ISSN 2047-3338

Implementation of MAC Layer Security Protocol in WiMAX Using OMNET++ Simulator

Abdul Maalik[1], Mamoona Naz[2] and M. J. Qureshi[2]
[1]University of Lahore, Lahore-Pakistan
[2]Department of Computer Science and Engineering, UET, Lahore-Pakistan
[1]abdul.maalik09@gmail.com

***Abstract*– WiMAX IEEE 802.16 is stated as worldwide interoperability for microwave access intended to facilitate the interoperation and adaptation of wireless MAN. WiMAX is capable of providing high data output and low delays in different modes of operation. For these reasons WiMAX is useful in organizational and end-client structures. Before the last upgrade in the standard of WiMAX i.e., privacy and key management version 2 (PKMv2) WiMAX was not considered fully secure. The standard cannot handle the security threats associated with MAC and physical layers even after mutually authenticating the base station and mobile station. There are two objectives for WiMAX security, first is to improve privacy over the wireless link and second obligation is delivering access control effectively to the network. To achieve these objectives a better key management scheme and effective security model of WiMAX must be designed. From security perspective the Identity Based Cryptography (IBC) has advantages in improving the security and effectiveness of the standard. Little research work has been devoted in a complete and competent security solution. Objective of this thesis is to handle different security issues like replay attack, man in the middle and denial of service, implement the security protocols and algorithms using OMNET++ simulator. Different security algorithms are implemented together to enhance the existing security protocol. Every attack is partitioned according to its type, its likelihood and its impact on the system so the protocol is implemented to deal with these attacks accordingly.**

***Index Terms*– Worldwide Interoperability for Microwave Access (WiMAX), Privacy and key Management Version 2 (PKMv2), Identity Based Cryptography (IBC) and Medium Access Control Layer (MAC)**

I. INTRODUCTION

IN WiMAX, both physical and MAC layers have risk of threats like jamming [7] and denial of service respectively. But there are no efficient procedures to deal with threats posed at PHY layer of WiMAX so, the emphasis of WiMAX security is entirely at the MAC level [3]. MAC layer security threats and vulnerabilities of the WiMAX networks [8] are discussed below:

There are numerous considerable deficiencies of 802.16 security implemented at the MAC layer. In order to establish secure connections between BS and SS, 802.16 [1] utilize a sequential two-way communication for controlling, authorization, and authentication. There are many problems faced during the connection face. One major problem is that, while setting up the primary connection, management messages by MAC are launched in plain-text and are not well authenticated. Therefore there is a strong possibility that they get hacked and can give way to other attacks. Second problem is that, 802.16 [2] uses X.509 certificate, the standard for Primary Key Identification (PKI), to identify a legitimate SS. a SS's certificate is provided by the manufacture and persistent on the machine making it possible for the attacker to steal it and exploits it.

WiMAX supports one-way device MAC based authentication, similar to Wireless-fidelity (Wi-Fi) MAC filtering based on hardware address. Therefore making it possible for the attacker to launch a SS masquerade attack through address sniffing and spoofing. Also, the lack of mutual authentication makes a Man-In-The-Middle attack possible from a fake BS. Though, a successful MITM attack is not that easy to launch due to the time division multiple access (TDMA) model in WiMAX. A much high power signal must be used and that too at the same time as the legitimate BS transmits in order to hide legal signal. In addition, WiMAX supports mutual authentication based on the generic EAP [9] and also supports its variants like EAP-TLS (transport layer security) (X.509 certificate based) and EAP- SIM.

Eavesdropping is a critical threat for the users as well as the system e.g. before carrying out the attack; an attacker exploits this weakness to confirm the existence of a victim within its range. Furthermore, a competitor may map the network based on this information. Another significant threat is posed by the encryption technique used, based on DES (data encryption standard). The DES key 56 bit in length can easily be busted by applying brute force attack. The DES encryption is susceptible to replay attacks due to lack of replay protection and message integrity. Therefore AES encryption which is more secure technique should be chosen over DES.

Authentication weaknesses make it possible to launch the masquerading threat of the BS or SSs. Identity theft and Rogues BS are precisely the techniques of masquerading. WiMAX supports unlicensed services that lead to Identity theft [5]. A rogue device can make use of the MAC address of a registered device by extracting information from management messages over the air and if succeeded, an attacker can turn into a rogue BS and can easily confuse associated SSs resulting in corrupted service or even service termination.

Finally, there is a probability for Denial-of-Service (DoS) attacks because authentication can set off the implementation of long procedures. For example, a SS has to deal with a high number of authentication messages flooded by DoS attack and as SS doesn’t has sufficient computational power, will not be able to get hold of a invalid message stream, making the DoS attack successful.

A) Objectives and Scope

The primary goal of this research will be the improvement in MAC layer security of WiMAX by implementing encryption and authentication strategies that will help improve secure transmission of overall network. It will also provide suitable cases and their simulation results to conclude the global optimization of the custom objectives. The adopted simulation tool may be an OMNET++.

II. COMPARISON BETWEEN DIFFERENT SIMULATORS

A computer simulator is a computer program that serves the purpose of replicating some sort of real world system in order to extract and visualize the system functionality. It may also simulate another computer. There are two types of simulators

- Full-System Simulators
- Instruction Set Simulator (ISS)

Simulators are often used to test new software on different hardware configurations without using physical computers. It basically regenerates a real world situation which otherwise was not possible and safe. A simulator provides time, resources efficiency and in depth analysis of systems which could have been impossible in real life. Simulators have their use in science and engineering works. At present with robust hardware and refined software, simulators can accomplish more complex tasks.

Here I shall compare different simulators i.e. OMNeT++ used in my thesis, OPNET, NS2 and J-Sim.

The comparisons made here explicitly explain that simulators differ in different aspects. Simulation of different network models is influenced by the basic aim and conception of these simulators. On the other hand due to limited comparisons available a statement regarding which simulator is accurate than the other is quite judgmental but the choice can be made depending upon the application area and in my thesis OMNeT++ is appropriate as a discrete event simulator for implementing the security protocol for MAC layer of WiMAX [24]. It is designed to support wireless networks and provides simulation platform better than the other simulators discussed above specially NS-2.In past few years OMNeT++ has proved to be a feasible choice for researchers to develop open source simulation models.

Table I: Comparison between Different Simulators

SIMULATORS	Simulator Description	Developer	Programming Language	Platform	Simulation Results
NS-2	NS-2 (Network simulator -2) is also a simulation tool for discrete event and is a freeware easily available on the internet	GNU General Public License	Tcl Script	Linux, Unix, Mac-Window (via Cygwin)	Log Files / GUI via NAM
OPNET	OPNET modeler is a freeware available throughout the world as a commercial product. OPNET supports protocol models like IPv6, Q0S, Ethernet, MPLS and OSPFv3 etc.	Commercial-OPNET Technologies	C/C++	Windows	GUI
J-Sim	J-Sim (JavaSim) is a compositional component based simulation environment and involves java for its implementation.	National Simulation Resource, University of Washington	Java/Tcl Script	JAVA- Web Applet	Log Files / GUI via gEditor
OMNeT++	OMNeT++ is also a freeware and serves as a discrete event simulation framework for versatile usage.	Free/András Varga	C++	Linux, Unix, Mac-Windows	GUI

III. PROPOSED SYSTEM METHODOLOGY

PKM (Privacy Key management) Protocol and Encapsulation protocol ensure the security of WiMAX. Encapsulation protocol consists of Set of cryptographic algorithms that is data encryption and authentication algorithms. Whereas Primary Key management (PKM) Protocol consists of Set of rules for embedding such algorithms in the MPDUs payload field. This protocol ensures the secure allocation of keying information between SS and BS, so that BS can enforce conditional access to its services. PKM protocol has two versions PKMv2 and PKMv1, in our proposed work, we will implement PKMv2 and its implementation is based on RSA encryption, AES cipher and CMAC as shown in Figure 1:

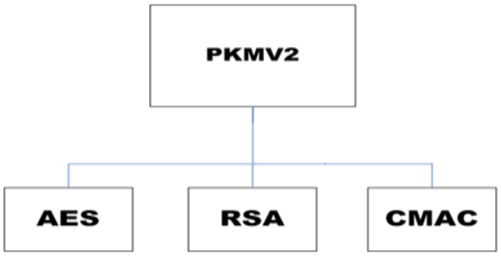


Figure 1: Implementation of PKMv2

A) RSA Algorithm

1) Key Generation Algorithm

Step 1: Select two prime numbers ‘i’ and ‘j’ large enough so that their product ‘n’ (modulus) n=i * j equals bit length of 1024 bits (not fix).

Step 2: Now calculate ‘φ’ i.e. $\varphi = (i-1)*(j-1)$

Step 3: Select an integer ‘ee’ (encryption exponent) such that it satisfies, $1 < ee < \varphi$ and Greatest Common Divisor (GCD) of ‘ee’ and ‘φ’ is 1.

Step 4: Compute an exponent ‘de’ (decryption exponent) such that $1 < de < \varphi$ and product ee * de approximately equals 1 (mod) φ.

Step 5: Now that we have calculated the values of ‘n’, ‘ee’ and ‘de’, public key is combination of ‘n’ and ‘ee’ whereas private key is combination of ‘n’ and ‘de’. It should be noted that values of ‘de’, ‘i’, ‘j’ and ‘φ’ are all undisclosed.

2) Encryption

MS encrypts the data in the following manner:

Step 1: First acquires the public key (ee , n) of BS.

Step 2: The plaintext message is represented as an integer ‘m’.

Step 3: Calculates the cipher text $C= m^{e} \bmod n$

Step 4: Sends BS the ciphertext C.

3) Decryption

BS on receiving the ciphertext responds in the following manner:

Step 1: Calculates $m= c^{d} \bmod n$ by making use of private key (de, n).

Step 2: Retrieves original plaintext message from the integer m.

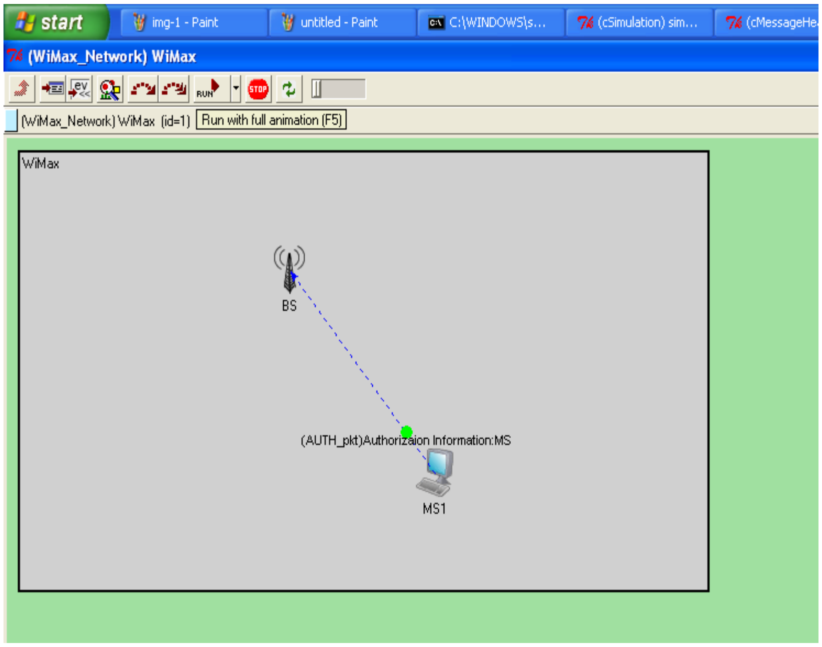


Figure 2: Authorization Information from SS

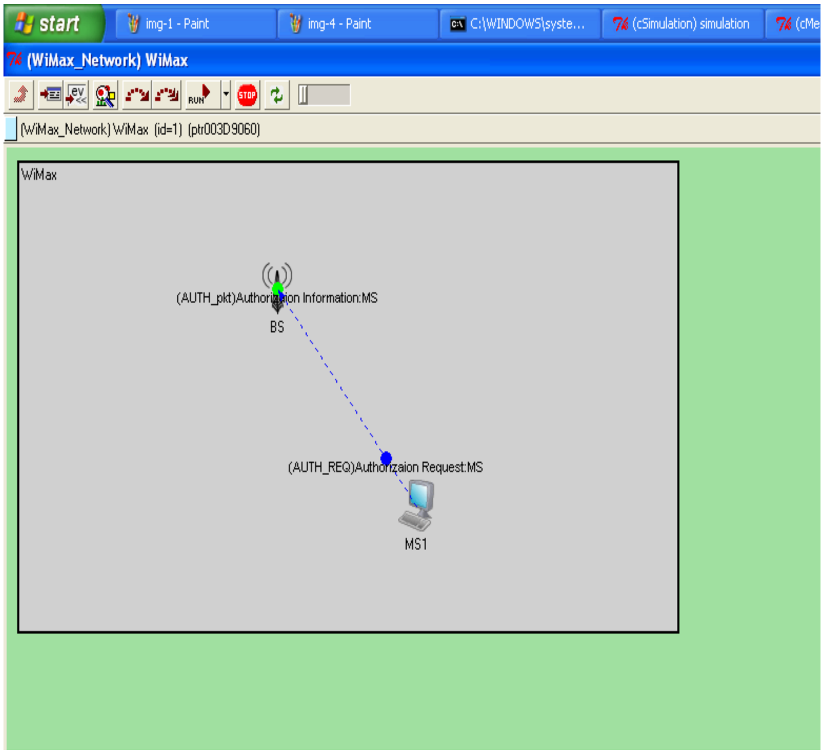


Figure 3: Authorization Request from SS

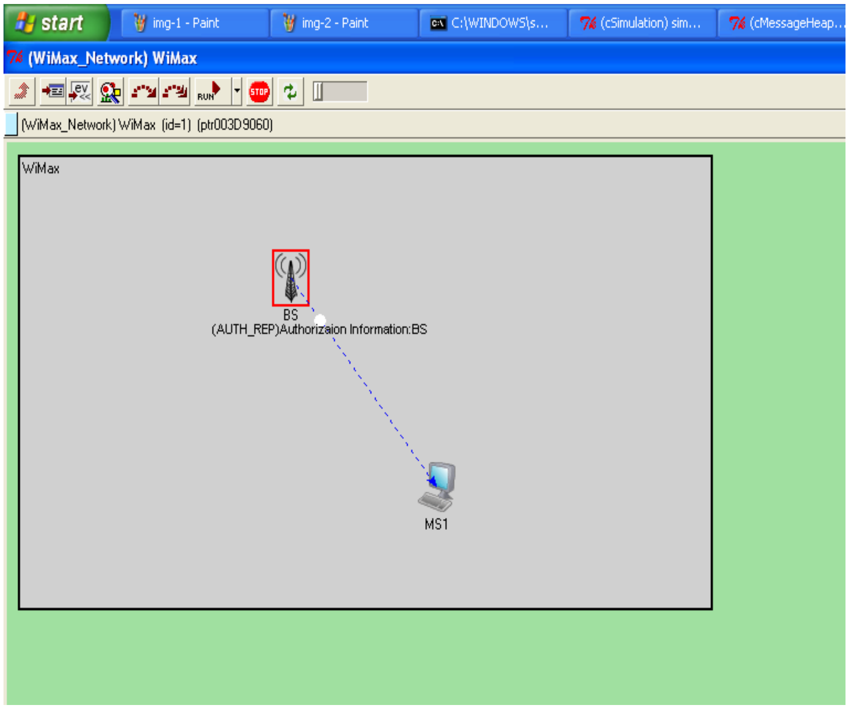


Figure 4: Encrypting Pre Pak & Generating Keys

4) Digital signature

SS performs the following steps to calculate digital signature:

Step 1: First a message digest of information is created.

Step 2: Symbolizes this message digest as an integer ‘m’ whose values range between 0 and n-1.

Step 3: Calculates the digital signature s= m ^{d} mod n by making use of its private key (n, de)

Step 4: Sends this signature to BS

5) Signature verification

BS verifies the digital signature in the following manner:

Step 1: Calculates integer 'v' such that v= s^{e} mod n y using its public key (n, ee).

Step 2: Integer 'm' helps to retrieve message digest.

Step 3: Calculates the signed message digest independently.

Step 4: If both message digests (the one it received and the one it calculated) are the same, the digital signature is valid.

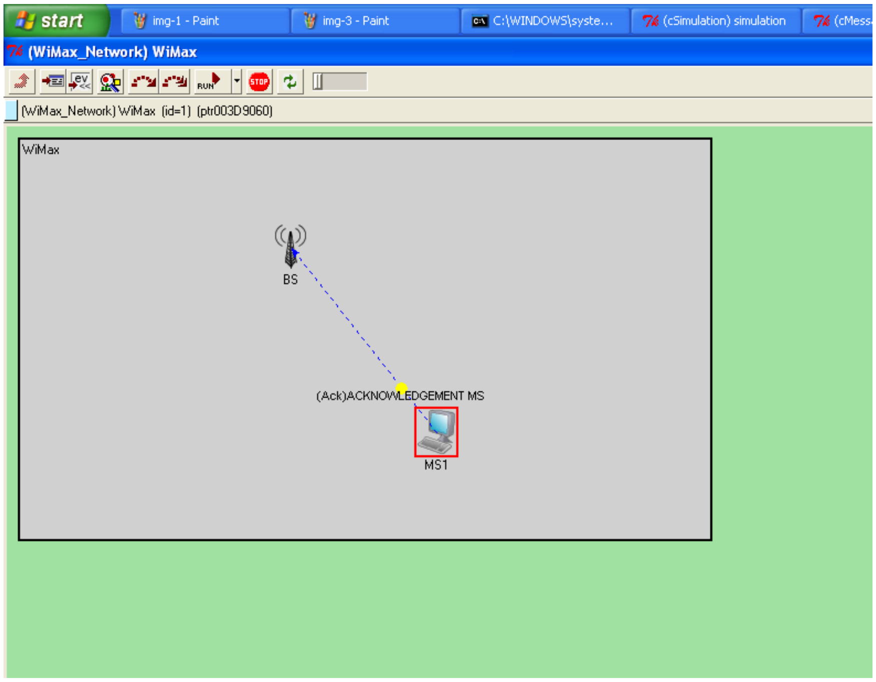


Figure 5: Key Establishment Successful

B) AES Algorithm

Advanced Encryption Standard (AES) [23] and Rijndael, often used interchangeably is a cipher (block) implemented as an encryption technique in this thesis. AES provides both the time and memory efficiency. It is easy to implement and widely being deployed, both in software and hardware.

1) Description of the cipher

AES and Rijndael are often treated as the same block ciphers but in reality there is a difference between the two, Rijndael has a support for larger key and block sizes whereas AES supports a fixed block length but 128, 192 and 256 bits key sizes. In this thesis 128 bits key size is being used. Rijndael uses key sizes and block sizes in multiples of 32.

In practice 1 byte equals 8 bits so a block or key size of 128 bits means 16 bytes which can be accommodated in a 4 x 4 matrix termed as state in AES. There are series of steps or rounds involved in the transition from input of plain test to output of cipher text at the sender end and there is a series of reverse steps to convert this cipher text back to plain text at the receiver end using the same encryption key.

2) High-level cipher algorithm

AES algorithm involves the following steps for transforming the plain text to ciphertext:

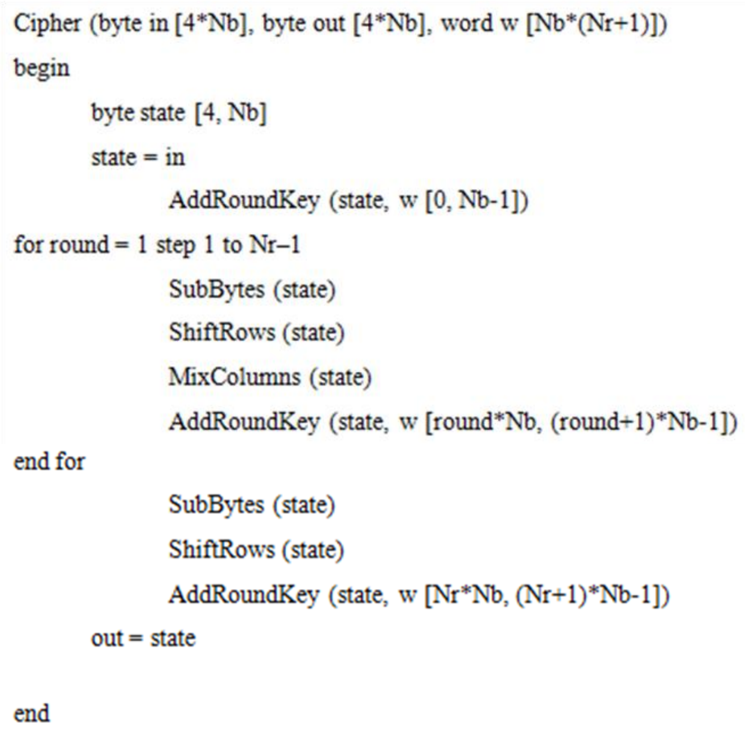
Step 1: Add Round Key to initial input matrix of bytes (i.e., 4 x 4 matrix)

Step 2: In the second phase four rounds are involved as described below:

1. *SubBytes*—each byte of matrix is replaced with another according to a table (S-Box) in a non-linear manner.
2. *ShiftRows*—repositioning of each row of the state in a circular pattern.
3. *MixColumns*—mixing of columns of the state, involve adding the four bytes together in each column.
4. *AddRoundKey*—a round key is added to each byte, round key is derived using a key schedule.

Step 3: Final phase is just the same as step 2 except that the MixColumns round is missing.

The pseudo code given below describes the cipher:



```
Cipher (byte in [4*Nb], byte out [4*Nb], word w [Nb*(Nr+1)])
begin
        byte state [4, Nb]
        state = in
                AddRoundKey (state, w [0, Nb-1])
for round = 1 step 1 to Nr–1
                SubBytes (state)
                ShiftRows (state)
                MixColumns (state)
                AddRoundKey (state, w [round*Nb, (round+1)*Nb-1])
end for
                SubBytes (state)
                ShiftRows (state)
                AddRoundKey (state, w [Nr*Nb, (Nr+1)*Nb-1])
        out = state

end
```

Figure 6: Pseudo Code for AES Cipher

i) The SubBytes step

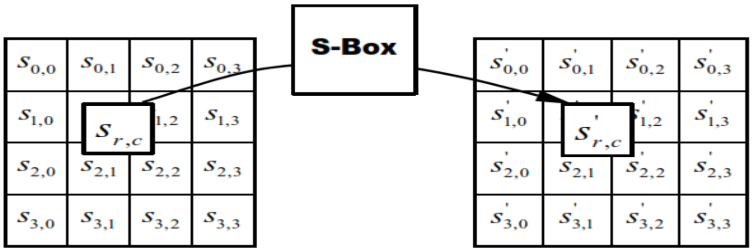

$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{r,c}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
S-Box
$s'_{0,0}$ $s'_{0,1}$ $s'_{0,2}$ $s'_{0,3}$
$s'_{1,0}$ $s'_{r,c}$ $s'_{1,3}$
$s'_{2,0}$ $s'_{2,1}$ $s'_{2,2}$ $s'_{2,3}$
$s'_{3,0}$ $s'_{3,1}$ $s'_{3,2}$ $s'_{3,3}$

Figure 7: SubBytes

In this step each byte in the state matrix is substituted with a byte derived from S-Box as shown in the Figure 7. The values derived from S-Box makes the cipher non-linear. S-Box is resultant of multiplicative inverse over GF (2^8) i.e., Galois field. S-Box is selected so as to restrict fixed points to avoid linearity and to avoid any predictive attack.

ii) The Shift-rows Step

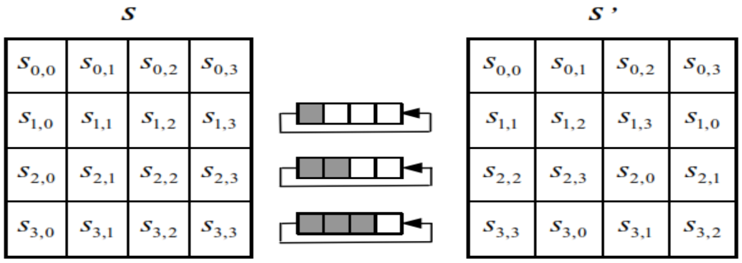

S
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
S'
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,1}$ $s_{1,2}$ $s_{1,3}$ $s_{1,0}$
$s_{2,2}$ $s_{2,3}$ $s_{2,0}$ $s_{2,1}$
$s_{3,3}$ $s_{3,0}$ $s_{3,1}$ $s_{3,2}$

Figure 8: ShiftRows

In this step a left shift is performed on each row of bytes in a circular pattern except the first row as described in the Figure 8. Second row is shifted by moving each byte to the left once, rows two and three are rotated to the left twice and thrice respectively.

iii) The MixColumns step

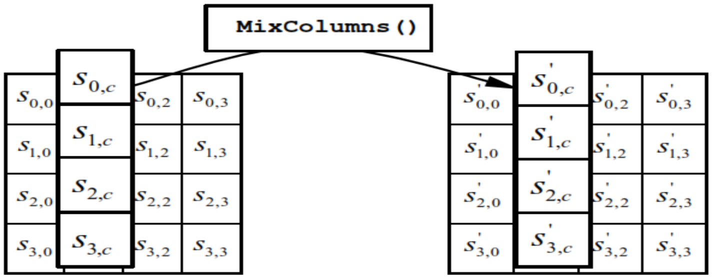

MixColumns()
$s_{0,0}$ $s_{0,c}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,c}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,c}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,c}$ $s_{3,2}$ $s_{3,3}$
$s'_{0,0}$ $s'_{0,c}$ $s'_{0,2}$ $s'_{0,3}$
$s'_{1,0}$ $s'_{1,c}$ $s'_{1,2}$ $s'_{1,3}$
$s'_{2,0}$ $s'_{2,c}$ $s'_{2,2}$ $s'_{2,3}$
$s'_{3,0}$ $s'_{3,c}$ $s'_{3,2}$ $s'_{3,3}$

Figure 9: MixColumns

In this step each byte a column is multiplied by a fixed polynomial described in the Figure 10:

$$s'_{0,c} = (\{02\} \bullet s_{0,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c}$$

$$s'_{1,c} = s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c}$$

$$s'_{2,c} = s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c})$$

$$s'_{3,c} = (\{03\} \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c}).$$

Figure 10: Multiplication

This function outputs four bytes affected by an input byte and provides diffusion in the cipher.

iv) The AddRoundKey Step

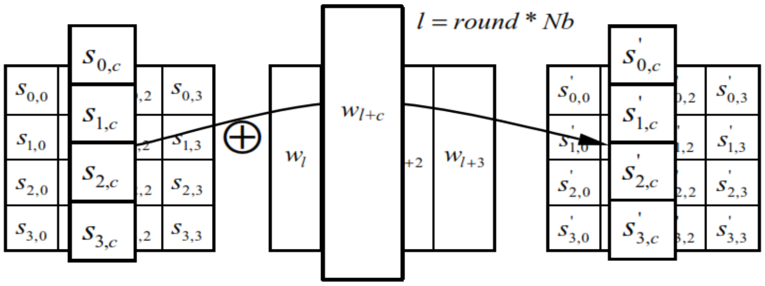

$l = round * Nb$
$s_{0,0}$ $s_{0,c}$ $s_{0,3}$
$s_{1,0}$ $s_{1,c}$ $s_{1,3}$
$s_{2,0}$ $s_{2,c}$ $s_{2,3}$
$s_{3,0}$ $s_{3,c}$ $s_{3,3}$
$\oplus$
w_l
w_{l+c}
w_{l+3}
$s'_{0,0}$ $s'_{0,c}$ $s'_{0,3}$
$s'_{1,0}$ $s'_{1,c}$ $s'_{1,3}$
$s'_{2,0}$ $s'_{2,c}$ $s'_{2,3}$
$s'_{3,0}$ $s'_{3,c}$ $s'_{3,3}$

Figure 11: AddRoundKey

In this step each byte is bitwise XORed with a sub-key to produce the resultant cipher as described in the Figure 11.

3) Optimization of the Cipher

Cipher can further be optimized keeping in my mind that system can support 32-bit words by adding the ShiftRows and SubBytes together with MixColumns; this would require a memory of 4KB, one KB for each lookup table. Each table will be 32 bit consisting of four 256-entries.

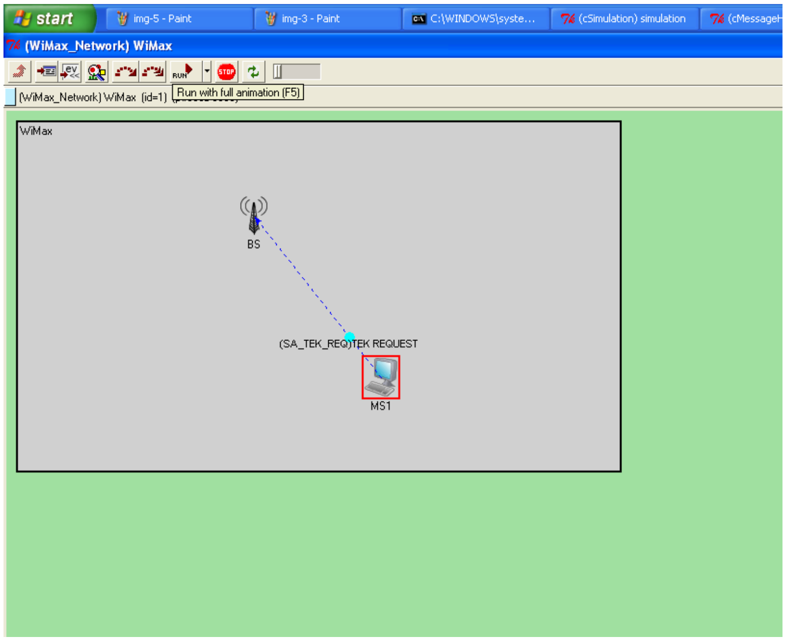


Figure 12: Encrypting TEK with KEK

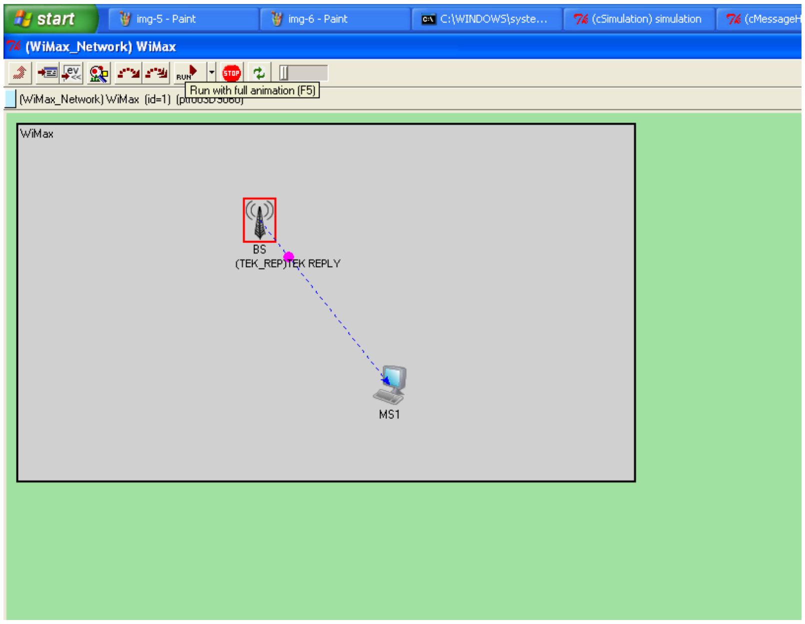


Figure 13: Generation of TEK using AES Cipher

C) CMAC Algorithm

The CMAC algorithm basically consists of a sub-key generation, MAC generation and MAC verification algorithm described comprehensively in the following text.

1) Sub-key Generation Algorithm

This algorithm works by taking a secret key similar to AES-128 and generates two output keys K1 and K2, both these keys are used in other two algorithms of CMAC.K1 is used for similar length block and K2 is used for variable length block. The steps involved in the working of this algorithm are described below:

Step 1: AES-128 is implemented on input block with all zeroes in it to produce an output variable L.

Step 2: Derivation of K1 depends on a conditional statement that if the most significant of L derived in step 1 is zero then it is shifted to left by 1 bit to produce K1 otherwise it is XORed with a constant 0x0^30 87 and shifted left by 1 bit.

Step 3: Derivation of K2 depends on a conditional statement that if the most significant of K1 derived in step 2 is zero then it is shifted to left by 1 bit to produce K2 otherwise it is XORed with a constant 0x0^30 87 and shifted left by 1 bit.

Step 4: Keys K1 and K2 are returned.

2) MAC Generation Algorithm

Step 1: In this step, sub-key generation algorithm is called to generate sub-keys K1 and K2.

Step 2: In this step, the number of blocks, n, is determined, dividing the length by the bock size i.e.128 bit.

Step 3: In this step the input message length is verified if it is zero the blocks for processing will be 1 and NOT-COMPLETE-BLOCK flag will return true and if it is 128 bits the COMPLETE-BLOCK flag is marked true.

Step 4: In this step, M_Last is determined by XORing M_n and sub-keys calculated in step 1.If COMPLETE-BLOCK flag returns true then K1 and M_n are XORed else K2 and padded M_n will be XORed.

Step 5: This step involves initialization of variable X.

Step 6: CBC-MAC is implemented with M_1….M_ (n-1), M_Last (Calculated in step 4)

Step 7: MAC from calculating AES-CMAC (K, M_len) is returned and is 128 bit in length. It is truncated before being returned.

3) MAC Verification Algorithm

The MAC verification algorithm generally involves recalculating MAC by applying MAC generation algorithm again. Four parameters are taken as an input in this algorithm a sub-key, input message, its length in 16 octets and MAC say T'. The output of this algorithm can be INVALID or VALID.

Step 1: Calculate T* by applying MAC generation algorithm to sub-key, M and M_len.

Step 2: Compare T* and T' if they are equal MAC verification algorithm returns VALID otherwise it returns INVALID.

If MAC verification algorithm returns INVALID it means message has been corrupted and is not originated from the source which actually generated the MAC T'.

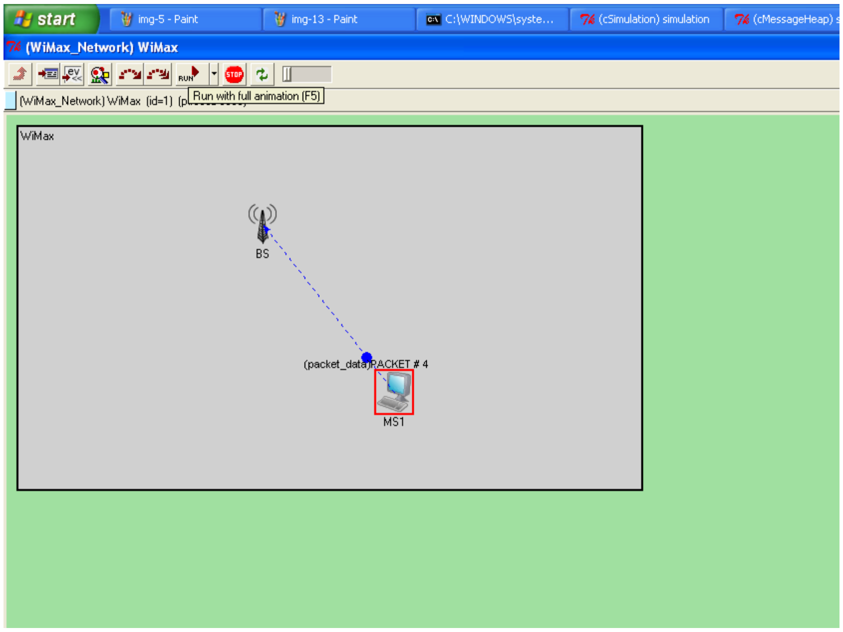


Figure 14: Message Encryption Using CMAC

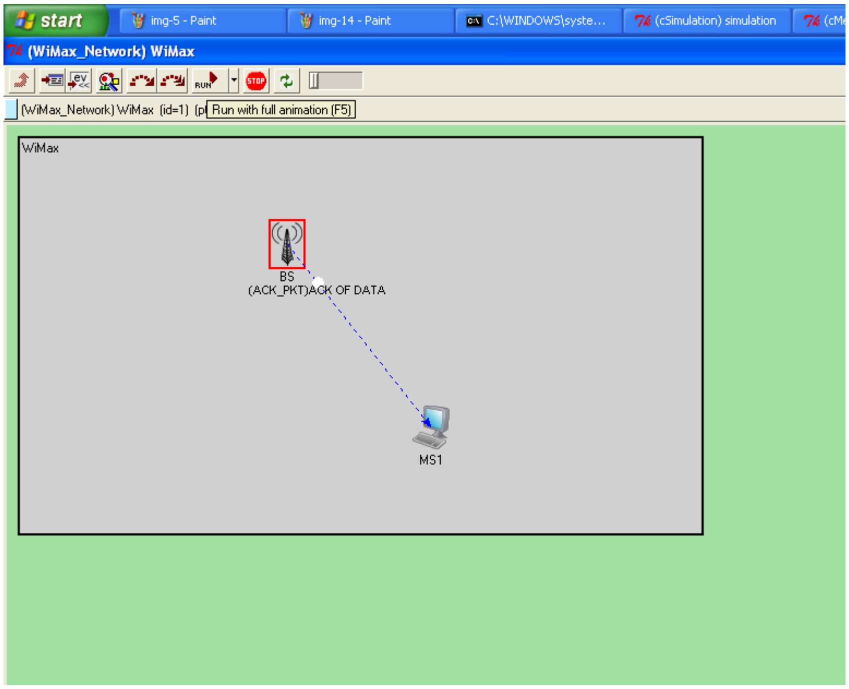


Figure 15: Acknowledgement of Encrypted Message

IV. CONCLUSION AND FUTURE RECOMMENDATIONS

We started the project with the objectives mentioned in the project motivation and achieved most of them. We have implemented the security solutions against most of the threats related to MAC layer. The threats like Rouge BS and Replay Attacks have totally been removed. Advance data protection is also being provided by using the latest in use algorithms. Issues regarding WiMAX security sublayer are conferred in detail, our primary goal was to maintain privacy and provide better authentication. We have successfully simulated the AES and RSA based security sublayer using OMNeT++. The security parameters used for the authentication by the BS and SS are their unique ID's, message type, SS manufacturer digital certificate, SAID and lifetime.

We would like to recommend the following features to be incorporated in our developed simulation model in order to further improve the security sublayer implementation with respect to latest WiMAX standard.

- Further modifications can also be made easily in OMNeT model to completely implement the whole MAC layer. The processes of bandwidth request and management messages can also be added.
- To show the complete MPDU format, the header and further related fields can also be added in messages definitions.
- Also the same simulation model should be practically implemented on other simulators and the parameters like time and memory efficiency be compared.

REFERENCES

[1] Bogdanoski, Mitko.; Latkoski, Pero.; Risteski, Aleksandar.; Popovski, Borislav.; "IEEE 802.16 Security Issues: A Survey," In proceeding of TELFOR, 16th Telecommunications Forum, pp., 25-27 Nov. 2008

[2] Sidharth, Sreejesh; Sebastian, M.P., "A Revised Secure Authentication Protocol for IEEE 802.16 (e)," In proceeding of Advances in Computer Engineering (ACE), pp.34-38, 20-21 June 2010.

[3] Habib, M.; Mehmood, T.; Ullah, F.; Ibrahim, M., "Performance of WiMAX Security Algorithm (The Comparative Study of RSA Encryption Algorithm with ECC Encryption Algorithm)," In proceeding of Computer Technology and Development, 2009 (ICCTD '09), Vol. 2, pp.108-112, 13-15 Nov. 2009.

[4] Sen Xu; Chin-Tser Huang, "Attacks on PKM Protocols of IEEE 802.16 and Its Later Versions," Wireless Communication Systems, 2006. ISWCS '06. 3rd International Symposium, pp.185-189, 6-8 Sept. 2006.

[5] Shahid, Adnan.; Fisal, Norsheila.; Hussain, Sazzad.; "Man-in-the-Middle Attack and Possible Solutions on WiMax 802.16j," In proceeding of Recent and Emerging Advance Technologies in Engineering (iCREATE), 2009.

[6] Shahzadi, Romana.; Shahzad, Asim.; "AES based security architecture of WIMAX using OMNET++," Video & Image Processing and Network Security, International Journal on, Vol. 9, No.10, pp.12-16, Dec. 2009.

[7] Barbeau,Michel.; "WiMax/802.16 Threat Analysis," Association for Computing Machinery, pp. Oct. 2005.

[8] Kolias, C.; Kambourakis, G.; Gritzalis, S., "Attacks and Countermeasures on 802.16: Analysis and Assessment," Communications Surveys & Tutorials, IEEE, No.99, pp.1-28.

[9] Lang Wei-min; Zhong Jing-li; Li Jian-jun; Qi Xiang-yu, "Research on the Authentication Scheme of WiMAX," In

proceeding of Wireless Communications, Networking and Mobile Computing, (WiCOM '08), pp.1-4, Oct. 2008.

[10] Lang Wei-min; Wu Run-sheng; Wang Jian-qiu, "A Simple Key Management Scheme Based on WiMAX," In proceeding of Computer Science and Computational Technology (ISCSCT '08), Vol. 1, pp.3-6, 20-22 Dec. 2008.

[11] S. Xu, M. Matthews, and C.-T. Huang, "Security Issues in Privacy and Key Management Protocols of IEEE 802.16", Proceedings of the 44th ACM Southeast Conference (ACMSE 2006), March 2006

[12] Altaf, A.; Javed, M.Y.; Ahmed, A., "Security Enhancements for Privacy and Key Management Protocol in IEEE 802.16e-2005," Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008. SNPD '08. Ninth ACIS International Conference on , vol., no., pp.335,339, 6-8 Aug. 2008

[13] Altaf, A.; Sirhindi, R.; Ahmed, A., "A Novel Approach against DoS Attacks in WiMAX Authentication Using Visual Cryptography," Emerging Security Information, Systems and Technologies, 2008. SECURWARE '08. Second International Conference on , vol., no., pp.238,242, 25-31 Aug. 2008

[14] William Stallings, “Cryptography And Network Security,” Prentice Hall; 5th edition, 2006

[15] Fan Yang, "Comparative Analysis on TEK Exchange between PKMv1 and PKMV2 for WiMAX," Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference, pp.1,4, 23-25 Sept. 2011

[16] Syed Ahson Mohammad Ilyas, “WiMAX Standards and Security,” CRC Press; 5th edition, 2008

[17] Luo Cuilan, "A Simple Encryption Scheme Based on WiMAX," E-Business and Information System Security, 2009. EBISS '09. International Conference, pp.1,4, 23-24 May 2009

[18] Alanazi. O. Hamdan, Zaidan .B.B, Zaidan .A.A, Jalab .A. Hami., Shabbir .M, Al-Nabhani .Y, “New Comparative Study Between DES, 3DES and AES within Nine Factors,” JOURNAL OF COMPUTING, vol. 2, no. 3, pp.154-157., March 2010.

[19] Reineck Karsten. M, “Evaluation and Comparison of Network Simulation Tools,” Institute for Open Communication Systems, August 2008.

[20] “OMNeT++ Documentation,” Available at: www.omnetpp.org

[21] “NS-2 Documentation,” Available at: www.isi.edu/nsnam/ns/

[22] “JavaSIM Documentation,” Available at: www.j-sim.org

[23] Federal Information Processing Standards Publication” Announcing the Advanced encryption standard (AES).” National Institute of Standards and Technology (NIST) on vol., no., pp., November 26, 2001

[24] “WiMAX Documentation,” Available at: www.wimaxforum.org

[25] IEEE Std. 802.16e, air interface for fixed and mobile broadband wireless access systems. IEEE Standard for local and Metropolitan Area Networks, February 2006.