers to the applications in a mobile computing environment the following security services: confidentiality, integrity, authenticity, non-repudiation as well as the combination of these services. The reuse of the FRAMESEC structure and implementation for the development of security mechanisms provide end-to-end security. Besides the security services, it also offers inter-process communication in a transparent way, which is shown below.

The FRAMESEC definition is based on the documentation of the Tropic pattern language [4] as well as Forwarder-Receiver [5], Strategy [6] and NullObject [7] patterns. UML (Unified Model Language) [8] is the notation used for designing class diagrams and other models necessary to the FRAMESEC definition. FRAMESEC instances are implemented in the J2ME MIDP/CLDC 1.0 platform [9], for example, the cryptographic algorithms used to compose the framework hot spots, which offer portability with a large variety of mobile devices (MD) types, such as Palms, mobiles telephones and Pocket PCs. The cryptographic algorithm implementations are freely distributed for the Legion of the Bouncy Castle group [10], which develops Java's APIs on the security area.

3.1 FRAMESEC Requirements

The general FRAMESEC architecture was defined through the Cryptographic Metapattern pertaining to the Tropic, combined to the Forwarder-Receiver pattern that provides the inter-process communication in a transparent way. On this architecture, the data communication involves two Peers (i.e., mobile device and application server), each one possessing a Forwarder and a Receiver, which are responsible for sending and receiving operations, respectively. A Peer can communicate with one or more Peers simultaneously. For this, it is necessary to keep a repository containing their physical locations in order to supply the transparent inter-process communication. Thus, the Forwarder and Receiver use the Entry repository, which is responsible for store and mapping Peer names to real address. At each entrance on this repository, it is informed the Codifier and the Decodifier, which will be used later by the Forwarder and Receiver.

On the sending process, the Forwarder consults the Entry repository by retrieving the physical location of the communicant Peer and the Codifier that will be used to perform the cryptographic transformations on the data to be sent, in accordance with the security requirements defined by the application. At its turn, the Receiver also

consults the repository by retrieving the Peer information from which it will get the data transmitted. When it receives the data, it requests to the Decodifier the execution of the inverse codification process, by retrieving the sent message. Codifier and Decodifier use the cryptographic algorithms to cipher and decipher the information.

At implementation level, a Forwarder, a Receiver and an Entry will be necessary for each different communication protocol (e.g., HTTP) used for the data transmission. Fig. 1 illustrates the preliminary class model of FRAMESEC.

Tropyc language [4] describes the following patterns: Information Secrecy, Message Authentication, Message Integrity, Sender Authentication, Secrecy with Authentication, Secrecy with Signature, Secrecy with Integrity, Signature with Appendix, and Secrecy with Signature with Appendix, that correspond to the confidentiality, integrity, authenticity and non-repudiation services, and the possible combinations between them. Therefore, the FRAMESEC proposes to offer the innovate services described in the Tropyc. For each service, a different strategy of encryption and decryption was defined (e.g., Secrecy, Integrity), using the Strategy pattern, as shown on the last FRAMESEC class diagram illustrated by the Fig. 2. In addition, the null encryption strategy was added through the NullObject standard.

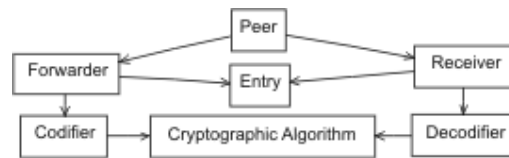


Fig. 1. Preliminary Class Model.

3.2 FRAMESEC Instantiation

In order to instantiate FRAMESEC to be incorporated in the application, a developer must observe the Use Cases illustrated on the Fig. 3. In the Use Case "Execute Algorithm Evaluation", the PEARL [11] tool is employed for performance evaluation of the cryptographic algorithms that will compose the hot spots of the FRAMESEC instances. Therefore it is possible to estimate the algorithm execution time for a determined entrance size, verifying the viability of the application of the algorithm on the FRAMESEC instance. The Use Cases are described:

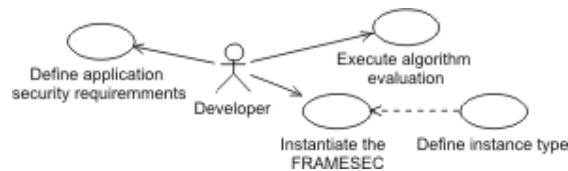


Fig. 2. FRAMESEC Use Cases.

Use Case: Define security application requirements
 Actor: Developer

Description: It is initiated when a developer wants to add security mechanisms in an application to be developed. In this phase, security requirements (or combination requirements) are identified, such as confidentiality and integrity.

Use Case: Execute Algorithm Evaluation

Actor: Developer

Description: It is initiated when a developer installs the PEARL[11] tool on the mobile device to perform the evaluation of the algorithms according to the security requirements defined on the Use Case "Define application security requirements". After the performance evaluation, an analysis of the results is performed in order to identify the cryptographic algorithms that will compose framework hot spots.

Use Case: Instantiate the FRAMESEC

Actor: Developer

Description: It is initiated when the developer wants to instantiate the FRAMESEC, having defined security application requirements, in addition to the algorithms that will compose the framework hot spots. However, the developer must decide if he/she wants to instantiate the framework completely (Use Case "Define instance type"), instantiating all transparent inter-process communication structure in addition to the encoding (Codifier) and decoding (Decodifier) structures. Otherwise, the FRAMESEC instance can be generated from Codifier and Decodifier, and the developer will be responsible for the connection establishment.

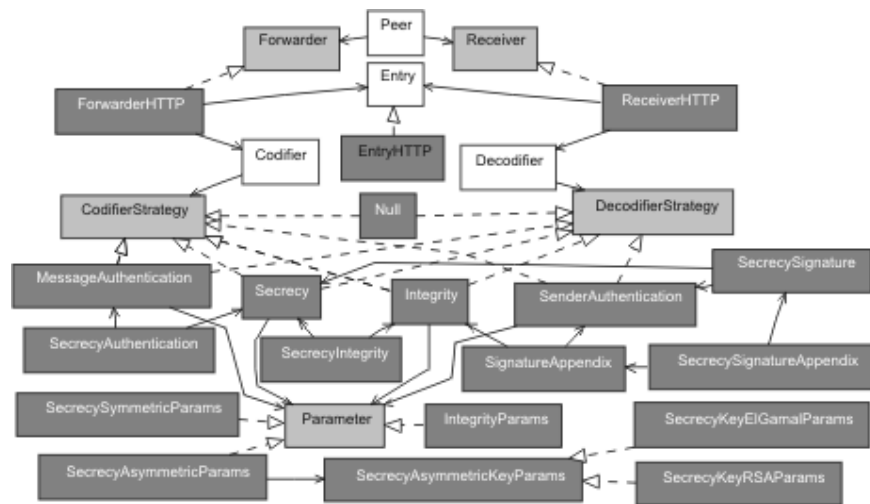


Fig. 3. Detailed FRAMESEC class diagram.

4 Case study

The MobileMulta application (implemented in [12]) was developed to allow users (i.e., the guard) to store transit infractions via a MD (e.g., Palm). The infractions registered on the MD (that operates off-line) are stored in a local data repository to be sent to the application server later on.

On the transmission of the infractions using Palms, a connection IrDA (Infrared Association Date) or Bluetooth is established with a mobile telephone that uses the GSM/GPRS system for the establishment of a data connection; in the case of the Sony Ericsson P800 smart phone, a data connection is established directly with the operator network. Fig. 4 illustrates the scenery of the application use.

Observing the Use Cases defined on Section 3.2, an application possesses confidentiality and integrity. The confidentiality is the scope of the InformationSecrecy standard in the Tropic language, which is defined on the FRAMESEC through of the SecrecyStrategy class. On the other hand, the integrity is the target of the MessageIntegrity standard, defined on the FRAMESEC through the IntegrityStrategy class. Thus, the SecrecyIntegrityStrategy class was used to combine the two services.

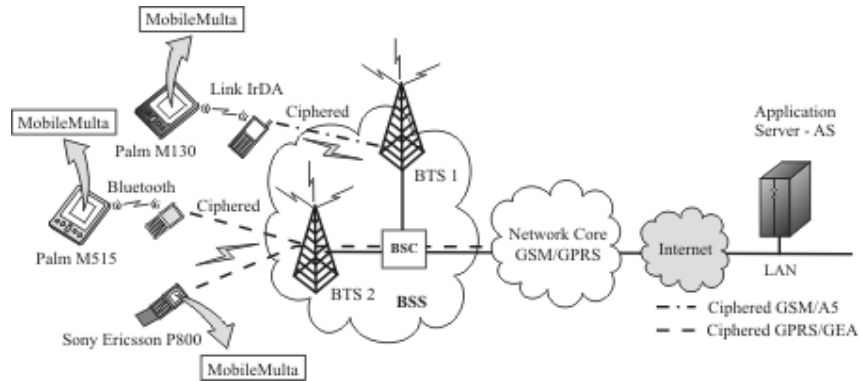


Fig. 4. Components of the MobileMulta application.

The next step consists on the identification of the cryptographic algorithms that will compose the FRAMESEC hot spots. To facilitate this task, the PEARL tool was used to get the performance parameters of the algorithms evaluated on these devices. Besides these parameters, it is necessary to study the security level supplied by the algorithms, searching for a balance between performance and security. Based on a study performed in [11] and the evaluation results, the AES (confidentiality) algorithm and the hash function Sha-1 (integrity) were chosen. The code of the application that uses the FRAMESEC is presented next:

```
// FRAMESEC Instantiation Code (MobileMulta)
import Framesc.Peer;
import FRAMESEC.Codifier;
import FRAMESEC.Decodifier;
```

```

import FRAMESEC.EntryHTTP;
import FRAMESEC.ForwarderHTTP;
import FRAMESEC.ReceiverHTTP;
import FRAMESEC.parameter.SecretcyParams;
import ramesec.codifierstrategy.SecretcyIntegrityStrategy;
Peer mobileMulta;
EntryHTTP peers;
Codifier codifier;
Decodifier decodifier;
SecretcySymmetricParams params;
byte[] plaintext, ciphertext, key;
codifier = new Codifier(
new SecretcyIntegrityStrategy("AES", "SHA-1"));
decodifier = new Decodifier(
new SecretcyIntegrityStrategy("AES", "SHA-1"));
params = new SecretcySymmetricParams(key);
codifier.initialize(params);
decodifier.initialize(params);
peers = new EntryHTTP();
peers.addEntry("SA", "http://200.17.37.12:8080", "post",
codifier, decodifier);
mobileMulta = new Peer(new ForwardHTTP(peers),
new ReceiverHTTP(peers));

```

When the user wants to send the fines registered on the device, the code described below is executed:

```
mobileMulta.send(plaintext, "SA");
```

On the other hand, in order to receive the transmitted information by the application servers, the following code is executed:

```
plaintext = mobileMulta.receive("SA");
```

Therefore, both the information encryption and decryption processes are encapsulated on the FRAMESEC, what makes easier and more transparent a safe communication end-to-end between the mobile devices and the application server. The code implemented on the server application, that also must instance the FRAMESEC, is similar to the code presented above, and, for this reason, will not be detailed in this case study.

5 Conclusion

The construction of security mechanisms to be incorporated to the applications is not an easy task, specially for lay developers on security. The definition of a framework that facilitates the construction of these provision mechanisms for the end-to-end security is, then, necessary. Therefore, the FRAMESEC considered on this work intend to supply the reuse of its structure and implementation for the construction of mechanisms for the provision of end-to-end security on the mobile computation environment.

FRAMESEC does not supply support for key exchange mechanism, as well as for the configuration of algorithm that will be used in the FRAMESEC instances. It is proposed then, in future works, the addition of these mechanisms. Moreover, the use of the FRAMESEC adds security mechanisms to the application layer, independently of the existence of security services in the inferior layers of the wireless communication systems, which can generate redundancy on the security services. Thus, it is also proposed the study of the integration of the FRAMESEC with security mechanisms inherent to wireless communication systems.

6 References

1. Dornan, Andy. The Essential Guide to Wireless Communication Applications, Prentice Hall Inc., 2001. ISBN 0-13-031716-0.1
2. Loureiro, A.; Minelli, A. Uma Ferramenta para Desenvolvimento de Aplicações para Dispositivos Móveis. 20º Simpósio Brasileiro de Redes de Computadores, v. 2, p. 571 - 586, 2002.
3. Stallings, W. Network Security Essentials: applications and standards. New Jersey: Prentice Hall, 1999.
4. Braga, A. M.; Rubina C. M. F.; Dahab, R.; Tropyc: A Pattern Language for Cryptographic Software. p. 1-27, jan. 1999.
5. Buschmann, F. Pattern-Oriented SoftwareArchitecture - A system of pattern. John Wiley, 1996.
6. Gamma, E. R. Helm; Johnson, R.; Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Reading, MA: Addison Wesley, 1995.
7. Anderson B., "Null Object", Pattern Language Of Programming – PloP '96.
8. Rumbaugh, J.; Jacobson, I.; Booch, G. The Unified Modeling Language Reference Manual. Addison-Wesley, 1998.
9. Java 2 Platform, Micro Edition (J2ME). Available in <<http://java.sun.com/j2me>>. Access: 10 mar. 2003.
10. Legion of The Bouncy Castle. Available in <<http://www.bouncycastle.org>>. Access: 20 jun. 2003.
11. Filho, Bringel; Viana, Windson, Castro, R. M. C. PEARL: a Performance evaluator of cryptographic algorithms for mobile devices. In: (MATA 2004). v.3284, To Appear, 2004.
12. Viana, Windson; Filho, Bringel; Magalhães, Katy; Giovano, Carlos; Castro, Javam de; Andrade, Rossana. Mobis: A Solution For The Development Of Secure Applications For Mobile Device. In: ICT, 11th, Fortaleza-CE, Brazil. Proceedings to Appear, 2004.
13. Itani, W.; Kayssi, A.I.; J2ME End-to-End Security for M-Commerce. Wireless Communications and Networking - WCNC 2003. v. 3, p.16-20, mar. 2003.