UML Model of the IEEE 802.11 Privacy Service using AES-CCM (Advanced Encryption Standard, Counter-Mode/CBC-MAC) and AES Implementation

Sugehi Marina Merino Higareda, Miguel Ángel León Chávez

Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación,
14 Sur y Av. San Claudio, C.P. 72570, Puebla, México.
msugehi@hotmail.com, mleon@cs.buap.mx

Abstract. Nowadays, mobility plays an important role in our society. Thanks to the wireless networks a real mobility in the devices has been obtained. The IEEE 802.11 standard [1] is one of the most deployed wireless technologies all over the world. Its main characteristics are simplicity, flexibility and cost effectiveness. However, the communication medium is insecure and in some applications the insecurity is an unwanted characteristic. This standard provides users with three security services: Authentication, Privacy, and Integrity. The privacy service is implemented by WEP (Wired Equivalent Privacy) protocol, which has demonstrated to be vulnerable [2, 3, 4]. This paper discusses the WEP vulnerabilities, and proposes to replace it by AES-CCM (Advanced Encryption Standard, Counter-Mode/CBC-MAC). The Unified Modeling Language (UML) model of the IEEE 802.11 privacy service using AES-CCM and AES implementation are presented.

1 Introduction

The purpose of the IEEE 802.11 standard [1] is to provide wireless connectivity to automatic machinery, equipment, or stations that require rapid deployment, which may be portable or hand-held, or which may be mounted on moving vehicles within a local area.

The IEEE 802.11 is part of a family of standards for local area networks (LANs), this standard defines two OSI layers only: Physical (PHY) layer and Data Link layer (DLL), this last one is divided in Logical Link Control (LLC) sub-layer and Medium Access Control (MAC) sub-layer. At the PHY layer, the standard defines the protocols and compatible interconnections of communication equipment via the air, radio or infrared like transmission medium. The MAC sub-layer defines two medium access coordination functions, one centralized (Point Coordination Function, PCF) and other distributed (Distributed Coordination Function, DCF) which implements Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol and Request To Send and Clear To Send (RTS/CTS) protocol, and should be implemented obligatorily in all the stations.

© A. Gelbukh, H. Calvo (Eds.)
Avances en la ciencia de la computación en México
Research on Computing Science 13, 2005, pp. 137-147

The MAC sub-layer provides users with the following services: MSDU delivery, Authentication, Deauthentication, Association, Disassociation, Reassociation, Distribution, Integration, and Privacy.

The IEEE 802.11 defines three security services, as follows: Authentication, Privacy and Integrity. The privacy service is implemented by the WEP (Wired Equivalent Privacy) protocol, which has demonstrated to be vulnerable [2, 3, 4].

This paper proposes to replace WEP protocol by AES-CCM (Advanced Encryption Standard, Counter–Mode/CBC-MAC). AES is based on the Rijndael algorithm, which is a powerful encryption and efficient algorithm. AES is a block cipher, with both keys and blocks of 128, 192, or 256 bits long.

Why do you use AES? Because AES is a cipher with a simple and elegant structure, and the key scheduling algorithm [5] shows a satisfactory security margin against cryptanalytic attacks.

AES, as the old standard DES, works in several modes. This paper uses the CCM (Counter–Mode/CBC-MAC) mode [6], because it provides authentication and encryption using a single block cipher key that is established beforehand. Thus, CCM requires a well-designed key management structure.

In order to implement the IEEE 802.11 privacy service using AES-CCM this paper presents the UML model of MAC sub-layer in DCF mode, which includes the privacy service.

The rest of this paper is organized as follows: Section 2 discusses the IEEE 802.11 security services, section 3 presents AES-CCM. Section 4 presents the UML model, which includes the analysis and design models, of the privacy service using AES-CCM. Section 5 presents the AES implementation. Section 6 presents the conclusions and our future work.

2 IEEE 802.11 Security Services

Three main security services are defined by the IEEE 802.11, as follows: Authentication, Integrity and Privacy. The *authentication* service verifies the supposed identity of a user or a system. The *integrity* service protects the data against non-authorized modifications, insertions or deletions. The *privacy* service protects the data against non-authorized revelations.

The authentication service is defined in two subtypes: Open System and Shared Key. In the former, no any authentication mechanism is used by the stations. In the latter, a shared secret key is used by the stations.

The integrity service is implemented by the Cyclic Redundancy Check (CRC)-32 algorithm, but the checksum that it creates is a noncryptographic value, known attacks, such as side-channel attacks, can compromise the data's integrity.

The privacy service is implemented by the WEP protocol, which is based on the RC4 symmetric algorithm. RC4 is a quite powerful crypto algorithm. Nevertheless, WEP takes a poor approach for using it.

Several vulnerabilities of the WEP protocol were discovered and analyzed in the last few years [2, 3, 4].

WEP uses a 64 key bit long, formed by an Initialization Vector (IV, 24 bits) and a secret key (typically, 40 bits long).

One of WEP's biggest downfalls is that its secret key is relatively shorter than other security protocols' key

Other problem is the IV because it is sent to the receiver in plaintext, which means that attackers can see the first 24 bits of every key used by WEP. Furthermore, the fact that the IV is so short nearly guarantees that it will be used for multiple messages. In fact, the same IV might be reused in as little as half of a day if there is significant activity over a company's WLAN. An attacker could easily collect an IV and use it to retrieve the key that the Access Point (AP) and wireless devices use.

WEP security also suffers from a poor solution for key management, which can leave the keys in a device unchanged for long periods of time. If the device were lost or stolen, an attacker could use the key to compromise not only that device but any other devices sharing the same key.

This is way this paper proposes to change the WEP protocol by the AES-CCM algorithm in order to implement the privacy service.

3 AES-CCM

Rijndael is a block cipher designed by J. Daemen and V. Rijmen and has been adopted as the Advanced Encryption Standard (AES). AES [5] is designed for use with blocks and keys of 128, 192 and 256 bits long. For simplicity, we use blocks and keys of 128 bits long.

AES consists of 10 rounds. Each round has a round key, derived from the original key. The original key is used in round zero. A round initiates with an input of 128 bits and produces an output of 128 bits. The algorithm treats 128-bit input block as a group of 16 bytes organized in a 4×4 matrix called State matrix. There are four basic steps, named layers, which are used to form the rounds:

1. The ByteSub (BS) Transformation: This nonlinear layer is for resistance differential and linear cryptanalysis attacks.
2. The ShiftRow (SR) Transformation: This linear mixing step causes diffusion of the bits over multiple rounds.
3. The MixColumn (MC) Transformation: This layer has a purpose similar ShiftRow.
4. AddRoundKey (ARK): The round key is or-exclusived with the result of the above layer.

Putting everything together, we obtain the Rijndael Encryption:

1. ARK, using the 0th round key.
2. Nine rounds of BS, SR, MC, ARK, using round keys 1 to 9.
3. A final round: BS, SR, ARK, using the 10th round key.

A round key is derived from the original key through a process called Key Scheduling. The Rijndael decryption algorithm operates similarly by applying inverse of all transformations described above in reverse order [5, 7].

Next we shall briefly describe the four AES transformations and the CCM mode.

3.1 The ByteSub Transformation

Each input byte of the State matrix is independently replaced by another byte from a look-up table called S-box. The S-box is a 256-entry table composed of two transformations: First each input byte is replaced with its multiplicative inverse in $GF(2^8)$ with the element {00} being mapped onto itself; followed by an affine transformation over $GF(2^8)$ [5, 7]. For decryption, inverse S-box is obtained by applying inverse affine transformation followed by multiplicative inversion in $GF(2^8)$ [5].

3.2 The ShiftRow Transformation

It is a cyclic shift operation where each row is rotated cyclically to the left using 0, 1, 2 and 3-byte offset for encryption while for decryption, rotation is applied to the right.

3.3 The MixColumn Transformation

In this transformation, the columns of the State matrix are considered as polynomials over $GF(2^8)$ and multiplied modulo x^4+1 with a fixed polynomial $c(x)$, given by:

$$c(x) = 3x^3 + x^2 + x + 2 . \quad (1)$$

This polynomial is coprime to x^4+1 and therefore invertible. This can be written as a matrix multiplication.

$$b(x) = c(x) \text{ XOR } a(x) . \quad (2)$$

Similarly, for decryption process, we compute Inverse MixColumn, by multiplying each column of State matrix by a constant fixed matrix.

3.4 The RoundKey Addition

The output of MC is XOR-ed with corresponding round sub-key derived from user key. The ARK step is essentially same for encryption and decryption processes.

3.5 The Key Schedule

The original key consists of 128 bits, which are arranged into a 4x4 matrix of bytes. This matrix is expanded by adjoming 40 more columns, as follows. Label the first four columns $W(0)$, $W(1)$, $W(2)$, $W(3)$. The new columns are generated recursively. Suppose columns up through $W(i-1)$ have been defined. If i is not a multiple of 4, then:

$$W(i) = W(i\text{-}4) \text{ XOR } W(i\text{-}1) . \tag{3}$$

If i is a multiple of 4, then:

$$W(i) = W(i\text{-}4) \text{ XOR } T(W(i\text{-}1)) . \tag{4}$$

Where $T(W(i\text{-}1))$ is the transformation of $W(i\text{-}1)$ obtained as follows. Let the elements of the column $W(i\text{-}1)$ be a, b, c, d. Shift these cyclically to obtain b, c, d, a. Now replace each of these bytes with the corresponding element in the S-box from the ByteSub step, to get 4 bytes e, f, g, h. finally, compute the round constant:

$$r(i) = 00000010^{(i-4)/4}. \tag{5}$$

In $GF(2^8)$ (recall that we are in the case where i is a multiple of 4). Then $T(W(i\text{-}1))$ is the column vector:

$$(e \text{ XOR } r(i), f, g, h) \ldots \tag{6}$$

In this way, columns $W(4), \ldots, W(43)$ are generated from the initial four columns. The round key for the ith round consists of the columns:

$$W(4i), W(4i+1), W(4i+2), W(4i+3) . \tag{7}$$

3.6 CCM Mode

The CCM mode is designed to use the AES block cipher, to provide authentication and encryption using a single block cipher key that is established beforehand. CCM is intended for use in a packet environment; the plaintext input includes a header, which is authenticated but not encrypted, and a payload, which is authenticated and encrypted. CCM operates on the whole packets; it does not support partial processing or stream processing. Each packet must be assigned a unique value, called a nonce. The size of the nonce determines the maximum number of packets that can be authenticated and encrypted with a single block cipher key.

CCM processing expands the packet size by appending an encrypted authentication tag. Successful verification of the authentication tag provides assurance that the packet originated from a source with access to the block cipher key. Consequently, successful verification of the authentication tag also provides assurance that the packet was not altered after the generation of the authentication tag. Failed verification of the authentication tag is designed to reveal intentional, unauthorized modifications of the packet, as well as accidental modifications.

4 UML Model of the IEEE 802.11 Privacy Service

This section presents the UML model of the IEEE 802.11, in DCF mode, privacy service using AES-CCM.

The UML model is composed of the analysis and design models. The analysis model is composed by the use cases diagram and the class diagram. The design model is composed by the refined class, interaction, sequence, and state diagrams.

There are two categories of MAC services: the Station Services (SS) and the Distribution System Services (DSS). SS are as follows: Privacy, Authentication, Deauthentication, MSDU delivery. DSS are as follows: Reassociation, Disassociation, Association, Integration, and Distribution.

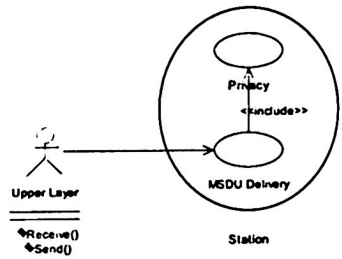

Fig. 1. Upper Layer Use Cases Diagram.. This shows an actor, who is an upper layer, which requests the SS; in this case only MSDU Delivery because the rest of the services are transparent for it. Note that MSDU Delivery includes the Privacy Service.

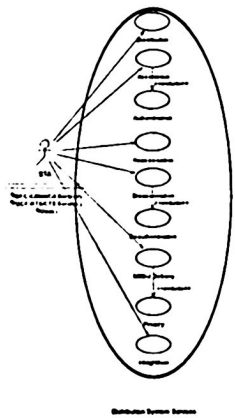

Fig. 2. Access Point Use Cases Diagram. This shows an actor, who is a Station (STA), which requests the DSS provided by an Access Point (AP).

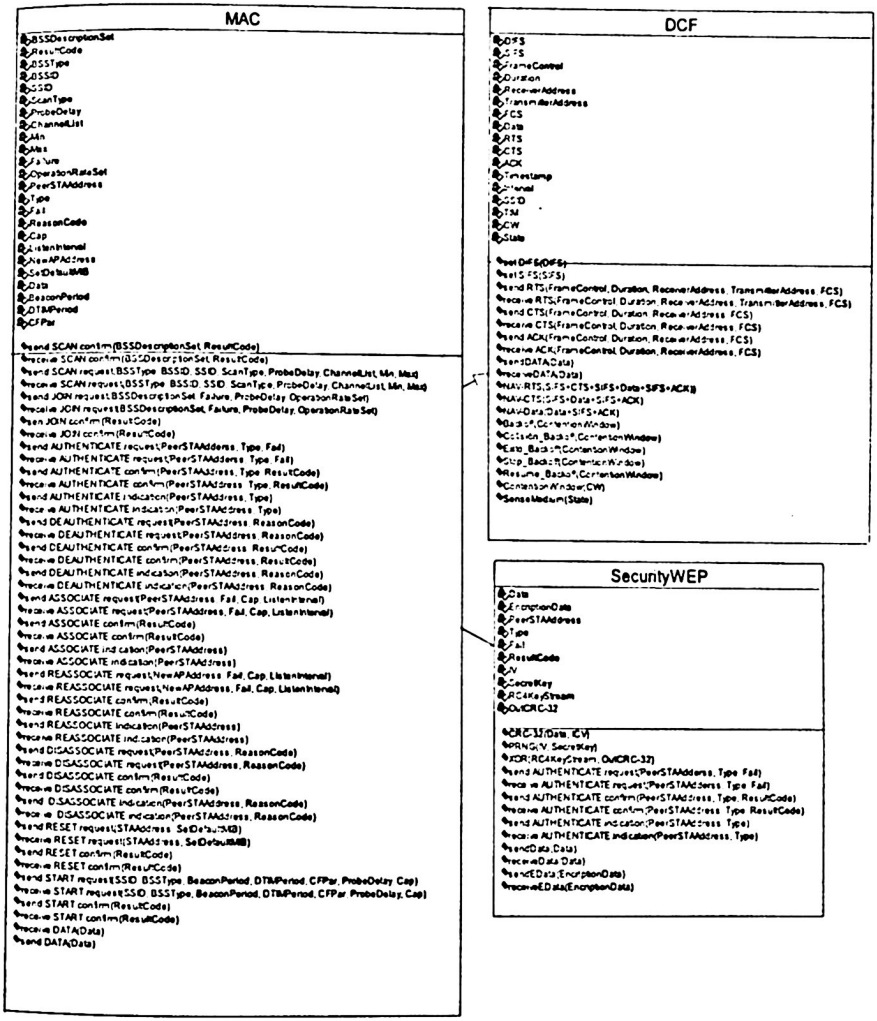

Fig. 3. IEEE 802.11 Class Diagram.. This shows the class diagram of the IEEE 802.11 MAC sublayer using DCF mode. The classes are as follows; *MAC*: It implements all the MAC services, SS and DSS. *DCF*: It implements the Distributed Coordination Function with the CSMA/CA and RTS/CTS protocol. *SecurityWEP*: It implements WEP protocol, RC4 encryption algorithm and CRC-32 algorithm.

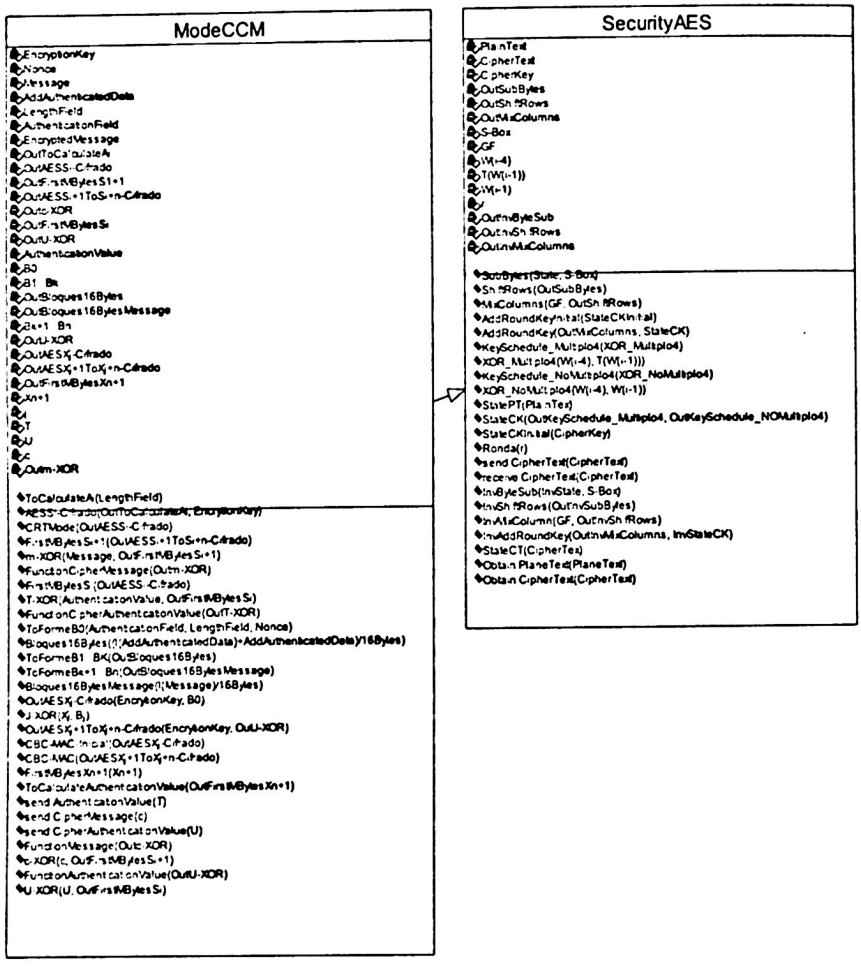

Fig. 4. AES-CCM Class Diagram. This shows the class diagram using AES-CCM, the classes are the following; CCMMode: It implements the CCM mode with an input block and key size of 128-bits. This class calls several methods of the PrivacyAES class for en/de-cryption. AESPrivacy: It implements the Rijndael algorithm, including the four basic layers and Key Scheduling process.

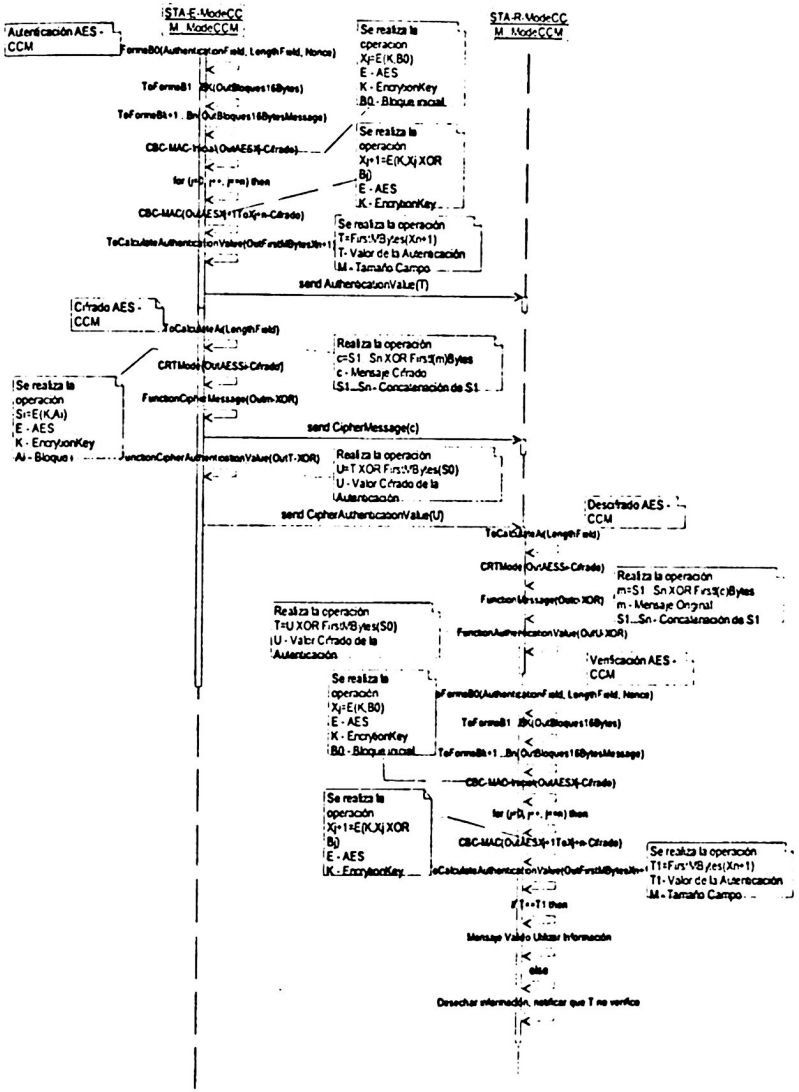

Fig. 5. AES-CCM Sequence Diagram. This shows the sequence diagram of the AES-CCM process, assuming that each STA has its secret key.

5 AES Implementation

The first step to implement AES-CCM is the AES implementation. This section presents the AES implementation is designed for use with blocks and keys of 128 bits long. This implementation used a single block cipher key that is established beforehand.

AES Program uses Microsoft Visual C++, Version 6 with platform Windows XP. This program is formed by two source files and two header files.

The MAIN.cpp source file is the main program, where the algorithm treats 128-bit input block (message) as a group of 16 bytes organized in a 4×4 matrix called State matrix and defined the key as a group of 16 bytes organized in a 4x4 matrix called CipherKey matrix.

The AES.cpp source file, it implements the Rijndael en/de-cryption algorithm, including the four basic layers and Key Scheduling process.

The AES.h header file is formed by the AES class, as follows:

```
class AES{
public:
          void ToObtainSetKey(unsigned char*Ckey);
          void CipherRijndael(unsigned char *State);
          void DecipherRijndael(unsigned char *State);
          void State(unsigned char *State);
          void RoundKeys();
private:
          unsigned char ExpKey[4][44];
          void KeyExpansion(unsigned char*Ckey);
          void ByteSub(unsigned char *State);
          void ShiftRow(unsigned char *State);
          void MixColumn(unsigned char* State);
          void AddRoundKey(unsigned char* State,int
Round);
          void InvByteSub(unsigned char *State);
          void InvShiftRow(unsigned char *State);
          void InvMixColumn(unsigned char* State);
          unsigned char Mult(unsigned char Byte, un-
signed char Val);
};
```

The AESTables.h header file, it is included S-Box, this is needed by the ByteSub transformation. Also it contains Inverse S-Box, this is used by the InvByteSub transformation.

6 Conclusions

This paper has presented the UML model of the IEEE 802.11, in DCF mode, privacy service using AES-CCM. The model includes the analysis and design models. We have discussed the IEEE 802.11 security services and the WEP vulnerabilities. For these reasons this paper has proposed to replace WEP protocol by AES-CCM.

This paper has presented the AES implementation. The implementation is designed for use with blocks and keys of 128 bits long. AES is a cipher with a simple and elegant structure, and the key scheduling algorithm shows a satisfactory security margin against cryptanalytic attacks.

This implementation is the first step to implement AES-CCM. To complete the implementation model is our future technology work as well as the performance analysis of AES-CCM into the IEEE 802.11g.

This paper uses the CCM (Counter–Mode/CBC-MAC) mode, because it provides authentication and encryption using a single block cipher key that is established beforehand. Thus, CCM requires a well-designed key management structure.

References

1. ANSI/IEEE Std 802.11, 1999 Edition. IEEE 802 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, (1999) 10-84.
2. J. R. Walker, Intel Corporation, Unsafe at any key size; An analysis of the WEP encapsulation, IEEE 802.11-00/362, October (2000) 1-9.
3. A. Stubblefield, J. Ioannidis, A. D. Rubin, Using the Fluhrer, Mantin, and Shamir Attack to Break WEP, Computer Science Dept. Rice University and AT&T Labs – Research, Florham Park, NJ, (1999) 1-11.
4. S. Fluhrer, I. Mantin and A. Shamir, Weaknesses in the Key Scheduling Algorithm of RC4, In 8th Annual Workshop on Selected Areas in Cryptography, August (2001) 1-23.
5. W. Stallings, Cryptography & Network Security: Principles and Practice, 2nd Edition, Upper Saddle River, NJ: Prentice Hall, (1998) 72-74, 121-130, 215-218.
6. D. Whiting, R. Housley and N. Ferguson, Counter with CBC-MAC (CCM)-AES Mode of Operation, NIST May (2004) 1-9.
7. J. Daemen, and V. Rijmen, The Design of Rijndael, AES-The Advanced Encryption Standard, Springer-Verlag Berlin Heidelberg, New York (2002) 31-50.