# Adaptive Message Authentication Protocol for Performance Improvement in ZigBee Wireless Sensor Network

**SAIF M. KH. ALALAK**
Babylon University, College of Science for Women
E-mail: saif.shareefy@gmail.com

**SAHAR A. KADHUM**
Babylon University, College of Science for Women
E-mail: salaa_38@yahoo.com

**MAJID J. JAWAD**
Babylon University, College of Science for Women
E-mail: majid_al_sirafi@yahoo.com

## Abstract

ZigBee is used as a low power and low rate wireless sensor network. Many applications utilize it for data transfer. It provides some security options, including message authentication, message encryption and defense against replay attack. The security algorithms used in the ZigBee are run for each packet transfer when the security mode is on. This study tested the impact of message authentication over the network performance in terms of the throughput and data packet delivery ratio. The message authentication leads to performance degradation for the network. The blocks of the message are encrypted by using Advanced Encryption Standard (AES) algorithm for authenticating in CBC-MAC algorithm. The performance of the network is improved over the original state when AES algorithm is replaced by MKP-AES algorithm and DMAC algorithm is used to parallelize the authentication. MKP-AES is tested within two and three secret keys. The network is simulated using NS-2 to measure the performance improvement.

**Keywords**: Data packet delivery ratio, Message Authentication, Security Improvement, Throughput, ZigBee

## الخلاصة

تعتبر شبكة الزك بي احدى شبكات المتحسسات اللاسلكية ذات الاستهلاك الواطئ لمعدل الطاقة. وقد تم استخدامها من قبل عدة تطبيقات لنقل البيانات، فهي توفر عدد من الامكانيات الامنية المتضمنة توثيق الرسائل وتشفيرها والدفاع ضد هجوم اعادة الارسال. ان الخوارزميات الحالية المستخدمة في شبكة الزك بي، تطبق على كل حزمة منقولة حسب احتياجاتها الامنية. في هذا البحث تم اختبار توثيق الرسائل على كفاءة الشبكة المتضمنة كمية الانجاز ونسبة البيانات المستلمة. ان توثيق الرسائل تؤدي الى تقليل نسبة كفاءة الشبكة حيث ان مقاطع الرسالة تشفر بأستخدام خوارزمية (AES) لغرض الموثوقية في خوارزمية CBC-MAC. في هذا البحث تم تحسين كفاءة الشبكة لتكون افضل من الوضع السابق لها من خلال استخدام خوارزمية MKP-AES و DMAC بدلا من خوارزمية AES الاعتيادية لتوثيق الرسائل بالتوازي. تم اختبار خوارزمية MKP-AES بحالتين، الحالة الاولى بأستخام مفتاحين والحالة الثاني بأستخدام ثلاثة مفاتيح امنية. متم استخدام برنامج NS2 لقياس درجة تحسن كفاءة الشبكة.

## 1. Introduction

The military, health and industry applications utilize the ZigBee sensors because of its low cost and power consumption. Security is one of the important features provided by ZigBee sensors. ZigBee sensors have multiple security options. Confidentiality, authentication and replay attack prevention are provided in MAC sub-layer. It utilizes Advanced Encryption Standard (AES) (NIST, 2001) algorithm for message ciphering and Cipher Block Chaining Message Authentication Code (CBC-MAC) (Rogaway, 2011) algorithm for message authentication.

The study measures the impact of message authentication over ZigBee network performance. The problem is that the securing of the message over the ZigBee sensors has high cost in terms of network performance. The research question is: how could Multiple-Key Protocol Advanced Encryption Standard (MKP-AES) (Al-alak et al., 2011) algorithm can reduce the security cost over network performance? Distributed Message Authentication Code (DMAC) (Al-alak et al., 2012a) algorithm is used to distribute the groups over the computing elements.

The network environment to test the network throughput and data packet delivery ratio is NS-2 simulator version 2.28. (Woon and Wan, 2008) The message authentication based on MKP-AES algorithm would be used to improve the network throughput and data packet delivery ratio for ZigBee network.

Many researchers have measured the performance of ZigBee sensors which is IEEE 802.15.4 based network. Zheng and Lee (2004) evaluated and compared the performance of 802.15.4 over 802.11 for low-rate wireless area network. Di Francesco et al. (2011) proposed an adaptively cross-layer framework that tunes the parameters of MAC layer to provide energy-efficient data collection for IEEE 802.15.4 standards. Charfi and Bouyahi (2012) developed the NS-2 simulator to measure the network performance when packet sending in Guaranteed Time Slot (GTS) of time. Daidone et al. (2011) performed an experimentally evaluation for the impact of security services on the performance of IEEE 802.15.4 standard.

The aim of this study is to evaluate the impact of message authentication operation on ZigBee performance. The study measures the ZigBee performnce when CBC-MAC/AES and DMAC/MKP-AES are used for message authentication.

## 2. Zigbee Background

The ZigBee wireless sensor is built over the IEEE 802.15.4 as a Low Rate Wireless Personal Area Network (LR-WPAN) standard, which is utilized in many platforms. ZigBee is used to transfer information over relatively short distances with limited power and relaxed throughput requirements. LR-WPAN data rates are 250 kb/s, 100kb/s, 40 kb/s and 20 kb/s. The topology is either star or peer-to-peer operation. It allocates 16-bit, short or 64-bit extended addresses. It has optional allocation of Guaranteed Time Slots (GTSs). The Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) protocol is utilized for channel access. It provides fully acknowledged protocol for transfer reliability and supports low power consumption, Energy Detection (ED) and Link Quality Indication (LQI). Different number of channels is provided depending on the radio frequency in the physical layer, 16 channels (11 to 26) for the 2450 MHz band, 10 channels (1 to 10) for the 915 MHz band and 1 channel for the 868 MHz band.

The ZigBee wireless sensor network system devices are either Fully Function Device (FFD) or Reduces Function Device (RFD). FFD device must be included in any network system. FFD can contact to FFD and RFD, but RFD can contact to FFD only. FFD has three service modes: personal area network (PAN) coordinator, coordinator, or device. RFD is intended to be a simple device which doesn't need to transfer a large amount of data.

Star and peer-to-peer topology can be used to construct ZigBee network systems. For star topology, the PAN coordinator is a FFD that has fixed 64-bit identification address and it is considered as the primary controller of the PAN in the network. In addition to PAN coordinator

association with a particular application, it can initiate, terminate, or route communication over the network. Meanwhile, in the peer-to-peer topology, the devices can communicate with each other through the potential range, where one of them is nominated as PAN coordinator. Within peer-to-peer topology, it is possible to do multiple hops for messages routing from a device to others in the network. A cluster tree network is an example of the peer-to-peer topology.

In MAC sub-layer of ZigBee sensor, data confidentiality, data authenticity and replay attack prevention are provided. It adopts a block cipher AES algorithm for data encryption, CBC-MAC algorithm for data authentication and nonce for defense against replay attack. The security is implemented in CCM security mode. It stands for Counter (CTR) Cipher Block Chaining (CBC) Message Authentication Code (MAC). The security mode is written as CCM*, which is a logical extension of CCM mode. ZigBee sensor has eight different security options as shown in table 1. It is possible to do both message encryption and authentication or only one of them. The authentication algorithm can be run with different tag length (32, 64 and 128). Additionally, it has a no security state.

Table 1: Security options of IEEE 802.15.4 standard

| Algorithms | Name | Encryption | Authentication |
|---|---|---|---|
| No security | | No | No |
| AES-CBC-MAC[1] -32 | MIC[2]-32 | No | Yes |
| AES-CBC-MAC-64 | MIC-64 | No | Yes |
| AES-CBC-MAC-128 | MIC-128 | No | Yes |
| AES-CTR[3] | ENC | Yes | No |
| AES-CCM-32 | AES-CCM-32 | Yes | Yes |
| AES-CCM[4]-64 | AES-CCM-64 | Yes | Yes |
| AES-CCM-128 | AES-CCM-128 | Yes | Yes |

## 3. MKP-AES Algorithm

The AES algorithm is replaced by the MKP-AES algorithm because the original AES has many types of attacks and the MKP-AES improved the randomness of ciphertext over normal AES algorithm as described in (Al-alak et al., 2012b).

In spite of the success of AES algorithm in passing the security tests, it does not mean that this algorithm is an immune. Although this security algorithm is very complex, the intruders are working on breaking it. For AES algorithm, the complexity of 128-bit key is $O(\log_2 64)$ and the attacker needs $2^{64}$ operation to expose it by traditional way (Housley et al., 2003). The time complexity of the secret key is based on the key length. In more details, there are many attacks over the AES algorithm, which detect the weakness point in the AES algorithm. The brute force attack (Bernstein, 2005) is a traditional way to break the secret key that performs one trial decryption for each key. The XSL attack (Cid and Leurent, 2005) is a theoretical attack based on the simple algebraic description for the AES. The related-key attack (Biryukov and Khovratovich, 2009) on the 192-bit and 256-bit versions of AES exploits the simple key schedule. The known-key distinguishing attack (Gilbert and Peyrin, 2010) is a developed version of the start-from-the-middle attack; it works on the 8-round version of AES-128. The chosen-key-relations-in-the-middle attack on AES is described in (Rijmen,

[1] AES-CBC-MAC: Advanced Encryption Standard – Cipher Block Chaining – Message Authentication Code
[2] MIC: Message Integrity Code
[3] AES-CTR: Advanced Encryption Standard - Counter
[4] AES-CCM: Advanced Encryption Standard – Counter CBC-MAC Mode

2009). The first key-recovery attack (Bogdanov et al., 2011) on full AES is faster than brute force by a factor of about four.

The MKP-AES has multiple secret keys against brute force attack in comparison with original AES. Furthermore, the 128-bit key length is still secure against related key attack. The robustness of MKP-AES is that it uses multiple secret keys with 128-bit length. The code of MKP-AES works as follows:

***BlockSize = 128***
***for i = 1 to Number_of_keys***
***k[i] = ECC(k[i-1], PK)***
***r[i] = ECC(r[i-1], PK)***
***for i = 1 to Number_of_keys***
***K[i] = f(k[i], r[n-i+1])***
***No_of_Blocks = Message / BlockSize***
***No_of_Blocks_in_Group = No_of_Blocks / Number_of_keys***
***count = 1***
***Key = K[count]***
***for j =1 to No_of_Blocks***
***C[j] = AES (B[j], Key)***
***if (j mod No_of_Blocks_in_Group == 0) then Key = K[count++]***

The number of secret keys is limited by the value of (Number_of_keys). Two lists of parameters are computed by ciphering the initial values ($r_0$ and $k_0$) using Elliptic Curve Cryptosystem (ECC) to produce parameters ($r_1$ and $k_1$) then it continues ciphering the computed parameters to produce new parameters ($r_i$ and $k_i$) until the end of the loop. PK refers to the public key of ECC from a trust center. Each secret key is computed by hashing (f) two parameters from different level ($r_i$, $k_{n-i+1}$) to prevent the attacker from tracking the secret key when one secret key is exposed, so that the secret keys are protected by ECC system. The algorithm groups the blocks according to a number of keys (it assumed that the number of blocks is divisible by the number of keys).

The MKP-AES algorithm employs multiple secret keys to cipher the plaintext; however, AES algorithm uses single secret key to cipher the plaintext. It is supposed that the plaintext consists of (*m*) blocks (block size is 16 bytes) then the blocks of plaintext can be classified into (*n*) groups as shown in eq.1 and the number of blocks per group is roughly equal. The number of groups must be equal to the number of secret keys. Each secret key ($K_i$) is assigned to a separate group ($G_i$) of blocks.

$$\text{Message} = \sum \text{Group\_i} \ (i = 1 \text{ to } n) \qquad (1)$$

The number of blocks in the plaintext may be not divisible by the number of groups so that sometimes the number of blocks in Group_i is not equal to the number of blocks in Group_j. In MKP-AES algorithm the number of blocks per group is at least one block and the total number of blocks in the plaintext must be greater than or equal to the number of groups.

In the encryption operation the MKP-AES algorithm ciphers the Block_j from group Group_i using its already assigned secret key Key_i where the cipher text block Cipher_j from Group_i is computed according to eq.2 where the encryption algorithm (Enc) is AES and the block size is 16-octet.

$$\text{Cipher\_j} = \text{Enc}_{\text{Key\_i}} (\text{Block\_j}) \qquad (2)$$

In the decryption operation the MKP-AES algorithm repeats the same steps from the encryption operation. It breaks the cipher text to the same number of groups of the encryption then assigns the secret keys to each group. Each block cipher Cipher_j belong to group Group_i is deciphering using secret key Key_i as shown in eq. 3 to produce the original plain text block Block_j. Because the AES algorithm is symmetric the MKP-AES used the same secret keys for blocks ciphering and deciphering.

$$Block\_j = Enc_{Key\_i}(Cipher\_j) \quad (3)$$

The MKP-AES provides a protocol to generate the secret keys inside message's sender and receiver and they don't need to transfer the keys over network. Both of them must agree upon the number of secret keys before they start generating the keys. MKP-AES adopts the ECC public key system to generate the secret keys because it requires less computational power, memory and communication bandwidth in compaison with other traditional public key crypto-algorithms

The MKP-AES assumes that there are two secure initial values $r_0$ and $k_0$ that have been generated by a trust center and have been established securely to the sender and receiver. The trust center uses the ECC algorithm with node's public key to ensure securely transfer of the initial values. Inside the sender and receiver nodes two lists of parameters $r_1, r_2, \ldots r_n$ and $k_1, k_2, \ldots k_n$ are computed by ECC with public key of trust center as shown in eq.4 and eq.5 where the number of secret keys is n.

$$r_i = ECC_{enc}(TC_{PKey}, r_{i-1}) \; 0 < i \leq n \quad (4)$$

$$k_i = ECC_{enc}(TC_{PKey}, k_{i-1}) \; 0 < i \leq n \quad (5)$$

When the number of secret keys is (n) a list of secret keys (Key_1…Key_n) is computed inside the node where the secret key Key_i is computed by a hash function (f) as shown in eq.6.

$$Key\_i = f(k_i, r_{n-i+1}) \; 1 \leq i \leq n \quad (6)$$

## 4. DMAC Algorithm

The DMAC algorithm improves the authentication time of CBC-MAC algorithm with MKP-AES over multiple computing devices system. It distributes the groups of blocks over computing devices. The number of groups is computed by the MKP-AES algorithm, where the number of groups must be equivalent to the number of secret keys.

The MKP-AES provides multiple-key AES where each key is used for authenticating single group of blocks (sub-message) and the output is XORing with authentication code computed from the previous group and so on. MKP-AES has a possibility of parallel the authentication computations of groups. The authentication code for individual group is computed in a single computing device because all the blocks of that group are sequentially computed to produce the authentication code of it. The DMAC algorithm distributes the groups over available devices as follows:

***If number_of_groups ≤ number_of_devices then***
***Assign each device_i to group_i where i = 1 to number_of_groups***
***Else For each group_i***
***If i ≤ number_of_device then***
***Assign device_i to group_i***
***Else assign device_j to group_i where j = i - number_of_devices***

Table 2 shows the ratio of authentication time improvement for the CBC-MAC by using DMAC algorithm. It improves CBC-MAC authentication time with MKP-AES. The system is partially utilizing the available computing devices when the number of groups is less than the number of computing devices. The number of working devices is limited to the number of groups. However, the system is fully utilizing the computing devices when the number of groups is equal to the number of cores. When the number of groups is greater than the number of cores and the number of groups is divisible by the number of cores then it fully utilizes the cores. While, when the number of groups is not divisible by the number of cores and the

number of groups is greater than the number of cores then the computing devices are partially utilized.

Table 2: Authentication time improvement rate using DMAC[5] algorithm (*Al-alak et al., 2012a*)

| | 2-key | 3-key | 4-key | 5-key |
|---|---|---|---|---|
| 2-core CPU | 40% | 28% | 40% | 30% |
| 3-core CPU | 40% | 50% | 40% | 53% |
| 4-core CPU | 40% | 50% | 55% | 36% |

## 5. Authentication Time Computing

Let Z be a function to compute the size of message as pointed in eq. (7). For MKP protocol, the number of secret keys is greater than 1, so that the function Z is computed in new equation when the MKP protocol is involved in CCM mode as shown in eq. (8).

$$\boldsymbol{Z}(\boldsymbol{L}) = \boldsymbol{2}\text{^}(\boldsymbol{8}.\boldsymbol{L}), \quad \boldsymbol{L} \in \{\boldsymbol{2},\boldsymbol{3},\boldsymbol{4},\boldsymbol{5},\boldsymbol{6},\boldsymbol{7},\boldsymbol{8}\} \qquad (\boldsymbol{7})$$

$$\boldsymbol{Z}(\boldsymbol{k},\boldsymbol{L}) = \boldsymbol{k}.\boldsymbol{2}\text{^}(\boldsymbol{8}.\boldsymbol{L}), \quad \boldsymbol{L} \in \{\boldsymbol{2},\boldsymbol{3},\boldsymbol{4},\boldsymbol{5},\boldsymbol{6},\boldsymbol{7},\boldsymbol{8}\}, \boldsymbol{k} \geq \boldsymbol{1} \qquad (\boldsymbol{8})$$

k: number of secret keys

The function Z is computed by two Equations (7) and (8) for two states, which are CCM mode and MKP protocol with CCM mode. Furthermore, the Equation (7) does not include (k) that means the number of secret keys does not influence the size of a message. However, the Equation (8) uses variable (k) that means the size of message is influenced by the number of secret keys. The MKP protocol improves the size of the transferred message in CCM security mode. Moreover, the increasing of number of secret keys improves the number of sub-messages that are transferred in the network. However, the increasing of secret keys without adopting the MKP protocol does not impact the size of the transferred message.
Since the variable L has specific range of values, then it sets as a constant value (α) as shown in Equation (9). Furthermore, the Z function is a linear function that increases depending on (k) variable (number of secret key).

$$\boldsymbol{Z}(\boldsymbol{k}) = \boldsymbol{\alpha}.\boldsymbol{k}, \qquad \boldsymbol{k} \geq \boldsymbol{1} \qquad (\boldsymbol{9})$$

Authentication time of ZigBee wireless sensor depends on the throughput of the processing unit, where the CPU throughput is the number of processed blocks per a unit of time as shown in eq. (10). Furthermore, the authentication time is computed by dividing the size of authenticated data over the CPU throughput that can be seen in eq. (11).

$$CPU\ Thr = \frac{Amount\ of\ Processed\ Data}{Time} \quad (10)$$

$$Auth\ Time = \frac{Amount\ of\ Authenticated\ Data}{CPU\ Thr} \quad (11)$$

Let T be a function to compute the authentication time for a blocks of data, where the number of blocks is (Blocks) as denoted by eq. (12). The authentication time for single block (AuthTime) is generated according to eq. (11).

$$T(Blocks) = Blocks\ .AuthTime \quad (12)$$

Blocks: number of blocks (*Blocks ≥ 1)*
AuthTime: authentication time for one block

[5] DMAC: Distributed Message Authentication Code

In the CCM security mode, the authentication operation is implemented in a sequential mode regardless of the number of processing unit. However, the DMAC algorithm exploits the available processing units to authenticate the data blocks in a parallel fashion. Furthermore, the data are divided into (n) groups as derived in eq. (1) and the number of processing units is (PrUnits).

The function T for DMAC algorithm is calculated as shown in eq. (13), where it includes two states based on the relationship between the number of blocks and the number of processing units in a system.

$$T(n, PrUnits) = \begin{cases} AuthTime & n \leq PrUnits \\ AuthTime(n\ div\ PrUnits + 1) & n > PrUnits, n\ mod\ PrUnits\ \neq 0 \\ AuthTime(n\ div\ PrUnits) & n > PrUnits, n\ mod\ PrUnits = 0 \end{cases} \quad (13)$$

AuthTime: authentication time for one group of blocks
n: number of groups
$PrUnits$: number of processing units

In a multi-processing computer system, the time of message authentication is based on the adopted authentication algorithm. The time of CBC-MAC algorithm is equivalent to a single and multi processing system since it works in a sequential fashion regardless of the number of processing units. Furthermore, the function of time (T) is increased linearly according to the incremental of message size.

## 6. Network Performance

This study emphasizes two performance metrics: throughput and data packet delivery ratio. The network perfection is measured by its performance metrics improvement. The performance metrics of the network are influenced by the network parameters. The next two subsections explain how the throughput and delivery ratio computation.

It refers to the amount of data that is sent from the source node and received by the destination node in a unit of time. The node throughput (*S*) evaluates the ability of that node to transfer data per time (see eq. 14). For the network that consists of (*n*) nodes, the network throughput is the average of throughput for all nodes (see eq.15).

$$Thr_{node} = \text{total data / simulation time (bps)} \quad (14)$$
$$Thr_{network} = (\sum Thr_i) / n \ (i = 1 \text{ to } n) \quad (15)$$

It is the percentage ratio of the sent data packet to the received data packet. The dropped packets are not taken in computation because some dropped packets are retransmitted and they are received by the destination node. It measures the delivery ratio rate to data packets only (see eq. 16).

$$\text{Del Rat} = \text{total sent packets / total received packets} \times 100\% \quad (16)$$

## 7. Methodology

The project adopted pretest-posttest design. In the first experiment, throughput and delivery ratio, are measured with normal condition (no change in simulator code). The same performance metrics is computed when data packet authentication is added to the network. The authentication is computed by the CBC-MAC with original AES algorithm. The experiment is tested using different traffic bit rate from 0.1 to 0.19 Mbps.

In the second experiment, the data packet authentication is measured using CBC-MAC with the MKP-AES algorithm. The experiment is tested with two and three secret keys. The two experiments are done by NS-2 simulator version 2.28.

### 7.1 Network Simulation

NS-2 simulator provides a batch to support IEEE 802.15.4 based connection. However, the code of IEEE 802.15.4 standard in NS-2 does not provide an optional security feature. It is impossible to expose the impact of different security features over the network. We developed a new event to the NS2. The event adds the security time to the simulation time. It checks the size of the payload and pads the last block to be 16-byte. The event chooses the accurate time needed to authenticate the payload from the already computed timetable. The developed event works as follows:

***temp = payloadSize***
***If temp mod 16 != 0 then temp = temp + (16 - temp mod 16)***
***authTime = tableTime[temp]***
***Scheduler::instance().schedule(&security_,&(security_.nullEvent),authTime)***

### 7.2 Simulation Parameters

The tested network consists of 15 nodes that are connected in star topology as illustrated in figure 1. One coordinator of type FFD device is connected to 14 nodes of type RFD. Each node has 8 flows and the traffic direction is from node toward the coordinator. The packet size is set to 70 bytes. For IEEE 802.15.4 devices the RO and SO are assigned to 3. The network frequency at the lowest level is assigned to 2.4GH. The simulator based on Constant Bit Rate (CBR) application and User Datagram Protocol (UDP). The simulation time is set to 1000 second.

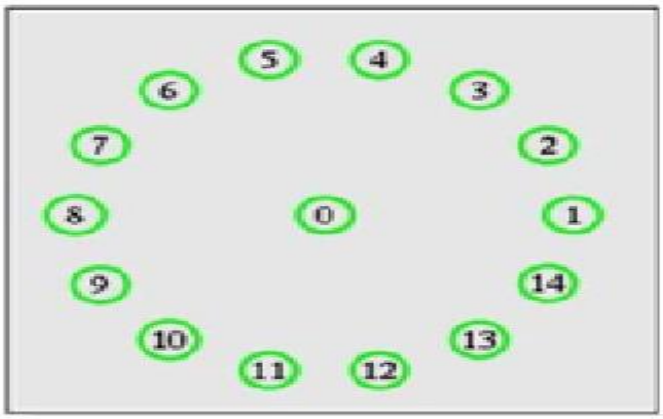


Figure 1 Star network topology (node 0 is a coordinator, nodes (1-14) are end devices)

## 8. Result

In the next paragraphs the computed throughput and data packet delivery ratio results are explained. Network throughput and data packet delivery ratio for ZigBee network are computed for four states: no security, CBC-MAC/AES data authentication algorithm, CBC-MAC/MKP-AES (2-secret key) data authentication algorithm, and CBC-MAC/MKP-AES (3-secret key) data authentication algorithm. The result from four states would show the impact of data authentication on network performance and how much the network performance could be improved when MKP-AES algorithm is used.

The throughput is degraded as shown in figure 2, which refers to the impact of authentication over the network throughput. This experiment has been conducted in two cases no security state and MIC 128 state. In the first case (No security) the results show that the throughput measurement for a range of traffic bit rate. Moreover, the data bit rate generator is started from 0.1 Kbps and it produced throughput 435 bps. The incremental of bit rate generator led to improve the network throughput regularly.

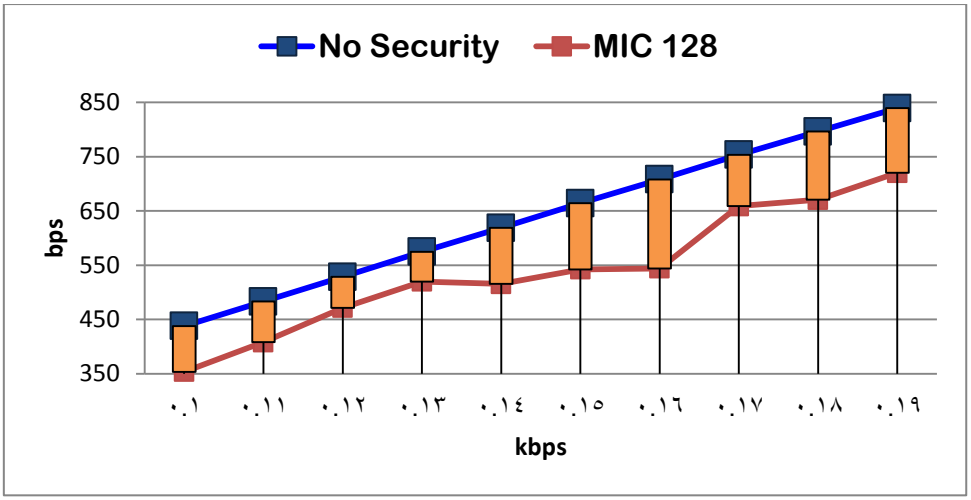


Figure 2 Authentication impact over Throughput

However, in the second case (MIC 128) the impact of data packet authentication over throughput is calculated. For the bit rate 0.1, the network throughput is 350 bps, which means the data authentication operation leads to decrease the network throughput. The increasing of bit rate generator improved the network throughput irregularly because in some cases the number of dropped packet is increased.

The authentication operation consumes time during packet authentication. It is shown that the message authentication leads to reduce the network throughput. The average of percentage ratio for throughput degradation is 15.58%.

The throughput is measured for network that uses MKP-AES algorithm in CBC-MAC authentication for the data packet improvement. The network throughput improvement is shown in figure 3 and figure 4, which are representing MKP-AES algorithm with two secret keys and three secret keys respectively.

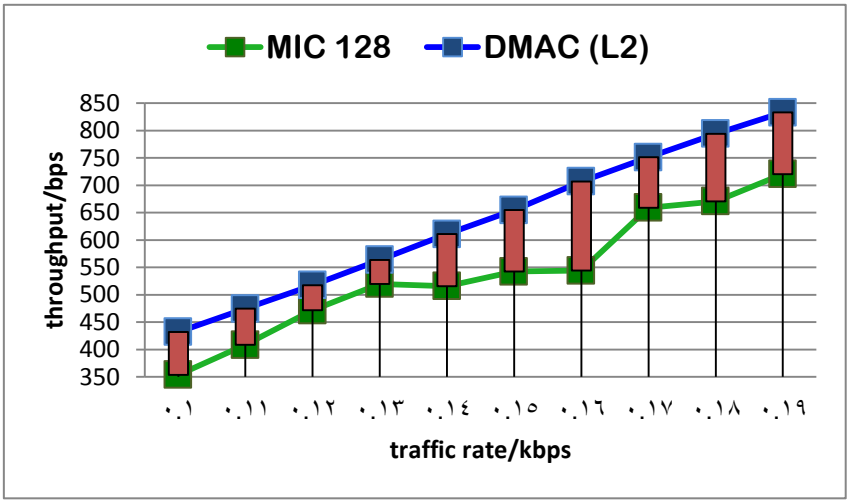


Figure 3 Comparison of authentication influence on network throughput for Advanced Encryption Standard (AES) and Multiple Key Protocol-Advanced Encryption Standard (MKP-AES) (two secret keys) algorithms

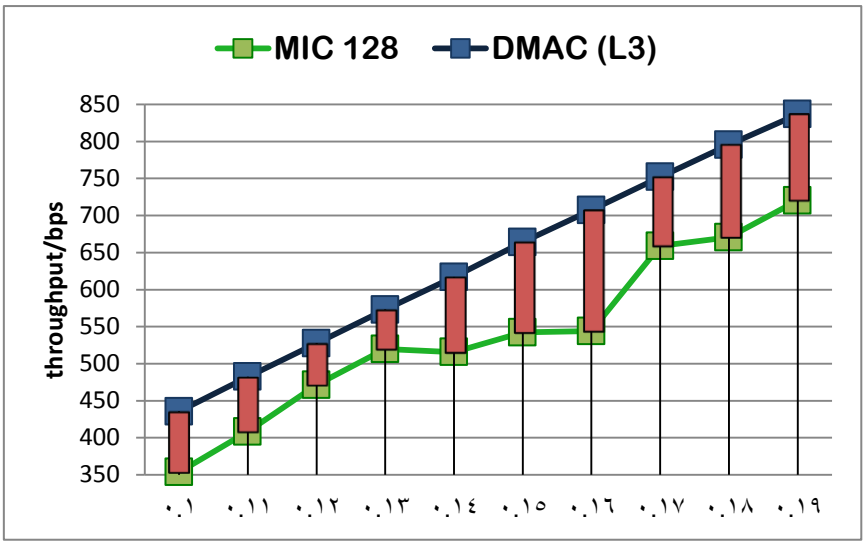


Figure 4 Comparison of authentication influence on network throughput for Advanced Encryption Standard AES and Multiple Key Protocol-Advanced Encryption Standard MKP-AES (three secret keys) algorithms

In figure 3, it is clear that the data bit rate incremental leads to improve the network throughput regularly for the network that utilizes MKP-AES algorithm (2-secret keys) with CBC-MAC algorithm for data authentication. The network throughput becomes close to the (no security) state. In the other side, the using of old authentication algorithm (CBC-MAC/AES) leads to irregularly throughput improvement because of dropping more packets.

In figure 4 the results of network throughput measurement that belong to two network systems are shown. Both of them are used CBC-MAC algorithm for data authentication, but first network utilized the original AES algorithm and the second system utilized the MKP-AES algorithm with three secret keys. The data bit rate incremental leads to improve the network throughput regularly when MKP-AES algorithm is used and irregularly when the original AES algorithm is used. The network throughput is higher when the MKP-AES (3-secret key) is used instead of original AES algorithm.

The throughput has been improved over original authentication when the MKP-AES is used. The average for percentage ratio improvement of the tests is about 17.36% for two secret keys where by in the three secret keys state, it is about 18.45%.

Figure 5 shows the impact of data authentication (MIC 128) over the ratio of data packet delivery. Moreover, the data bit rate generator is started from 0.1 Kbps and most of packets are delivered. However, in the second case when data packets are authenticated by CBC-MAC/AES algorithm the data packet delivery ratio is degraded irregularly. For the bit rate 0.1 the data packet delivery ratio is 80%. When the bit rate is increased the data packet delivery ratio is not stable because the number of dropped packet is not stable. It is clear that the authenticating of data packets produces degradation in the ratio of data packet delivery. The average of degradation of data packet delivery ratio is 15.5%.

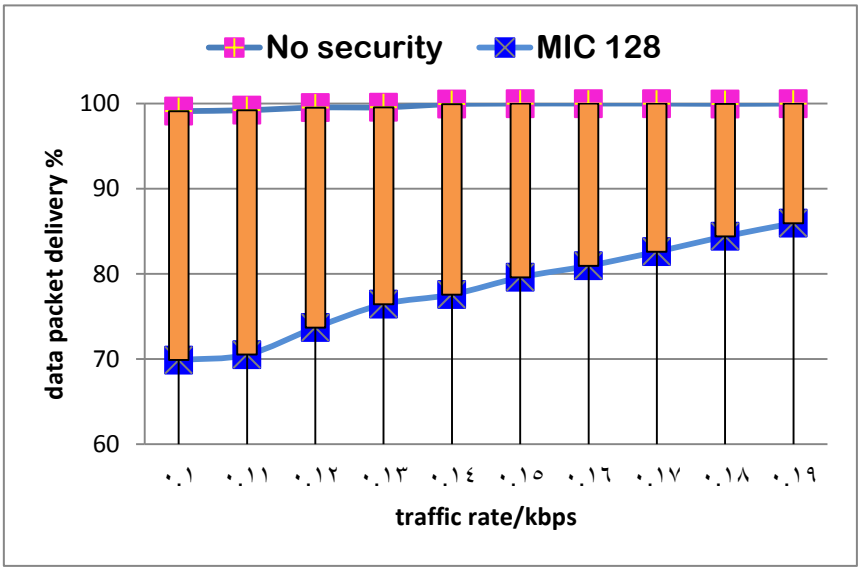


Figure 5 Data packet delivery ratio degradation by Message Integrity Code MIC 128

However, by using MKP-AES algorithm with CBC-MAC algorithm, it leads to reduce the degradation of the data packet delivery ratio. Figure 6 shows the reduction of the degradation in the data packet delivery ratio when MKP-AES algorithm (2-secret key) is used for CBC-MAC authentication. The data bit rate incremental leads to improve the data packet delivery ratio regularly for the network that utilizes MKP-AES algorithm (2-secret keys) with CBC-MAC algorithm for data authentication. The data packet delivery ratio becomes close to 100%. In the other side, the using of old authentication algorithm (CBC-MAC/AES) leads to irregularly data packet delivery ratio because of dropping packets. The average data packet delivery ratio is 98.77% that indicates the average of degradation is 1.23%.

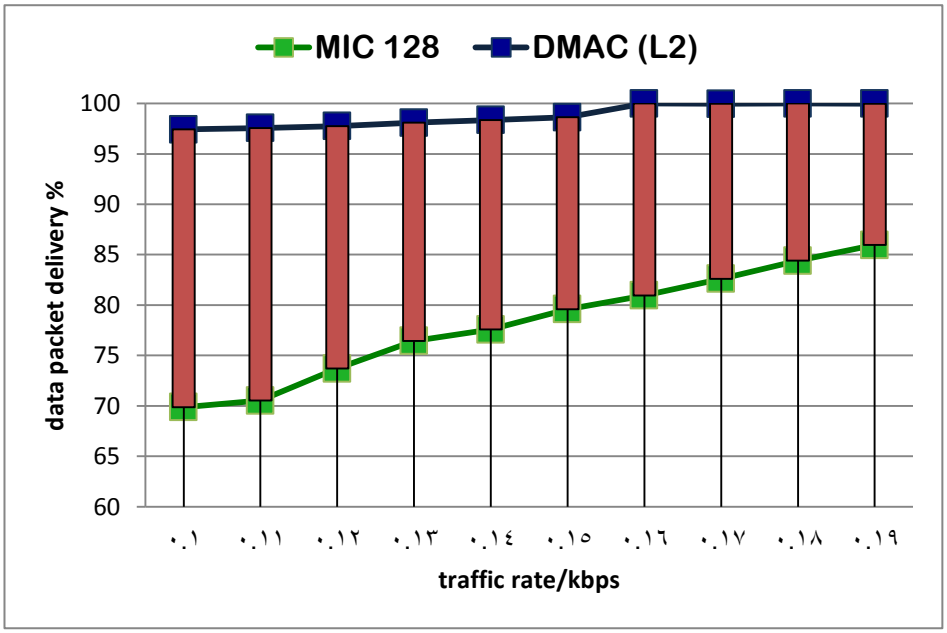


Figure 6 Data packet delivery ratio improvement by Multiple Key Protocol-Advanced Encryption Standard MKP-AES (2 secret keys) algorithm over Advanced Encryption Standard AES algorithm

In figure 7, the data packet delivery ratio with MKP-AES (3 secret keys) algorithm based is evaluated and compared to the original AES algorithm based. The data bit rate incremental leads to improve the data packet delivery ratio regularly when MKP-AES algorithm is used and irregularly when original AES algorithm is used. The data packet delivery ratio is very close to

100% when the MKP-AES (3-secret key) is used instead of original AES. The average data packet delivery ratio becomes 99.58%, which means the average of degradation is 0.42%.

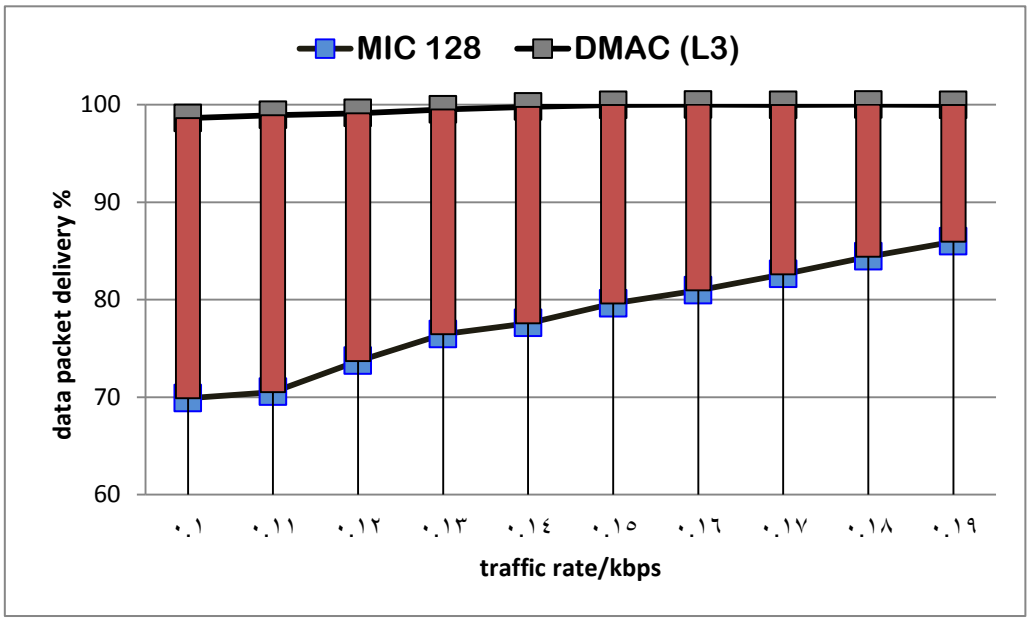


Figure 7 Data packet delivery ratio improvement by Multiple Key Protocol-Advanced Encryption Standard MKP-AES (3 secret keys) algorithm over Advanced Encryption Standard AES algorithm

## 9. Discussion

To compare the result of this study with previous studies, it is clear that the current study is unique that develops a protocol to improve the network performance via message authentication algorithm. However, none of previous studies (Zheng and Lee, 2004) (Di Francesco et al. (2011) (Charfi and Bouyahi, 2012) (Daidone et al., 2011) has developed an authentication algorithm to improve the network performance.

The time of data packet authentication leads to reduce the node throughput because it is added to the simulation time. Each data packet should be authenticated first then it is sent to the physical layer. When the data packet is received, the receiver should match the authentication code with the received packet. The code from the received packet must be equivalent to the authentication code received from the sender. The authentication time in both sides (sender and receiver) is measured in them. Normally, throughput for node is computed as shown in eq. 18.

However, when the data packet is authenticated, the authentication time is considered in the simulation and the equation for throughput is denoted in eq. 17.

$$Thr_{node} = \text{total data} / (\text{simulation time} + \text{authentication time}) \qquad (17)$$

Based on the number of secret keys and number of computing elements, the MKP-AES algorithm reduces the authentication time. It improves the node throughput and network throughput. The project tested two statuses, two secret keys with 2-core CPU and three secret keys with 3-core CPU. In both states, the authentication time is reduced and the network throughput is improved.

Furthermore, the data packet delivery ratio is reduced by adding data packet authentication to the network. The sender node sends the data packet to the receiver at a specific time so that the receiver node expects to receive the data packet at that particular time.

The authentication operation delays the packet for authentication that causes the time out for the packet at receiver side. The authentication time will reduce the number of received data packet. The delivery ratio of data packet is reduced when the data packet is authenticated. When CBC-MAC utilizes MKP-AES algorithm for authentication, the authentication time is improved and that will increase the number of delivered data packet in the receiver side.

## 10. Conclusion

The network performance degradation happens when the transferred messages are secured. For ZigBee network, the network throughput and data packet delivery ratio are reduced when the data packet is authentication by CBC-MAC algorithm. The research work simulated using NS-2 simulator to show the impact of data authentication over the network performance.

The network improvement is done when MKP-AES is imposed as a block cipher algorithm over the original AES algorithm with CBC-MAC, for data packet authentication. The results show that the network throughput and the data packet delivery ratio are improved. In future work, we will test the impact of MKP-AES algorithm over performance of the ZigBee network with beacon packet authentication and command packet authentication.

## 11. References


Al-alak, S., Z. Zukranain, A. Abdullah and S. Subramiam, 2011. AES and ECC Mixed for ZigBee Wireless Sensor Security. In: International Conference of Sensor Networks, Information and Ubiquitous Computing (ICSNIUC'11), Singapore September 2011, pp 535-539.

Al-Alak, S., Z. Zukranian, A. Abdullah and S. Subramiam, 2012a. Authentication time of IEEE 802.15.4 with multiple-key protocol using distributed message authentication code algorithm. Res. J. Inform. Technol., 4: 140-154.

Al-Alak, S., Z. Zukranain, A. Abdullah and S. Subramiam, 2012b. Randomness improvement of AES using MKP. Res. J. Inform. Technol., (In Press).

Bernstein, D.J., 2005. Understanding brute force. Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, May 22-26, 2005, Aarhus, Denmark, pp: 1-10.

Biryukov, A. and D. Khovratovich, 2009. Related-key cryptanalysis of the full AES-192 and AES-256. Proceedings of the 15th International Conference on the Theory and Application of Cryptology and Information Security, December 6-10, 2009, Tokyo, Japan, pp: 1-18.

Bogdanov, A., D. Khovratovich and C. Rechberger, 2011. Biclique cryptanalysis of the full AES. Proceeding of the 17th International Conference on the Theory and Application of Cryptology and Information Security, December 4-8, 2011, Seoul, South Korea, pp: 344-371.

Charfi, F. and M. Bouyahi, 2012. Performance evaluation of beacon enabled IEEE 802.15.4 under Ns2. Int. J. Distrib. Parallel Syst., 3: 67-79.

Cid, C. and G. Leurent, 2005. An analysis of the XSL algorithm. Proceedings of the 11th International Conference on the Theory and Application of Cryptology and Information Security, December 4-8, 2005, Chennai, India, pp: 333-352.

Daidone, R., G. Dini and M. Tiloca, 2011. On experimentally evaluating the impact of security on IEEE 802.15.4 networks. Proceedings of the International Conference on Distributed Computing in Sensor Systems and Workshops, June 27-29, 2011, Barcelona.

Di Francesco, M., G. Anastasi, M. Conti, S.K. Das and V. Neri, 2011. Reliability and energy-efficiency in IEEE 802.15.4/zigbee sensor networks an adaptive and cross-layer approach. IEEE J. Selected Areas Commun., 29: 1508-1524.

Gilbert, H. and T. Peyrin, 2010. Super-Sbox cryptanalysis: Improved attacks for AES-like permutations. Proceedings of the 17th International Workshop on Fast Software Encryption, February 7-10, 2010, Seoul, Korea, pp: 365-383.

Housley, R., S. Drive, D. Whiting and N. Ferguson, 2003. Submission to NIST: Counter with CBC-MAC (CCM) AES mode of operation. http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/ccm/ccm.pdf

LAN/MAN Standards Committee, 2006. IEEE Standard for Information technology-Telecommunications and information exchange between systems- Local and metropolitan area networks- Specific requirements--Part 15.4: Wireless MAC and PHY Specifications for Low-Rate WPANs. IEEE. Available from: http://profsite.um.ac.ir/~hyaghmae/ACN/WSNMAC1.pdf

Woon, W. and T. Wan, 2008. Performance evaluation of IEEE 802.15.4 wireless multi-hop networks: simulation and testbed approach. Int. J. Ad Hoc Ubiquitous Comput., 3: 57-66.

NIST, 2001. 197: Announcing the advanced encryption standard (AES). Information Technology Laboratory, National Institute of Standards and Technology, Nov. Available from: Avcsrc.nist.gov/publications/fips/fips197/fips-197.pdf [Accessed 1 January 2012]

Rijmen, V., 2009. Practical-titled attack on AES-128 using chosen-text relations. Proceedings of the 15th International Conference on the Theory and Application of Cryptology and Information Security, December 6-10, 2009, Tokyo, Japan.

Rogaway, P., 2011. Evaluation of Some Blockcipher Modes of Operation. Available from: http://www.cryptrec.go.jp/estimation/techrep_id2012_2.pdf

Zheng, J. and M. Lee, 2004. A comprehensive performance study of IEEE 802.15.4. Sensor Network Operat., 4: 1202-1206.