Applied Mathematics, 2014, 5, 581-592
Published Online March 2014 in SciRes. http://www.scirp.org/journal/am
http://dx.doi.org/10.4236/am.2014.54055

Scientific Research

Increase Throughput of CCM Security Mode Using MKP

Zuriati Ahmad Zukarnain

Department of Communication Technology and Networks, Faculty of Computer Science and Information Technology, University Putra Malaysia, Serdang, Selangor, Malaysia
Email: zuriati@fsktm.upm.edu.my

Received 25 September 2013; revised 25 October 2013; accepted 5 November 2013

Abstract

A security protocol is used to ensure the messages transferred over a network free from attack. The CCM security operation mode adopts CTR, and CBC-MAC schemes to implement message ciphering and authentication respectively. There are two limitations of CCM including the secret key which should refresh for each 2^{64} blocks in CTR mode and the authentication operation of CBC-MAC works in sequential fashion on multi-processor system. In this paper, we have proved that the Multiple Key Protocol (MKP) can be used within CTR mode to increase the size of the transferred message. Also, we have showed that the DMAC algorithm is able to decrease the time of authentication operation on multi-processing units system.

Keywords

CBC-MAC; CCM; CTR; Message Authentication Throughput; Parallel Message Authentication

1. Introduction

Recently, there are many security systems adopting the CCM [1] security operation mode to ensure that the messages are securely transmitted over the wireless networks. The Wireless Sensor Network (WSN) systems are utilizing the CCM mode in providing the security requirements. The Wi-Fi (802.11) [2], LR-WPAN, and IP-Sec's are using the CCM mode for ciphering and authenticating the data [3]. The IEEE 802.15.4 [4], which is the basic for many wireless applications, such as, Zig Bee technology [5], 6LoWPAN and WirelessHart, is adopting the CCM security mode to provide its security suite. The IEEE 802.15.4 security suite includes the message authentication, message encryption, replay resistance, and freshness.

The CCM security operation mode includes Counter (CTR) [6] mode for encryption and Cipher Block Chaining Message Authentication Code (CBC-MAC) [7] for authentication. The size of the message in CCM

How to cite this paper: Zukarnain, Z.A. (2014) Increase Throughput of CCM Security Mode Using MKP. *Applied Mathematics*, **5**, 581-592. http://dx.doi.org/10.4236/am.2014.54055

mode is limited to the number of keys [3]. The authentication operation is conducted in sequential fashion, which means that the increasing of processing units in a system does not improve the authentication operation.

The hypotheses of the research are the Multiple Key Protocol (MKP) [8] increasing the size of the transmitted message over the network, and the DMAC [9] algorithm decreasing the authentication operation time. The message size and authentication time improvement lead to increase of the maximum authentication throughput.

The original CCM security operations mode has two major disadvantages. The first one is that CTR, which is used for message encryption in CCM, can only be used for transmitting a short message. The other one is that CBC-MAC in CCM, which is used for authentication, is performed in a sequential way and thus it could not take full use of a multi-processor system. The main idea of the paper is parallel encryption. More precisely, the message is divided into several sub-messages, and each sub-message is further encrypted with its own secret key independently. Thus, a long message can be processed in parallel and a multi-processor system can deal with it efficiently.

It seems that if we divide the message into several pieces and then apply the original CCM method to each piece independently, we may obtain the same advantages as those of the proposition. The difference between the original scheme and the proposed scheme is that in the original scheme, the secret keys must be transferred through the network; however, in the proposed scheme, the secret keys are generated by MKP protocol.

2. CCM Security Operation Mode

The CCM security mode implements two security operations that are consist of the message ciphering and the message authentication. This security mode adopts two security schemes such as the Counter (CTR) to conduct the ciphering operation and the Cipher Block Chaining Message Authentication Code (CBC-MAC) to conduct the authentication operation. Besides, the CBC-MAC is a security operation mode that authenticates a message within the wireless media. The CTR mode is a security operation mode that is used for message encryption. Both of CBC-MAC and the CTR are integrated in building a CCM mode as illustrated in **Figure 1**.

There are two important parameters that used in implementing the CCM mode of security in the wireless network. The two parameters are L and M, where L is belong to {2, 3, 4, 5, 6, 7, 8} and M is belong to {4, 6, 8, 10, 12, 14, 16}. The parameter of L is stand for the size of a message length field (message size (l (m)) is $< 2^{8*L}$ bytes). The parameter of M is stand for the size of a message authentication code (tag length = (8*M) bits). The parameter L is coded as $L - 1$, and the M is coded as $(M - 2)/2$.

2.1. Message Ciphering

The CTR mode is used for the ciphering operation of CCM mode. The CCM adopts one of the block cipher methods for data ciphering such as Data Encryption Standard (DES) [10], International Data Encryption Algorithm (IDEA) [11], RC5 [12], blowfish [13], and AES [14]. Most of the network security systems employ the AES

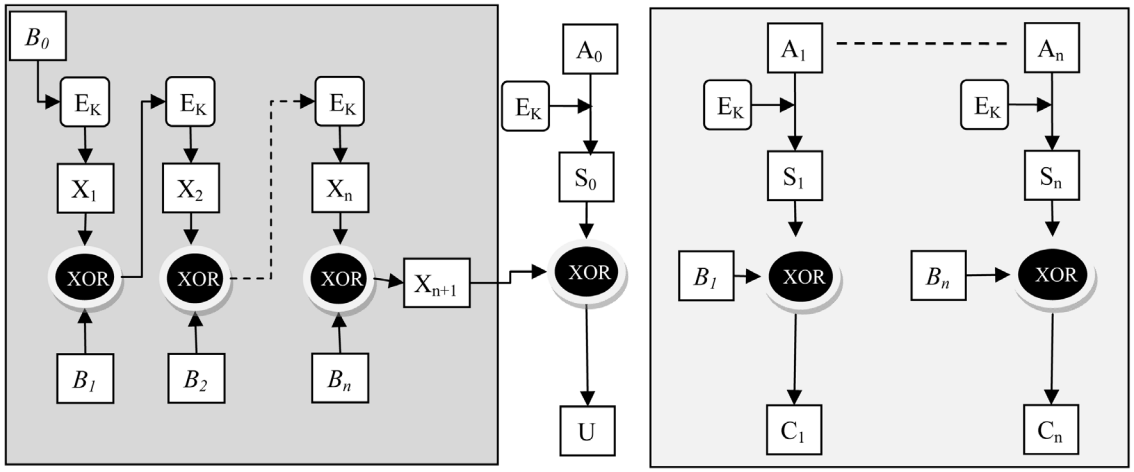

Figure 1. Message encryption and authentication in CCM. (a) CBC-MAC; (b) CTR.

block cipher algorithm for data ciphering because the block cipher algorithm is faster than the stream cipher algorithm for encrypting and decrypting operations [15]. A sequence of blocks consists of $S_1\ S_2 \cdots S_i$ is generated by Equation (1). The block of (A_i) is structured as illustrated in **Figure 2(a)**. In more details, in the flag octet, the bits from 0 to 2 are assigned for L and the bits from 3 to 7 are set to zero as illustrated in **Figure 2(b)**. The (A_i) block is distinct from (B_0) block in the message authentication operation because the values of three bits that are assigned to M is non-zero values in (B_0) block, but the same bits are set to zero in (A_i) block. The octets from 16 – L to 15 are assigned to the counter and the octets from 1 to 15 – L are assigned to nonce. The second step breaks the message into equivalent size blocks ($B_1\ B_2 \cdots B_h$). The blocks of the message are XOR with the blocks ($S_1\ S_2 \cdots S_h$) to produce the ciphertext blocks as computed in Equation (2).

$$S_i = E_k\left(A_i\right),\ i \geq 0 \tag{1}$$

S_i: computed block
E: block cipher encryption method
k: symmetric key

$$C_i = B_i \oplus S_i,\ i \in [1, h] \tag{2}$$

B_i: plaintext block
h: number of plaintext blocks
S_i: block computed from A_i

2.2. Message Authentication

The CBC-MAC algorithm implements the authentication operation for the CCM mode. The blocks B_1, $B_2 \cdots B_h$ are generated from breaking the message into h equal size blocks, which are the input to CBC-MAC. The last block may be padded with zeros. The block B_0 is structured as illustrated in **Figure 3(a)**. Furthermore, the first octet is assigned for the flags, where (15 – L) octets are assigned for nonce, and (L) octets are assigned to store the message length (l (m)). The octet of flags includes (L) and (M) parameters in the first six bits (each parame-

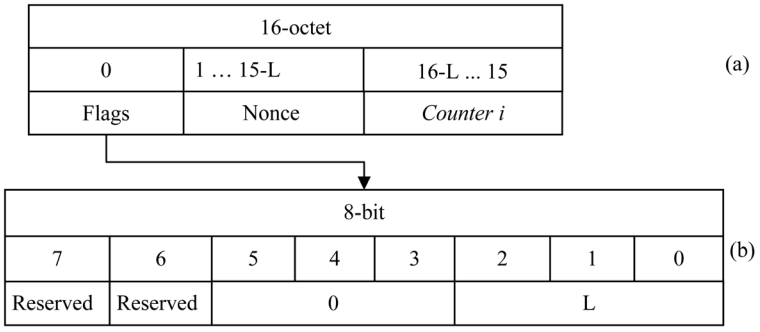

Figure 2. Format of block (A_i) and flags of encryption in CCM [1]. (a) Block format; (b) Flags.

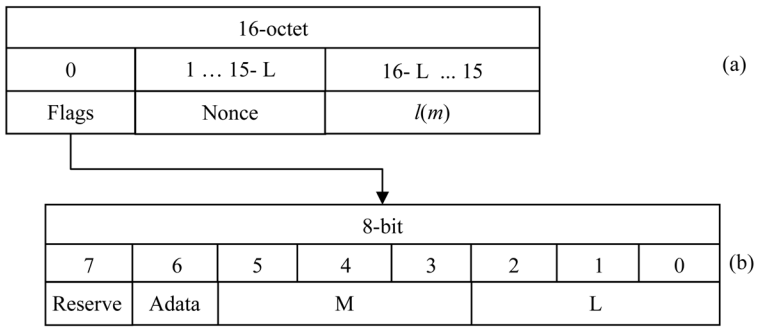

Figure 3. Format of block (B_0) and flags of authentication in CCM [1]. (a) Block format; (b) Flags.

ter coded in three bits). The Adata bit is set to one when the additional authentication data is enabled else it is set to zero, and the last bit is reserved for future use that is set to zero as illustrated in **Figure 3(b)**.

Moreover, the temporary block of X_1 is computed by ciphering the block B_0 as computed in Equation (3). The authentication code for a message is computed by the Equation (4) and Equation (5). Furthermore, each block of B_i is a plaintext that can be XOR with its partner temporary block X_i, and then the output is ciphered with a block cipher algorithm to produce a new temporary block X_{i+1}, until it receives the last block X_{h+1}. The authentication code length could be 128, 64, and 32-bit that is generated from X_{h+1}.

$$X_1 = E_k\left(B_0\right) \tag{3}$$

E: block cipher algorithm
k: secret key

$$X_{i+1} = E_k\left(X_i \oplus B_i\right),\ i \in [1,h] \tag{4}$$

B_i: plaintext block
h: number of plaintext block

$$\text{Tag} = \text{First\_}M_{\text{Bytes}}\left(X_{h+1}\right) \tag{5}$$

3. Multiple Key Protocol

The Multiple Key Protocol (MKP) is used within the block cipher algorithm. It provides multiple secret keys for a block cipher algorithm. It adopts a public key algorithm to generate a sequence of secret keys such as Elliptic Curve Cryptosystem (ECC) [16]-[18]. The MKP adopts the ECC public key system to generate the secret keys since it requires for the less computational power, memory, and communication bandwidth compared to the other traditional public key crypto-algorithms [19]. The security feature of the node does not enquire to exchange all the secret keys with other connected nodes. With regards to thus, only two initial parameters may be exchanged. Moreover, MKP may generate a sequence of secret keys according to the desired level of security. The two communicated nodes agree about the number of secret keys. The key generation operation computes two lists of parameters including ($r_1\ r_2 \cdots r_n$ and $k_1\ k_2 \cdots k_n$) from the two initial parameters (r_0 and k_0) as shown in Equation (6) and Equation (7). Each secret key (K_i) is produced by XOR two parameters, as computed in Equation (8), and it is illustrated in **Figure 4**.

$$r_i = PKC\left(r_{i-1}\right),\ i \in [1,n] \tag{6}$$

PKC: public key ciphering
n: number of secret keys

$$k_i = PKC\left(k_{i-1}\right),\ i \in [1,n] \tag{7}$$

$$K_i = k_i \oplus r_{n-i+1},\ i \in [1,n] \tag{8}$$

The MKP protocol breaks the message into n groups of blocks as pointed in Equations (9) and (10). The number of groups must equivalent to the number of secret keys. The last the group is not necessary to be equal to the other groups. The secret keys are assigned to the created groups of blocks as illustrated in **Figure 5**. The key that belongs to the created group is used for ciphering and authenticating the blocks of that group.

$$\text{Message} = \sum G_i,\ i \in [1,n] \tag{9}$$

$$G_i = \sum B_j,\ j \in [1,m] \tag{10}$$

G: group (set) of blocks
B: block of bytes
n: number of groups
m: number of blocks per group

In the encryption operation, the CCM uses CTR mode in which the counter value starts from zero and incremented by one for each block of the message. The maximum message size should not exceeding 2^{61} blocks in IEEE 802.15.4 standard, which adopts CCM security mode [20]. However, the MKP protocol breaks the mes-

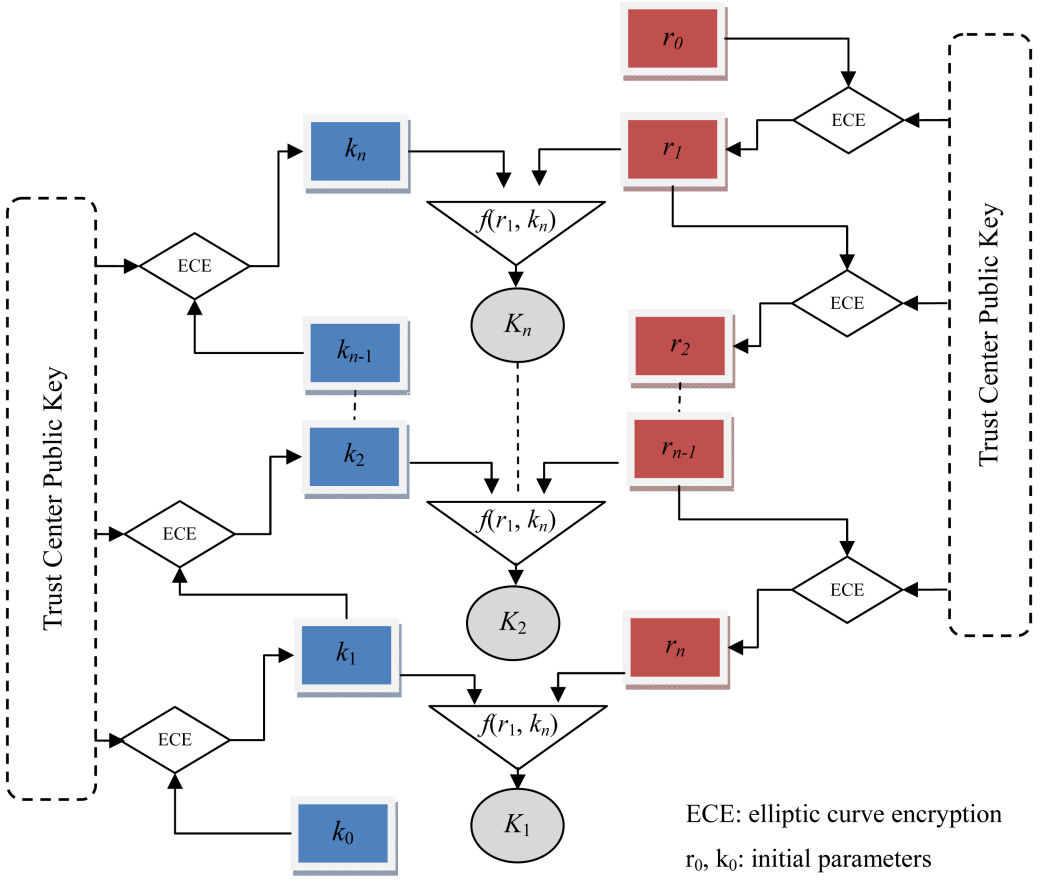

Figure 4. Secret keys generation for MKP.

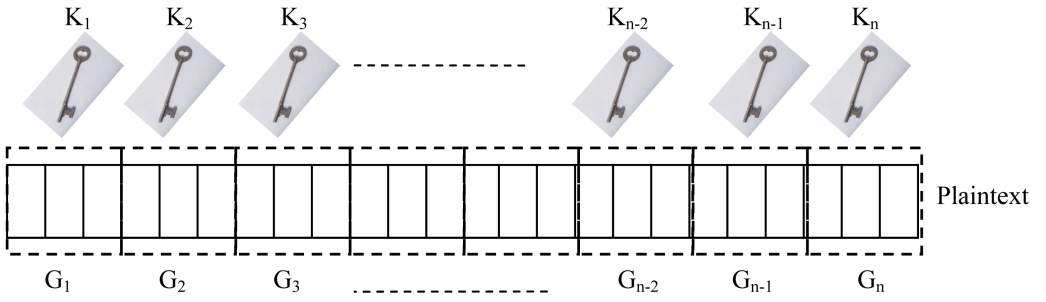

Figure 5. Keys assignment over plaintext groups.

sage into *n* groups, where each group has its own counter that equivalent to the CTR counter. The MKP protocol improves the message size about *n* times. Each group is ciphered by its own secret key.

The level of security refers to the number of secret keys and to the number of sub-messages in the encryption system. In addition, the number of secret keys should equal to the number of sub-messages.

4. Distributed Message Authentication Code

The CCM mode derives the authentication operation by CBC-MAC algorithm that is work in a sequential fashion. Furthermore, the increasing of the processing units in the system does not improve the authentication operation performance [3]. However, the Distributed Message Authentication Code (DMAC) algorithm authenticates a message in a parallel fashion. The DMAC improves the authentication time of a message on a multiple processing units system. It distributes the groups of blocks over computing devices. The number of group is

computed using MKP protocol, where the number of groups must be equivalent to the number of secret keys.

Moreover, the MKP provides multiple secret keys, where each key is used to authenticate a single group of blocks (sub-message), and the output is XOR with authentication code of other group. MKP supports the possibility of parallel authentication computations of groups. Since all the blocks of a group are authenticated in a sequential fashion, then the authentication code of an individual group is computed in a single processing unit. The authentication of each sub-message is computed independently, and the outputs from sub-messages authentication are XOR in producing the message authentication code as illustrated in **Figure 6**.

The DMAC algorithm improves the authentication operation performance according to the number of processing units and the number of sub-messages. The DMAC algorithm distributes the groups over available devices as following:

1. ***If number_of_groups ≤ number_of_devices then***
2. ***Assign each device_i to group_i*** (where i = 1 to number_of_groups)
3. ***Else For each group_i***
4. ***If i ≤ number_of_device then***
5. ***Assign device_i to group_i***
6. ***Else assign device_j to group_i*** (where $j = i$ - number_of_devices)

The DMAC algorithm assigns the groups to a number processing units according to their sequence. When the system has enough free processing units, it assigns each unit to its partner group; otherwise, the groups are assigned in a circular fashion when the number of processing units is less than the number of groups. Two states in the number of processing units (p) versus the number of groups (n) are considered. The first state ($p \geq n$), in which the number of units is greater than or equal to the number of groups. The second state ($m < n$) shows which the number of units is less than the number of groups.

5. Mathematical Measurement

This section provides the mathematical model that increases the message size, decreases the authentication operation performance, and increases the authentication operation throughput.

5.1. Message Size

Let Z be a function to compute the size of a message as pointed in Equation (11). For MKP protocol, the number of secret keys is greater than 1, so that the function Z is computed in the new equation. When the MKP protocol is involved in CCM mode, the function Z has two variables, which are the number of secret keys (k) and the pa-

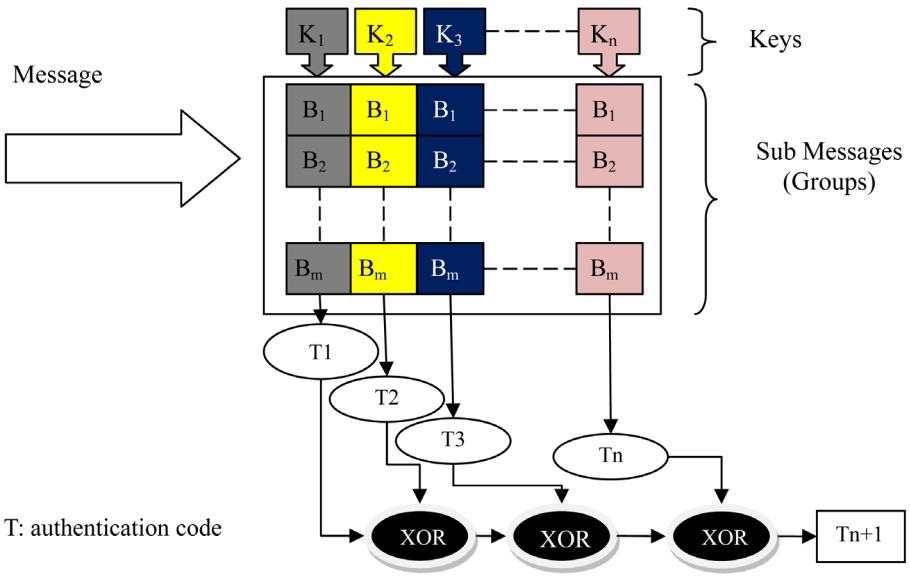

Figure 6. DMAC message authentication.

rameter L as shown in Equation (12).

$$Z(L) = 2^{(8 \cdot L)},\ L \in \{2,3,4,5,6,7,8\} \tag{11}$$

$$Z(k,L) = k \cdot 2 \wedge (8 \cdot L),\ L \in \{2,3,4,5,6,7,8\}, k \geq 1 \tag{12}$$

k: number of secret keys

Furthermore, the Equation (11) does not include the (k) value that indicates the number of secret keys computed in the function Z may not influence the size of a message. However, the Equation (12) uses a variable (k) to compute the function Z that means the size of a 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 the number of secret keys increases the number of sub-messages that are transferred in the network.

Since the variable L has a specific range of values, then the term (2^{8L}) is set as a constant value (α) as shown in Equation (13). Furthermore, the function Z is a linear function that is based on (k) variable (number of secret key).

$$Z(k) = \alpha \cdot k,\quad k \geq 1 \tag{13}$$

5.2. Authentication Operation Time

The authentication time is based 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 Equation (14). Furthermore, the authentication time is computed by dividing the size of authenticated data over the CPU throughput that can be seen in Equation (15).

$$\text{CPU Throughput} = \frac{\text{Size of Processed Data}}{\text{Total Time}} \tag{14}$$

$$\text{Authentication Time} = \frac{\text{Size of Authenticated Data}}{\text{CPU Throughput}} \tag{15}$$

Let T be a function to compute the authentication time for a blocks of data, where the number of blocks is (h) as denoted by Equation (16). The authentication time for single block (H) is generated according to Equation (15).

$$T(h) = h \cdot H \tag{16}$$

h: number of blocks ($h \geq 1$)

H: authentication time for one block

In the CCM security mode, the authentication operation is implemented in a sequential mode regardless the number of processing unit. However, the DMAC algorithm exploits the available processing units to authenticate the data blocks in a parallel fashion as illustrated in **Figure 7**. Furthermore, the data are divided into (n) groups as derived in Equation (9) and the number of processing units is (P).

The function T for DMAC algorithm is calculated as shown in Equation (17), where it includes two states

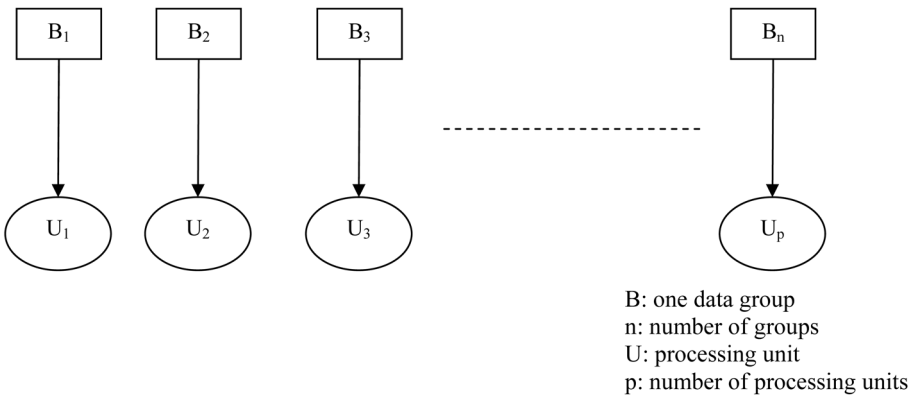

Figure 7. Parallel data processing of DMAC algorithm.

based on the relationship between the number of blocks and the number of processing units in a system.

$$T(n,p)=\begin{cases} H & n \le p \\ H\cdot(n\ div\ p+1) & n>p, n\ mod\ p\neq 0 \\ H\cdot(n\ div\ p) & n>p, n\ mod\ p=0 \end{cases} \quad (17)$$

H: authentication time for one group of data
n: number of groups
p: number of processing units

5.3. Authentication Throughput

The message authentication throughput is the number of authenticated blocks per unit of time as derived in Equation (18). The maximum authentication throughput of CBC-MAC is the maximum message size per unit of time. The maximum size of the message in CCM mode is generated by the value of parameter L as shown in Equation (11). The maximum value for parameter L is 8. The authentication time is based on the CPU throughput as concluded from Equations (14)-(16).

$$TH=\frac{Z}{T} \quad (18)$$

TH: authentication throughput
Z: message size
T: time

By doing the Equations of (11), (16), and (18), then the maximum authentication throughput of CBC-MAC algorithm is computed as pointed in Equation (19).

$$TH_{\max}=\frac{\alpha}{h\cdot H} \quad (19)$$

α: maximum message size
H: authentication time for one block
h: number of blocks

The maximum authentication throughput of DMAC algorithm is calculated by substitute Equations (13), and (17) into (18) as shown in Equation (20). Since the number of secret keys equivalents to the number of submessage, then the DMAC authentication throughput equation include only the number of secret keys variable. Then the Equation (20) is simplified into Equation (21).

$$TH_{\max}(k,p)=\frac{\alpha\cdot k}{\begin{cases} H & k\le p \\ H\cdot(k\ div\ p+1) & k>p, k\ mod\ p\neq 0 \\ H\cdot(k\ div\ p) & k>p, k\ mod\ p=0 \end{cases}} \quad (20)$$

k: number of secret keys
p: number of processing units

$$TH_{\max}(k,p)=\begin{cases} \beta\cdot k & k\le p \\ \beta\cdot k/(kdivp+1) & k>p, k\ mod\ p\neq 0 \\ \beta\cdot p & k>p, k\ mod\ p=0 \end{cases} \quad (21)$$

β: α/H

6. Result and Discussion

This section discusses the results of the implementation of the mathematical models in measure the size of message, authentication operation time, and maximum authentication throughput. The two considered systems are the CCM, and the MKP-CCM modes.

6.1. Message Size

The message ciphering in CTR mode is based on the counter value, where the maximum value of the counter is 2^{8L}. Each counter value computes one block of cipher text. The value of the counter is set to zero in CTR mode of the network security system, and the secret key of the encryption operation should be refreshed to ensure that distinct cipher blocks are generated [21].

The message size per single secret key in the CCM security mode is computed in Equation (11). The incremental of the value of parameter L increases the size of the transferred message in CCM mode. However, the increasing of level of security does not increase the size of the message as illustrated in **Figure 8(a)**.

However, the MKP protocol increase the size of the transferred message, since the level of security is increased, and the value of parameter L is increased. For any value of parameter L, the incremental of the level of security duplicated the size of the transferred message from previous level of security as illustrated in **Figure 8(b)**.

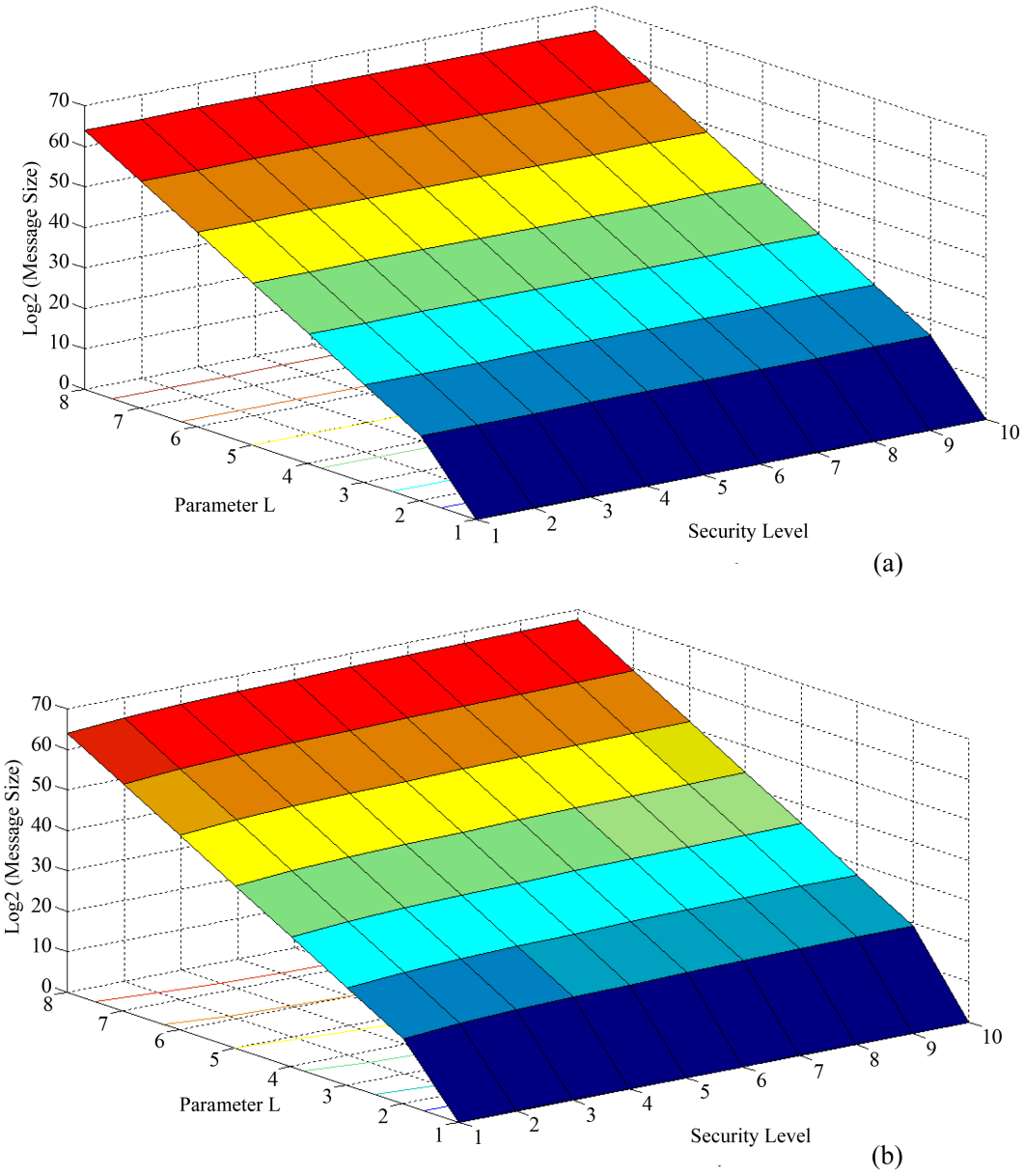

Figure 8. Size of transferred message in CCM mode. (a) Without MKP; (b) With MKP.

6.2. Message Authentication Time

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 for 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 as illustrated in **Figure 9(a)**.

However, the time of DMAC algorithm is based on the number of processing units. Furthermore, the DMAC algorithm works in parallel by exploiting the available processing units to authenticate each sub-message independently. In the single processing unit system, the time is increased linearly according to the message size. In two processing unit system, the time is decreased 50% when the number of sub-messages is even, and it is decreased about 34%, 40%, 43%, and 45% when the number of sub-messages is 3, 5, 7, and 9 respectively. In three processing units system, the time is decreased about 50%, 67%, 60%, 57%, and 63% when the number of sub-messages is (2 and 4), (3, 6 and 9), (5 and 10), 7, and 8 respectively. In four processing units system, the time is decreased about 50%, 67%, 75%, 60%, 71%, and 70% when the number of sub-messages is 2, (3, 6 and 9), (4 and 8), 5, 7, and 10 respectively. In five processing system, the time is decreased about 50%, 67%, 75%, 80%, 71%, and 78% when the number of sub-messages is 2, (3 and 6), (4 and 8), (5 and 10), 7, and 9 respectively as illustrated in **Figure 9(b)**.

Furthermore, the authentication time of DMAC algorithm on the single processor system is equivalent to the time of CBC-MAC algorithm. The increasing of processing units by one unit leads to decrease the time of authentication of DMAC 50% when the number of sub-messages is even. The incremental of processing units to three decreases the time of authentication 67% when the number of sub-messages is 3, 6, and 9. In four processing units system, DMAC decreases the time of authentication 75% when the number of sub-messages is 4 and 8. In five processing units system, DMAC decreases the time of authentication 80% when the number of sub-messages is 5 and 10. In conclusion, best authentication time decreasing of DMAC when the number of groups is divisible by the number of processing time.

6.3. Maximum Authentication Throughput

The maximum authentication throughput of CBC-MAC algorithm is equivalent for a single and multi processing units systems, since it works in a sequential fashion. The incremental of the number of secret keys does not influence the authentication throughput of CBC-MAC algorithm as illustrated in **Figure 10(a)**. Moreover, in a single processing system, the maximum authentication throughput of CBC-MAC is equivalent to DMAC algorithm in any level of security. In multi processing units system, the maximum authentication throughput of CBC-MAC is equivalent to DMAC algorithm in the first level of security.

However, since the maximum size of the message of the MKP is based on the number of the secret keys and the authentication time of the message by the DMAC algorithm is based on the number of secret keys and the number of processing units, then the number of secret keys and the number of processing units influences the

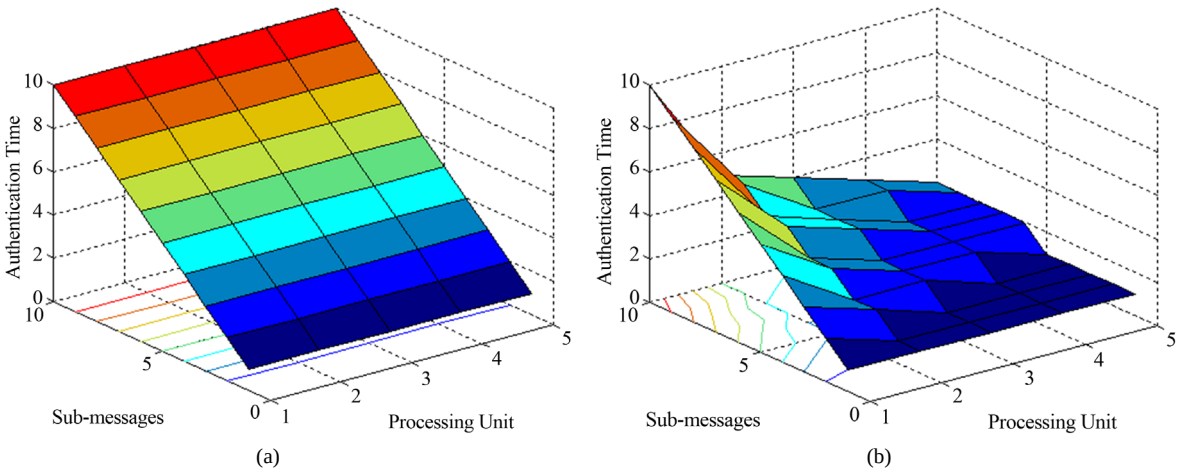

Figure 9. Authentication time on multi-processing system. (a) CBC-MAC; (b) DMAC.

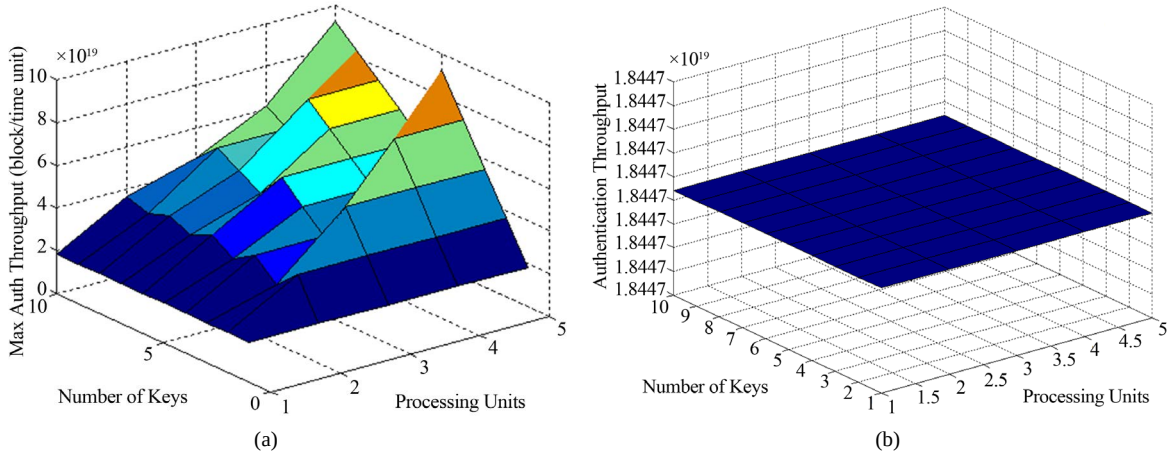

Figure 10. Maximum authentication throughput. (a) DMAC; (b) CBC-MAC.

maximum authentication throughput of DMAC algorithm.

In more details, in the second level of security (two secret keys), the DMAC increases the maximum authentication throughput 100% for system with up to two processing units. In the third level of security (three secret keys), the DMAC increases the maximum authentication throughput 50% for system with two processing units and 200% for system with up to three processing units. In the fourth level of security (four secret keys), the DMAC increases the maximum authentication throughput 100% and 300% for system with (2 and 3) and (4 and 5) processing units correspondingly. In the fifth level of security (five secret keys), the DMAC increases the maximum authentication throughput 67%, 150%, and 400% for system with 2, (3 and 4), and 5 processing units relatively. In the sixth level of security (six secret keys), the DMAC increases the maximum authentication throughput 100% for system with two processing units and 200% for system with up to three processing units. In the seventh level of security (seven secret keys), the DMAC increases the maximum authentication throughput 75%, 133%, and 250% for system with 2, 3, and (4 and 5) processing units separately. In the eighth level of security (eight secret keys), the DMAC increases the maximum authentication throughput 100%, 167%, and 300% for system with 2, 3, and (4 and 5) processing units correspondingly. In the ninth level of security (nine secret keys), the DMAC increases the maximum authentication throughput 80%, 200%, and 350% for system with 2, (3 and 4), and 5 processing units separately. In the tenth level of security (ten secret keys), the DMAC improves the maximum authentication throughput 100%, 150%, 233%, and 400% for system with 2, 3, 4, and 5 processing units respectively, as illustrated in **Figure 10(b)**.

7. Conclusions

The MKP protocol increases the message size of CCM security mode, since it provides multiple secret keys for ciphering blocks in CTR mode. The MKP supports the DMAC algorithm to authenticate a message in parallel fashion on multi-processing units system. The MKP breaks a message into a set of sub-messages (groups), and it assigns a distinct secret key to each group for authentication and ciphering independently. The level of security denotes the number of sub-messages and the number of secret keys.

The MKP increases the size of the transferred message in CCM mode. The size of the transferred messages is increased according to the level of security. The DMAC algorithm decreases the time of message authentication operation according to the level of security and the number of available processing units in the system. The relationship between the number of groups and the number of processing units evaluates the reduction of the time of authentication operation and they evaluate the incremental maximum authentication throughput on the multi-process units system.

References

[1] Whiting, D., Housley, R. and Ferguson, N. (2003) Counter with CBC-MAC (CCM).

[2] Lei, J., *et al.* (2007) Comparative Studies on Authentication and Key Exchange Methods for 802.11 Wireless LAN.

Computers & Security, **26**, 401-409. http://dx.doi.org/10.1016/j.cose.2007.01.001

[3] Rogaway, P. (2011) Evaluation of Some Block Cipher Modes of Operation. Technical Report, Cryptography Research and Evaluation Committees (CRYPTREC).

[4] IEEE Inc. (2003) IEEE 802.15.3 Working Group—Part 15.3: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPAN). *IEEE Draft Standard.*

[5] Yüksel, E., Nielson, H.R. and Nielson, F. (2008) ZigBee-2007 Security Essentials. *Proceedings of* 13*th Nordic Workshop on Secure IT-Systems*, 65-82.

[6] Dworkin, M. (2001) Recommendation for Block Cipher Modes of Operation. Methods and Techniques. DTIC Document.

[7] Ehrsam, W.F., *et al.* (1978) Message Verification and Transmission Error Detection by Block Chaining. Google Patents.

[8] Al-Alak, S., *et al.* (2013) Randomness Improvement of AES Using MKP. *Research Journal of Information Technology*, **5**, 24-34. http://dx.doi.org/10.3923/rjit.2013.24.34

[9] Al-Alak, S., *et al.* (2012) Authentication Time of IEEE 802.15.4 with Multiple-Key Protocol Using Distributed Message Authentication Code Algorithm. *Research Journal of Information Technology*, **4**, 140-154. http://dx.doi.org/10.3923/rjit.2012.140.154

[10] Standard, N. (1999) Data Encryption Standard (DES). Federal Information Processing Standards Publication.

[11] Schneier, B. (1993) The Idea Encryption Algorithm-The International Data Encryption Algorithm (IDEA) May Be One of the Most Secure Block Algorithms Available to the Public Today. Bruce Examines Its 128-Bit-Long Key. *Dr Dobb's Journal-Software Tools for the Professional Programmer*, **18**, 50-57.

[12] Rivest, R. (1995) The RC5 Encryption Algorithm. Springer, Berlin.

[13] Schneier, B. (1994) The Blowfish Encryption Algorithm. *Dr Dobb's Journal-Software Tools for the Professional Programmer*, **19**, 38-43.

[14] Fips, N. (2001) 197: Announcing the Advanced Encryption Standard (AES). Information Technology Laboratory, National Institute of Standards and Technology.

[15] Aladdin, K.S.L. (2000) The Enduring Value of Symmetric Encryption. *White Paper*.

[16] Koblitz, N. (1987) Elliptic Curve Cryptosystems. *Mathematics of Computation*, **48**, 203-209. http://dx.doi.org/10.1090/S0025-5718-1987-0866109-5

[17] Vanstone, S.A. (2003) Next Generation Security for Wireless: Elliptic Curve Cryptography. *Computers & Security*, **22**, 412-415. http://dx.doi.org/10.1016/S0167-4048(03)00507-8

[18] Kırlar, B.B. (2011) On the Elliptic Curves y2 = x3 − c with Embedding Degree One. *Journal of Computational and applied Mathematics*, **235**, 4724-4728. http://dx.doi.org/10.1016/j.cam.2010.08.020

[19] Edoh, K.D. (2004) Elliptic Curve Cryptography: Java Implementation. *Proceedings of the* 1*st Annual Conference on Information Security Curriculum Development*, Kennesaw, 8 October 2004, 88-93. http://dx.doi.org/10.1145/1059524.1059542

[20] LAN/MAN, S.C. (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. Control, 1-203.

[21] Qianqian, M. and Kejin, B. (2009) Security Analysis for Wireless Networks Based on ZigBee. *International Forum on Information Technology and Applications*, Chengdu, 15-17 May 2009, 158-160.