Integrating EAP in SIM-IP smartcards

[1]*Pascal Urien,* [1]*Adel Tizraoui,* [1]*Marc Loutrel,* [2]*Karen Lu.*

Smartcard Research Center

[1]SchlumbergerSema,
36-38 rue de la Princesse,
BP 45, 78431 Louveciennes Cedex, France

[2]Austin Product Center, Schlumberger,
8311 North FM 620 Rd,
Austin, TX 78726,USA

urienp@slb.com, atizraoui@slb.com, mloutrel@slb.com, lu@austin.apc.slb.com

Abstract

This paper reviews authentication issues in wireless networks. It suggests to integrate EAP protocol in internet smartcards, so-called SIM-IP, in order to support multiple authentication schemes. Smartcard content is described in XML syntax. Several wireless LAN client profiles, e.g. a set of files and procedures, are stored in a high secured and portable web server in the smartcard, and invoked through HTTP messages.

Keyword: WLAN, security, smartcard, EAP.

1. Introduction.

This paper presents new security issues, associated with the exponential growth of cheap wireless networks. We propose to integrate Extended Authentication Protocol (EAP [1]) in SIM-IP smartcards that work as high secure and tamper resistant internet nodes.[2].

Until now companies have deployed their local area networks without taking care of access points security. Typically network is organized around a tree of switching HUBs, which connect workstations by means of RJ45 plugs (called access ports in IEEE 802.1x standard [3]). Because company entrances are secured and reserved to authorized staff, network access ports are freely used, in particular to enable connections of nomad laptops computers.

Users mobility and geographic constraints of IP networks lead to use DHCP protocol in order to perform dynamic addresses allocation, compatible with visited intranet architecture. As mentioned in [4], *DHCP in its current form is quite insecure,* because no password or challenge mechanism are currently supported. Company restricted services (private web, email,…) are freely available from any computers plugged to the society network.

In wireless network, the link between intranet and computer is done by electromagnetic waves, which create virtual plugs, even outside company walls, for example in its parking lot [11]. For that reason security features like user authentication and data privacy are mandatory in order to deploy trusted radio networks.

Here are examples of some architectures under study,

The AAA [5] IETF working group aims at defining an architecture in which *Service Providers* deliver and charge services, to nomad users visiting *User Home Organizations.*

An authentication architecture dedicated to local 802 wired or wireless networks [3] (802.1X) is currently under standardization by the IEEE organization. This architecture could equally support new internet services like QoS and traffic billing.

OWLAN ([6] Operator Wireless Local Area Network) is an architecture in which wireless LANs (802.11,…) are utilized as an extension to GPRS networks. Access points are used in a way similar to base stations, and are connected to operator packets network.

As SIM smartcards showed their efficiency to ensure mobile communication security and accounting, it seems logical to work with these silicon chips in order to provide AAA services. In addition we recently introduce an innovative smartcard architecture ([2] SIM-IP) dedicated to these functions. Basically SIM-IP card is a trusted web server, internal resources like data, procedures, or protocols being identified by URLs.

A service profile is a collection of information required by a foreign network. It contains user identifiers ([8] NAI Network Access Identifier), authentication procedures (using secret shared keys) and session keys, deduced from authentication protocols, which ensure data privacy and integrity.

Another advantage of this approach is to store in SIM-IP modules software profiles which customize applications available on multiple computers (PC, PDAs, …), like VoIP phones, with bearer or operator information (NAI, private data).

2. Wireless security requirement.

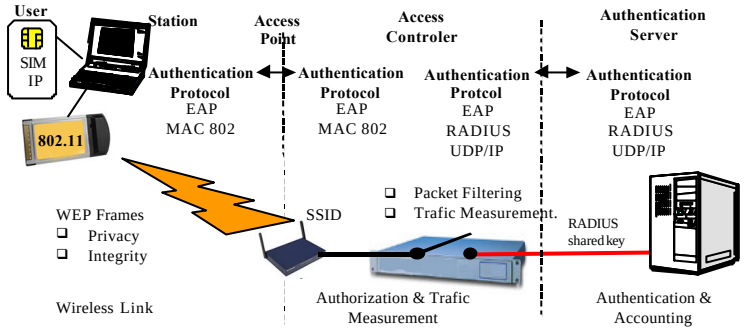

Figure 1. 802.11 security architecture.

Basic Architecture.

We shall distinguish three kinds of elements (figure 1) dealing with security aspects in 802.11 networks,

- *Wireless link security*. Frames exchanged between station and access point need privacy and integrity services. Keying materials are computed according to an authentication protocol, like WEP or EAP.
- *Station packets filtering*. Data which are sent or addressed to a given wireless node are filtered according to station credentials deduced from a previous authentication process. The device, performing this operation is usually referred as the *access controller* [6].
- *Authentication server*. The access controller acts as a proxy between station and remote authentication server. Authentication elements are shuttled between station and access controller either via MAC [3] or IP packets [6], and then are forwarded by the access controller to the authentication server, for example by means of RADIUS protocol [9].

Wireless Link Security: WEP.

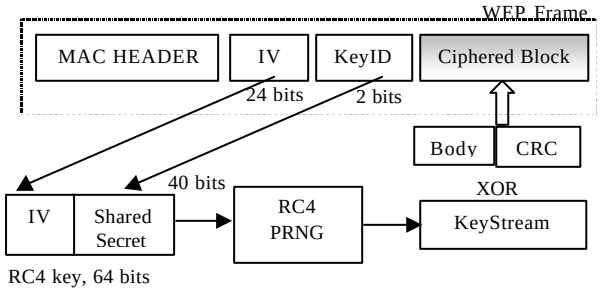

Figure 2. WEP frames.

The IEEE 802.11 standard has defined the *Wireless Equivalent Privacy* protocol that provides the following services,

- User authentication.
- Information privacy
- Information integrity.

Unfortunately as mentioned in [12,13], this protocol exhibits major threats, mostly due to the lack of key management infrastructure.

The WEP protocol deals with RC4 cryptographic algorithm of which we are going to briefly remind some properties. RC4 works with a pseudo random number generator (PRNG), initialized with a seed value (RC4 key) of which size ranges between 8 and 2048 bytes. This entity generates a byte stream (Ksi - called keystream), used to cipher a plaintext (Mi) according to the Vernam's algorithm which produces a cipher stream (Ci) deduced from a byte-to-byte exclusive-or (xor) between plaintext (Mi) and keystream (Ci = Mi xor Ksi).

It is important to notice that the knowledge of a plaintext (Mi) and a cipher stream recording (Ci) allows to recover a keystream, because

Ci xor Mi = Ksi xor Mi xor Mi = Ksi.

Therefore it is not safe to reuse an RC4 key.

WEP frames (figure 2) shuttle security procedures between station and access point, its includes the following fields,

- A MAC header, indicating that the WEP format is used.
- A three-bytes IV-value.
- A KeyID field (2 significant bits) that identifies a key index.
- A ciphered block, resulting from frame body and CRC RC4 encoding.

A 64-bit key obtained by concatenation of the 40-bit secret value (one among four, according to the KeyID field) and the IV vector (24 bits), is used to compute the ciphered block.

The IEEE 802.11b standard defines a shared-key authentication protocol in which the access point sends a challenge text packet (Ti) that the client encrypts with one of its RC4 keys (Ci= Ti xor Ksi) . An attacker can easily deduced the keystream (Ksi) associated to the IV vector specified in the WEP frame by performing xor operations between clear text and ciphered challenge(Ksi = Ci xor Ti). Therefore spoofing attacks are possible as long as the same RC4 key is used.

Only 16 millions (2^{24}) of keystreams are available, therefore if the secret shared between the station and the access point key remains unchanged for a sufficient time amount, it's easy to design software dedicated to keystreams recovering and recording.

One general property of all CRC checksums [13] is that the CRC of a frame (z), built from two frames x & y, according to xor byte to byte operations(z = x xor y), can be deduced from the CRC values of these two frames

z = x xor y => CRCz = CRCx xor CRCy.

As a consequence an attacker can easily modified a ciphered WEP frame (Ci) which will keep a correct CRC value, by performing xor operations with any well formed frame (Mi') [13].

3. Extended Authentication Protocol – EAP

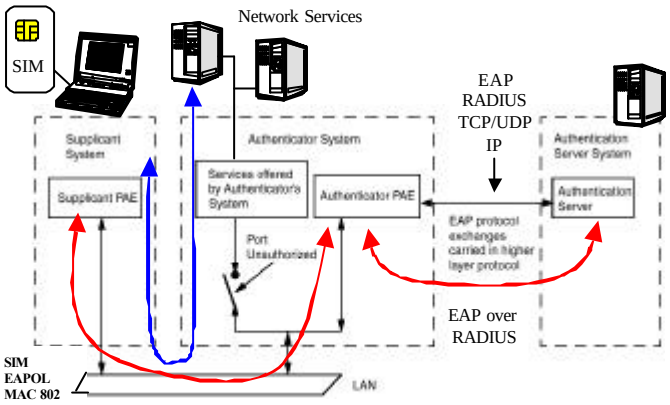

Figure 3. 802.1x authentication architecture.

As proposed by Cisco [14] a centralized authentication infrastructure, working with authentication servers and using RADIUS protocol, allows to securely distribute WEP keys over wireless networks.

The basic idea is to conduct an authentication process between the user and the authentication server, according to simple or mutual authentication schemes. Mutual authentication avoids the connection to rogue access points potentially responsible of denial-of-services attacks through station hijacking.

A shared and secret session key Ks is computed by the two parties (AS & AP), and is transmitted through the RADIUS protocol, from authentication server to access point. The access point then encrypts a WEP key (called broadcast key, Kw) with Ks, and sends the result to the station through a clear frame. The station, which knows the Ks key resulting from the authentication process, decrypts the broadcast key Kw.

In the RADIUS protocol the exchanged messages are secured by means of MD5 digests deduced from transported information and a secret shared between the radius server and its client. This means that access controller is configured to work with one or more authentication server, and therefore that a security association exists between this two entities.

An authentication architecture has been recently proposed by the IEEE 802.1x group [3]. The main idea is to forbid network access to non authenticated (wireless) nodes. In a logical way a network user (*supplicant system*) reached its services provider by means of access ports that are able to block or forward its IP packets.

The EAP protocol, shuttled by 802 frames, transports various authentication schemes between user station and access controller. The RADIUS protocol is used by access controller to forward EAP messages to an appropriate authentication server.

EAP can be seen as an umbrella protocol because it may support various authentication schemes, like

- MD5 challenge, a digest is computed from a random value and a shared secret.
- *PPP EAP TLS* [15] a protocol based on SSL mechanisms.
- IAKERB, adaptation of Kerberos V5 procedures.
- EAP SIM [16], reuse of SIM smartcards (GSM 11.11).
- EAP AKA [17], support of USIM smartcards (UMTS security modules).

EAP packets (figure 4) are identified by code and type fields. There are four kinds of packets, each of them is associated to a particular value, 1 for request, 2 for response, 3 for success, 4 for failure.

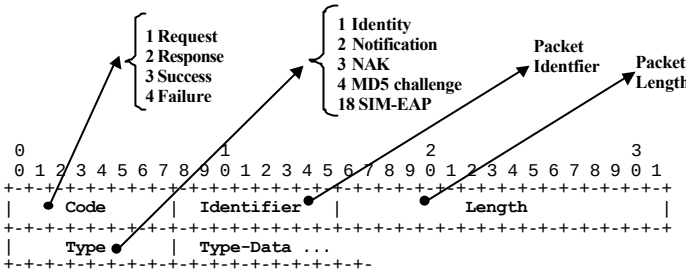

Figure 4. EAP message structure.

The type field is one byte and identifies the structure of an EAP request or response packet. Each implementations must support at least four types, 1 for Identity, 2 for notification (displayable text), 3 for nak (not supported type), 4 for MD5 challenge scenario. Type 18 has been proposed for EAP-SIM authentication [16].

As suggested in EAP-SIM, Network Access Identifier (NAI) could apply for station Identity. It is made of two parts, the domain name (*realm*) associated to an authentication server, and the user login (*userID*) known in this domain. A GSM *userID* is built with the character "1" appended to subscriber IMSI (that consists of no more than 15 decimal digits), the right part indicates the service provider domain name. As an illustration a subscriber NAI looks like 1imsi@service_provider.com.

4. EAP-SIM.

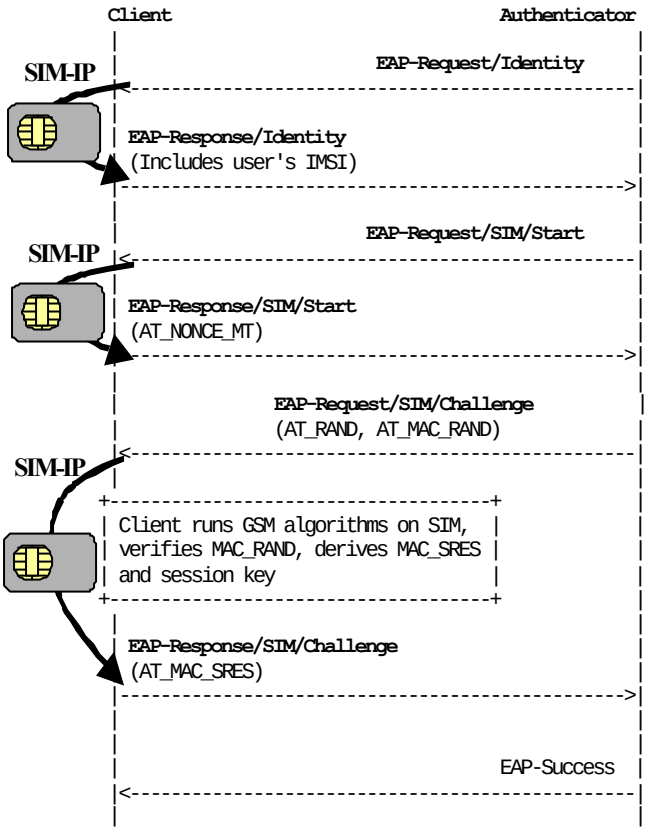

Figure 5. EAP-SIM protocol [16].

As we previously mentioned it SIM module can work with EAP protocol. Three roundtrips messages are used in order to generate session keys, between client (station) and authenticator (an access point and an authentication server).

The authenticator initiates an EAP-Request/Identity, which gets the client NAI. From this NAI it's possible to deduced the authentication server address.

Then four specific messages, identified by dedicated sub type values (10 & 11), are generated.

- Authenticator sends an EAP-Request/SIM/Start message, which optionally requires the user IMSI. Client returns a response (EAP-response/SIM/Start) including a nonce value (AT_NONCE_MT) and optionally its IMSI.
- Authenticator generates a new request (EAP-request/SIM/Challenge) which includes the followings, A list (AT_MAC_RAND) of GSM random number (RANDi). As specified in GSM 11.11 each record includes 16 bytes. This list is signed by a keyed MAC (AT_RAND), which is computed from a one-way hash function (SHA-1) and a secret key (K_int) deduced from the master key K_Master.

This master key is obtained from the nonce value, and a list of Kci values, computed with the A8 algorithm stored in a SIM card (figure 6),

KCi = A8(RANDi), K_Master = F(Kci, NONCE_MT).

- Client computes the Kci values (from RANDi list), derives the master key, and checks the EAP-response message signature (AT_MAC). By means of well known GSM algorithms A3 & A8 computed by SIM card, it deduces a set of SRESi values.

SRESi = Run_Gsm_Algorithm(RANDi)

A 20-byte digest (AT_MAC_SRES) deduced from SRESi is sent back in the EAP-Response/SIM/Challenge message, and is checked by the authenticator. Upon success additive specific application keys may be obtained from the master key.

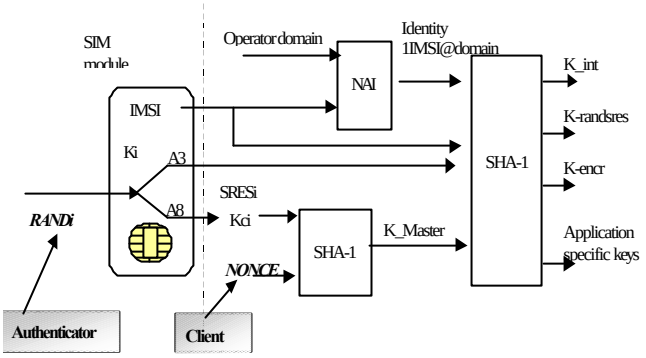

Figure 6. SIM use in EAP-SIM

In summary we notice that

- Authenticator generates the RANDi values.
- SIM card stores client IMSI and computes A3 and A8 algorithms (Kci and SRESi values).
- Client builds the user NAI, computes a master key deduced from Kci(RANDi) and AT_NONCE value.

5. SIM-IP.

Internet smartcard is an emerging technology in which these tamper resistant devices run server and client applications. In particular they work as high trusted web servers and support various protocols like HTTP or LDAP clients. An internet smartcard [18,19] is organized around five layers (C.f figure 7),

1. A communication stack, which manages data exchange with the internet network.
2. A web server acting as an interface to all smartcard objects.
3. An optional XML parser, which execute scripts written in XML syntax.
4. A security interface checking user credentials for objects access.

5. A set of smartcard objects either public (controlled by card bearer) or private (e.g. controlled by mobile communication operator), classified in four categories,

- *Data* stored in files, like phonebook, NAI or cryptographic keys.
- *Procedures*, computing cryptographic algorithm associated to secret keys.
- *Client* applications (HTTP, LDAP…) started from an HTTP request. In this case internet card works as a trusted proxy [18].
- *Agents*, for example Java mobile code stored in the smartcard, and executed on client side. They may be used when smartcard computing performances are insufficient. They execute non secure protocol elements but can invoke smartcard secure objects through TCP sessions.

A SIM-IP module is an internet smartcard dedicated to security purposes in wireless LAN environment. It may store user identities (NAI), authentication keys, procedures and QoS protocol like COPS or RSVP.

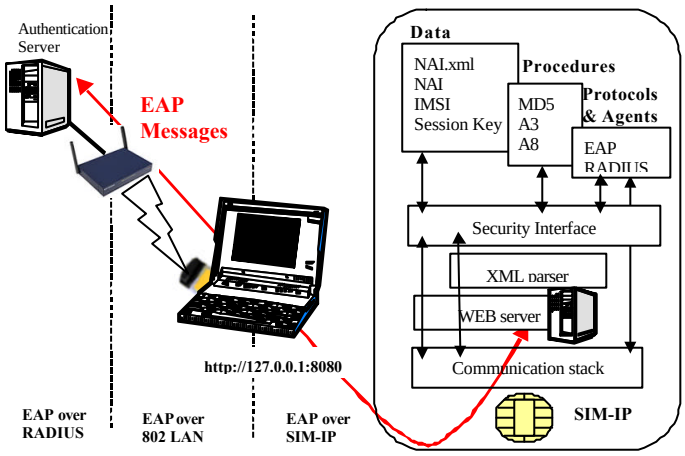

Figure 7: EAP support in SIM-IP.

For a given network, a user profile is a set of information, shuttled by SIM-IP, which is required by AAA services.

As we previously mentioned it, authentication is required for evaluating user credentials. Because EAP is a convenient way to classify and standardize authentication protocols we suggest to support or integrate it in SIM-IP cards.

First we should remark that a smartcard bearer may use several identifiers (NAIs) in various networks.
Second authentication procedures could be different, in order to support existing security architecture. For example a MD5-challenge (EAP_Type = 4) mechanism could be mandatory in a corporate wireless network, while SIM-EAP (EAP_Type=18) may be used in an OWLAN context.

We believe that there are two ways of supporting EAP in SIM-IP module. One alternative is to follow an RPC (Remote Procedure Call) paradigm, where NAIs and authentication procedures are identified and invoked through URLs. Another possible choice, which requires more computing resources is to process EAP requests in smartcards, which will build the EAP responses (as shown in figure 5).

Identity setting.

We suggest to describe SIM-IP bearer identities in an XML file (NAI.xml), stored in SIM-IP card (figure 8). In this example an element named NAI_List stores a list of NAI empty elements of which attributes are identifier index, value, and EAP_Type.

```
<NAI_List nb="2">
<NAI index="1" value="urienp@slb.com" EAP_Type = "4"/>
<NAI index="2" value="1imsi@operator.com" EAP_Type = "18"/>
<NAI_List />
```

Figure 8. **NAI.xml** file

A wireless software gets this file from SIM-IP, then it will typically display a menu and invite the terminal user to select an identity. An URL like,

http://127.0.0.1:8080/Set_NAI?index=1.

sets the current user identity. According to HTTP protocol an authorization may be required which needs user login and password.

Attribute	**URL http://127.0.0.1:8080/**	**Returned message**
MD5	EAP4_MD5?index=2&challenge=1234	Digest hexadecimal value
MD5 XML interface	X?x=<Copy_11 s="2" fD="index"/> <Copy_11 s="1234" fD="challenge"/> <Call_7 s="EAP4_MD5"/> <Send_0 f="Out"/>	Digest hexadecimal value

Figure 9. EAP4 profile

6. RPC paradigm.

An EAP profile, is a set of attributes (data and procedures) required for authentication purpose. Each profile identified by an URL returns an XML body (Content_Type: text/xml). If SIM-IP supports an XML parser, then procedures may be executed via XML messages [8]. We shall describe two EAP profiles, of which are respectively 4 (MD5 challenge) and 18 (SIM-EAP)

- Type 4. MD5 digest challenge is computed from a challenge value and a NAI index. Figure 9 presents MD5 call be means of URL or XML message.
- Type 18 (EAP-SIM). This profile includes at least IMSI and NAI files, two procedures (SRES, KC); and an optional NONCE generator may be supported. Each of these objects is identified by an URL (figure 10). Additional functions could be integrated, in order to compute all keying materials (K_int, K_randsres, K_encr, Application_specific_keys,…).

Attribute	URL http://127.0.0.1:8080/	Returned Message
IMSI	EAP18_IMSI?index=2	IMSI ascii value
SRES	EAP18_SRES?index=2&rand=1234ABCD	SRES hexadecimal value
KC	EAP18_KC?index=2&rand=1234ABCD	KC hexadecimal value
NONCE	EAP18_NONCE?index=2	NONCE hexadecimal value

Figure 10: EAP18 profile.

Processing EAP request in SIM-IP.

According to smartcards computing abilities and memory sizes, EAP requests could be completely handle by SIM-IP modules.

Incoming EAP requests are sent to SIM-IP web server through HTTP post messages, then corresponding EAP responses are built and returned to the client.

If, like in EAP-SIM several requests are processed, it implies that SIM-IP manages an EAP state machine and checks that requests are submitted in the right order. EAP message could be identified by URL like

http://127.0.0.1:8080/EAP?message="….;"

which returns an EAP response expressed in an hexadecimal format. First packet (like EAP-request/Sim/Start initializes an EAP authentication state machine, and last packet (like EAP-request/SIM/Challenge) ends it. If an error occurred an HTTP status error is produced and EAP state machine is reset.

7. Conclusion.

In this paper we have demonstrated that introduction of SIM-IP modules in wireless networks could lead to support multiple user profiles for various environments. Because it is likely that authentication will be mandatory in most WLANs, a trusted and tamper resistant web server is a well adapted solution in order to support EAP features and to describe available profiles in XML syntax.

8. References.

[1] «PPP Extensible Authentication Protocol (EAP)" RFC 2284.

[2] Pascal Urien, "SIM-IP, Smartcard benefits for wireless applications" Application and Services in Wireless Networks ASW'2001, Juillet 2001, Hermès Sciences Publication ISBN 2-7462-0305-7.

[3] IEEE *Draft* P802.1X/D11, « Standard for Port based Network Access Control», Standards for Local and Metropolitan Area Networks, mars 2001.

[4] «Dynamic Host Configuration Protocol" rfc 2131 march 1997.

[5] "AAA Authorization Framework" rfc 2904 august 2000

[6] J.Ala-Laurila, J.Mikkonen, J.Rinnemaa "Wireless LAN Access Network Architecture for Mobile Operators", IEEE Communications Magazine November 2001 pp 82,89.

[7] Pascal Urien " Internet Card, a smart card as a true Internet node", Computer Communication, volume 23, issue 17, October 2000.

[8] "The Network Access Identifier" rfc 2486 January 1999.

[9] «Remote Authentication Dial In User Service (RADIUS)" rfc 2138 April 1997.

[10] W. Arbaugh, N. Shankar, and Y. Wan, Your 802.11 «Your Wireless Network has No Clothes. http://www.cs.umd.edu/~waa/wireless.pdf »

[11] N. Borisov, I.GoldBerg, D.Wagner «Intercepting Mobile Communications: The Insecurity of 802.11 » Proceeding of the Eleventh Annual International Conference on Mobile Computing And Network, July 16-21, 2001.

[12] J. Walker, "Unsafe at any key size: An analysis of the WEP encapsulation", Tech Rep. 03628E, IEEE 802.11 committee, March 2000. http://grouper.ieee.org/groups/802/11/Documents/DocumentHolder/0-362.zip.

[13] CISCO - "Wireless LAN Security" http://www.cisco.com/warp/public/cc/pd/witc/ao350ap/prodlit/a350w_ov.pdf

[14] «PPP EAP TLS Authentication Protocol" rfc 2716 October 1999.

[15] "EAP SIM Authentication" draft-haverinen-pppext-eap-sim-02.txt November 2001

[16] "EAP AKA Authentication" draft-arkko-pppext-eap-aka-01.txt, November 2001

[17] Pascal Urien "Programming internet smartcards with XML scripts", Springer Verlag, LNCS 2140, e-Smart 2001 september 2001.

[18] "SmartTP, smart transfer protocol" *draft-urien-SmartTP-00.txt.*