ISSN 2278-3091
Volume 9 No.2, March - April 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse143922020.pdf
https://doi.org/10.30534/ijatcse/2020/143922020

Secure and Lightweight Encryption Model for IoT Surveillance Camera

Mohammed Abbas Fadhil Al-Husainy[1] , Bassam Al-Shargabi [2]
[1,2] Department of Computer Science, Middle East University, Amman, Jordan
[1]mal-husainy@meu.edu.jo, dralhusainy@gmail.com,
[2]bshargabi@meu.edu.jo, bassam20_152@yahoo.com

ABSTRACT

We are witnessing the era of Internet of Things (IoT), where its applications such as smart cities and smart homes catch sensitive data gathered mostly by IoT surveillance cameras among other sensors or devices. Therefore, security and privacy protection is a key concern during transmitting such sensitive data across the IoT network to be processed and stored on Cloud. In this paper, we proposed a lightweight encryption model that complies with the limited resources of IoT devices in terms of process and memory. Also, the encryption model also provides a high level of security for the transmitted data through a constant change of the key used for encrypting of transmitted IoT data. In addition, the key size used to encrypt transmitted data in the proposed model is large enough which makes it hard to break by the attackers. The experimental results show outstanding results with an average of 170.7 ms of encryption time for a key size 80 bits where the key size is relatively large and with an average PSNR of 7.7 compared to other algorithms.

Key words : Data Encryption, Smart Cities, Edge Computing, Internet of Things, Security, Surveillance Camera

1. INTRODUCTION

The Internet of Things (IoT) is extending at a quick rate and is anticipated to grow more over the next few years which makes past innovation receptions look irrelevant. Expectations are that by 2025 there will be approximately over 25 Billion connected IoT devices [1]. As the expected that IoT devices are to be connecting every aspect of our life, such devices are range from any computing devices, mechanical and digital equipment, and objects with the capability to transferring and generating data across a network without the need for human or computer interactions such as Closed-circuit television (CCTV) cameras, vehicles, and home devices, through electronic tags, sensors, actuators [2], [3]. The IoT devices have the capabilities to collect data and convey this information along with other connected IoT devices, where these data can be stored, analyzed, and can be exploited by any user with a mobile device. Therefore, IoT devices have the capacity to detect, gather, dissect, screen, and convey information on a monstrous scale. These objects or devices additionally increment the measure of information that's gathered, prepared, put away, and moved between IoT devices and Cloud.

The IoT devices have their very own unique digital identities all together for these devices to communicate in a consistent manner. Nowadays, with the significant progression in IoT empowering innovations starting with RFID, connectivity, Cloud, and Big data analytics have been presented in numerous fields and applications such as smart homes, smart cities, water, power, traffic control, surveillance.

The introduction of IoT devices in-home or street surveillance cameras helped to create safer cities, and homes by permitting both private and public businesses to securely and remotely monitor facilities and public spaces in real-time with smart security and surveillance IoT enabled solutions. Nevertheless, as it's conceivable to get data of IoT surveillance cameras or devices when this data are sent to the Cloud, such images of people's faces, the vehicle plate numbers among the checking zone, the protection of such data is then risked through the surveillance cameras in open regions. Also, if theses cameras are installed in private places such homes are at risk against the reconnaissance recordings as well. Moreover, such IoT devices can be exploited for real-time monitoring of the property, and engagements and behavior of people inside their homes [4]. Unfortunately, security and privacy dangers have not gotten as much consideration with the IoT industry. Since IoT surveillance cameras will commonly be installed somewhere in systems, they are going to be easily targeted and may turn into the weakest point for breaking into a safe information technology framework [5].

Nevertheless, processing and analyzing the tremendous amount of such delicate data created by IoT surveillance cameras is faced with many security and privacy challenges and concerns with the presence of unauthorized users who tries to reach this sensitive data [6], [7],[8]. We are on the verge of smart cities through realizing IoT application with the existence of cloud computing for storing and analyzing data generated by IoT. On the other hand, there are a few issues in cloud-based IoT applications, some of these issues relates to the latency, and security, which can be tackled with existing of fog computing and edge computing paradigms [9],[10],[11]. Due to the major distinction between fog and cloud computing makes the security issue for cloud services is not appropriate for fog computing services that are accessible for users or IoT surveillance cameras [6],[12],[13].

A variety of cryptographic methods can be used to deal with the security problem; these methods are not suitable for constrained IoT devices. The IoT devices have limited resources in terms of CPU and memory, this means they often transmit data only when there is something important was triggered or sensed. As a solution the security issue is offloaded to the fog or the edge nodes within IoT networks, which security and data analysis directly handled by the edge of the network. Besides, in broker-based architecture where the IoT devices or cameras can be set to be an MQTT publisher, the cloud or data centers that need to receive these camera data can be set as the MQTT subscriber, and any intermediary node can be set as a broker to relay the data between the publisher and the subscriber. As a result, the broker-based architecture must guarantee the confidentiality of the publisher and subscribers. Furthermore, some authentication methods have taken into consideration the IoT constraint characteristics through the use of a lightweight authentication protocol by the use of a public key encryption infrastructure [14]. Although, homomorphic encryption has newly used because of its exceptional strength and capability on computations of encrypted data, but it is impractical to be used within IoT networks due its computational cost [6],[15],[16]. Therefore, any cryptography method to use for the encryption of sensitive data collected by IoT devices such as surveillance images should consider the resources available within IoT devices. Moreover, it must be adequately fast to satisfy the needs of real-time surveillance images applications and also should be convenient and efficient techniques to provide security for such data transferred IoT network [17], [18],19].

The major contribution of this paper is uncovering a fast enough algorithm to encrypt and decrypt real-time streaming of images and video in terms of processing time and memory space as we know the limitations of IoT surveillance cameras. The proposed model in this paper achieves a high level of security for the transmitted data through a constant change of the key used for encrypting of transmitting data. In addition, the key size used to encrypt transmitted data in the proposed model is large enough which makes it hard to break by the attackers.

The rest of the paper organized as follows: Section 2 outlines the most recent related work. The elaboration of the methodology of the proposed model is presented in section 3. Testing and evaluation of the proposed model with comparisons to other models are presented in section 4. Finally, the conclusion about this paper is drawn in Section 5.

2. RELATED WORK

Many approaches have been proposed to tackle the issue of securing the data transmission within IoT network with consideration of the limited resources of IoT devices.

A lightweight based on extracting Region of Interest (RoI) encryption approach that employs binary sequences for each RoI within the video, where each RoI block of the video as the initialization of an 8-layer of Layered Cellular Automata (|LCA) [20]. Given that each layer of the LCA can be considered as an arrangement of a progression of 1D CAs, at this point an arbitrarily selecting the reversible Elementary Cellular Automata (ECA) standards for LCA training and after that acquire the LCA's final state that can be transformed to a pixel grid of the encrypted image. Achieving enhanced confusion of encrypted image and for each layer, they applied half transformation just as in neighbour layers. In addition, they applied an irregular shift change on each 1D CA. The authentication process within this approach allows users with an on-request way to retrieve the surveillance videos. Since LCA is a profoundly parallel system, LCA-based encryption with the basic principles and transformations is naturally efficient and simple to be executed. Also, every RoI is sliced into a series of binary blocks and every one of the blocks are synchronously encrypted, Surely, the approach complies with the constant operation of surveillance cameras and satisfies the secrecy of data generated by IoT cameras but the computational cost for such approach is considered a major problem regarding the resources available within IoT devices. An approach was presented in [21] to the leaking of information over IP traffic of IoT surveillance cameras, even if the payload was encrypted. They have tried to examine the leakage through processing the metadata of network traffic such as packet size, and video bandwidth. Their examination using metadata is conceivable regardless of whether a regular encryption strategy is used for data streaming of IoT surveillance cameras. They have come to a result that there is leakage information observed in the camera's data.

A Chaos-Based cryptosystem was proposed in [22] for encrypting the streaming of surveillance cameras. They used a case study on health care with the mobile camera used to monitor a diabetic patient and sending encrypted real-time videos about the patient to a specific data center. They used chaotic map algorithm Arnold cat map, their experiments show that it takes 0.0071 seconds to encrypt and decrypt a video. Although the result shows impressive speed, but chaos-based methods it is not yet mature to use for image and video encryption as compared to plain text in term of speed. Another lightweight approach encrypts real-time audio-visual based on chaotic map, Chebyshev map, secure hash to secure transmission of audio-visual hearing-aid signals to Cloud but their approach requires much power and high computation .

In [23], the authors developed a system for encrypting multimedia medical data by exploiting the Feistel Encryption Scheme, an Advanced Encryption Standard (AES) along with a genetic algorithm to comply with the computation time by utilizing the GPU. This system was assessed on IoT multimedia medical data to benchmark the encryption approaches, for example, MARS, RC6, 3-DES, DES, and Blowfish regard computational running time and throughput for both encryption and decryption just along with avalanche effect. Their outcomes demonstrated that their system has the least computation time and most noteworthy throughput for

both encryption and decryption procedures and most elevated avalanche effect with contrasted with the current encryption methods, but their approach does not go well with the encryption and decryption of real-time streaming of IoT surveillance cameras.

A security approach for transmitting media packet routing with IoT network was applied to the smart city scenario presented in [24]. This security approach relies on merging Identity, Route and Location (IRL), and Privacy Algorithm for routing at IoT sensor level along with intermediate nodes, and for the entire security of IoT network they used and adapted Practical Algorithm for Data Security (PADS) [25] to be appropriate to the original standard of video compression (HEVC) for media files transfer .This approach was only used on the concept of request and response communication model between the IoT sensor and the IoT network which is not applied on a real-time streaming of sensor data images.

An authentication schema based on public key encryption [14], where it was used to develop a mutual lightweight authentication protocol to be appropriate for IoT devices with low power networks due to the reason that this encryption schema does not involve high computations and calculations. The mutual authentication protocol is compared to other approaches such as Elliptic Curve Cryptography (ECC), Algebraic Erase (AE), and NTRU in term of computation time, and ciphertext size the authors also claims that the schema work better without a trust third party to manage setup phase between different IoT devices, but they never tested their schema with real-time streaming of images due to the reason of high computation time and the distribution of private key.

An instant encrypted sensitive data transmission between any two devices within a network of IoT devices is presented in [26]. They applied their approach on smart home scenario, where there is a key generation center to generate private keys of all home IoT devices (sensor or actuators) network based on the identity of each IoT devices within home IoT network, as a matter of fact this way increases the computation time for their approach. Thereafter, all of the IoT devices inside the home are collecting data some of these data are sensitive data that needed to be transmitted securely by encrypting data using the private key.

An approach in [27] that is based on the use of IoT multi-view surveillance cameras to enhance processing capability to be exploited to examine and identify the keyframes of the video stream and discard irrelevant redundant data. Therefore, only a small amount of data to be transmitted, because keyframes only needed to be encrypted through the use of lightweight probabilistic keyframes encryption Algorithm. The encryption is based on the use of a 2D chaotic map to generate PRNG for image encryption alongside with an RBG image encryption algorithm for keyframe, on the other hand, the secret key is used to decrypt transmitted data to retrieve the original keyframe.

Aggregate-Signcryption with decryption Signcryption that carries out data signing and encryption is proposed in [28]. They claim that Signcryption fairness with a lightweight security method for surveillance camera IoT that relies on EC used to protect sensitive data captured by the cameras and secure transmission of such the data from multiple cameras to other monitoring data centers. Another approach motivated by the use of multi-receiver Signcryption in [29], where there is a generation center for the private key to distribute the pseudo partial private key through public channels to users throughout the key extract algorithm, and the designated user uses it to decrypt the transmitted data.

3. METHODOLOGY

A large number of cameras are distributed and installed by governments to monitor the streets and locations in cities and within institutions. These cameras are continuously sending a stream of images and videos to a dedicated database on the server. Some of these cameras are installed around highly sensitive areas such as embassies, military institutions, and ministries. Therefore, it becomes necessary to send videos and images taken by these cameras in unrecognizable form or in an encrypted form.

To resolve this problem, a proposed model has been developed that involved a lightweight encryption algorithm that is satisfying the following goals:

1) Fast enough to encrypt and decrypt live images and video.
2) Does not require a lot of resources (memory space and processor time).
3) Achieves a high level of security for the transmitted data. This is achieved through:
    a) Periodical change of the key used.
    b) Use strong and large keys.

Figure 1 shows the general framework of the suggested model. The main components used in this model are listed here:

- **IoT Cameras:** a set of IoT cameras used to monitor different places.
- **Central Server:** a computer system connected to edge servers , that implements a protocol used to determine the keys that will be used by IoT cameras to encrypt video/images.
- **Edge Server:** a computer system that distributes different keys on connected IoT cameras and receives encrypted videos/images taken by these cameras.
- **Key_i:** a secret key used by IoT cameras to encrypt videos/images taken by these cameras. The same key is used by the central server to decrypt the received videos/images.
- **Encryption algorithm:** a lightweight algorithm used by IoT cameras to encrypt videos/images using a key received from the edge server, where this key is periodically changed by the edge server .
- **Decryption algorithm:** a lightweight algorithm used by the central server to decrypt the received videos/images from the IoT cameras using the predetermined key.

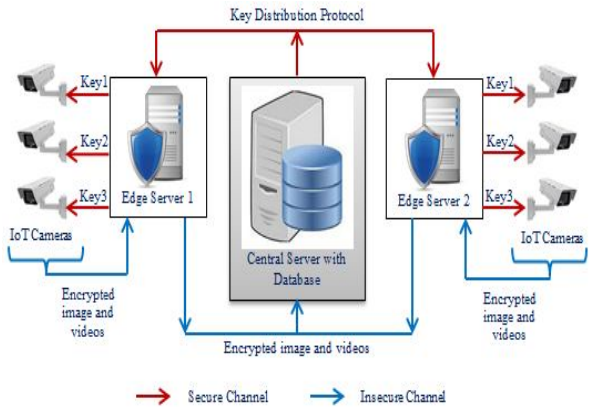


Figure 1: General framework of the proposed model

The implementation of the proposed model involves a set of stages that can be summarized below:

Stage 1: The central server sends a table to each edge server using secure protocol. This table contains information used by IoT cameras as a secret key to encrypt videos/images. The values in each row of the table are created randomly by the server and implemented starting from the time specified in that row. The period between times in each row of the table, for each edge server , depends on the sensitivity level of places monitored by IoT cameras. Table 1 shows an example of that table; each value in each row represents a portion of the secret key with its size in byte. Where CameraID is a number used to identify each camera, Time is a number that represents the start time for the key execution, XORValue is a number used by the encryption algorithm to perform XOR logical operation on the bytes of videos/images, ROTATEValue is a number used by the encryption algorithm to perform ROTATE logical operation on the bytes of videos/images.

Stage 2: Depending on the time determined in each row of the table, the edge server sends the portions of the secret key to the specified camera.

Stage 3: Each IoT camera uses the portions of the key received from the edge server to encrypt the videos/images taken. The implementation of the encryption algorithm used in the proposed model contains the following operations:

A) Substitution (XOR) operation: The implementation of the XOR operation is applied on each byte in the source data (starting with Index = 0 to (2^{32}-1) and repeat) using equations (1), (2) and (3), where CurrentTime variable represents the global time starting from Time and increasing periodically, Value1 represents the initial 2-bytes value results from performing XORing logical operation between three 2-bytes values and Value2 represents 1-byte value after performing XORing logical operation between two halves of Value1. The resulted Value2 has different values for each byte of Data.

$$\text{Value1} = \boldsymbol{XOR}(\text{CameraID}, \text{XORValue}, \text{CurrentTime}) \quad (1)$$

$$\text{Value2} = \boldsymbol{XOR}(\text{LeftHalf}(\text{Value1}), \text{RightHalf}(\text{Value1})) \quad (2)$$

$$\text{Data}_{\text{new}}[Index] = \text{Data}_{\text{old}}[Index] \; \boldsymbol{XOR} \; \text{Value2} \quad (3)$$

Table 1: An Example of the table used by edge server .

Key Portion (Size in byte)			
CameraID (2 bytes)	**Time (2 bytes)**	**XORValue (2 byte)**	**ROTATEValue (4 bytes)**
34900	12:35	17780	678943
34900	01:00	200	8765314
34900	01:45	9067	981183
:	:	:	:
2509	23:35	12987	986701
2509	23:45	45001	9865010
:	:	:	:

B)Transposition (ROTATE) operation: The implementation of the ROTATE operation is applied to each byte in the source data (starting with Index = 0 to (2^{32}-1) and repeat). At the first, the value of the ConditionValue variable is calculated using equation (4) for the byte of data at Index.

$$\text{Data}_{\text{new}}[Index] = \text{Data}_{\text{old}}[Index] \; \boldsymbol{XOR} \; \text{Value2} \quad (4)$$

Then, if the values of the two halves of ConditionValue (each half represents 2 bytes) are different, then the equation (5) is applied. Otherwise, the byte value in Data in Index is not changed.

$$\text{Data}_{\text{new}}[Index] = \boldsymbol{RotateRight}(\text{Data}_{\text{old}}[Index], 4) \; \boldsymbol{OR} \; \boldsymbol{RotateLeft}(\text{Data}_{\text{old}}[Index], 4) \quad (5)$$

The change in the byte value in the Data is completely different from byte to byte because it is depending on the value of ROTATEValue and the *Index*.

Stage 4: The encrypted bytes of the source data of the videos/images that are resulted from Stage 3 are sent from the IoT camera to the edge server and then to the central server . After the central server received the encrypted data from the IoT camera, the central server implements the same two operations (substitution and transposition) that were implemented in the encryption phase but in reverse order to decrypt the received encrypted data and recover the source data.

4. TESTING, EVALUATION & DISCUSSION

Three main objectives have been set for the proposed cryptographic model. In this section, we will assume that the source data were images just to test the proposed encryption algorithm. First, the encryption algorithm used must be lightweight enough to be executed on the IoT camera processor. This means that the resources needed (processor time and memory used) as little as possible. Second, the key used in the encryption algorithm must be constantly changed to provide a high level of protection for the data transferred. Third, the size of the key (in bits) used in the encryption algorithm must be as large as possible to make it hard to break by the attackers. Fourth, the encryption algorithm must cause the greatest proportion of distortion in the data. This effect can be tested numerically by calculating the Peak Signal to Noise Ratio (PSNR) of the encrypted image and statistically by comparing the histogram of the source and encrypted images.

The proposed cryptographic model has been implemented using a C# programming language in Visual Studio 2011. And using a computer system has Intel (Core-i3) 2.40 GHz processor and 4.0 GB memory.

In this section, the objectives of the proposed cryptographic model are analyzed and tested. Evaluation of each test is performed by making comparisons with known algorithms. A discussion of the results helped to draw some conclusions.

4.1 Speed of the Encryption Algorithm

Most encryption algorithms try to use complex mathematical and logical formulas to encrypt data. This will make the algorithm execution time too long, but at the same time it adds more difficulties for the attackers. In addition, the use of mathematical formulas takes more execution time than the use of logical operations.

Admittedly, the strength of any cryptographic system does not mainly depend on the algorithm but on the key used. Therefore, all the operations used in the proposed encryption algorithm are logical operations (XOR and ROTATE operations), this will helps to reduce the time required to complete the encryption operation.

Regarding the memory used in the encryption algorithm, the XOR and ROTATE operations are performed on each byte of data separately; this means that the processor does not need a lot of memory to perform these operations.

In order to test the speed of the proposed algorithm, many color images were encrypted using both the proposed encryption algorithm and some known encryption algorithms such as DES (Data Encryption Standard), AES (Advanced Encryption Standard). Figure 2 shows some of the images used in the experiments and Table 2 shows the encryption time for these images.

Bank (284×177) **Palace (275×183)** **ATM (300×168)**

Figure 2: Some images used in the experiments**.**

Table 2: The encryption time of the proposed, DES and AES algorithms

Image	Encryption time (msec)		
	Proposed algorithm	DES	AES
Bank	210	367	405
Palace	134	271	276
ATM	168	314	266

4.2 The Key used in the Encryption Algorithm

Because of the confidentiality of videos/images transmitted by IoT cameras, this makes it necessary to change the key used by these cameras to encrypt the transmitted data continuously. This is really implemented in the proposed algorithm, where the edge server periodically receives a table that contains randomly generated keys from the central server . Each key in that table is sent, by the edge server , based on the implementation time of that key. As mentioned before, the period of change of the key used by each IoT camera depends on the sensitivity level of places monitored by that camera, where it decreases as the sensitivity of the place increases.

4.3 The Size of the Key used in the Encryption Algorithm

To achieve a high level of protection for the transmitted encrypted data, it is necessary to use the largest possible size of the key used. The key size used in encryption techniques is measured by the number of bits in that key. The size (in byte) of each portion of the key used in the proposed encryption algorithm is mentioned in Table 1. The total size, in bits (where 1 byte = 8 bits), of the key used in the proposed algorithm is 80 bits and it is calculated using equation (6).

$$\begin{aligned}&\text{TotalSizeOfKey} = \\ &\quad (\text{SizeOf(CameraID)} + \text{SizeOf(Time)} + \\ &\text{SizeOf(XORValue)} + \text{SizeOf(ROTATEValue)})\end{aligned} \tag{6}$$

Table 3 shows the size of the keys (in bits) used in the proposed algorithm and other known encryption algorithms. It is clear from the table that the proposed algorithm uses a relatively large key size; this makes the key hard enough to break by attackers.

Table 3: The size of the keys used in the proposed algorithm and other known encryption algorithms [22], [29], and [30].

Encryption Algorithm	Key Size in bits
Proposed algorithm	80
DES	56
AES	128, 192 or 256
Twofish	128, 192, or 256
Blowfish	32 to 448
RC4	40 to 2048
[20]	10 to 172
[27]	90

4.4 The Proportion of Distortion in the Encrypted Data

The calculated PSNR values of the encrypted images for the corresponding source images in Figure 2 are mentioned in Table 2. The PSNR is calculated using equation (7) and (8). The PSNR values in experiments indicate that the proposed algorithm achieved a great proportion of distortion in the encrypted data as shown in Table 4.. And it is relatively close to the PSNR of known encryption algorithms.

$$NMAE = \frac{\sum_{k=0}^{I_{Size}-1} |I(k)-E(k)|}{I_{Size}} \times 100 \quad (6)$$

$$PSNR_{db} = 10.\log_{10}\left(\frac{Max_I^2}{NMAE}\right) \quad (7)$$

Where: Max_I is the maximum possible pixel value of the image $\boldsymbol{I}$. And *db* refers to a decibel.

Table 4: The PSNR values of the encrypted images.

Image	PSNR (db)		
	Proposed algorithm	**DES**	**AES**
Bank	8.00	8.01	8.05
Palace	7.13	7.15	7.15
ATM	8.11	6.79	8.09

The distortion effect in the image that is appeared when implementing the proposed algorithm can be tested visually by comparing the source and encrypted images. Figure 3 shows the encrypted images of the source images in Figure 2.

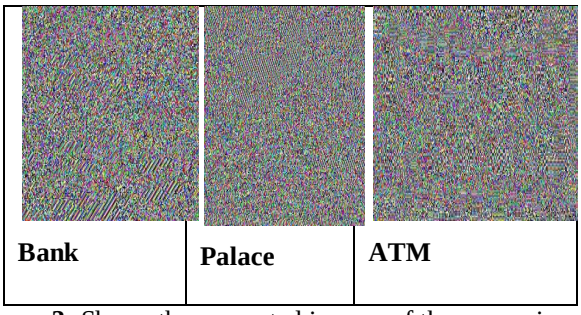
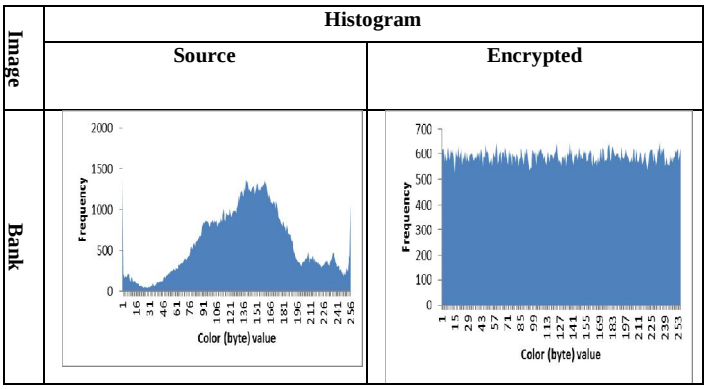


Figure 3: Shows the encrypted images of the source images in Figure 2

Also, the histograms of the values of bytes for the source and encrypted image (data) are given in Figure 4. The histogram of the encrypted images shows that the proposed algorithm succeeded to make a great change in the statistical distribution of the values of the bytes in the encrypted image.



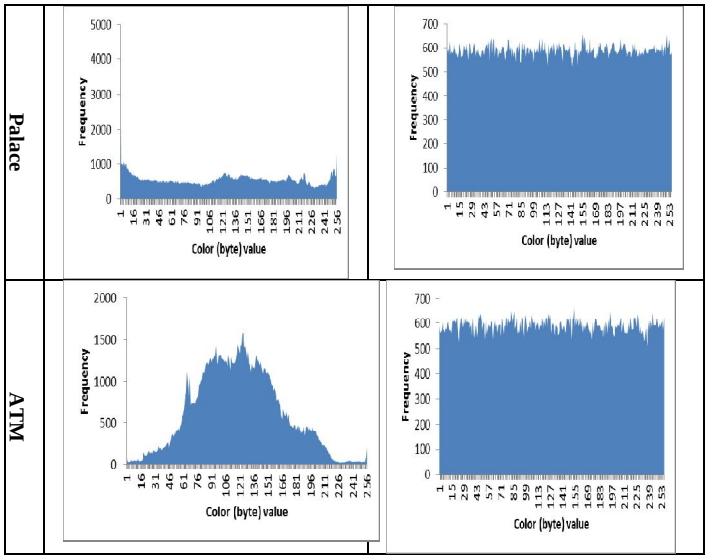


Figure 4: A histogram of the source and encrypted images used in the experiments.

5. CONCLUSION

A new encryption model was proposed to secure data transmission for IoT surveillance cameras because these data are considered sensitive as these cameras installed in highly sensitive areas such as embassies, military institutions, and ministries. Furthermore, the proposed encryption model complies with the limitation of resources of IoT devices in terms of processing time, memory space and power consumption. Experimental results reveal that our proposed model achieves less processing time and memory spaces compared to other approaches, while ensures a high-level of security of transmitted data through a constant change of the key used for encrypting of transmitted IoT data. Besides, the key size used to encrypt transmitted data in the proposed model is large enough which makes it hard to break by the attackers.

ACKNOWLEDGMENT

The authors are grateful to the Middle East University, Amman, Jordan for the financial support granted to cover the publication fee of this research article.

REFERENCES

1. J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, Internet of Things (IoT): A **vision, architectural elements, and future directions,** *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013. https://doi.org/10.1016/j.future.2013.01.010
2. J. Lloret, I. Bosch, S. Sendra, and A. Serrano, **A wireless sensor network for vineyard monitoring that uses image processing**, *Sensors*, vol. 11, no. 6, pp. 6165–6196, 2011. https://doi.org/10.3390/s110606165

3. I. Mehmood, M. Sajjad, W. Ejaz, and S. W. Baik, **Saliency-directed prioritization of visual data in wireless surveillance networks**, *Information Fusion*, vol. 24, pp. 16–30, 2015. https://doi.org/10.1016/j.inffus.2014.07.002
4. T. Yu, V. Sekar, S. Seshan, Y. Agarwal, and C. Xu, **Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the internet-of-thing**s, in *Proceedings of the 14th ACM Workshop on Hot Topics in Networks*, 2015, pp. 1–7.
5. [M.-H. Maras, **Internet of Things: security and privacy implications**, *International Data Privacy Law*, vol. 5, no. 2, p. 99, 2015. https://doi.org/10.1093/idpl/ipv004
6. F. Al-Turjman, H. Zahmatkesh, and R. Shahroze, **An overview of security and privacy in smart cities' IoT communications**, *Transactions on Emerging Telecommunications Technologies*, no. June, pp. 1–19, 2019, doi: 10.1002/ett.3677.
7. K. Zhang, J. Ni, K. Yang, X. Liang, J. Ren, and X. S. Shen, **Security and privacy in smart city applications: Challenges and solutions**, *IEEE Communications Magazine*, vol. 55, no. 1, pp. 122–129, 2017. https://doi.org/10.1109/MCOM.2017.1600267CM
8. Y. He, F. R. Yu, N. Zhao, V. C. M. Leung, and H. Yin, **Software-defined networks with mobile edge computing and caching for smart cities: A big data deep reinforcement learning approach**, *IEEE Communications Magazine*, vol. 55, no. 12, pp. 31–37, 2017. https://doi.org/10.1109/MCOM.2017.1700246
9. Z. Maamar, T. Baker, M. Sellami, M. Asim, E. Ugljanin, and N. Faci, **Cloud vs edge: Who serves the Internet□of□Things better?**, *Internet Technology Letters*, vol. 1, no. 5, p. e66, 2018.
10. R. Mahmud, R. Kotagiri, and R. Buyya, **Fog computing: A taxonomy, survey and future directions**, in *Internet of everything*, Springer, 2018, pp. 103–130.
11. B. Al-Shargabi and O. Sabri, **Internet of Things: An exploration study of opportunities and challenges**, in *Proceedings - 2017 International Conference on Engineering and MIS, ICEMIS 2017*, 2018, vol. 2018-Janua, pp. 1–4, doi: 10.1109/ICEMIS.2017.8273047.
12. B. Al-Shargabi, S. Al-Jawarneh, and S. M. A. Hayajneh, **A cloudlet based security and trust model for e-government web services,** *Journal of Theoretical and Applied Information Technology*, vol. 98, no. 1, pp. 27–37, 2020.
13. N. Li, D. Liu, and S. Nepal, **Lightweight mutual authentication for IoT and its applications**, *IEEE Transactions on Sustainable Computing*, vol. 2, no. 4, pp. 359–370, 2017. https://doi.org/10.1109/TSUSC.2017.2716953
14. B. Al-Shargabi and O. Sabri, **A study of Adopting Cloud Computing from Enterprise Perspective using Delone and Mclean IS Success Model,** *International Journal of Computer Science and Information Security*, vol. 14, p. 32, 2016.
15. D. Chialva and A. Dooms, **Conditionals in homomorphic encryption and machine learning applications**, *arXiv preprint arXiv:1810.12380*, 2018.
16. H. Abualese, T. Al-Rousan, and B. Al-Shargabi, **A New Trust Framework for E-Government in Cloud of Things**, *International Journal of Electronics and Telecommunications*, vol. 65, no. 3, pp. 397–405, 2019, doi: 10.24425/ijet.2019.129791.
17. [S. R. Masadeh, H. A. Al-Sewadi, and M. A. F. Al-Husainy, **Embedded key cryptosystem for cloud computing applications,** in *Proceedings of the 2nd International Conference on Future Networks and Distributed Systems*, 2018, pp. 1–7. https://doi.org/10.1145/3231053.3231078
18. X. Zhang, S. H. Seo, and C. Wang, **A Lightweight Encryption Method for Privacy Protection in Surveillance Videos**, *IEEE Access*, vol. 6, pp. 18074–18087, 2018, doi: 10.1109/ACCESS.2018.2820724.
19. C. Wampler, S. Uluagac, and R. Beyah, **Information leakage in encrypted ip video traffic**, in *2015 IEEE Global Communications Conference (GLOBECOM)*, 2015, pp. 1–7.
20. N. Mekki, M. Hamdi, T. Aguili, and T. H. Kim, **A real-time chaotic encryption for multimedia data and application to secure surveillance framework for IoT system**, *Proceedings - 2018 International Conference on Advanced Communication Technologies and Networking, CommNet 2018*, pp. 1–10, 2018, doi: 10.1109/COMMNET.2018.8360271.
21. A. Adeel, J. Ahmad, and A. Hussain, **Real-Time Lightweight Chaotic Encryption for 5G IoT Enabled Lip-Reading Driven Secure Hearing-Aid**, pp. 1–14, 2018.
22. S. Aljawarneh, M. B. Yassein, and W. A. Talafha, **A resource-efficient encryption algorithm for multimedia big data**, *Multimedia Tools and Applications*, vol. 76, no. 21, pp. 22703–22724, 2017, doi: 10.1007/s11042-016-4333-y.
23. V. A. Memos, K. E. Psannis, Y. Ishibashi, B.-G. Kim, and B. B. Gupta, **An efficient algorithm for media-based surveillance system (EAMSuS) in IoT smart city framework**, *Future Generation Computer Systems*, vol. 83, pp. 619–628, 2018.
24. R. A. Shaikh, H. Jameel, B. J. d'Auriol, H. Lee, S. Lee, and Y.-J. Song, **Achieving network level privacy in wireless sensor networks**, *Sensors*, vol. 10, no. 3, pp. 1447–1472, 2010.
25. D. Christin, A. Reinhardt, P. S. Mogre, and R. Steinmetz, **Wireless sensor networks and the internet of things: selected challenges**, *Proceedings of the 8th GI/ITG KuVS Fachgespräch Drahtlose sensornetze*, pp. 31–34, 2009.
26. C. Wang, J. Shen, Q. Liu, Y. Ren, and T. Li, A **Novel Security Scheme Based on Instant Encrypted Transmission for Internet of Things,** *Security and Communication Networks*, vol. 2018, 2018, doi: 10.1155/2018/3680851.
27. K. Muhammad, R. Hamza, J. Ahmad, J. Lloret, H. Wang, and S. W. Baik, **Secure surveillance framework for**

IoT systems using probabilistic image encryption, *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3679–3689, 2018, doi: 10.1109/TII.2018.2791944.

28. S. Ullah, L. Marcenaro, and B. Rinner, **Secure smart cameras by aggregate-signcryption with decryption fairness for multi-receiver IoT applications**, *Sensors (Switzerland)*, vol. 19, no. 2, 2019, doi: 10.3390/s19020327.
29. L. Pang, M. Kou, M. Wei, and H. Li, **Anonymous Certificateless Multi-Receiver Signcryption Scheme Without Secure Channel**, *IEEE Access*, vol. 7, pp. 84091–84106, 2019, doi: 10.1109/ACCESS.2019.2924654.
30. M. A. F. Al-Husainy and H. A. A. Al-Sewadi , **Implementing Binary Search Tree Concept for Image Cryptography**, *International Journal of Advanced Science and Technology*, Vol. 130, pp. 21- 32, 2019. https://doi.org/10.33832/ijast.2019.130.03