Improved Image Security in Internet of Thing (IOT) Using Multiple Key AES

Khadeeja Gebor Salim*  Saif M. Kh. Al-alak Majid Jabbar Jawad

Department of Computer Science, College of Science for Women, University of Babylon, Hillah, Iraq
*Corresponding author: khadeeja88@gmail.com, saif.shareefy@gmail.com, majid_al_sirafi@yahoo.com
ORCID ID: https://orcid.org/0000-0003-1990-3691, https://orcid.org/0000-0002-0042-9400, https://orcid.org/0000-0001-9142-3815

Received 26/3/2020, Accepted 23/8/2020, Published Online First 11/1/2021, Published 1/6/2021

Abstract:
Image is an important digital information that used in many internet of things (IoT) applications such as transport, healthcare, agriculture, military, vehicles and wildlife, etc. Also, any image has very important characteristic such as large size, strong correlation and huge redundancy, therefore, encrypting it by using single key Advanced Encryption Standard (AES) through IoT communication technologies makes it vulnerable to many threats, thus, the pixels that have the same values will be encrypted to another pixels that have same values when they use the same key. The contribution of this work is to increase the security of transferred image. This paper proposed multiple key AES algorithm (MECCAES) to improve the security of the transmitted image through IoT. This approach is evaluated via applying it on RGB bmp images and analyzing the results using standard metrics such as entropy, histogram, correlation, Peak Signal-to-Noise Ratio (PSNR) and Mean Square Error (MES) metrics. Also, the time for encryption and decryption for the proposed MECCAES is the same time consumed by original single key AES is 12 second (the used image size is 12.1MB therefore time is long). The performance experiments show that this scheme achieves confidentiality also it encourages to use effectively in a wide IoTs fields to secure transmitted image.

Key words: AES, Asymmetric, Cryptography, IoT, MECC, RGB Color Image, Symmetric, Security, WiFi, WiMAX.

Introduction:
The IoT involves diverse link layer technologies and an enormous collection of devices (1). The number of IoT applications is growing. The applications comprise smart home, smart city, smart agriculture, utilities, healthcare monitoring, and animal farming, smart water, security and emergencies, industrial control, smart transportation, environment monitoring (2). It tackles the highly sensitive information about people and companies, which must not be revealed to those unauthorized persons and attackers.

Security is a very important issue that could confront the IoT development and providing security for IoT technology is a great real challenge. As the IoT technology has a wide spread scope, there are many variables areas of research that focuses on security challenges. Security requirements in the IoT environment are confidentiality, integrity, authentication, authorization, access control, and availability.

One of the most important methods to provide data security, especially to protect end-to-end data sent over networks is Cryptography. Cryptography is the system in which text or any type of data is converted from readable data to unreadable (encrypted) data that can only be decrypted by someone who has the decryption key therefore, the purpose of the encryption is to ensure that the privacy of the data is not allowed to tamper with or view it, because it is either confidential or private and very important. Symmetric (the key used to encrypt and decrypt is the same) and asymmetric (the key used to encrypt is differed from the key used to decrypt) are basic types in cryptography and lightweight appear as modern type which is focused on the lower end devices.

A few lightweight security mechanisms currently in use are based on number of security techniques and cryptographic algorithms. They include hash functions like access control, XOR encryption, lightweight public-key cryptographic schemes based on ECC (Elliptic Curve Cryptography) (3) and symmetric AES 128 algorithm (that used as default in Wireless Fidelity
Elliptic Curve Cryptography (ECC) is one of the most symmetric block cipher algorithms that is used commonly in ciphering. When encrypting data by AES algorithm, it is very difficult for hackers to get the original data. There are three key sizes allowed for the AES algorithm: 128, 192, and 256 bit; but the block size of the message allowed is fixed which is 128 bit.

Security in transmission of the digital images has its importance in the present image communications due to the increasing use of image throughout the industrial process, transport, healthcare like Connected MRI (Magnetic Resonance Imaging) scanner, agricultural, military, vehicles and wildlife etc. It is necessary to protect the confidentiality of image data from unauthorized access, therefore image security is a critical issue and that is obvious from researchers who tried in many papers to enhance image security by use different encryption schemes.

The Advanced Encryption Standard (AES) is one of the most secure public key cryptography algorithm because it uses fully exponential computations to solve the problems. It is used to generate smaller, faster, and more efficient cryptographic keys. ECC generates the keys through the properties of an elliptic curve equation instead of the conventional generation method as a result of very large initial numbers. Because ECC helps create equivalent security with less power to use the computer and use battery resources.

The proposed MECC technique with two sets key generator model is demonstrated in Fig.1. Algorithm 1 shows the proposed generated keys. In this model, a different number of initial sequences and different ECC public keys are used, two initial random key seeds $r_0$ and $s_0$ are used to generate two sets of keys: $r_1, r_2 \ldots r_n; s_1, s_2 \ldots s_n$, where the sets are generated according to the Equations (1 and 2).

Then, the xor function $f$ equation (3) is used to obtain the final keys, which are used in the encryption and decryption process.

Where:

- $ECCen$ is an elliptic curve ciphering (ECC algorithm).
- TCK1 and TCK2: are trust center public keys which are different keys.
- The trust center must satisfy the values of $r_0$ and $s_0$ in the sender and receiver sides.

The main objectives of this work are improving the security and the accuracy of the transmitted image through IOT. The work suggested a multiple key symmetric scheme. At the first use MECC technique to generate multiple secret keys which is used with AES for improving the security and accuracy of image of IOT.

**MECC Technique**

The proposed MECC (14) technique can generate $N$ keys by using multiple ECC public keys. The proposed MECC is two sets key generator. In this model, a different number of initial sequences and different ECC public keys are used, two initial random key seeds $r_0$ and $s_0$ are used to generate two sets of keys: $r_1, r_2 \ldots r_n; s_1, s_2 \ldots s_n$, where the sets are generated according to the Equations (1 and 2).

The proposed MECC technique with two sets key generator model is demonstrated in Fig.1. Algorithm 1 shows the proposed MECC technique with two sets key generator model.

**Algorithm (1): MECC Algorithm (Two Sets)**

| Input: $s_0, r_0$; key sets $PK_1, PK_2$; generated public keys |
| Output: $K_1$: proposed generated keys |

1. **Begin**
2. For $i = 1$ to $n$ (Where $n$: is number of the secret keys)
3. $s_i = ECC(s_{i-1}, PK_1)$
4. $r_i = ECC(r_{i-1}, PK_2)$
5. **End for**
6. For $i = 1$ to $n$
7. $K_{1i} = f(s_i, r_n-i+1)$
8. **End for**
9. **End**

This study tries to enhance the security of transmitted image through IOT which is secured by using AES 128 single key. The proposed MECC - AES (Multiple key Elliptic Curve Cryptography) achieves high levels of security and integrity according to statistical metrics such as Entropy, PSNR, MSE and correlation.
The MECC -AES technique is suggested to improve the strength of the security system that is available when using keys generated by the MECC technique. The MECC -AES technique consists of a MECC technique that generates secret keys and the AES algorithm which uses these secret keys in the encryption and decryption operations. In order to encrypt the plain image by the MECC -AES technique after separating the plain image into three matrix of R,G,B colors, each matrix must be divided into the same number of n blocks, as demonstrated in Equation (4). If the number of secret keys equals the number of image blocks then encrypting each block directly with the corresponding key. But if the number of keys is less than the number of blocks. Then these blocks are distributed into groups. The number of these groups is equal to the number of secret keys. All blocks belonging to the same group i are encrypted by the secret key Keyi. The encryption operation using AES algorithm is demonstrated in Equation (5). The decryption operation is in reverse way.

\[
I_R = \sum_i B_i, \quad I_G = \sum_i B_i, \quad I_B = \sum_i B_i
\]  

\[
C_R = AES(BIR) 
C_G = AES(BIG) 
C_B = AES(BIB)
\]  

Where:
- \(C_R\): red color matrix of encrypted image.
- \(C_G\): green color matrix of encrypted image.
- \(C_B\): blue color matrix of encrypted image.
- AES: AES algorithm.
- Key: Secret key which is generated by the MECC technique.

Figure 2 shows a general view of the proposed MECC-AES encryption and decryption technique.
Algorithms 2 and 3 demonstrates the encryption of the proposed MKE-AES technique.

| Algorithm (2): M ECC-AES Encryption | Algorithm (3): MECC-AES Decryption |
|------------------------------------|-------------------------------------|
| **Input**: RGB Image (separate RGB color image into three matrix Red, Green, Blue); Fifty (50) keys; secret keys | **Input**: RGB Ciphered Image (separate RGB color image into three matrix Red, Green, Blue); Fifty (50) keys; secret keys |
| **Output**: $C_1, C10, C20, C30, C40, C50$; the ciphered image (depending on the number of secret keys) | **Output**: $DC_1, DC10, DC20, DC30, DC40, DC50$; the Deciphered image (depending on the number of secret keys) |

1. **Begin**

   2. Block-size = 16 byte
   3. No-of-Blocks ($R,G,B$) = (Height*width) / block-size
   4. No-of-Blocks-in-Group = No-of-Blocks / n (Where n: is number of the secret keys)
   5. $i = 1$
   6. For $j = 1$ to No-of-Blocks ($R,G,B$)
      7. $CR_j = AES(B_j, Key_i)$
      8. $CG_j = AES(B_j, Key_i)$
      9. $CB_j = AES(B_j, Key_i)$
      10. If ($j$ mod (No-of-Blocks-in-Group) = 0)
          11. $i = i + 1$
      12. End If
      13. Reconstruct image
      14. End For

15. **End**

   (A)                                                                                   (B)

Figure 2. The Proposed MECC-AES Technique (A) Encryption (B) Decryption
Result and Discussion:

Based on the algorithms discussed in the previous section, an encryption quality is evaluated by using Sun image shown in Fig.3 which has high redundancy and it is taken from website (https://www.hlevkin.com/06testimages.htm) (15). The proposed scheme is implemented on configuration such as OS: Windows 7 ultimate 32 bit, Processor: Intel Celeron, Memory: 4 GB, Software used: Java, Matlab 2008a, Excel and Input image: RGB image with Dimensions 2096*2030 and size 12.1 MB Bit depth 24 bit. After applying the proposed scheme, Fig.4 shows the ciphered images that belongs to the original RGB Sun image.

![Ciphered images](image)

Figure 3. Original Sun image.

| Ciphered 1 key | Ciphered 10 keys | Ciphered 20 keys |
|---------------|------------------|------------------|
| ![Ciphered 1 key](image) | ![Ciphered 10 keys](image) | ![Ciphered 20 keys](image) |
| Ciphered 30 keys | Ciphered 40 keys | Ciphered 50 keys |
| ![Ciphered 30 keys](image) | ![Ciphered 40 keys](image) | ![Ciphered 50 keys](image) |

Also, in this section the results of the proposed MKE-AES method are discussed as follows: Figure 4. The ciphered images with different number of secret keys corresponding to plain Sun image.

Entropy

Entropy (H) is a statistical measure of uncertainty (randomness) in communication theory (16). Information theory (Entropy) is concerned with network security, cryptography, data compression, communication systems, error correlation and other related topics. The Eq. (6) is used to calculate the entropy as follows:

\[ H = - \sum p(x_i) \log_2 p(x_i) \]  

Where \( p(x_i) \) is the probability density function of the occurrence of the symbol \( x_i \). Entropy of an
image indicates how grey level values are distributed. When an image has equal probabilities, the entropy will be calculated to 8 which is an ideal result. If the entropy is less than eight (8) this means there is a certain degree of predictability (17). For a cryptosystem to resist the entropy attacks.

Table 1 for Sun image illustrates how the entropy is increased when number of keys is increased. The entropy values in each ciphered image is increased and be more close to ideal value(8) which mean highly randomness. As a result all entropy values are better than entropy for ciphered image with one key.

| Entropy          | Red       | Green     | Blue      | AVG(RGB)   |
|------------------|-----------|-----------|-----------|------------|
| Original Sun     | 7.350198419 | 4.706794447 | 1.5760462 | 4.5443464  |
| Ciphered 1 key   | 7.999960541 | 7.9949104  | 7.9981814 | 7.9976841  |
| Ciphered 10 keys | 7.99993581  | 7.999339879 | 7.997501  | 7.996812   |
| Ciphered 20 keys | 7.999959587 | 7.99963732  | 7.998085  | 7.998019   |
| Ciphered 30 keys | 7.999958022 | 7.999772259 | 7.998683  | 7.998662   |
| Ciphered 40 keys | 7.999960651 | 7.999807619 | 7.999037  | 7.998906   |
| Ciphered 50 keys | 7.99995235  | 7.999836309 | 7.999058  | 7.998982   |

PSNR & MSE

PSNR is abbreviation to Peak Signal-to-Noise Ratio that reveals the encryption quality and displays the changes in the pixel values between the plain image and the ciphered image (18). The result of PSNR is a single number in decibels (dB) and if this number is smaller than 30dB that mean the encrypted image is dissimilar to the original. Mathematically PSNR is founded by putting the Mean Squared Error (MSE) in relation to the maximum possible value of luminance (for a typical 8-bit value this is $2^8 - 1 = 255$) as follows:

$$\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (f(i,j) - F(i,j))^2 \quad \ldots \quad (7)$$

$$\text{PSNR} = 10 \cdot \log_{10} \left( \frac{\text{MAX}^2}{\text{MSE}} \right) \text{dB} \quad \text{(formula no)} \ldots \quad (8)$$

Where $f(i,j)$ represent the original pixel (i, j), while $F(i,j)$ is the ciphered pixel, and $M \times N$ refer to the picture size. MSE as part of equation 6, provides a quantitative score to describe the degree of similarity between two signals (19).

Table 2 show the values of PSNR is highly below 30dB and that prove the encrypted image is highly secure and MSE values is very high and far from 0 with significant value. Therefore PSNR and MSE proved that the ciphered image is completely different from original one. At the same time the values of PSNR & MSE are Inf and 0 successively prove that the deciphered image is identical to the original image ,by which this scheme is lossless way.

| Image                      | Avg.MSE | Avg.PSNR |
|----------------------------|---------|----------|
| Original Sun with decrypted image | Inf dB  | 0        |
| Ciphered 1 key            | 16262.76 | 6.1761 dB|
| Ciphered 10 keys          | 16329.59 | 6.1598 dB|
| Ciphered 20 keys          | 16335.59 | 6.1586 dB|
| Ciphered 30 keys          | 16365.47 | 6.1511 dB|
| Ciphered 40 keys          | 16341.59 | 6.1571 dB|
| Ciphered 50 keys          | 16347.52 | 6.1551 dB|

Histogram

Histogram is one of the significant statistical characteristics that illustrates how many times the pixel intensity value occurs inside an image. A respectable encryption scheme provides a uniform histogram of ciphered image contrasting to the original image histogram. (20)

The below Fig.5 shows the histogram of original and ciphered images. Histogram of ciphered image in contrast to the original image histogram is uniform in RGB channels and when number of keys is increased histogram is more uniform and this proves the encryption with multiple keys increases defense toward statistical attacks.
| Image name       | Red channel | Green channel | Blue channel | RGB image |
|------------------|-------------|---------------|--------------|-----------|
| Original Sun     | ![Image](image1.png) | ![Image](image2.png) | ![Image](image3.png) | ![Image](image4.png) |
| Ciphered 1 key   | ![Image](image5.png) | ![Image](image6.png) | ![Image](image7.png) | ![Image](image8.png) |
| Ciphered 10 keys | ![Image](image9.png) | ![Image](image10.png) | ![Image](image11.png) | ![Image](image12.png) |
| Ciphered 20 keys | ![Image](image13.png) | ![Image](image14.png) | ![Image](image15.png) | ![Image](image16.png) |
| Ciphered 30 keys | ![Image](image17.png) | ![Image](image18.png) | ![Image](image19.png) | ![Image](image20.png) |
| Ciphered 40 keys | ![Image](image21.png) | ![Image](image22.png) | ![Image](image23.png) | ![Image](image24.png) |
| Ciphered 50 keys | ![Image](image25.png) | ![Image](image26.png) | ![Image](image27.png) | ![Image](image28.png) |

**Figure 5. Histogram of original and encrypted Sun image with different number keys**

**Correlation**

Highly correlation inherently is guaranteed in plain image data between pixels (close to one) in horizontal, vertical or diagonal directions. The encryption aim is to destroy the strong correlation between neighboring pixels of the ciphered image (near to zero). It is strongly related if $|r_{xy}| > 0.8$, otherwise weakly related if $|r_{xy}| < 0.3$. In image, the correlation coefficient between each pair of pixels uses equation (9) to be calculated as follow:
\[ D(x) = \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))^2 \]

Where,

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

\[ E(x) = \frac{1}{N} \sum_{i=1}^{N} x_i \]

\[ r(x,y) = \frac{Cov(x,y)}{\sqrt{D(x)D(y)}} \]

\[ \text{(9)} \]

Where \(cov(x,y), D(x)\) and \(D(y)\) are covariance and variances of variable \(x\) and \(y\) respectively. \(E(x)\) and \(E(y)\) are the expected values of the variables \(x\) and \(y\). Where \(N\) is the whole pels of an image, \(N=\text{rows} \times \text{cols}\), \(x\) is a vector of length \(N\) and \(x_i\) is the intensity values of original image \((20,21)\).

Tables 3, 4, 5 and Figs. 6, 7, 8 show correlation between pixel pairs adjacent. Original Sun image show strong correlation while encrypted images show low correlation in three directions horizontal, vertical and diagonal for selected 3000 pixel pairs in a random way. The low values of correlation nearly to zero and less than 0.3 between pixels in ciphered image proved the power of MECC-AES scheme.

### Table 3. Evaluation of correlation metric in horizontal, vertical and diagonal for Sun image in Red channel.

|        | horizontal | vertical | diagonal |
|--------|------------|----------|----------|
| Original Sun | 0.9977     | 0.9969   | 0.9953   |
| Ciphered 1 key | 0.0005    | -0.0234  | -0.0152  |
| Ciphered 10 keys | 0.0049   | 0.0583   | -0.0022  |
| Ciphered 20 keys | 0.0089   | -0.0066  | 0.004    |
| Ciphered 30 keys | -0.0098  | -0.0409  | -0.0492  |
| Ciphered 40 keys | 0.0057   | 0.0038   | 0.01     |
| Ciphered 50 keys | 0.0152   | 0.0101   | -0.0138  |

### Table 4. Evaluation of correlation metric in horizontal, vertical and diagonal for Sun image in Green channel.

|        | horizontal | vertical | diagonal |
|--------|------------|----------|----------|
| Original Sun | 0.9937     | 0.994    | 0.9892   |
| Ciphered 1 key | -0.0168   | 0.0224   | 0.0066   |
| Ciphered 10 keys | -0.0302  | -0.0072  | -0.0256  |
| Ciphered 20 keys | 0.04     | 0.007    | 0.0017   |
| Ciphered 30 keys | 0.0078   | -0.005   | -0.0172  |
| Ciphered 40 keys | -0.0231  | -0.0083  | -0.012   |
| Ciphered 50 keys | 0.0034   | 0.0149   | 0.0375   |

### Table 5. Evaluation of correlation metric in horizontal, vertical and diagonal for Sun image in Blue channel.

|        | horizontal | vertical | diagonal |
|--------|------------|----------|----------|
| Original Sun | 0.9754     | 0.9857   | 0.9551   |
| Ciphered 1 key | 0.0187    | -0.0038  | 0.0027   |
| Ciphered 10 keys | -0.00005 | 0.0074   | -0.0153  |
| Ciphered 20 keys | -0.018   | 0.0241   | -0.0095  |
| Ciphered 30 keys | 0.0084   | -0.0284  | -0.0367  |
| Ciphered 40 keys | 0.0112   | 0.0034   | 0.0178   |
| Ciphered 50 keys | 0.018    | -0.0031  | 0.0029   |
| Image name | Red_horizontal | Red_vertical | Red_diagonal |
|------------|----------------|--------------|--------------|
| Original Sun | ![Image](image1.png) | ![Image](image2.png) | ![Image](image3.png) |
| Ciphere d 1 key | ![Image](image4.png) | ![Image](image5.png) | ![Image](image6.png) |
| Ciphere d 10 keys | ![Image](image7.png) | ![Image](image8.png) | ![Image](image9.png) |
| Ciphere d 20 keys | ![Image](image10.png) | ![Image](image11.png) | ![Image](image12.png) |
| Ciphere d 30 keys | ![Image](image13.png) | ![Image](image14.png) | ![Image](image15.png) |
| Ciphere d 40 keys | ![Image](image16.png) | ![Image](image17.png) | ![Image](image18.png) |
| Ciphere d 50 keys | ![Image](image19.png) | ![Image](image20.png) | ![Image](image21.png) |

Figure 6. Correlation in Red channel of original and encrypted(Sun) image with different number keys
| Image name | Green_horizontal | Green_vertical | Green_diagonal |
|-----------|------------------|----------------|----------------|
| Original Sun | ![Image](Original_Sun.png) | ![Image](Original_Sun.png) | ![Image](Original_Sun.png) |
| Ciphere d 1 key | ![Image](Ciphere_d_1_key.png) | ![Image](Ciphere_d_1_key.png) | ![Image](Ciphere_d_1_key.png) |
| Ciphere d 10 keys | ![Image](Ciphere_d_10_keys.png) | ![Image](Ciphere_d_10_keys.png) | ![Image](Ciphere_d_10_keys.png) |
| Ciphere d 20 keys | ![Image](Ciphere_d_20_keys.png) | ![Image](Ciphere_d_20_keys.png) | ![Image](Ciphere_d_20_keys.png) |
| Ciphere d 30 keys | ![Image](Ciphere_d_30_keys.png) | ![Image](Ciphere_d_30_keys.png) | ![Image](Ciphere_d_30_keys.png) |
| Ciphere d 40 keys | ![Image](Ciphere_d_40_keys.png) | ![Image](Ciphere_d_40_keys.png) | ![Image](Ciphere_d_40_keys.png) |
| Ciphere d 50 keys | ![Image](Ciphere_d_50_keys.png) | ![Image](Ciphere_d_50_keys.png) | ![Image](Ciphere_d_50_keys.png) |

Figure 7. Correlation in Green channel of original and encrypted (Sun) image with different number keys
**Conclusion:**

It is concluded that the standard encryption method which has been proposed to enhance image security and integrity which uses multiple keys proved that security is increased throughout using entropy, MSE, PSNR and correlation. Also it is concluded that the accuracy of transmitted image increases by using this approach which is lossless thus the decrypted image is the identical to the original image. At the end, the suggested MECC_AES will be perfect and appropriate to be used for securing image in a widespread range of visual applications in IoT.

As mentioned above, the limitation of this work is that it will be used in IoT technologies that use or have the ability to use AES algorithm for security, thus, there are IoT technologies that need lightweight security scheme. The future work is
using MECC-AES technique in parallel mode to enhance power consumption and using MECC with lightweight encryption algorithms.

Authors' declaration:
- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are mine ours. Besides, the Figures and images, which are not mine ours, have been given the permission for republication attached with the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee in University of Babylon.

Reference:
1. Vadivukarasi K, Krithiga S. Home security system using IOT. IJPAM. 2018;119(15):1863-8.
2. Medagliani P, Leguay J, Duda A, Rousseau F, Duquennoy S, Raza S, et al. Internet of Things Applications - From Research and Innovation to Market Deployment, The River Publishers, 2014, 287-313, Series in Communications. (hal-01073761).
3. Samaila MG, Neto M, Fernandes DA, Freire MM, Inácio PR. Challenges of securing Internet of Things devices: A survey. Secur Priv. 2018;1(2):e20.
4. Lackner G. A Comparison of Security in Wireless Network Standards with a Focus on Bluetooth, WiFi and WiMAX. Int. J Netw. Secur. 2013;15(6):420-36.
5. Loksai Pothineni GSSK, V R Venu Maadhow, A R S Yaswantb4, Dr.Manikandan K, Wi-Fi, WiMAX & WiGig: A Comparative Study, International Research Journal of Engineering and Technology (IRJET). Mar-2018 Volume: 05( Issue: 03): 2228
6. Lu D, Liu T. The application of IOT in medical system. In2011 IEEE International Symposium on IT in Medicine and Education 2011 Dec 9 ; 1: 272-275. IEEE.
7. Ciara RA, Simion CM. Industrial applications of image processing. Acta Universitatis Cibiniensis. 2014;64(1):17-21.
8. Lakshmi K, Gayathri S. Implementation of IoT with image processing in plant growth monitoring system. IJISR. 2017;6(2):80-3.
تحسين أمنية الصورة في انترنت الأشياء (IoT) باستخدام خاصية متعددة المفاتيح AES

خديجة جبر سالم الجنابي
سيف محمود خلف العلاك
ماجد جبار جواد الفهراوي
قسم علوم الحاسبات، كلية العلوم للبنات، جامعة بابل، الحلة، العراق.

الخلاصة:

الصورة هي معلومات رقمية مهمة تستخدم في العديد من تطبيقات إنترنت الأشياء (IoT) مثل النقل والرعاية الصحية والزراعة والتطبيقات العسكرية والمرفقات والحياة البحرية.. إلخ. كذلك تتميز الصورة بسمات مهمة جدًا مثل الحجم الكبير والارتباط القوي والتكرار الهائل وبالتالي تثير اهتمامها باستخدام معيار التشفير المتقدم (AES) بمفتاح واحد من خلال تقنيات اتصالات إنترنت الأشياء تجعله عرضة للتغيرات من التهديدات. مما عزز هذا العمل هو زيادة، أمنية الصورة المنقلة. لذلك اقترح هذا العمل خاصية متعددة المفاتيح AES (MECCAES) لتحسين الأمان للصورة المرسلة من خلال إنترنت الأشياء. يتم تقييم هذا النهج من خلال تطبيقه على صور RGB bmp وتحليل النتائج باستخدام المقاييس القياسية مثل الارتباط (correlation)، الاختلاف (entropy)، المدرج التكراري (histogram)، المقاييس نسبة الدورة للإشارة إلى الضوضاء (PSNR) ومتوسط مربع الخطأ (MES). تظهر نتائج التجارب أن الطريقة المترسحة تحقيق مستوى عالٍ من السرية كما أنها استخدمت بشكل فعال في مجالات واسعة من تشفير الصور في إنترنت الأشياء.

الكلمات المفتاحية: معيار التشفير المتقدم، غير متماثل، التشفير، انترنت الأشياء، تشفير منحنى الأفلاج المتعدد، الصور الملونة (RGB)، WiFi، WiMAX، MIMO، الأمان.