International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 04 | Apr 2020 www.irjet.net p-ISSN: 2395-0072

MAES: Modified Advanced Encryption Standard for Resource Constraint Environments

P.CHARISHMA[1], Dr. A.MAHESWAR REDDY[2], P.ANJANEYA[3], S.SALEEM[4]

[1]P.G Student, Dept of ECE, Annamacharya Institute of technology and sciences, Kadapa, Andhra Pradesh, India
[2]Professor, Dept of ECE, Annamachrya Institute of Technology and Sciences, kadapa, Andhra Pradesh India Author
[3]Professor, Dept of ECE, Annamachrya Institute of Technology and Sciences, kadapa, Andhra Pradesh India Author
[4]Professor, Dept of ECE, Annamachrya Institute of Technology and Sciences, kadapa, Andhra Pradesh India Author

---

Abstract - *Internet of things (IoT), internetworking of smart devices, embedded with sensors, software, electronics and net- work connectivity that enables to communicate with each other to exchange and collect data through an uncertain wireless medium. Recently IoT devices are dominating the world by providing it's in different functionality and real-time data communication. Apart from versatile functionality of IoT devices, they are very low- battery powered, small and having revealing or involving, and experience lots of challenges due to unsafe communication medium. Inspite the fact of many challenges, the energy issue is now becoming the prime concern. Optimization of algorithms in terms of energy consumption has not been explored specifically; rather most of the algorithms focus on hardware area to minimize it extensively and to maximize it on security issue as possible. But due to recent emerge of IoT devices, the main concern are shifting to moderate security and less energy consumption rate. We present MAES, a lightweight version of Advanced Encryption Standard (AES) which meets the demand. A new 1-dimensional Substitution Box is proposed by create or prepare methodically a novel equation for constructing a square matrix in affine transformation phase of MAES. Efficiency rate of MAES is around 18.35% in terms of packet transmission which indicates MAES consumes less energy than AES and it is applicable for Resource Constraint Environments.*

Keywords—AES, IoT, Energy Consumption, Resource Constraint Environments (RCEs), TelosB , Cryptography.

INTRODUCTION

Internet of Things (IoT) is the next generation of the internet which brings intense impact on our everyday lives. IoT is the extension of the Internet to connect just about everything on the planet. This includes real and physical objects vary from household accessories to industrial engineering [1]. As such these "things" that are connected to the Internet will be able to take actions or make decisions based on the information they gather from the Internet with or without human interaction. In addition, they also update the Internet with real-time information with the help of various sensors. IoT works with resource-constraint components such as sensor nodes, RFID tags etc. These components have low computation capability, limited memory capacity and energy resources, and capable to physical capture. Also, they communicate through the wireless communication channel which is not secured [2] and transmit real-time information through the presenting wireless medium. In certain applications, the state of keeping private, authentication, data freshness, and data integrity might be extremely important. Therefore, encryption of data is becoming a major concern [3]. But due to resource-constraint nature of the components, short-term security can meet the demand [4]. By the term resource-constraint means low battery power, less computing speed etc. Encrypting data using standardized code algorithms may consume more energy which extremely reduces the lifetime of the components. Two main approaches are followed to design and implement security primitives which are fitted with extremely constraint devices [5]. Firstly, designing new lightweight cryptosystem. For example, [6] - [12] are some recently proposed lightweight code algorithms. Secondly, modifying the existing standard cryptosystem in a lightweight fashion. Possible examples of the second approach are modification of the Advanced Encryption Standard Algorithm (AES) [13], SHA-256 [14] etc.

With respect to the particular appearance and implementation complexity, AES is considered as one of the strongest and efficient algorithms. Inspite of other symmetric encryption algorithms, the secret key distribution is still considered as a critical issue [15]. Again to encrypt or decrypt a single block (128-bit) of data, an essential amount of computational processing has to be done which consumes high quality battery power. As components of IoT have resource-constraint characteristics, consuming huge power may cause expiration of such components. Analyzing related work [5], we come to know that Substitution Layer is the most energy consuming portion of AES in the round based design. Considering energy consumption of resource-constrained components of IoT, we are proposing MAES, a lightweight version of AES where we reduce the computation of Substitution Box (S-Box) of AES.

The contribution of the proposed work can be give a brief statement of the main points as follow:

- We propose 1-dimensional Substitution Box (S-Box) which is constructed by formulating a novel equation for constructing a square matrix in affine transforma- tion phase of MAES.
- We implement both original AES and MAES algorithms on TinyOS 2.1.2 platform in TelosB sensor mote using MTS400 as sensor board along with TI MSP430 microcontroller and CC2420 radio chip.

- After examine the result of our experiment we conclude that MAES is well efficient than AES around 18.35% and 23.983 milliseconds in terms of number of packet transmission and latency, respectively.

The rest of the paper is organized as follows: Section II provides an overview of related works regarding degree of energy consumption of IoT devices. The preliminaries on the AES block cipher are given in Section III. The information regarding the methodos used to reduce the energy con- sumption of IoT devices by modifying Substitution Box (S- Box) of AES along with Rijndael S-Box generation method is described in Section IV. Performance analysis is discussed in Section V. Finally concluding remarks and future works are endowed in Section VI.

1. REATED WORKS

In recent years, many research works have been undergone and significant achievements have been found by many known researchers with respect to Resource Constraint Environments (RCEs). However, most of them have attack low area and low power as their concerned aspect and are hardware oriented. To provide implementation choice of the block cipher, Bogdanov et. al. [5] have investigated how the variation in (a) the architecture of S-Box/MixCloumn, (b) frequency of the clock cycle and (c) unrolling the design can affect the energy consumption. Their model is accuracy of estimating the energy consumed by several lightweight algorithms. With the help of figures, the proposed model is compared with respect to the different degree of unrolling. Moreover, they have proved that total energy consumption in a circuit during an encryption operation has roughly a quadratic relation with the degree of unrolling and the most energy consuming part is Substitution Box (S-Box) in 2-round unrolled design.

Feldhofer et. al. [16] and Moradi ET. al. [17] have proposed an implementation of AES and its basic transformations (such as S-Box) attacked low area and low power. All the implementations are mostly hardware oriented. [16] Design is based on an 8-bit datapath and approximately occupies 3400 Gate Equivalents (GE) and [17] design features a mixed datapath and requires 2400 GE respectively. The silicon implementation of low power AES is discussed in the work of Hocquet et. al. [18] which illustrates that 740 pj energy is consumed per encryption.

Kerckhof ET. al. [19] have presented a comparative study of different algorithms based on area, throughput, power and energy and applied state of the art techniques for reducing power consumption. Batina ET. al. [20] have explored the area, power, and energy consumption of several recently-developed lightweight block ciphers csonsidering possible optimization for the non-linear transformation and compared these with the AES algorithm. However, the effects of energy consumption for different design choices, such as the size of the data path, amount of serialization, and effects of architectural optimized are not considered by any of the works.researchers, Kong et. al. [22] have surveyed modern symmetric cryptographic solutions for Resource Constraint Environments (RCEs). Authors have provided collective surveys from other literature on the topic of hardware, applications, design re- quirements and security trends of RCEs.

Felicisimo et. al. [23] have proposed two Substitution Boxes, where the first S-Box is the Rijndael S-Box and the second S-Box, replacing the MixColumns operation of the original AES, is constructed through an XOR operation and affine transformation. Based on action of pretending testing, it is found out that the modified AES algorithm using multiple S-Boxes has better speed performance compared to the original cipher. Kawle et. al. [24] have proposed a modified AES using state- of-art techniques to decrease computational overhead of the original AES algorithm by encrypting large size data, for instance, multimedia data.

2. PRELIMINARIES

The Advanced Encryption Standard (AES) [13], a sym- metric key block which is published by the National Institute of Standards and Technology (NIST) in December 2001. It is a non-Feistel block cipher that encrypts and decrypts a fixed data block of 128-bits. There are three different key lengths. The encryption/decryption consists of 10 rounds of processing for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys.

AES performs several rounds where each round is made of several stages. A data block is changed from one stage to another. Before and after each stage, the data block is referred to as a state. Each round, except the last, performs four trans- formations which are invertible. The last round performs the rest three transformations except the MixColumns stage. Figure 1 shows the AES cipher structure.

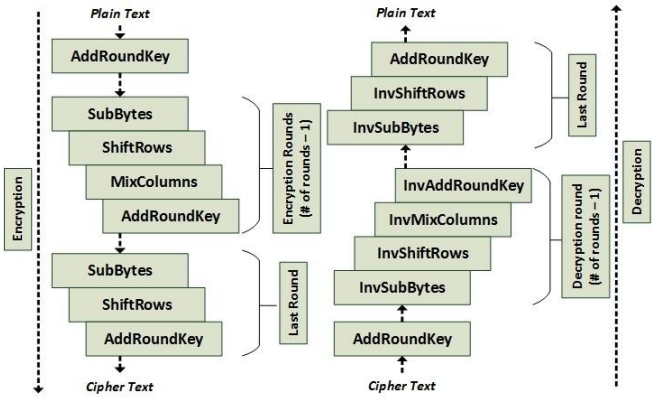

Fig. 1. General design of AES encryption and decryption

4 stages of each round are:

1)Substitute Bytes*:* The first transformation, Sub Bytes, is used at the encryption site. It is a non-linear byte substitution it controls independently on each byte of the state using a substitution table (S-Box). All the 16 bytes of the state are substituted by the equivalent

values which are found from the lookup table. In decryption, InvSubBytes is used. Bytes of a state are substituted from InvSubBytes table. Figure 2 shows the Sub Bytes operation.

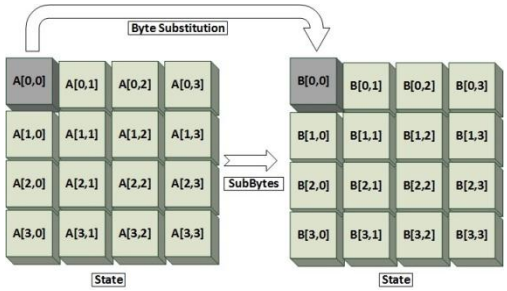

Fig. 2. Sub Bytes

2)Shift Rows*:* Fig 3 shows, the state bytes are shifted left in each row. It is called Shift Rows operation. The number of the shifts depends on the row number (0, 1, 2 or 3) of the state matrix. Row 0 bytes are not shifted and row 1, 2, 3 are shifted to 1, 2, 3 bytes left accordingly. The figure shows the Shift Rows operation.

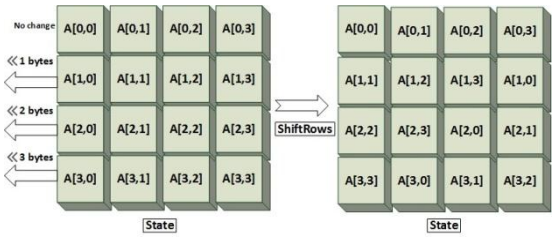

Fig. 3. ShiftRows

3)Mix Column: The MixColumns transformation operates at the column level. It forward each column of the state to a new column. The transformation is actually the matrix multiplication of a state column by a stable square matrix. All the arithmetic operations are conducted in the Galois Field (Finite Field). The bytes are treated as polynomials rather than numbers. Figure 4 shows the MixColumns operation.

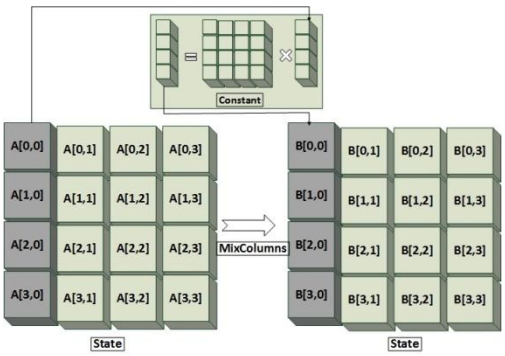

Fig. 4. MixColumns

4)Add Round Key: AddRoundKey proceeds one column at a time. It is similar to MixColumns in this respect. Ad-AddRoundKey adds a round keyword to each column matrix. Matrix addition operation is performed in the AddRoundKey stage. Figure 5 shows the AddRoundKey operation.

In encryption, Sub Bytes, Shift Rows, MixColumns, and AddRoundKey are performed in all rounds except the last round.

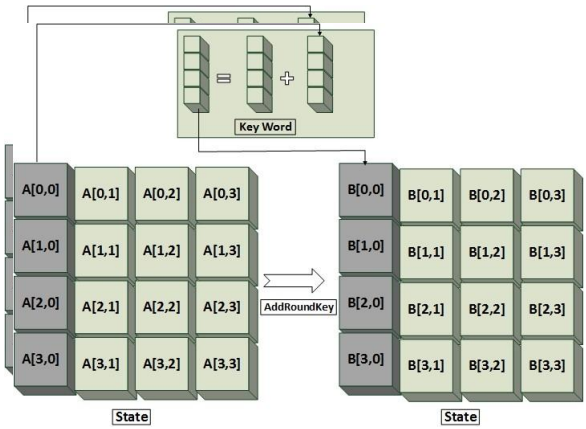

Fig. 5. AddRoundKey

MixColumns transformation operation is not performed in the last round of encryption. The decryption process essentially follows the same structure as the encryption, in addition to the nine rounds of Inverse ShiftRows, Inverse SubBytes, Inverse AddRoundKey and Inverse MixColumns Transformation. In the final round, Inverse MixColumns is no longer performed.

4. MODIFIED ADVANCED ENCRYPTION STRANDARD

According to previous research observation, we have found out that S-Box and MixColumns are the most energy con- summing stages in encryption and decryption process. We have examined the S-Box generation process of the Rijndael AES. The 16x16 2-dimensional lookup table is formed through the multiplicative inverse phase and affine transformation phase in the original AES. We put forwarded another new 1-dimensional lookup table as S-Box. It also follows the same generation process as the original one. However, substitution of one complete byte requires two times substitution from the S- Box. First four bits of the state byte is replaced first then the remaining four bits are substituted from the S-Box.

A. Rijndael S-Box Generation Method :

The Rijndael S-Box is a square matrix which is used in the Rijndael cipher. The S-Box performs duties as a lookup table. It is generated by determining the multiplicative inverse for a given number in GF (2^8) and they passing the information to the multiplicative inverse using affine transformation.

1) Multiplicative Inverse Phase: In multiplicative inverse phase, the input byte is inversed by substituting

value from multiplicative inverse table.

2) Affine Transformation: The most suitable of the simplified polynomial and the designated byte are the two most important factors of affine transformation phase. In Rijndael AES, $x^8 + x^4 + x^3 + x + 1$ is used as the irreducible polynomial and as the constant column matrix 0x63 specially designated byte is chosen. Basically, the affine transformation consists of two operations. Firstly, 8x8 square matrix's multiplication and secondly, 8x1 constant column matrix addition. The 8x8 square matrix is constructed using the following

$$d_i = b_i \oplus b_{(i+4)\%8} \oplus b_{(i+5)\%8} \oplus b_{(i+6)\%8} \oplus b_{(i+7)\%8} \oplus C_i \quad (1)$$

$b_i = i^{th}$ *bit of multiplicative inverse of input byte(2)*

$C_i = i^{th}$ *bit of a specially designated byte(3)*

Figure 6 illustrates the generation process of Substitution Box (S-Box) of the original AES.

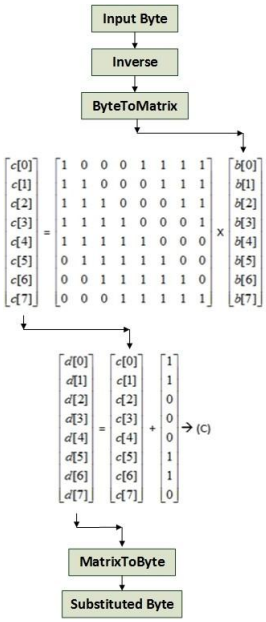

Fig. 6. Original S-Box generation process

B. Modified AES S-Box Generation :

Our modified AES S-Box generation process follows to build an establishment of the original AES. The whole process differs only in the selection of the simplified polynomial and specially particular byte.

1) Multiplicative Inverse Table: In the Rijndael AES, all the arithmetic operations are performed over the Galois Field (2^8). In our work, the Galois Field (2^4) is considered. The number of irreducible polynomials of degree 4 over GF(2) are $x^4 + x + 1$, $x^4 + x^3 + x^2 + x + 1$ and $x^4 + x^3 + 1$. All the generated values of the multiplicative inverse table and substitution box depend on the selection of irreducible polynomial. For our experiment purpose, we choose $x^4 + x + 1$ as our irreducible polynomial but we can select any of the simplified polynomials which are mentioned above. Following the Extended Euclidean Algorithm, 1-dimensional multiplicative inverse table is formed. Figure 7 illustrates the multiplicative inverse table of the proposed algorithm.

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1	9	E	D	A	7	6	F	2	C	5	A	4	3	8

Fig. 7. Multiplicative inverse table

2)Affine Transformation: This affine transformation pro- cess also follows two phases. Firstly, 4x4 square matrix's mul- tiplication and secondly, 4x1 constant column matrix addition. The 4x4 square matrix is to build following the equation 1 and equation 2 refers to the value of d_i.

$$d_i = b_i \oplus b_{(i+2)\%4} \oplus b_{(i+3)\%4} \oplus C_i \quad (4)$$

$C_i = i^{th}$ *bit of a specially designated byte which is*

Hexadecimal of $3, 8, 10, 13, 15$ *as they don't generate any fixed points.*

Selection of the constant value is a little bit not secured position. As we are calculating over the GF(2^4) where the value of the constant column matrix ranges from 0x00 to 0x0F, we can only select 5 values from there as these values do not generate any fixed point after transformation. The fixed point refers to the generation of the output value same as the input value. Figure 8 shows the generation process of proposed MAES.

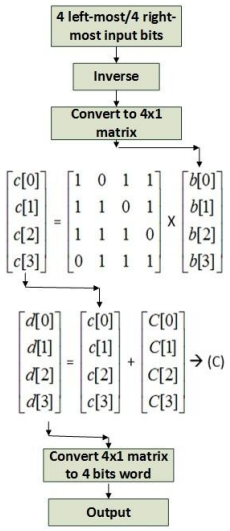

Fig. 8. Proposed MAES S-Box generation process

Different S-Boxes and inverse S-Boxes for different values of the constant value C is given below from figure 9 to 13:

Case-1: When C = 0x03

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
3	4	F	B	2	1	7	0	C	D	5	9	6	E	A	8

Inverse S-box:

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
7	5	4	0	1	A	C	6	F	B	E	3	8	9	D	2

Fig. 9. Case-1: When C = 0x03

Case-2: When C = 0x08

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	F	4	0	9	A	C	B	7	6	E	2	D	5	1	3

Inverse S-box:

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
3	E	B	F	2	D	9	8	0	4	5	7	6	C	A	1

Fig. 10. Case-2: When C = 0x08

5.PERFORMANCE ANALYSIS:

A.Environment Setup

The value of our experiment, we have gone through some environmental setup procedures. We have used telosB mote

Case-3: When C = 0x0A

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A	D	6	2	B	8	E	9	5	4	C	0	F	7	3	1

Inverse S-box:

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
B	F	3	E	9	8	2	D	5	7	0	4	A	1	6	C

Fig. 11. Case-3: When C = 0x0A

Case-4: When C = 0x0D

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
D	A	1	5	C	F	9	E	2	3	B	7	8	0	4	6

Inverse S-box:

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
D	2	8	9	E	3	F	B	C	6	1	A	4	0	7	5

Fig. 12. Case-4: When C = 0x0D

Case-5: When C = 0x0F

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
F	8	3	7	E	D	B	C	0	1	9	5	A	2	6	4

Inverse S-box:

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	9	D	2	F	B	E	3	1	A	C	6	7	5	4	0

Fig. 13. Case-5: When C = 0x0F

Which is fully able to exist with IEEE 802.15.4 standard and TinyOS 2.x. It has TI MSP430F1611 microcontroller and CC2420 RF chip. We implement MAES in nesC language which is supported in TinyOS 2.1.2. AA 1.5V batteries are used for the experiment.

We integrate MAES in TelosB sensor motes then transmit the encrypted temperature data while the sensor motes are placed in two different locations. Then we run our experiment for a week and analyze the packet transmission rate, the energy consumption rate, and the delay before a transfer of data begins following an instruction transfer of data transmission.

B. Implementation :

As we propose a lightweight version of AES for RCEs, first analyze the actual mechanism of AES and implement AES in structured programming language before implementing it in nesC. The same process is followed for the proposed MAES. After implementing both the algorithms (AES and MAES) in nesC, we integrate these in telosB sensor mote. Humidity, temperature, and luminance can be sensed by telosB sensor mote. Room temperature is used as our input set for both the algorithms. The experimental approach construct in two phases. In the first phase, we implement the original AES in the telosB sensor mote and build two sets of transmitted data packets. The first data set is to pass a encrypted data packets using the original AES encryption techniques and the second data set is composed of without converted a data packets. In the second phase, we implement both the original AES and the proposed MAES in the telosB sensor mote. Unauthorized data packets are not considered in the second phase of the experiment. Same as the previous phase, we construct two sets of transmitted data packets which are encrypted using

the original AES and the proposed MAES. The internal voltage sensor of telosB uses the microcontrollers 12-bit Analog to Digital Converter (ADC). For both the phases, the raw value of ADC is considered. Equation 6 is used to convert the raw value of the ADC to the corresponding voltage value.Since we conduct our experiment using AA 1.5V batteries, the values of reference voltage is 1.5V.

$$\text{Raw value}/4096*\text{Vref} \quad (6)$$

C. Comparison:

The Figure 14 shows the comparison between the number of packets of the first phase. Analyzing the result of figure 14, the black curve which is constructed from the original AES encrypted data set is steeped down compared with the red curve from the unencrypted data set. In this case, the voltage is very low accurate from 1.48V (raw value =4040) to 1.32V (raw value = 3600). The comparison between the number of packets of the phase 2 of our experiment is shown in fig 15. As we have mentioned earlier, transmission of the unencrypted data packets are not considered in the second phase. From the data log, we analyze that 18915 packets are sent for AES encryption when the voltage is dropped approximately from 1.50V (raw value = 4096) to 1.30V (raw value = 3550). For the same amount of voltage decreasing, 22385 packets are sent in terms of MAES which is is important way as a better choice while working in RCEs. The quality rate is 18.35%. Our proposed algorithm gives more energy efficiency than the original one. Table 1 shows the comparison of the number of transmitted packets between the Rijndael AES and the proposed MAES and also the efficiency. We calculate the latency of the proposed MAES and compare the result with the original AES. We observe the time needed to transmit 1000 packets which are encrypted by both the original AES and the proposed MAES. Considering the average transmission time, the latency of proposed MAES is less than the original AES. Table 2 illustrates the comparison of the delay before a transfer of data begins following instructions transferred between the original AES and the proposed MAES.

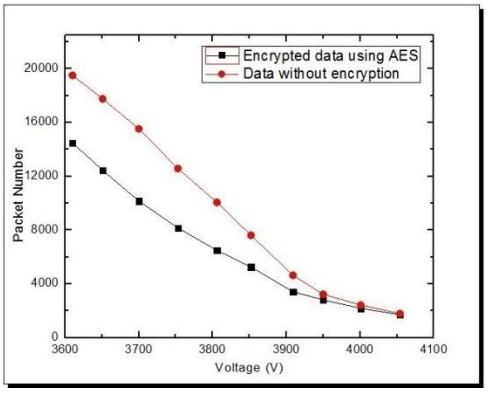

Fig. 14. Number of packets comparison between the encrypted and the unencrypted data

TABLE I. EFFICIENCY OF THE MAES ALGORITHM

Algorithm	Number of Packets	Voltage Degradation	Efficiency
Rijndael AES	18915	1.50 to 1.30	18.35%
Modified AES	22385	1.50 to 1.30	

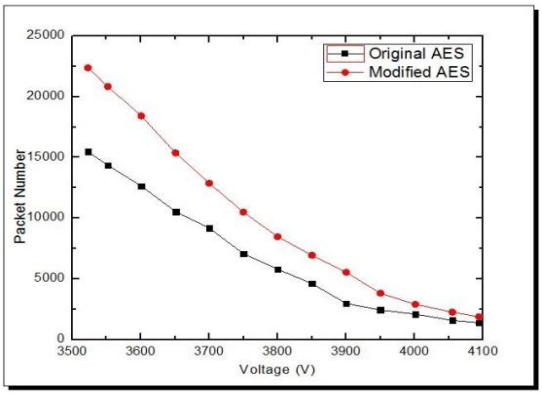

Fig. 15. Number of packets comparison between the modified and the original algorithm

TABLE II. LATENCY COMPARISON BETWEEN THE ORIGINAL AES AND THE MAES

Algorithm	Avg. transmission time (millisec)	Efficiency (millisec)
Rijndael AES	4999.879	29.983
Modified AES	4969.896	

6. CONCLUSION & FUTURE WORK

In this paper, we present a modified version of AES for Resource-Constraint Environments. A new Substitution Box is proposed which works over the Galois Field (2^4) to build remarkable an unique affine transformation equation. One notable feature of MAES is extending the battery life of low powered devices by consuming less amount of energy. The proposed method shows 18.35% efficiency when encrypted packets are transmitted using the proposed MAES to the sink node and the number of transmitted packets has increased. Similarly,29.983 milliseconds is found in terms of the delay before a transfer of data begins following an instruction is transferred . In future, the security issue and space complexity will be considered to make the proposed modification more applicable. Also, we plan to investigate multipath routing scheme while transmitting the encrypted data to the sink node. We will further delve to integrate Public Key Cryptosystem, specially Elliptic-curve cryptography (ECC) to reach able to linked to another efficiency in terms of number of packet transmission and the delay before a transfer of data begins following of data begins following an instruction is transfer with better security.

7. ACKNOWLEDGMENT

Authors of this paper are extremely grateful to the anonymous reviewers for their rigorous criticism about this

work which help the authors to reorganize the work in excellence.

REFERENCES

[1] Madakam, Somayya, R. Ramaswamy, and Siddharth Tripathi. "Internet of Things (IoT): A literature review." Journal of Computer and Commu- nications 3, no. 05 (2015): p.164.

[2] Wang, Yong, Garhan Attebury, and Byrav Ramamurthy. "A survey of security issues in wireless sensor networks." IEEE Communications Surveys Tutorial (2006).

[3] Veeramallu, B., S. Sahitya, and Ch LavanyaSusanna. Veeramallu, B., S. Sahitya, and Ch LavanyaSusanna. "Confidentiality in Wireless sensor Networks." International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-6, January 2013.

[4] Banik, Subhadeep, Andrey Bogdanov, and Francesco Regazzoni. "Ex- ploring energy efficiency of lightweight block ciphers." International Conference on Selected Areas in Cryptography. Springer, Cham, 2015.

[5] Bogdanov, Andrey, et al. "PRESENT: An ultra-lightweight block cipher." CHES. Vol. 4727. 2007.

[6] Borghoff, Julia, et al. "PRINCEa low-latency block cipher for pervasive computing applications." International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg, 2012.

[7] Beaulieu, Ray, et al. "The SIMON and SPECK lightweight block ci- phers." Design Automation Conference (DAC), 52nd ACM/EDAC/IEEE. IEEE, 2015.

[8] Suzaki, Tomoyasu, et al. "TWINE: A Lightweight Block Cipher for Multiple Platforms." Selected Areas in Cryptography. Vol. 7707. 2012.

[9] Li, Wei, et al. "Security analysis of the LED lightweight cipher in the internet of things." Jisuanji Xuebao(Chinese Journal of Computers) 35.3 (2012): p.434-445.

[10] Shibutani, Kyoji, Takanori Isobe, Harunaga Hiwatari, Atsushi Mitsuda, Toru Akishita, and Taizo Shirai. "Piccolo: An ultra-lightweight blockci- pher." In CHES, vol. 6917, pp. 342-357. 2011.

[11] Wu, Wenling, and Lei Zhang. "LBlock: a lightweight block cipher." In Applied Cryptography and Network Security, pp. 327-344. Springer Berlin/Heidelberg, 2011.

[12] Daemen, Joan and Rijmen, Vincent. "The design of Rijndael: AES-the advanced encryption standard.", Springer Science & Business Media, 2013.

[13] Descriptions of SHA-256, SHA-384, and SHA-512. http://csrc.nist.gov/groups/STM/cavp/documents/shs/sha256-384- 512.pdf.

[14] Al Hasib, Abdullah, and Abul Ahsan Md Mahmudul Haque. "Acomparative study of the performance and security issues of AES and RSA cryptography." Third International Conference on Convergence and Hybrid Information Technology, 2008. Vol.2.

[15] Feldhofer, Martin, Johannes Wolkerstorfer, and Vincent Rijmen. "AES implementation on a grain of sand." IEE Proceedings-Information Secu- rity 152, no. 1 (2005): p.13-20.