Area And Power Optimized AES For IoT Applications Using Dual Port ROM Based S-Box And Security By Optimized Key Scheduling Algorithm.

Rekha C, Krishnamurthy G N

Abstract: Data encryption plays major role in communication and security applications. Advanced encryption standard plays major role in internet of things (IoT) based applications. To apply encryption standard in real-time environment it is necessary to consider the parameters such as power consumption, speed and implementation area in a chip. AES encryption algorithm can be used in the IoT application due to moderate implementation complexity and better performance. AES can be easily deployed in the real-time environment using FPGA and embedded processors for high speed IoT based applications. Hardware and complexity can be balanced in the real-time environment by using AES standard and security with optimized key scheduling algorithm. In this work, optimized AES for IoT applications using dual port rom based S-Box (DPROM-SBOX-AES) and optimized key scheduling algorithm is proposed to increase the security and performance in the real-time environment. Proposed system gives better performance in term of security, hardware and timing performance when compared to conventional AES implementation. The AES architecture data path is optimized by replacing dual port RAM in the S-box which can improve the performance in terms of speed, delay and throughput and the key scheduling algorithm is optimized can improve the security in terms of avalanche effect, key sensitivity and coefficient correlation. To compare the performance, parameters such as area, power and delay parameters are considered. The analysis results show that the proposed technique attains 5% of reduction in power consumption and 10% of reduction in chip implementation area and improvement in the security.

Index Terms: Advanced Encryption Standard (AES), Internet of Things (IoT), S-Box, Dual Port Random Access Memory (DPRAM, Sub bytes, Data Encryption, Key Scheduling Algorithm.

1 INTRODUCTION

Nowadays IoT is a fast-growing technology applied for various real-time communication applications. The major purpose of IoT is developing a new world with help of smartphones, computers, consumer electronic devices, sensors that are connected in internet. Using this smart technology, we can access and control any device using smartphone or a tablet computer that are connected in internet. The enormous growth of the Internet leads to several innovations, Internet of things is one among them basically IoT is collaboration of several technologies. IoT is used in home automation, industrial factories, medical fields, transportation systems, applications [1]. The Internet of Things (IoT) is beginning to get a bad reputation every day, it seems like we hear of another way an insecure device of IoT was compromised. One of the main ways that the IoT can turn into an increasingly secure is through the best possible utilization of cryptography. The best way to utilize the cryptography is in securing the communication channels. According to the Statistics Portal research the total installed device connected in IoT reaches to 75.44 billion in worldwide by 2025 [2]. So, more vigilant about data security and need to achieve data confidentiality, data integrity and data availability. The most important mater is to provide secure communication protocol for protecting the cyber-attacks. Include security in the physical layer in the IoT protocols. Most of the communication protocols are lies under the physical layer both wireless (WIFI, Bluetooth,) and physical wired connections like coaxial and optical fiber cables [3]. However, most of the IoT devices not providing inbuilt security in it, the IoT devices basically low cost and have simple architecture. So, they don’t have higher memory or higher processing speed and not required large amount of energy conception it very difficult to implement security in those devices with low processing speed. In this paper provide an optimization of hardware arrangements for AES for high speed and low power with multilevel security in IoT device. The AES is formal encryption or decryption technique, adopted by the National Institute of Standards and Technology of the US Government and is acknowledged worldwide. The AES is more advanced than Data Encryption Standard (DES), it uses key length of 256 much larger than the key length of DES that can be broken easily because it has known vulnerabilities but AES is more secure than the DES cipher. It is found at least six time faster than triple DES. AES has been using in many applications including Banking, Military and Communication systems. The AES encryption or decryption algorithm is a block cipher, encrypted with several rounds with encryption key. In standard AES encryption or decryption process, block size is 128 bits or 16 bytes and the word “rounds” means the way in which the encryption or decryption algorithm combines to data and repeatedly encrypting 10-14 times dependent on the key. AES encryption uses a single key (128 bits, 192 bits or 256 bits in length) as a part of the encryption or decryption process. AES uses same key to perform both encryption and decryption. This is called a Symmetric Encryption algorithm. An encryption key, a binary data, is used in both encryption or decryption process [4]. To keep the encryption key secret, generation of keys/sub keys is important and use these subkeys since difficult to predict. The key scheduling algorithm
in AES implementation is used for generating the keys. AES key scheduling algorithm uses a single key to generate 44 32-bit keys. In conventional AES implementation uses simple structure and we can predict the first 4 four keys which are derived directly from 128-bit key. Depending upon these four keys we can easily derive remaining keys easily. In this work, dual port rom based s-box (DPROM-SBOX-AES) replaces the conventional ROM based S-box for reducing consumption of power and area. DPROM-SBOX-AES reduces the overall chip area power conception because the 2 DPROM is replaced instead of 4ROM based S – box, and an optimized key scheduling algorithm which provides better security. The proposed design is suitable for hardware-critical1 applications, such as network security, ATM Machines, smart card, PDA, and mobile phone, etc. The structure of the paper is given as follows: Section-II is the literature survey, it is brief description about the previous methods and its properties. Section III explains the proposed DPROM base S box implementation and optimized key scheduling algorithm. Section IV explains the result and discussion and compare our proposed method with other method and Vth section concludes the paper.

2 LITERATURE SURVEY
Kun-Lin T et al., have introduced AES-128 based Secure Less Power Communication for LoRaWAN, a long range wide area network, long distance communication protocol appropriate for IoT applications. Using this protocol, developed AES-128 based high securer and less power conception by reducing encryption cycles to make sure its communication security. A secure, low-power consumption communication method, SeLPC for LoRaWAN is suggested. Encryption key and D-Box update method in the SeLPC, which is provided to improve the level of security and makes simpler the AES encryption/decryption process so that further lowered consumption of power can be achieved. Relating both the customary AES and proposed method, the analysis result shows that the proposed method with SeLPC can reduce the power of encryption is up to 26.2% and also has the resistance towards three attacks, which are known-key, replay, eavesdropping attacks and is practically helpful for use in LoRaWAN IoT environments [6]. However, in this study, author has discussed on only application layer’s data encryption/decryption. The encryption key of the MAC layer (NwkSKey) is used to generate MIC code which is not updated periodically. Besides, the MIC-code generation process has not been simplified. Yuhao W et al. have proposed a b1ock-level-memory architectural design for advanced-encryption-standard(AES). In this method a Domain-Wall-Nano-wire based AES for higher Throughput and efficient consumption of Energy in Data-Encryption in Non-Volatile-Memory [7] is proposed. The logical operations needed for the evaluation of AES-cipher are realized within a homogeneous, high-density, and standby power free nonvolatile Spintronic-based memory array without revealing the intermediate results to external I/O interface using the DW-Nano-wires that features lower leakage power dissipation. The DW-XOR gates are used instead of XOR gates in CMOS based AES, DWWall-shifter are used instead of the Shift Rows transformation and the S-box function is replaced by DWall-LUT. The spintronic devices which are same devices that are used for storage elements (nonvolatile), which makes highly uniform and compact by the combination of the in-memory AES architecture. The throughput of the proposed work DWall-AES can be improved by nano-wire based techniques which are pipeline and multi-issue pipelining with the introducing a SW-FIFO and additional DW-nano-wires computing units for balancing the stage. Here the presented pipelined DWall-AES approach provides the better energy efficiency (22pJ/bit) among its rivals, with 3× and 5.2× improvements over the AES implementations based on CMOS, ASIC and CMOL based, respectively. Ali Akbar et al. in the proposed work and authentication based on transformation of Matrix and encryption/decryption implemented on Multicore Processor that achieves an exhaustive performance, high throughput and secure AES-CCM, yet less overheads of the power dissipation and the latency [8]. The high throughput of 8.32Gbps is achieved by implementing matrix multiplication GF(2^8) computation which transforms 16 plaintexts to 1 plaintext for authentication process. The low-overheads and highly resistance of SCA over >5×105 traces are attaining concurrently by influence the multi-core-ANOc with this propose AMPMP. The additional1 feature of security like adjusting key approach, is implemented to provide secure AES-CCM which is against leakage patterns of HW and the future computational attacks and HD during updating the key and MMA. With all these advantageous, applications of IoT with secure is proposed AMPMP in AES CCM. Weize Y et al. implemented a secure IoT against CPA attacks using false key-based lightweight masked AES. A false key and WDDL assisted AES process is used to avoid the leaking of stored secret key from the S_box under CPA attacks in the absence of major power and area overhead. False round key is added in each round of encryption/decryption to reduce the correlation amongst dynamic power consumption and actual key [9]. Simultaneously, WDDL-based-XOR gates are used during the reconstruction of block to conceal the intermediate data and remove the mask at the end of the encryption/decryption process. Attacker necessities to apply a CPA attack in two distinct stages during whole encryption/decryption process and the side-channel dynamic power consumption profile is correlated with the false key. After implementing the false key assisted AES technique and WDD, the MTD value becomes over 1.5×108. In this proposed technique the aspects like power, area, and performance overheads are negligible, as related to an unprotected AES engine. Yongchuan N et al. have proposed two novel attacks collision attack on linear layers, and scalable collision attack on block ciphers. They carried out practical attack experiments focusing on the edge computing scene, to verify the proposal method [10]. In the propose method, the leakage of linear layers will be find by analyzing the rotation amongst mask and masked data and further author proposes the collision-attack using mask against software execution of symmetric ciphers, LLCA, by using this 1eakage. It is the first time that the collision attack takes masked linear layers as the attack target. In gener1, the method capable of breaking first-order masking strategy that uses uniformly distributed random masks. Bui, Duy-Hieu, et al [11], have implemented an AES 32-bit data path architecture for low power and high throughput with multiple-level security. In this proposed method, number of registers are minimized and control logic and reduced the area and power. S. Mathew et al. presented a Nano AES hardware accelerator, an on-die lightweight, fabricated in CMOS with 22 nm tri gate high k/metal gate, aim for ultra-less-power symmetric-key encryption/decryption on mobile SOCs. This implementation utilizes a single 8-bit S-box circuit along with Shift Rows byte order data processing.
towards computing all rounds in AES in native GF \((2^{128})^2\) composite-field. The system archives high working speed but it consumes more power than the existing systems [12]. Sanu Mathew et al. [13] introduced a 2 stage pipeline data-path architecture for reconfigurable AES hardware accelerator fabricated in 45 nm CMOS. This method achieved very high-power conception and area. Shirai, Taizo, et al., have proposed a 128-bit block cipher CLEFIA with key-lengths of 128, 192 and 256 bits, which is compatible with AES. CLEFIA achieves enough resistance against known attacks as well as flexibility for efficiently carried out in hardware or software [14].

3 PROPOSED METHOD

Performance of AES implementation can be validated by using various parameters such as area, power, delay, throughput and security. For real-time application it is necessary to balance all these parameters for better performance. Fig.1 shows proposed method block diagram. It is having various blocks such as key-expansion unit, s-box, shift-rows, mix-columns etc. The final output of the AES is depending on the round which is performed while doing encryption process. The AES algorithm can be implemented by using parallelly as well as serially. Parallel implementation may take more area so that the power conception will be drastically increased more. In this work serial architecture is implemented to balance power, throughput and speed along with optimizing the key scheduling algorithm. Detailed information about each and every module with optimized key scheduling algorithm are explained in the following section.

3.1. Optimized Key Expansion Unit

3.1.1. Conventional Key Scheduling Algorithm

The 128-bit cipher key is fed into the key-expansion unit, the cipher key takes the form of 4x4 matrix. The first word from first row of the cipher key is taken to the key-array first column, and so on. Each round consumes four words from the key schedule and continues up to 44 rounds. Figure 2 shows cipher-key arrangement in the words form each with 4-byte and expansion of cipher-key is derived by using key scheduling algorithm consisting of 44 words each with 4-bytes. From these, before any round-based processing can begin the first four words are used for adding to the input state array, and the remaining 40 words are derived from the ten rounds of processing that are required for a 128-bit case encryption key that means depends over encryption key length. Before beginning of any round based encryption/decryption process, the input state-array is XORed with the first four key-schedule words in key-scheduling algorithm.

3.1.2. Optimized Key Scheduling Algorithm

The input for key expansion unit is 128-bit cipher key which is used as 4x4 matrix as shown in Figure 3. Each element of the first row is XORed to get TK(0), and each element in the second row is XORed to get TK(1), similarly TK(2) and TK(3) like

\[
\begin{align*}
K(0) \oplus K(1) \oplus K(2) \oplus K(3) &= TK(0) \\
K(4) \oplus K(5) \oplus K(6) \oplus K(7) &= TK(1) \\
K(8) \oplus K(9) \oplus K(10) \oplus K(11) &= TK(2) \\
K(12) \oplus K(13) \oplus K(14) \oplus K(15) &= TK(3)
\end{align*}
\]

The 32-bit value [TK0 TK1 TK2 TK3] is then XORed with the previous generated key value and stored it in temp variable. Depending upon the ith value the temp value is passed to rotation function and substitution function to get a key value. This value is then XORed with the [TK(0), TK(1), TK(2), TK(3)] in the next iteration to come up with more security on the key value. The flowchart of the proposed modified key-expansion algorithm provided in figure3 and algorithm for proposed key expansion algorithm 1 is given below.

Algorithm 1 Enhanced key expansion algorithm for AES
1: procedure Key Expansion ALGORITHM
2: for I = 0 to 44.
3: shift = 0
4: for j = 0 to 3
5: for k = 0 to 3
6: 
7: end for
8: 
9: end for
10: temp1 = temp1 \& (tk(0 \& tk(1 \& tk(2)\& tk(3)));
11: if((i % 4) == 0)
12: temp1 = substitute(rotate(temp1), s_box);
13: end if
14: expanded_key[i] = temp1;
15: temp = temp >> 8;
16: for k = 0 to 3
17: key1[4*i+k] = temp&0x000000ff;
18: temp = temp >> 8;
19: end for
20: end for
21: Repeat these steps to generate all entries of 44 key values
22: end procedure

3.1.3. Shift Rows

Shift Rows is used for shifting the rows of the state array circularly in the course of forward process and modification of Shift Rows contains:
1. In state array, the first row array remains same means no shifting at all
2. In state array shift the second row circularly left by one byte.
3. In state array shift the third row circularly left by two bytes
4. In state array shift the last row circularly left by three bytes.

This action over state-array represented by
Fig. 1. Original key scheduling algorithm with key, 128-bit, being expanded up to 44 words

<table>
<thead>
<tr>
<th>s00</th>
<th>s01</th>
<th>s02</th>
<th>s03</th>
</tr>
</thead>
<tbody>
<tr>
<td>s10</td>
<td>s11</td>
<td>s12</td>
<td>s13</td>
</tr>
<tr>
<td>s20</td>
<td>s21</td>
<td>s22</td>
<td>s23</td>
</tr>
<tr>
<td>s30</td>
<td>s31</td>
<td>s32</td>
<td>s33</td>
</tr>
</tbody>
</table>

Fig. 2. Modified key scheduling algorithm with key, 128-bit, being expanded up to 44 words.

(a)

Fig. 3. (a) Conventional ROM based S Box, (b) Proposed DPROM based S-box

3.1.4. Mix column
In this step, a function of all the bytes in column is replaced by each byte in a column in the same column is shown in below.

\[ s'_{ij} = (0 \times 02 \times s_{0j}) \otimes (0 \times 03 \times s_{1j}) \otimes s_{2j} \otimes s_{3j} \]

3.1.5. DPROM Based S-Box
Conventional DES and AES algorithm uses ROM based S-boxes for the sub word transformation. The proposed method replaces the 2 DPROM instead of 4 ROM. There for the overall chip area and power dissipation are reduced in the proposed system. The ROM based S-boxes are consuming more power, area and generate heat during the operation. DPROM-SBOX-AES is implemented using parallel architecture. Standard table which is used in the AES implementation is stored in the LUT to substitute the input bytes. Fig. 4. (a) shows the architecture of the existing s-box implementation which uses four single port ROM for the implementation. It takes more power consumption in terms of static power. S-box’s power-consumption is directly proportional to the number of clock circuit used in it. The number of clock signal in an implementation is directly proportional to the dynamic power consumption. In DPROM-SBOX-AES design, number of clock signal used is two which is lesser than the clock signal which is used in the conventional techniques. Power consumption of the DPROM-SBOX-AES can be reduced by a factor of ½. Fig. 4. (b) shows the architecture of S-box used in DPROM-SBOX-AES design. It is implemented using dual port ROM to store the bytes. It can access two memory location in a single clock cycle.

A 4-byte (32-bit) data is passed as the input for the S-Box. Fig. 5. Shows the dual port ROM used inside the S-Box. A bit splitter is used presented in Fig. 5. (a) and (b) to convert input data in to 4 individual bytes. The converted input bytes will be used as the input of S-box to extract the output. Similarly, a bit combiner block is used in the output side to combine the individual bytes into 4-bytes. By using DPROM hardware efficient architecture for the S-box for AES algorithm is implemented. Hardware and throughput balancing can be achieved by using dual port ROM based implementation of AES s-Box. The analytical results are explained in the following section by using various parameters such as area, power delay and throughput.

Fig. 4. (a) shows the architecture of the existing s-box implementation which uses four single port ROM for the implementation. It takes more power consumption in terms of static power.

Fig. 5. Shows the dual port ROM used inside the S-Box. A bit splitter is used presented in Fig. 5. (a) and (b) to convert input data in to 4 individual bytes. The converted input bytes will be used as the input of S-box to extract the output. Similarly, a bit combiner block is used in the output side to combine the individual bytes into 4-bytes. By using DPROM hardware efficient architecture for the S-box for AES algorithm is implemented. Hardware and throughput balancing can be achieved by using dual port ROM based implementation of AES s-Box. The analytical results are explained in the following section by using various parameters such as area, power delay and throughput.
4 ANALYSIS AND DISCUSSION

Standard 45nm technology library files are used to synthesize the code using cadence RTL compiler in Linux environment. The maximum operating frequency is set to 10MHz. Integrated environment such as MATLAB co-simulation using ModelSim is used to verify the encrypted and decrypted output in an easy and time saving manner. For analysis, DPROM-SBOX-AES is compared with other existing methods such as Mathew JSSC’11 [12], Mathew JSSC’15 [13], Zhao TVLSI’s [14] and Duy-Hieu Bui [11]. The detailed analysis and its results are presented in the following sections.

Table 1. Comparison with other Implementations of AES.

<table>
<thead>
<tr>
<th>Design</th>
<th>Block size (bit)</th>
<th>Key (bit)</th>
<th>Arch (data path)</th>
<th>Tech (nm)</th>
<th>Cycle/Encryption</th>
<th>Area (kGEs)</th>
<th>Freq (MHz)</th>
<th>Power (µW)</th>
<th>Througput (Mb/s)</th>
<th>Energy/bit (pj/bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPR-OMSBOX-AES</td>
<td>12, 8</td>
<td>8, 92</td>
<td>8</td>
<td>45</td>
<td>44.60</td>
<td>7.2</td>
<td>10</td>
<td>18 (@0.9 V)</td>
<td>30</td>
<td>0.5</td>
</tr>
<tr>
<td>Duy-Hieu Bui</td>
<td>12, 8</td>
<td>8, 19, 2, 25</td>
<td>32</td>
<td>28</td>
<td>44.52</td>
<td>8.6</td>
<td>10</td>
<td>20 (@0.6 V)</td>
<td>28</td>
<td>0.65 (-0.8 @0.6V)</td>
</tr>
<tr>
<td>Zhao TVLSI’s</td>
<td>12, 8</td>
<td>8</td>
<td>8</td>
<td>65</td>
<td>160</td>
<td>4</td>
<td>32</td>
<td>61.7 (@0.6 V)</td>
<td>25.6</td>
<td>2.3 (@0.6 V)</td>
</tr>
<tr>
<td>Mathew JSSC’15</td>
<td>12, 8</td>
<td>8</td>
<td>8</td>
<td>22</td>
<td>336</td>
<td>1.9</td>
<td>47</td>
<td>170 (@0.3 V)</td>
<td>29</td>
<td>5.6 (@0.34 V)</td>
</tr>
<tr>
<td>Mathew JSSC’11</td>
<td>12, 8</td>
<td>12, 19, 2, 25</td>
<td>8</td>
<td>45</td>
<td>5.6</td>
<td>10</td>
<td>31</td>
<td>409 (@0.3 V)</td>
<td>800</td>
<td>0.51 (0.34 V)</td>
</tr>
</tbody>
</table>

4.1 Result Analysis

Fig.10. depicts systems analysis performance by means of the other prevailing AES implementations. It is clear that our proposed method achieves highest performance as well as more security over the other methods. The above graph shows that the proposed method achieves low energy per bit about 0.5pj/bit. Also, it requires a smaller number of cycles around 44 cycles to complete the encryption. Power consumption and area of our proposed methodology is very smaller than the other methods this is achieved by replacing the ROM with DPROM. Apply original AES algorithm [5] and modified AES algorithm using same key on an image cman.bmp. The comparison is made on both original AES and alternate AES algorithms by using Avariance-Effect, Sensitivity of key and statistical analysis like correlation of horizontal pixels. In Figure5, the cman.bmp is encrypted or decrypted by using original AES-algorithm and same original image is encrypted or decrypted with modified AES algorithm. In Figure6 and Figure7, the result of encrypted or decrypted image by means of AES-original-algorithm and Figure8 and Figure9 shows the result of encryption/decryption images as a result of consuming modified-AES.

Fig. 5 Plaintext image cman.bmp

Fig. 6. Encrypted Image of cman.bmp using AES-Original-Algorithm

Fig. 7. Decrypted Image of cman.bmp using AES-Original-Algorithm
4.1.1. Area

Area is one of the important parameters in the chip level implementation of an RTL design. Minimum area of an RTL implementation is better for real-time working of a chip. Table I shows the various performance measure for DPROM-SBOX-AES with various techniques. It shows that, when compared to the other techniques such as Duy-Hieu Bui [11], Zhao TVLSI’s [14], Mathew JSSC’15 [13], Mathew JSSC’11 [12] DPROM-SBOX-AES is giving better performance in term of area.

4.1.2. Power

Power consumption is directly affecting the battery life of the system. Lesser power consumption increases the discharging rate of energy storage device and cost. DPROM-SBOX-AES techniques uses 45nm library with 0.7 – 0.9 V, obtained 18µ at 0.7V. it is very less comparatively other techniques such Duy-Hieu Bui [11], Zhao TVLSI’s [14], Mathew JSSC’15 [13], Mathew JSSC’11 [12] DPROM-SBOX-AES is giving better performance in term of area.

4.1.3. Throughput

Speed of a processor is directly proportional to the throughput of the system. Through put defined the difference between the processing time between the in put and output. Increases in the throughput increases the speed of the system. Balanced throughput is obtained in DPROM-SBOX-AES when compared to the other works. Fig.6 shows the Throughput and other performance metrics which is used to measure hardware performance.

4.1.4. Key Sensitivity

The Key sensitivity test, [15] has been carried out as follows
1. Applying 128-bit key, Key1, to Encrypt an image Arms.bmp by original AES algorithm.
2. Change any one bit randomly selected from Key1. Using this modified key, say Key2, same image will be encrypted by applying to original algorithm. Ex : BD F2 98 DE 5E 26 4A D1 F5 DE C9 4A AD F2 5E 28, from this key1 we have randomly selected F as shown in bold and changed to B as BD F2 98 DE 5E 26 4A D1 F5 DF C9 4A AD F2 5B 28 which is key2.
3. Apply these two keys to modified AES algorithm using the same image and then compared.
4. The result is shown in below Table 2, comparison of both ciphered images which are encrypted by original as well as modified AES algorithm using same key at all rounds.

<table>
<thead>
<tr>
<th>Number of Rounds</th>
<th>AES-Original</th>
<th>AES-Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>78.854</td>
<td>78.578</td>
</tr>
<tr>
<td>4</td>
<td>78.583</td>
<td>78.538</td>
</tr>
<tr>
<td>6</td>
<td>78.538</td>
<td>78.599</td>
</tr>
<tr>
<td>8</td>
<td>78.551</td>
<td>78.572</td>
</tr>
<tr>
<td>10</td>
<td>78.562</td>
<td>78.546</td>
</tr>
</tbody>
</table>

4.1.5. Avalanche Effect

<table>
<thead>
<tr>
<th>No of Rounds</th>
<th>CASE1</th>
<th>CASE2</th>
<th>CASE3</th>
<th>CASE4</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>886</td>
<td>826</td>
<td>819</td>
<td>876</td>
</tr>
<tr>
<td>4</td>
<td>835</td>
<td>882</td>
<td>813</td>
<td>1390</td>
</tr>
</tbody>
</table>
The avalanche effect in [15] means if there is a small modification in the plaintext/key then there will be significant different form in the ciphertext. To compute avalanche we need to change one bit from the plain text (image cman.bmp), named as an image Bcman.bmp, and then encrypt this image using both original AES and modified AES algorithms. Here the proposed algorithm is compared with original algorithm at different rounds along with four cases. The Table 3 provides which algorithm gives better avalanche effect

**Case 1:** Comparing Avalanche-effect for encrypted image of cman.bmp using AES-original and AES-modified algorithm with key1 and key2

**Case 2:** Comparing Avalanche-effect for encrypted image B1cman.bmp using AES-original and AES-modified algorithm with key1 and key2.

**Case 3:** Comparing Avalanche-effect for encrypted image of cman.bmp and B1cman.bmp using AES-original and AES-modified algorithm with same key1

**Case 4:** Comparing Avalanche-effect for encrypted image of cman.bmp and B1cman.bmp using AES-original and AES-modified algorithm with same key2

4.1.6. **CORRELATION COEFFICIENT**

The correlation is a relationship between two pixels which are adjacent in an image [16], [17]. To find the correlation of horizontal-adjacent pixels in an image (cman.bmp) is as follows

1. N pairs of horizontal-adjacent pixels from the original image cman.bmp and cipher text image by using AES original algorithm and modified algorithm.

2. Grey scale values, x and y, of horizontal-adjacent pixels are used to calculate correlation coefficient, \( r_{xy} \) formulae, in an image.

\[
r_{xy} = \frac{COV(x, y)}{\sqrt{D(x) \times D(y)}}
\]

where variance of x and y is represented by \( D(x) \) and \( D(Y) \) and the mathematical formula for calculating \( COV(X, Y) \) is

\[
COV(x, y) = \frac{1}{N} \sum_{i=0}^{N} (x_i - E(x))(y_i - E(y))
\]

Where the mean values of \( \{x, y\} \) are represented as \( E(X) \) and \( E(Y) \) and number of neighboring pixels, say N, which are selected randomly. The correlation test is implemented for these random selected horizontal-adjacent pixels from plain image cman.bmp and ciphered images and is as shown in below Figure 12, Figure 13, and Figure 14. The correlation-coefficient of original image is 0.935614 and for ciphered image encrypted by AES is -1.616538 and for modified AES is -1.687496. In the original AES and modified AES algorithm the correlation-coefficients for plain-image with respect to the cipher-images are far apart.

**Figure 12. Analysis of Correlation Coefficient for Original Image cman.bmp**

**Figure 13. Analysis of Correlation Coefficient using original AES algorithm**

**Figure 14. Analysis of Correlation Coefficient using modified AES algorithm**
5 CONCLUSION
In this work, an optimized AES for applications of IoT using dual port rom based S-box (DPROM-SBOX-AES) and modified key expansion algorithm is implemented for real-time applications. DPROM-SBOX-KEY EXPANSION-AES design achieves a lower-cost higher-throughput, ultra-lower-power, ultra-lower-energy with multi-level security. The power consumption is reduced to 18µW at 0.5v and area reduced to 7.2 kGE with the help of DPROM-SBOX. Also, DPROM-SBOX-AES depicts optimized procedure is not only favorable for the consumption of throughput, area and energy or power and as well adds security that provides by modifying key expansion algorithm. With the analysis of key-sensitivity, avalanche-effect, horizontal1-correlation coefficient, higher throughput, ultra-low-power utilization, and ultra-low-energy utilization, our proposed method is obviously suitable for future ultra-low-power applications of IoT with multiple level of security.

6 REFERENCES