University Authentication System Based on Java Card and Digital X.509 Certificate

Maria Ortega[1], Sergio Sánchez[2]

[1] Faculty of Computer Systems Engineering, Technological University of Panama, Panama, Panama

[2] School of Telecommunications Engineering, Polytechnic University of Madrid, Madrid, Spain

Abstract

This article presents a solution to the problem of strong authentication, portable and expandable using a combination of Java technology and storage of X.509 digital certificate in Java cards to access services offered by an institution, in this case, the technology of the University of Panama, ensuring the authenticity, confidentiality, integrity and non repudiation.

Keywords: *Authentication, Java Cards, PKI, Smart Cards, X.509 Certificate.*

1. Introduction

With technological advances and increased processing power of today's computers, there is the need to increase security in authentication processes and incorporate new technologies to increase and provide a higher level of security. Various institutions have authentication systems for data access and authentication applications characterized by the use of username / password [1], called access classic, presenting the problem that can be easily violated with current technology reducing security applications. However, there are other institutions that have decided to have a safer technology in protecting access to applications and information [2].

In the specific study case discussed in this article, the Technological University of Panama (UTP), it has a Public Key Infrastructure (PKI) [1] used only by teachers to record grades and by administrative for the annual evaluation, but that is not available at present to all members of the university community. The aim of this work is to improve the setting of access to services in the UTP trying to extend the use of PKI [3] and performing an integration of technologies that provide greater certainty for all users (teachers, administrators and students) and to ensure a flexible, secure and guaranteed access to services.

2. Methodology of work

The methodology used in conducting this work consists of several phases. The first is the analysis. It has identified the need or demand, areas for improvement, and has made a study of the current situation of the problem, taking into account the specific case study of the Technological University of Panama (UTP) which has a PKI structure, used only by teachers to record grades, by administrative for the annual evaluation but today is not available to all members of the university community.

After the analysis phase it has addressed the design phase. It makes a proposal that tries to meet the identified demands and solve all the problems and / or deficiencies identified in the analysis. This design, from the technical point of view, is a modeled logical architecture through diagrams of a unified modeling language (Unified Modeling Language - UML).
As a result of the design there are models from the view component showing the set of entities in the proposed architecture as a solution and the relationship between them. Also, this point addresses the integration of X.509 certificate and Java Card as part of the authentication system. Likewise, there are two patterns of communication from the view of diagrams sequence, one to obtain the certificate, it means the mechanism by which the user gets an X.509 identity certificate that stores in its smart card, and another for access to a generic service, that is how the user uses the certificate to securely access and with guaranties to the offered services.

After the design, goes the implementation phase. This phase is to develop a small demonstrator of the architecture that can be used as a reference implementation and for verifying the functionality of the design. These individual demonstrator tests are carried out based on the

results obtained, will influence the design stage to provide feedback and improve the solution.

3. Related Works

In related works, we can mention the work of Smart Cards with PKI to data access control, by [4], where they use smart cards with support for managing public key certificates and PKI asymmetric cryptography to implement access control to data in health information systems and to protect data confidentiality, integrity, authentication and non repudiation.

This presents several benefits to support the implementation of PKI smart cards:
a. The private key is generated and stored on the smart card. The operating system of the smart card key prevents exposure outside the card.
b. The digital certificate is stored in the smart card itself, instead of having it on a hard drive. This makes the authentication and non repudiation possible because the user of the card has a PIN that controls access to the required keys to provide non-repudiation and data signing.
c. The smart card information is encrypted, which means secure data recovery, transfer and storage.

Another related work is that of [2], which deals with web authentication. Below are the two most important results obtained:
1. Steel, on the server side. It is the system able to generate and manage digital certificates and is the entity in which you can trust, to be sure that the certificate belongs to the right person and not an intruder.
2. JCCM (Java Card Certificate Management) at the client side. Manages user agents to generate and verify digital signatures using the Netscape browser and uses a smart card, specifically the Java Card as a cryptographic device for storing certificates.

Furthermore, other related work is [5], where the problem consists in improving the authentication mechanism because of several attacks on the web. Here are some percentages:
a. Request Forgery (11% of websites).
b. Insufficient Authentication (10% of websites).
c. HTTP Response Splitting (9% of websites).

Therefore, in this work the traditional mechanism for client authentication is similar to server authentication. The Web server requests a digital certificate to the client to check if it is the person who claims. It is noteworthy that only you can use client authentication when the server requests a certificate from a client and it should be noted that not all servers support client authentication.

4. Proposal Architecture

To develop a system that satisfy each necessary requirements for providing security to a user application using Java Card smart cards and X.509 certificates, it has developed a based component architecture that provides an overview of the entities involved, its sequence of actions and communication.
The components that make up this architecture (Fig. 1) are described below:

• User: A person who accesses a resource or service offered by an application or system.

• Service Provider: Entity that will provide the services required by users.

• PC device from which the user will access the service. The user uses the PC to interact with other system components.

• Java Card Reader: You are connected to the PC and facilitates communication between the card and Java Card user applications.

• Java Card: Stores keys and certificates of identity. Through reader communicates with the Java Card runtime environment to provide access to the applications required by the user, manage user access control and provide information on the digital certificate to access applications or services offered by the institution.

• PKI: public key infrastructure that will be associated with the organization and will be responsible for delivering the certificates to users. Consist of a Registration Authority (RA) to validate the user record and a CA to issue and check the status of a user's X.509 certificate.

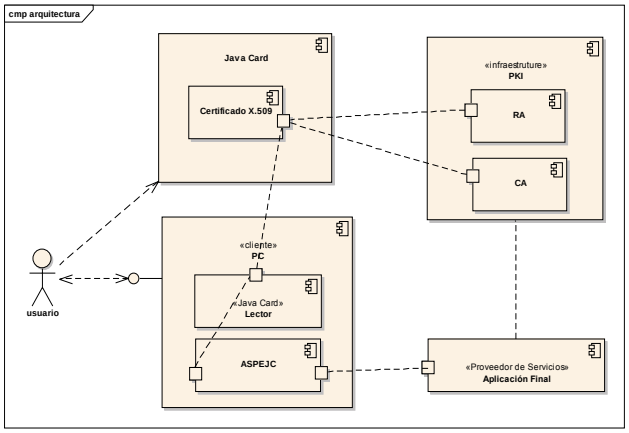

Fig. 1: View of Components: Architecture Logic and relationships.

The user has a Java card that communicates with the PC, through the Java card reader to make use of the offered services by the service provider (Final Application).
Before accessing the services of the final application requires a first phase of user authentication based on a Personal Identification Number (PIN) that allows you to access the services of the Java Card. Once validated as the owner of the card it is necessary to authenticate with the service provider, for it makes use of PKI associated with the institution, which is responsible for issuing certificates to users using the CA and RA. The user is authenticated using the certificate that is stored in the Java card before entering the final system.

4.1 Communication scheme

The Figure 2 shows the communication architecture of the Java card with card reader or CAD. The exchange of information and commands between the board and the DAC is done through Protocol Data Units Application or APDUs (Application Protocol Data Units). The APDUs are packets of information with a specific format according to ISO 7816 [6]. We define two types of APDU, the Command APDU, which are sent to the card, and the Response APDU, which are sent from the card in response to a Command APDU.

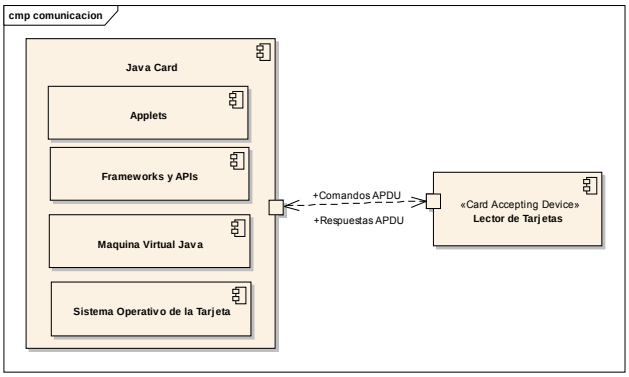

Fig. 2: Communication card and CAD.

4.2 Processes ASPECJ

The mechanism ASPEJC defines three processes essential for authentication and access to the final application, the first is the storage of digital certificates within the Java Card, the second to obtain the certificate stored in the Java Card and the third to access to services offered by the institution using the X.509 digital certificate stored in the Java Card.

4.3 X.509 Certificate store Java Card

To ensure the portability of the X.509 certificate, it must be stored in a portable media, from which it can be claimed and validated. The portable storage way to use is Java Card.

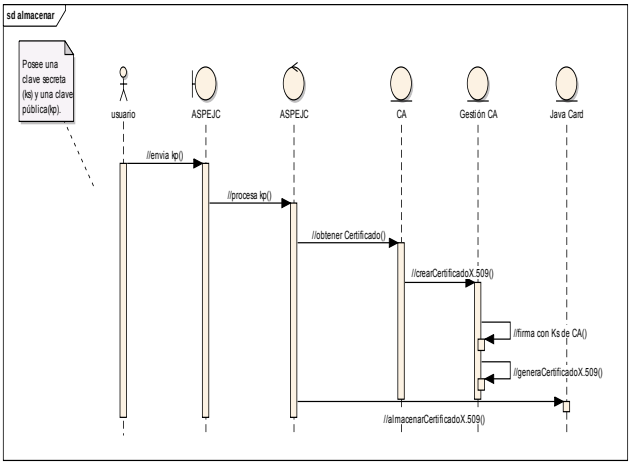

Fig. 3: Sequence storage of X.509 Certificate

The Figure 3 shows the sequence of X.509 certificate store within the Java Card, in this process involves the component ASPEJC who receives the keys of the user and provides them to the CA, which is responsible for signing and certificate generation, which is transformed by the ASPEJC and bytes sent through commands APDUs to the Java Card.

Member has a public key (Kp) and secret key (Ks), previously generated by cryptographic algorithms, but needs a trusted third party (TTP) as the RA to validate the user registration and the CA, which is the entity that can issue and check the status of an X.509 certificate.

To create the certificate, the user sends his Ks to CA, this makes the process of issuing the certificate and signs his own secret key for the record that this certificate is valid to all members of that environment. The user Kp is made known to all, while his Ks is private, only the user must know the owner of the certificate. It is noteworthy that it is assumed that it is impossible to obtain the Ks from Ks.

Once issued the certificate, the Ks is stored inside the card and Kp Java within the digital certificate that is stored in the Java Card for safety being used after in accessing any services needed by the user. Therefore, the user has a portable authentication element, safe and flexible.

4.4 Obtaining X.509 Certificate

Once the X.509 certificate is stored in the Java Card, it can be used in a portable and secure because you cannot get the same information, unless you use the ASPEJC mechanism.
To get X.509 certificate it must be used the ASPEJC and request a digital certificate. ASPEJC communicates with the Java Card, which asks the user validation credentials (PIN), once the user has access to Java Card services, ASPEJC extracts the X.509 certificate using APDUs requests that return a buffer of bytes that are assembled to form ASPEJC X.509 certificate.
ASPEJC design lets extracting the X.509 certificate of the card, but also allows operations such as signing documents using the stored certificate, without being removed from inside the Java Card. These operations are executed by the applet stored in the Java Card.

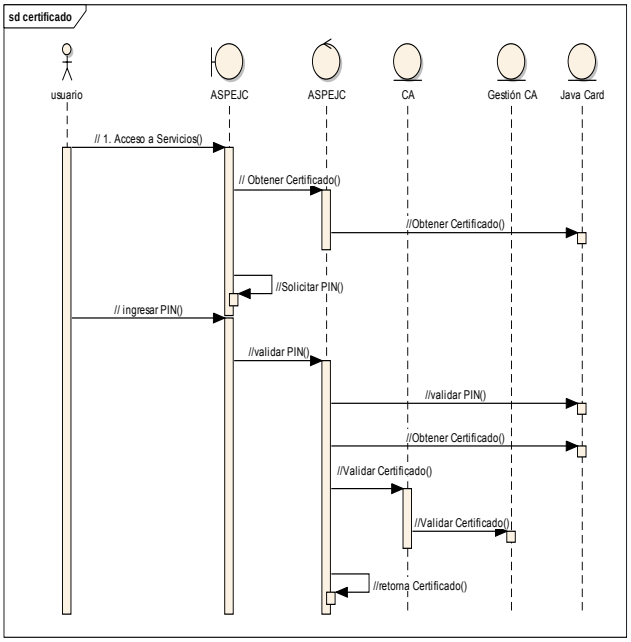

Fig. 4: Obtaining X.509 certificate from the Java Card

The Figure 4 shows the diagram sequence of how the user gets the X.509 digital certificate stored on the identity smart card.

4.5 Access to Service

The ASPEJC component is responsible for providing secure access to a service provided by the final application through a series of features and commands (APDUs).
The sequence diagram for the access to a service is shown in Figure 5. It shows how users access and authenticate services security using X.509 digital certificate stored on the smart card.

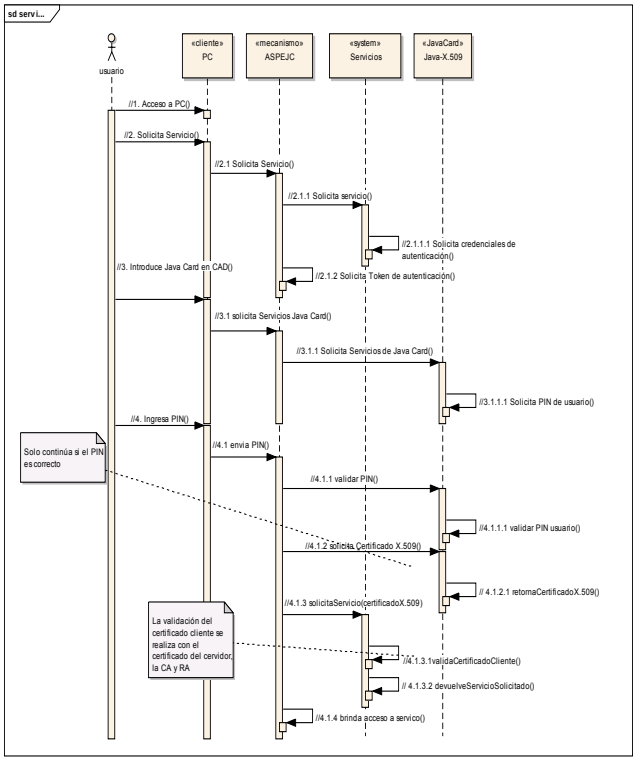

Fig. 5: Sequence Diagram of access to a service

The user accesses the PC to make use of a service offered by the final system. Needs to authenticate to access the service and introduces Java card which in turn prompted for authentication to verify that the user is the owner of the card (this authentication can be performed with the use of biometric technology, which was not applied in this project due to financial constraints, technology and time, but is planned as a future upgrade).

The user enters his PIN to identify himself to card. The card validates the user's PIN, if the PIN is correct provides access to the services offered by the installed applets in Java Card. In this second authentication, end service access, it uses the X.509 digital certificate stored on the card that shows the holder's identity to any entity through a mechanism, such as challenge-response that involves both the Ks and Kp of the user, both stored on his card. This will authenticate the user and can securely access the final system.

Once submitted the results of the overall design phase then examines the implementation phase, in which three parts are considered. The first is the development of basic authentication module. The second is the integration of

X.509 certificate identity in Java card and the third includes the development of the final application module, which consists of a development of a module of the final application in a small demonstrator to verify the functionality of the proposed solution.

5. Implementation and Results

To verify the correct functioning of the architecture described in previous sections, we have developed 3 projects that represent the Java Card and Java Card applet, the card reader and Broker (Middleware ASPEJC) and End-User Application (Host). Each of these projects fulfills a specific purpose and communicates with each other to recreate a complete environment to validate the proposed architecture.

5.1 Applet Java Card

This project aims to define the required functionality by the Java Card applet for the storage and retrieval of X.509 digital certificate and the signed and authentication methods inside the card.

Table 1: Operations ASPEJC Applet

COMANDO O PETICION APDU							
Cabeza				***Datos***			
CLA	*IN*	*P 1*	*P 2*	*LC*	*DATA FIELD*	*LE*	
0x80	0x22						Validar PIN
	0x26						Firmar Documento
	0x38						Extracción del Certificado.
	0x40						Almacenamiento del Certificado

The Table 1 summarizes the main operations carried out for java card applet ASPEJC mechanism. Once implemented these features are deployed to the emulator through the JCWDE simulator interface, as shown in the figure below:

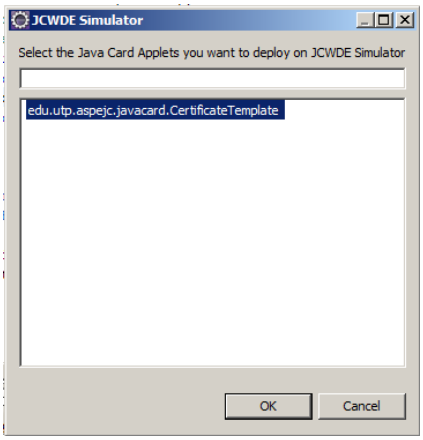

Fig. 6: Interface Simulator JCWDE

Once deployed the applet, you can make use of the offered services. Figure 7 shows the main screen emulator JCWDE, where you can see the java card platform used and the port and protocol listening emulator to user requests.

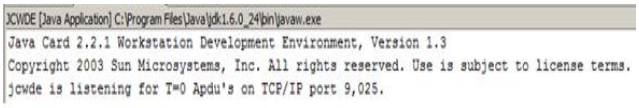

Fig.7: Window JCWDE initial simulator

5.2 ASPEJC

Aspejc project aims to provide communication services between Java Card and the end user application; it is responsible for sending and receiving command APDUs to and from the Java Card reader.
This project makes use of the libraries of the Java Card framework apduio.jar and bcprov-jdk16-146.jar of the BouncyCastle.

To validate the architecture and the proposed mechanism was implemented with console module to simulate the interaction between the user, the Java middleware card (ASPEJC) and the final application. This component is called ASPEJC Test Console. Figure 8 shows the initial screen test console.

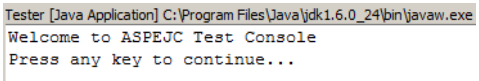

Fig. 8: Initial screen of the console ASPEJC

The test console can see part seeing part of the communication process between the component and the

Java Card ASPEJC, and communication with the final application.

The Figure 9 shows the command APDUs of ASPEJC component interaction with the Java Card. These commands represent the functionality of X.509 certificate stored in the Java Card and the command to sign a text. These commands are displayed through ASPEJC Test Console.

```
.....................................................
Saving Certificate...
CLA: 80, INS: 40, P1: 00, P2: 00, Lc: 50, 30, 82, 01, d4, 30, 82, 01, 7e, 02, 09, 00, ba, eb, a8, bf, 82, 48, fc, ca,
CLA: 80, INS: 40, P1: 00, P2: 00, Lc: 50, 0c, 30, 0a, 06, 03, 55, 04, 0a, 13, 03, 55, 54, 50, 31, 0d, 30, 0b, 06, 03,
CLA: 80, INS: 40, P1: 00, P2: 00, Lc: 50, 31, 31, 31, 31, 31, 33, 31, 39, 31, 32, 35, 32, 5a, 17, 0d, 31, 32, 31, 31,
CLA: 80, INS: 40, P1: 00, P2: 00, Lc: 50, 04, 0a, 13, 03, 55, 54, 50, 31, 0d, 30, 0b, 06, 03, 55, 04, 0b, 13, 04, 46,
CLA: 80, INS: 40, P1: 00, P2: 00, Lc: 50, 00, c1, ab, a3, 76, 07, 70, 35, 9c, 48, 4a, 75, a6, 10, 58, 38, 1c, 73, 89,
CLA: 80, INS: 40, P1: 01, P2: 01, Lc: 48, 01, 01, 05, 05, 00, 03, 41, 00, 3a, f7, a3, aa, 23, 32, f3, 37, b6, c1, 9b,
CLA: 80, INS: 50, P1: 01, P2: 01, Lc: 40, c1, ab, a3, 76, 07, 70, 35, 9c, 48, 4a, 75, a6, 10, 58, 38, 1c, 73, 89, 52,
CLA: 80, INS: 54, P1: 01, P2: 01, Lc: 03, 01, 00, 01, Le: 00, SW1: 90, SW2: 00
CLA: 80, INS: 42, P1: 00, P2: 00, Lc: 40, c1, ab, a3, 76, 07, 70, 35, 9c, 48, 4a, 75, a6, 10, 58, 38, 1c, 73, 89, 52,
CLA: 80, INS: 46, P1: 01, P2: 01, Lc: 40, ab, e4, e3, 98, 86, d1, 99, 43, b5, 06, 21, 95, db, 3b, f1, d1, 8e, da, 7b,
-------------------------------------------------
CLA: 80, INS: 58, P1: 00, P2: 00, Lc: 00, Le: 00, SW1: 90, SW2: 00
Press any key to continue...
.....................................................
Sign text...
CLA: 80, INS: 38, P1: 00, P2: 00, Lc: 00, Le: 64, 30, 82, 01, d4, 30, 82, 01, 7e, 02, 09, 00, ba, eb, a8, bf, 82, 48,
CLA: 80, INS: 38, P1: 01, P2: 00, Lc: 00, Le: 64, 04, 0b, 13, 04, 46, 49, 53, 43, 31, 0e, 30, 0c, 06, 03, 55, 04, 03,
CLA: 80, INS: 38, P1: 02, P2: 00, Lc: 00, Le: 64, 02, 50, 41, 31, 0b, 30, 09, 06, 03, 55, 04, 08, 13, 02, 50, 41, 31,
CLA: 80, INS: 38, P1: 03, P2: 00, Lc: 00, Le: 64, 06, 09, 2a, 86, 48, 86, f7, 0d, 01, 01, 01, 05, 00, 03, 4b, 00, 30,
CLA: 80, INS: 38, P1: 04, P2: 00, Lc: 00, Le: 48, 01, 01, 05, 05, 00, 03, 41, 00, 3a, f7, a3, aa, 23, 32, f3, 37, b6,
CLA: 80, INS: 38, P1: 05, P2: 00, Lc: 00, Le: 00, SW1: 90, SW2: 00
```

Fig.9: Command APDUs implemented by ASPEJC to store a X.509 certificate

5.3 Prototype of final application

Finally, the project prototype application purpose is to create a test application that works as end-user application and use the offered services by the Java Card to digitally sign a document and transmit to the board a new certificate or read it.

For integration with the application X.509 digital certificates were created with Keytool device and open ssl. Figure 10 shows the contents of X.509 digital certificate stored in the Java Card and visualized with ASPEJC Test Console.

```
Tester [Java Application] C:\Program Files\Java\jdk1.6.0_24\bin\javaw.exe
Jan Vossaert
Aftanse steenweg 3
9000
Gent
man
1987
[
[
  Version: V3
  Subject: EMAILADDRESS=cautp@utp.ac.pa, CN=CAUTP, OU=FISC, O=UTP, L=Panama, ST=PA, C=PA
  Signature Algorithm: SHA1withRSA, OID = 1.2.840.113549.1.1.5

  Key:  Sun RSA public key, 512 bits
  modulus: 11819011242183582052027873662154906123351356552703036785559046232043801316364972668497515
  public exponent: 65537
  Validity: [From: Sun Nov 13 14:04:43 COT 2011,
              To: Wed Nov 10 14:04:43 COT 2021]
  Issuer: EMAILADDRESS=cautp@utp.ac.pa, CN=CAUTP, OU=FISC, O=UTP, L=Panama, ST=PA, C=PA
  SerialNumber: [    a0a4ba03 51899309]

Certificate Extensions: 3
[1]: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: FF B9 2A 5A B8 0B 3C 35   1C 69 4C 3D 0D A0 0D F1  ..*Z..<5.iL=....
0010: 9F D5 0D AD                                        ....
]
]

[2]: ObjectId: 2.5.29.35 Criticality=false
```

Fig.10: X.509 Certificate processed by ASPEJC

The Figure 11 shows the integration with the final application, this example validates the authenticity of the user and the digital certificate. This validation is performed using client certificate (stored in the Java Card), the server's certificate (end use), and the CA certificate. The example is a text signed by the customer and the corresponding verification by the server application.

```
  Algorithm: [SHA1withRSA]
  Signature:
0000: 6E F8 97 0B 57 62 32 26   E0 82 66 88 CF 0F 4E 6C  n...Wb2&..f...Nl
0010: 18 E8 78 E1 0D A7 23 76   10 7E 46 B4 4F 42 26 DD  ..x...#v..F.OB&.
0020: 9F 0B 59 0A 65 ED 2C 0C   6C 25 32 4A 52 48 D9 6D  ..Y.e.,.l%2JRH.m
0030: 33 C7 00 F7 99 F3 75 29   45 3C 5C 2B EB D7 91 F5  3.....u)E<\+....

]
Valid certificate
CN=Universidad Tecnologica, OU=FISC, O=UTP, L=Panama, ST=PA, C=PA
Text to sign: [B@375212bc
signature length: 64
signature content: [B@6d4c1103
```

Fig.11: Validation of credentials and signature of the text by the ASPEJC

6. Conclusion and Future Works

Having completed this study, we could validate the feasibility of using smart cards, Java Card specifically as a means to develop mechanisms to enable secure access to services offered by an application or institution. The combination of technologies such as digital certificates and smart cards can handle different credentials and functions through a single token.

The solution developed is based on the use of PKI for access to the services offered by the institution, using asymmetric algorithms for creating the user's key. It has created a security infrastructure that consists of Certificate

Authority and Registration Authority that are in charge of the generation and registration of digital certificates used by the users according to their role. Also, it is included the use of Java cards for storage of certificates and user authentication. The flexibility, convenience and comfort are some of the advantages of this technology.

To verify the functionality of the proposed mechanism, we have developed a demonstrator user / server. This is the X.509 certificate store on the card Java and an application to access the resource by authenticating the card and user certificates. With the integration of two technologies, we have obtained the benefits of each as scalability, interoperability, portability and information security in your applications.

This has allowed a more secure communication to ensure the authenticity, confidentiality, integrity and non-repudiation of origin is an important point because it is a security service that lets you test the participation of an individual or entity in a communication.

One possible future application niches ASPEJC mechanism consists of the implementation of a VPN infrastructure that allows users access to a private network with a set of privileges defined by profile / role. To this effect, the first tests using OpenVPN, which was established by the client infrastructure and server infrastructure. Authentication is performed using X.509 digital certificates. The next item is the certificate stored in the Java Card, which was achieved through the mechanism ASPEJC. Finally there remains the task of configuring the VPN client for reading the X.509 certificate from the Java Card and Java Card applet to redirect users based on their profile / role to a specific network configured on the server and whose credentials are loaded from the Java Card.

References

[1] V. Nicussor, "Public Key Infrastructure for Public Administration in Romania", Communications (COMM), 2010 8th International Conference, IEEE, 2010, pp. 481-484.

[2] D. A. Ignacio, G. A. Arturo, C. V. Maria, " Autenticación en la Red: ACerO y JCCM*: Java Card Certicate Management", III Jornadas de Ingeniería Telemática. JITEL, IEEE, 2001, pp. 405-412.

[3] E. Nazar, A. Yaqoob, "An Approach for Multi Factor Authentication for Securing Smart Cards' Applications", Proceedings of the International Conference on Computer and Communication Engineering IEEE, 2008, pp. 368-372.

[4] W. Yewel, Y. Huiming, Y. Xiaohong, "Case Study: Using Smart Cards with PKI to Implement Data Access Control for Health Information Systems." Proceding of the IEEE SoutheastCon 2010 (SoutheastCon) IEEE, 2010, pp. 163-167.

[5] A. Shadi, D. Mader, H. Hiba, "A Web Client Authentication System Using Smart Card for e-Systems: Initial Testing and Evaluation", Fourth International Conference on Digital Society, IEEE, 2010, pp. 192-197.

[6] Ch. Zhu, "Java Card™ Technology for Smart Cards: Architecture and Programmer's Guide". Addison-Wesley, California, USA, 2010.

Maria Ortega Master of Information Technology and Communication of Technological University of Panama. I am currently a student researcher at the University, Faculty of Computer Systems. Current research interests include information security.

Sergio Sanchez PhD in Telecommunications from the Polytechnic University of Madrid (UPM). Spain. I currently work as a professor in the PSU Department of Telematic Engineering and Architectures (DIATEL) and I am a researcher in the group TSIC (Telematic Systems for the Information Society and Knowledge) in the School of Telecommunications Engineering of the Polytechnic University of Madrid (EUITT).