UML Model and Implementation of the IEEE 802.11 Authentication and Privacy Services using Elliptic Curves Cryptosystems

Rocío A. Aldeco Pérez, Miguel A. León Chávez

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
14 sur y Av. San Claudio, C.U., Puebla, Pue.
Tel: 2295500 ext. 7213 Fax. 2295672
raldeco@mail.cs.buap.mx, mleon@cs.buap.mx

Abstract. In the last years a spectacular development in the mobile communications has been produced, specifically in Wireless Local Area Networks (WLANs). The main function of this kind of networks is provide connectivity and access to the traditional wired networks, such as Ethernet, like it will be an extension of these, but with more mobility and flexibility. One of the most important standards in this area is the IEEE 802.11. This standard provides users with three security services (authentication, privacy, and integrity) based on WEP (Wired Equivalent Privacy) protocol and CRC32. WEP, in turn, is based on the RC4 symmetric algorithm, which is a quite powerful crypto algorithm. Nevertheless, WEP takes a poor approach for using it. This paper proposes the usage of the Elliptic Curves Cryptosystems (ECC) in order to implement the authentication and privacy services into the IEEE 802.11, and presents the modelling of these services using UML (Unified Modelling Language).

1 Introduction

The IEEE 802.11 standard [1] defines two OSI layers, Physical (*PHY*) and Data Link (*DLL*), the latter is divided in two sublayers, Logical Link Control (*LLC*) and Medium Access Control (*MAC*). The MAC sublayer provides upper layers with three main services, control of the medium access, mobility, and security. Two access methods are defined by the MAC sublayer, one centralised (Point Coordination Function, *PCF*) and another distributed (Distributed Coordination Function, *DCF*).
PCF uses an access scheme based on polling, where the Access Point (*AP*) acts as Point Coordinator (*PC*). The PC cyclically polls all the stations and grants them rights to transmit.

DCF is based on Carrier Sense Multiple Access with Collision Avoidance (*CSMA/CA*) which is mandatory and has to be presented in all stations.
The mobility services are provided by the Distribution System, such as association, reassociation, and disassociation.

© A. Gelbukh, H. Calvo (Eds.)
Avances en la ciencia de la computación en México
Research on Computing Science 13, 2005, pp. 149-158

The MAC sublayer provides users with three security services, authentication, integrity, and privacy. However, the authentication scheme is quite weak [2]. The integrity scheme is restricted to use the CRC-32 algorithm, and the privacy scheme, named Wired Equivalent Privacy (WEP), does not carry out its purpose [3,4,5].
The new version of the standard, i.e. IEEE 802.11i, will be based on 802.x and extensible authentication protocol (*EAP*). In the long term, 802.11i might provide a framework for using the Advanced Encryption Standard (AES) [2].
The goal of this paper is to use the Elliptic Curves Cryptosystems (*ECC*) to implement the authentication and privacy services into the PCF method of the IEEE 802.11 standard.

ECC are public-key mechanisms based on the elliptic curve discrete logarithm problem, whose best known algorithms run in exponential time. This means that a desired security level can be attained with smaller keys in ECC than in other public-key mechanisms, such as RSA.

In order to implement the authentication and privacy security services into IEEE 802.11 using ECC, this paper presents the modelling of the authentication and privacy services using UML (Unified Modelling Language) and the implementation of this algorithms in C++.

The rest of the paper is organized as follows, section II discusses the security services of the IEEE 802.11 standard. ECC are presented in section III, the UML model of the authentication and privacy services are presented in section IV, section V presents a brief introduction about the issues of the implementation in C++, conclusions and future research work are presented in section VI.

2 IEEE 802.11 Security Services

The IEEE 802.11 standard provides upper layers with three security services, authentication, privacy, and integrity. For authentication it uses the WEP protocol. WEP encrypts the data frames with the RC4 algorithm, each frame has to be encrypted with a different key, which consists of the field IV (Initialization Vector, 24 bits) and the WEP shared key (40 bits) concatenated forming a 64 bits key. This key is used to create the RC4 key stream, it takes one bit of this stream and one bit of the original message then it applies an XOR forming a cipher bit [1].

In October 2000, it was shown the vulnerabilities of WEP [3] using any size of key. Fluhrer, Mantin, and Shamir [4,5] describe how WEP takes a poor approach for using RC4 algorithm.

The integrity is provided by the CRC-32 mechanism and it is not a cryptographic mechanism, reason why this service is not guaranteed. The authentication service is implemented by the AP, since it only has to accept frames from the stations previously authenticated.

The IEEE 802.11 specifies two kinds of authentication.

Open System authentication: This is a null authentication algorithm. It accepts all the stations.
Shared Key Authentication: The shared key is the same used in WEP. The AP sends a plaintext exchange to the station; the station encrypts it and returns it to the AP. The

AP does the same and compares both results. If they are equal it allows the access, otherwise returns error [1]. Fig. 1 shows the sequence diagram of the authentication and association services. Park and Dicoi [3] discuss the authentication problem in these networks, causes by the bad management and distribution keys.

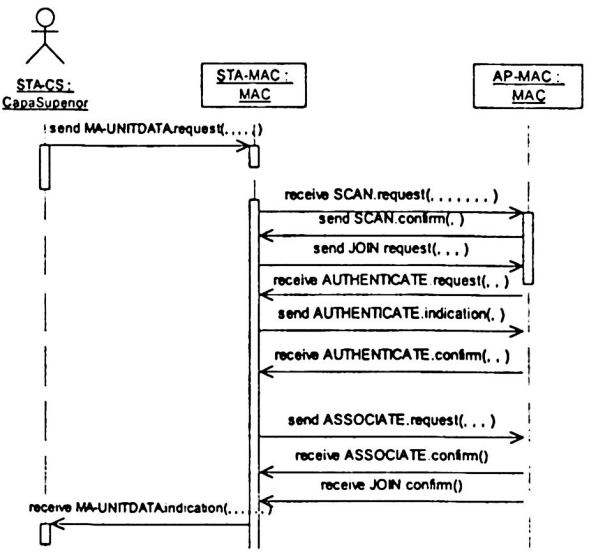

Fig. 1. Authentication and Association Services.

3 Elliptic Curve Cryptosystems

ECC are public-key mechanisms based on the elliptic curve discrete logarithm problem, whose best known algorithms run in exponential time. This means that a desired security level can be attained with smaller keys in ECC than in other public-key mechanisms, such as RSA.

Let q be a prime number, and let F_q denote a finite field (integers modulus n, Galois field, rational numbers, complex numbers) with order q. An elliptic curve E over F_q is defined by the equation:

$$y^2 = x^3 + ax + b \tag{1}$$

where $a, b \in F_q$. The point at infinity denoted by O, is also said to be on the curve. The set of all the points on E is denoted by $E(F_q)$. Elliptic curve points can be added (O plays the role of the neutral element in this operation) but not multiplied. It is, however, possible to perform scalar multiplication, which is another name for repeated addition of the same point. If k is a positive integer and $P \in E(F_q)$, the scalar multiple kP is the result of adding $k - 1$ copies of P to itself.

If G ∈ E(Fq) then ⟨G⟩ denotes the set {O, G, 2G, 3G}, if we have a point Q∈⟨G⟩, there is a positive integer number k such that kG=Q [6]. The discrete logarithm problem for Elliptic Curves consists of finding the number k starting from G and Q.

Given the huge computational complexity that this problem represents, it is possible to obtain with ECC security levels similar to the provided by other cryptosystems, for the price of operations over finite fields much smaller than the required for the others systems. These operations allows us to use smaller public and private keys that gives us higher velocity, lower memory requirements, and computer power in the implementations of the following algorithms [7].

3.1 ECC Algorithms

3.1.1 System configuration and key generation

First, it chose a finite field *GF(q)*, an elliptic curve *E* over *GF(q)* and a point *P* over E of prime order *n*, these are the system parameters and are publics [7]. The process for key generation is the follow:

1. Select a random integer number *d* such that $1 \leq d < n$
2. Compute the point $Q=dP$.
3. The public key is the point Q.
4. The private key is the integer *d*.

3.1.2 Encryption Scheme:

If *M* is the plaintext, the next steps are necessaries for cipher it

1. Obtain the public key Q.
2. Represent the message *M* as an element of the field $m \in GF(q)$.
3. Select a random integer number *k* such that $1 \leq k < n$.
4. Compute the point $(x_1,y_1)=kP$.Compute the point $(x_2,y_2)=kQ$. If $x_2=0$ then go to step 3.
5. Compute $c=m\,x_2$.
6. Send the ciphertext (x_1,y_1,c).

3.1.3 Decryption Scheme:

If we have the ciphertext (x1,y1,c) the next steps are necessaries for decipher it.

1. Compute the point $(x_2,y_2)=d(x_1,y_1)$, using the private key *d*.
2. Recover the message *m* computing $m=c\,x_2^{-1}$

3.1.4 Digital Signature

The next steps are necessaries for to sign the message *M*.

1. With some hash algorithm compute $e=H(M)$.
2. Chose a random integer number *k* such that $1 \leq k < n$.
3. Compute the point $(x_1,y_1)=kP$ and $r=x_1 \bmod n$.

4. Use the private key for compute $s=k^{-1}(e+rd) \bmod n$.
5. The signature of the message M is (r,s).

3.1.5 Digital Signature Verification

For verify the digital signature (r,s) in the message M.

1. Obtain the public key Q.
2. If $(r \bmod n)=0$ then reject the signature
3. With the hash algorithm compute $e=H(M)$.
4. Compute $s^{-1} \bmod n$.
5. Compute $u=s^{-1}e \bmod n$ and $v=s^{-1}r \bmod n$.
6. Compute the point $(x_1,y_1)=uP+vQ$.
7. Accept the signature for the message M if and only if $(x_1 \bmod n) = r$.
8. If $r=0$ then the sign equation $s=k^{-1}(e+rd)$ does not involve the private key d, and reject it.

4 UML Model

UML has a Semi-formal semantic meta-model that defines basic modelling concepts (object, class, etc.) and includes well-formed rules expressed as formal constraints. UML has a graphical notation for modelling concepts; it is the standard language for visualizing, specifying, constructing, and documenting the artefacts of a software-intensive system. It can be used with all processes, throughout the development life cycle and across different implementation technologies. UML allows creating several diagrams such as use case, class, sequence, collaboration, etc. These diagrams are the elements of the models. This section presents the analysis and design models of the authentication and privacy services into the PCF mode of the IEEE 802.11 using ECC.

Fig. 2 shows the use case diagram, where the actor is a station, which request any

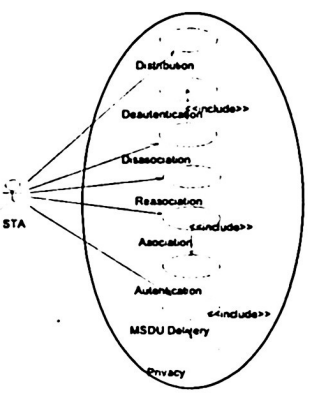

Fig. 2. Use Cases Diagram.

MAC service.

Fig. 3 shows the class diagram that models ECC and figure 4 shows the sequence diagram of encryption and decryption scheme explained in the previous section.

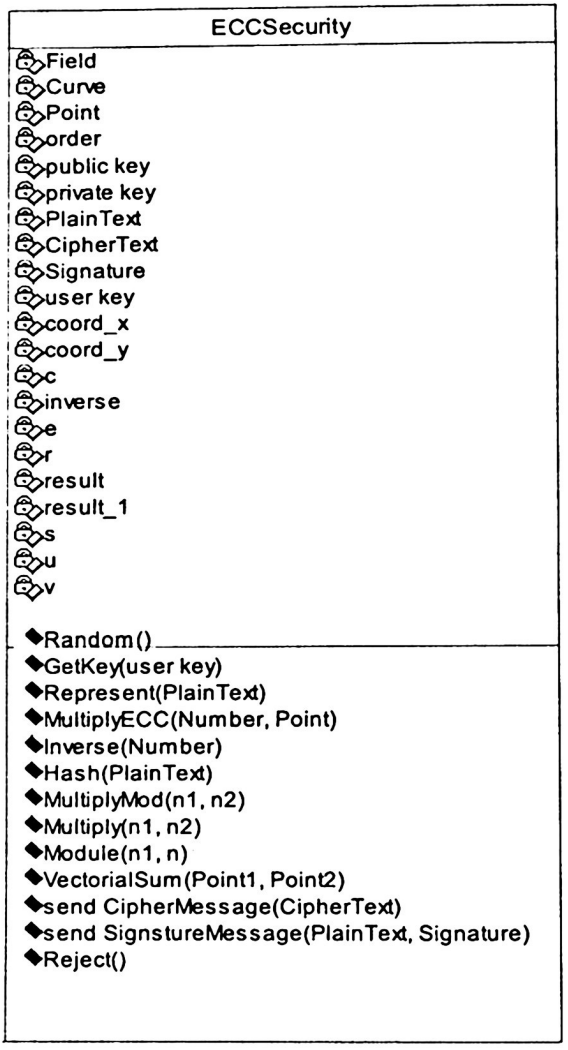

Fig. 3. Class Security ECC.

Both users, A and B, compute their keys (public and private) and then exchange only the public key. After this all the processes continue as before.

Fig. 5 shows the sequence diagram of the signature and verification scheme with ECC. User A signs a message in plaintext with his private key and then sends it to user B.

The user B checks the signature using A's public key. Both users represented in fig. 4 and 5 can be a station or an AP for the encryption – decryption scheme, or they can be a station authenticating to an AP for the signature – verification scheme.

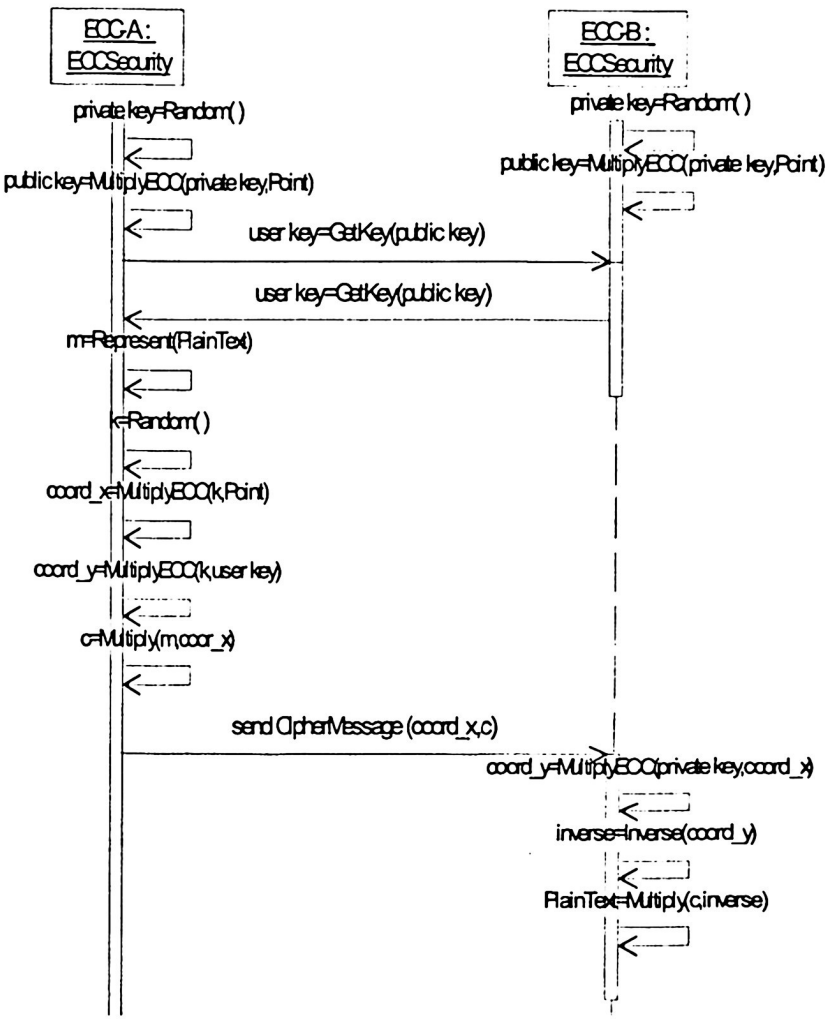

Fig. 4. Encryption – Decryption Scheme.

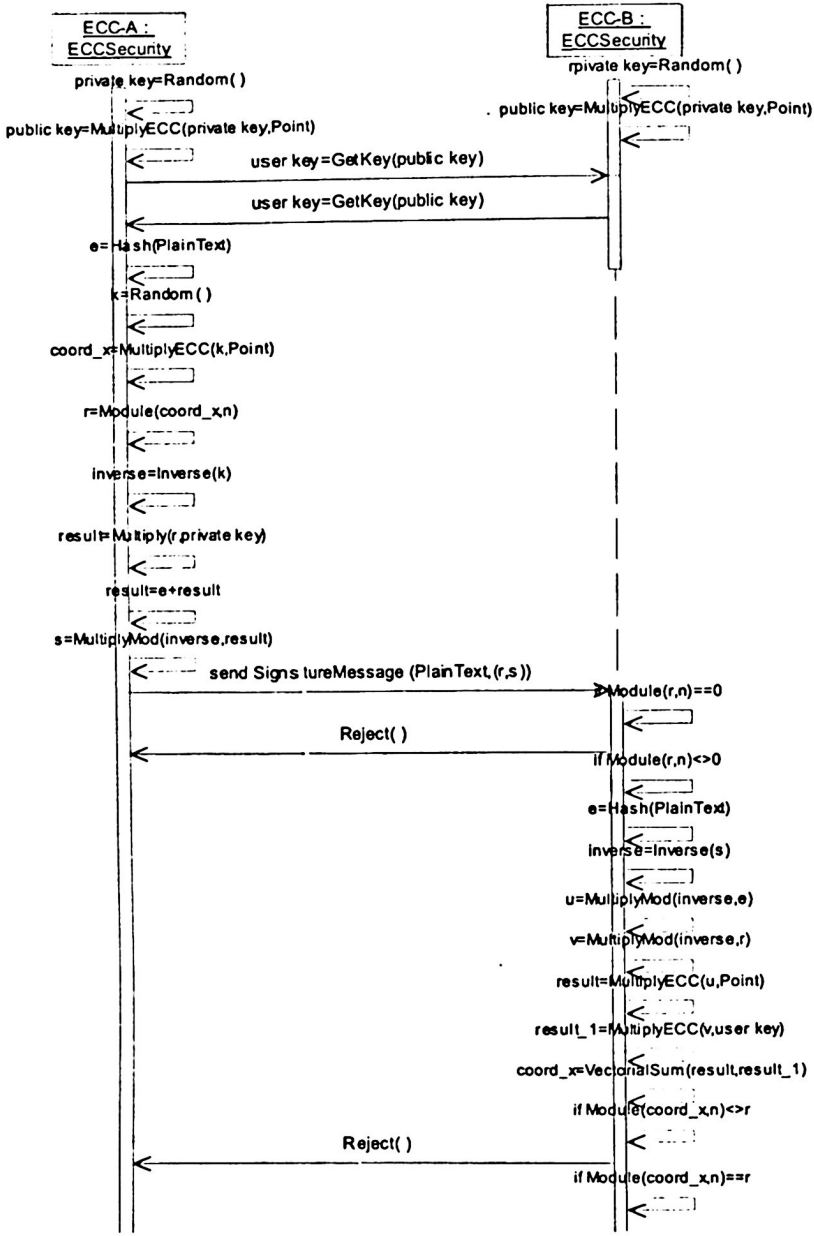

Fig. 5. Signature – Verification Scheme.

5 Implementation

We decided to make the implementation under the programming language C++. Besides to be a great language due to its flexibility, supporting data abstraction and supporting generic programming, also it is an object oriented programming language. The most important issue is, existing a lot of tools, programming in C++, that provide the mathematical background necessary to implement the elliptic curves algorithms.

One of these tools is NTL [8]. NTL is a high-performance, portable C++ library providing data structures and algorithms for manipulating signed, arbitrary length integers, and for vectors, matrices, and polynomials over the integers and over finite fields.

We use NTL for creating our own library for elliptic curves. This library implements all the necessary operations of elliptic curves algorithms (addition, multiplication, inverse, etc.). Consequently, this library doing easier the process of programming.

After that, creates the program that implements both the Encryption – Decryption Scheme and Signature – Verification Scheme.

This program shows all the process modeling in UML. The next step is including these programs in a driver of a wireless card and doing performance test.

6 Conclusions

This paper has discussed the security services defined by the IEEE 802.11 and it has reviewed the vulnerabilities of the authentication, privacy and integrity services. This work has proposed to replace the cryptographic mechanisms implemented in the authentication and privacy services by public key cryptosystems based on Elliptic Curves due to its theoretical efficiency, and that the key size, digital signatures and cipher messages are smallest.

In fact, the cryptosystems based on EC provide the same security level than the systems based on the factorization problem or discrete logarithm problem reducing considerably the number of digits.

ECC can be implemented with efficient in hardware or software and be able to compete in speed with systems like RSA.

The paper has presented the analysis and design models of IEEE 802.11 in PCF mode extended with the authentication and privacy security services using ECC, its implementation in a lower layer, doing performance test and the definition of the key exchange protocol is our future research work.

References

1. IEEE Std 802.11, 1999 Edition. "IEEE ANSI E 802 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", 1999.

2. Park, J. and Dicoi, D. "WLAN Security: Current and Future". IEEE Internet Computing, September – October, 2003, pp. 60-65.
3. Walker, J. "Unsafe at any key size: An analysis of the WEP encapsulation". IEEE 802.11-00/362. October 2000.
4. Fluhrer, S., Mantin, I., and Shamir, A. "Weaknesses in the Key Scheduling Algorithm of RC4", 8th Annual Workshop on Selected Areas in Cryptography, August 2001.
5. Stubblefield, A., Ioannidis, J., and Rubin A. "Using the Fluhrer, Mantin, and Shamir Attack to Break WEP", Computer Science Dept. Rice University and AT&T Labs – Research, Florham Park, NJ.
6. Jurisic, A. and Menezes, A. "Elliptic Curves and Cryptography: Strong digital signature algorithms" Dr. Dobb's Journal, April 1997.
7. Hankerson, D., Menezes, A., and Vanstone, S. "Guide to Elliptic Curve Cryptography", Springer-Verlag, New York, 2004.
8. Shoups V, NTL: A Library for doing Number Theory (version 5.3.2) available in http://shoup.net/ntl/