A scalable and secure model for surveillance cameras in resource constrained IoT systems

Unsub Zia[1],Bryan Scotney[1],Mark McCartney[1],Jorge Martinez[1],Mamun AbuTair[1] and Ali Sajjad[2]
[1]School of Computing, Ulster University, Northern Ireland, UK
[2]Applied Research, British Telecom, Ipswich, UK
Email: zia-smu,bw.scotney,m.mccartney,j.martinez-carracedo,m.abu-tair@ulster.ac.uk,
ali.sajjad@bt.com

ABSTRACT

In wireless multimedia surveillance networks (WMSNs), the sensors generate continuous data streams which are saved on cloud servers for processing and future use. Large-scale security and monitoring decisions are based on the video data fed back to the cloud by the visual sensors. Internet of Things (IoT) networks are deployed in certain resource constrained scenarios where edge computing is not available for data analytics and security considering the IoT devices have also limited resources such as memory, power and processing capabilities. In this paper, the problems of limited resources and data security are addressed by the proposal of a secure model for video surveillance systems working in resource constrained IoT assisted scenarios. The suggested approach comprises of the following four main stages: i) save only those video frames in which any activity is detected, ii) encrypt the saved frames for secure transmission, iii) synchronize the encrypted frames between cloud and sensor node and iv) remove the transmitted frames from sensor and decrypt the stored data on cloud. The performance of the encryption process for resource constrained devices is analyzed on different types of sensor nodes during experimentation. The results prove that the proposed method automates and speeds up the process of live video data extraction and occupies less space on cloud when compared to the conventional approach for saving surveillance videos. Furthermore, adding encryption to video frames ensures integrity of video data during their journey from sensor to the cloud.

CCS Concepts

• Security and privacy → Database and storage security → Management and querying of encrypted data

Keywords

IoT; security; scalability; resource constrained video surveillance.

1. INTRODUCTION

The development of embedded devices with computing capabilities has resulted in the emergence of the internet of things (IoT), where smart sensors gather data from their surroundings and communicate it over the internet for processing, storage and large-scale decision making [1]. These ubiquitous intelligent IoT devices provide services in different fields of life ranging from healthcare, industry, transport, military and academia to the everyday home chores [2]. Wireless multimedia surveillance networks (WMSNs) can be visualized as subnets of IoT-assisted networks, comprising of visual sensors that monitor and record activities in their surroundings continuously, thereby generating extensive amount of visual data [3]. In a recently conducted survey it is estimated that there will be around 45 billion cameras by 2022 collecting the information from their surroundings [4]. The British Security Industry Authority (BSIA) published a thorough report in 2015 that approximated the number of closed-circuit television cameras in the United Kingdom (UK) to be around 5.9 million, out of which 750,000 cameras are in sensitive areas like hospitals, schools and places of worship [5]. The BSIA report also estimated that there is one camera for every 14 people in UK. The rapidly increasing trend of security cameras can lead to serious problems including privacy breach and storage difficulties for preserving the large amount of video data generated by the smart devices. The IoT sensor nodes are designed to work in remote conditions, therefore they are equipped with limited resources such as memory, power and processing capabilities.

The recording of continuous video streams utilizes extra processing resources and then sending this large amount of data over the network occupies additional bandwidth which can lead towards data congestion and resource scarcity. The research community has consensus on developing intelligent ways to reduce the excessive use of resources like storage space. For instance, storing large amount of redundant visual data on cloud would not be a rational approach. Several schemes are proposed to somehow improve the methods of storing humungous amount of video data generated by IoT devices. A CouchDB data server-based storage infrastructure is proposed for heterogeneous multimedia data in IoT [6]. A data management system for massive IoT data is introduced using NoSQL for increased performance and availability [7]. There are several other solutions that suggest improvement in data storage systems for large volume of IoT data [8-10], but alongside enhancing the data storage systems, it is also important to reduce the amount of redundant data that is injected to the cloud for storage. Some important pre-requisites for ideal working of WMSN-based monitoring systems include efficient resource utilization and assurance of data security [11]. It is crucial to preserve the integrity of the data generated by the IoT sensors as the large-scale decisions solely rely on them.

In this paper, the previously outlined shortcomings are addressed through a combined solution for video surveillance systems that would only record essential information from the video stream and ensure safe transmission of video data to the destination for storage. The proposed model design allows the visual sensor nodes to pipe the live stream video to the display system but save only those frames that contain any difference in information between two consecutive frames using OpenCV tool. The frames are saved with respective date and time stamp on them and secured using encryption for transmission. The results and analysis prove that the proposed model provides a secure solution with efficient resource utilization for surveillance cameras with limited resources, such as processing and memory capabilities. The suggested model is also scalable in nature hence the best fit for scenarios with more than one surveillance camera recording video streams. The rest of the paper comprises the following sections: Section II summarizes the related work, Section III presents the proposed model, Section IV illustrates experiments and results, and Section V concludes the paper and future work.

2. RELATED WORK

WSMN-based networks are often confronted with resource limitations due to inherent characteristics of IoT devices; simultaneously they also face security vulnerabilities when the video data is transmitted wirelessly on the public channels. For efficient resource allocation in cloud based live video streaming applications, a game winning strategy was introduced in [12] that uses Nim game approach to pair the users and cloud resources. The results show that the suggested approach is quite efficient in terms of resource allocations to users, but the authors did not consider any security measures to safeguard the live stream video hosted at the cloud server. During the transmission of video data from visual sensor to the cloud, it can be manipulated by man in the middle attacks and the data sent to cloud might not be in its genuine form.

An efficient encryption method was introduced for video streaming which relies on low-cost FPGA devices to encrypt only half of the most significant video data bits [13]. The authors claimed that the proposed method reduces the resource utilization as only half of the most significant bits were encrypted. This encryption algorithm might perform faster and utilize less processing resources, but in terms of memory footprint, the cipher text generated for continuous video stream would be very large in size, which would not be viable to transmit and store. A study on securing video data streams using permutation-based encryption was conducted for improved efficiency and security [14]. The comparison of the proposed scheme was done with Hénon chaotic map system and phase modulation approach. The results indicated that the proposed method takes almost the same time during encryption as Hénon chaotic systems, and thereby not increasing the efficiency for live stream videos. A design of smart home security systems for intrusion detection was done using object recognition and PIR sensors [15]. The suggested model captures the image of the intruder and triggers the alarm when it receives the signal from the motion sensor. Even though it is a model for home-based security system, no safety method was applied to the picture of the intruder that is captured. The intruder might reach the camera and remove it somehow if he gains access inside the house. Othman et al. also introduced a similar approach, using computer vision-based security scheme for combined body detection of people [16]. The picture of the detected person is sent to the owner's mobile phone but no encryption method is used to ensure that the picture received over the network is genuine or manipulated by man-in-the-middle attack [17]. Another method based on motion detection technique was introduced to identify the presence of people which uploads specific video frames instead of the whole video stream to the cloud [18].

The suggested scheme utilizes lesser processing resources but needs improvement in the storage method for saving the video frames. There should be a mechanism to remove the detected frames from the sensor device after they have been successfully uploaded to the cloud. As an extension of this previous work, a robust and resource effective model for live video streaming application is proposed in this paper. The detailed architecture and working of proposed model are discussed in the following section.

3. THE PROPOSED SECURITY MODEL FOR RESOURCE CONSTRAINED LIVE VIDEO STREAM

The hierarchy of the proposed framework can be visualized from the network diagram shown in Figure 1. The camera sensor projects the live video stream to the display system for live monitoring and surveillance. The displayed video is analyzed for any change in the activities and only the video frames that detect the happening of any event are extracted from the video stream. The detected video frames undergo encryption process to ensure the security of video data. In the next phase, an automated synchronization tool is used to send the encrypted video frames to the cloud and remove them from the sensor node, this in turn frees up space for new video frames on the sensor node. Finally, the frames that recorded activities are decrypted and stored on cloud with respective date and time stamp for future use. The proposed framework is assumed to be working in resource constrained scenario in which the edge computing facility is not available and all processing is done on the sensor node and the cloud.

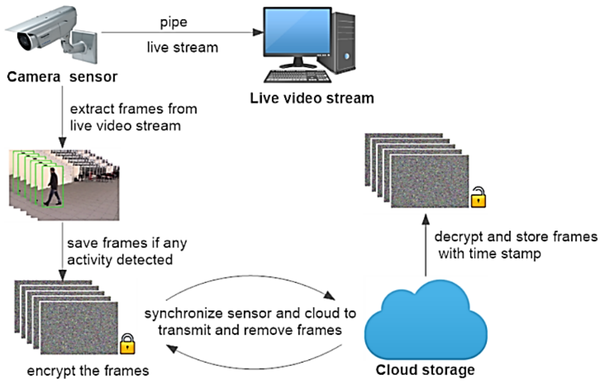

Figure 1. Network diagram of the proposed framework.

The flow of steps for the process of detecting activity and securing data in resource constrained live stream videos is shown in Figure 2. Once the video stream is initialized, we use OpenCV to grab each frame from the live stream for processing. The frames are resized, converted to grayscale, and Gaussian smoothing [19] is applied to reduce the uneven distribution of pixel intensities due to noise, which might result in false activity detection. The weighted average of the frames F_{av} is calculated and if it is the start of the video and no previous frame is in the pipeline then the current frame F_c is initialized as F_{av}. In next stage, difference between the frames F_d is calculated to detect any activity occurrence in video frames using equation (1).

$$F_{d\ (av,c)} = |\, F_{av} - F_c \,| \qquad (1)$$

Ideally, the difference between the frames should be zero if there is no change in pixels, but practically it is not possible due to variable pixel intensity values, and changes in light conditions. Therefore, thresholding is applied on frames to convert them into binary which would ease the process of edge detection in upcoming steps. A suitable value of threshold T is selected (T = 25 in this case) depending on the lighting conditions to reduce the false positives that might occur due to noise or lighting conditions. If F_d is greater than threshold value T, the particular pixel is set to white, and it is assumed to be foreground of the image. Whereas, if F_d is lesser than threshold than the pixel is discarded and set to black, which appears as background of the image. The contour detection method is applied on the thresholded frame for edge detection. The minimum area A_{min} is the minimum pixel area (A_{min} = 4000 in this case) which is set as criterion to detect any activity and ignore false detections. If the contour area A_c is greater than the minimum area A_{min}, consider it as a change in activity otherwise loop over contours unless the condition is satisfied.

This method detects and saves frames from a continuous live stream video that contain the information about some happening that does not convey redundant information. The frames extracted from video stream contain important data that cannot be directly sent over the wireless link publicly, as it could be easily accessed by the attackers. Therefore, in the next step symmetric encryption is applied using a private key to secure the video frames from the exposure to the un-intended audience. The private key is derived from the password chosen by the sender, and the key is shared through any other safe channel instead of the same channel carrying video data. Public key encryption could have been used instead to share the key publicly but as we are assuming the availability of resources is limited, private key encryption serves quite efficiently in that case. The encryption of frames is performed using equation (2), where *sym_enc* shows the function for symmetric encryption, F is the frame to be encrypted and K is the secret key that is used by encryption algorithm to produce encrypted frame F_e .

$$F_e = sym\_enc\ (F, K) \qquad (2)$$

In this study, Advanced Encryption Standard (AES) algorithm is used to encrypt the video frames with 256-bit key length and 128-bit block size in cipher block chaining (CBC) mode. The hashing function SHA-256 is used to generate hash digest for secure key derivation from the chosen password, in order to ensure the integrity of the secret key for decryption. OpenSSL version 1.1.1d is an open source linux library, used for the implementation of the symmetric encryption function using AES-256 and SHA-256. The encrypted frames are synchronized using Rsync Linux tool for synchronization of the sensor node and the cloud. Rsync uses the delta-transfer algorithm to send reduced amount of data and avoid any file duplication. The feature that makes Rsync suitable in the proposed model is that it automatically removes the frames once successfully transferred to the cloud server. This method frees up the memory space on the sensor node, making it available for new data to be stored in case of live stream videos. Finally, on the cloud server the encrypted frames are decrypted using equation (3), where *sym_dec* is the method for decryption cryptographic algorithm, F_e is the encrypted frame received from the sensor node, K is the secret key for decryption and F is the recovered original video frame. The recovered frames are saved with respective date and time stamp which can be later accessed if required.

$$F = sym\_dec\ (F_e, K) \qquad (3)$$

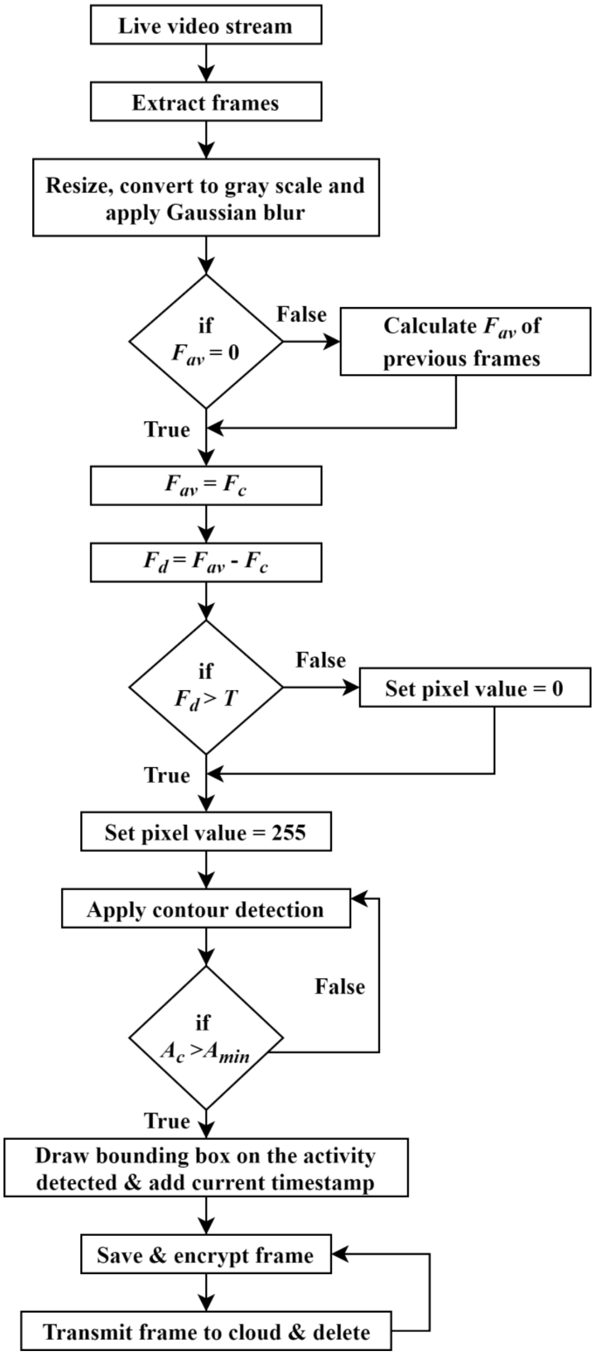

Figure 2. Flow of steps used during the process of securing resource constrained live stream videos.

4. RESULTS & DISCUSSION

The proposed algorithm is tested on real-time video stream generated by 8 megapixel Sony IMX219 Raspberry Pi official camera but for fair analysis a publicly available dataset by the computer vision research lab in École polytechnique fédérale de Lausanne (EPFL) is used for experimentation [20]. The dataset contains video clips of varying duration recorded by multi-cameras in a variety of situations, with variable number of pedestrians and different light conditions. The frame rate form videos used here is 25fps and is encoded using MPEG-4 codec and Indeo 5. Figure 3 (a) shows a frame detected from one of the video clips in the data sets. An activity is recorded as soon a person walked in the camera view due to change in the pixel values. The difference between the average of the previous frames and the current frame is represented in Figure 3 (b), but it is difficult to detect activity from this due to variation in the intensity values. The edges of the image containing frame difference have been sharpened by thresholding for contour detection as shown in Figure 3 (c). The original image is encrypted so that attackers cannot extract information, as shown in Figure 3 (d).

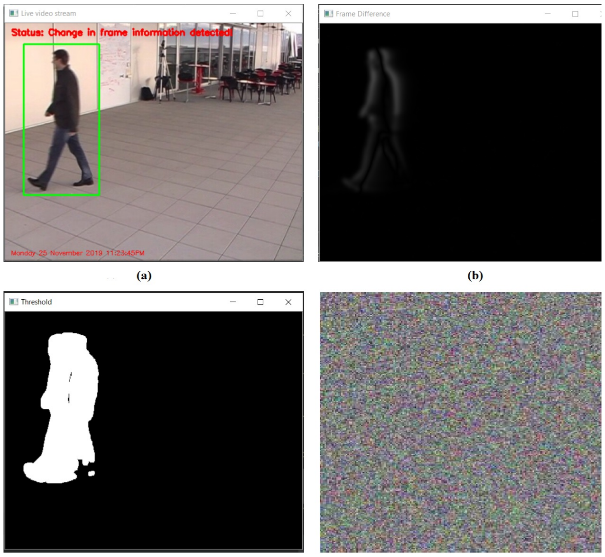
Live video stream
Status: Change in frame information detected!
Monday 25 November 2019 11:23:45PM
Frame Difference
(a)
(b)
Threshold

Figure 3. Shows (a) activity detection in the frame, (b) difference between the average and current frames, (c) contour detection applied, and (d) encrypted bitmap of the frame

Experiments are performed to analyze the resource utilization in video streams without the use of any optimization technique to record, store and transmit video frames when compared with the proposed method. Table 1 shows the comparison in terms of number of frames detected and the memory footprint occupied by the saved frames when using the proposed model. The results clearly show that the number of frames detected by the proposed method for storage is much lesser in number of frames that are conventionally extracted by video streams without using activity detection method. As shown in Table 1, the proposed method outperforms the conventional method of storing frames, even in worst case scenarios. For instance, in the video clip 'Campus Sequences Seq.1, cam.0', there is little movement of pedestrians in the video while most part of the video clip shows the still view of the campus entrance. Thereby, the proposed model detects only 699 useful frames containing activity information and skips 1020 redundant frames which contribute to almost 59 percent reduction in memory footprint required by the conventional method. Whereas, the video sequence 'Laboratory sequences, Cam 0' comprises of video recorded in a laboratory where continuous activity is taking place, motion was detected in a total of 3914 frames out of 3693. In this case the percentage difference in number of frames detected between proposed and conventional method was about 5.6 percent, which is not a very significant difference, but still proposed method saves 4.3 percent of the memory footprint required by the conventional approach. It can be deduced from these results that the proposed method saves more memory in scenarios where lesser activity takes place, for instance classrooms, hospital rooms and smart-home surveillance, compared to the traffic on a highway or a busy marketplace where continuous events are happening.

Keeping in view the heterogeneous and scalable nature of IoT network, experiments are conducted on devices with different characteristics, such as memory and processing capabilities. The comparative analysis of the time required by two different sensor nodes to encrypt increasing number of video frames is shown in Table 2. The sensor nodes used during experimentation are computing devices equipped with limited resources namely, Raspberry pi zero W and Raspberry pi 3B+. Raspberry pi 3B+ encrypts 1000 frames almost five times faster than raspberry pi zero W. This fact is very important to consider when processing video data streams. During experimentation Raspberry pi zero W lags processing of videos as it has only a single processing core whereas, Raspberry pi 3b+ performed better for the video processing task as it has four processing cores and uses pipelining simultaneously. But in terms of memory storage both Raspberry pi devices had limited memory space (maximum available 32 giga bytes). The proposed algorithm performs effectively to accommodate new incoming frames, as the Rsync tool removed the encrypted frames from the sensor node once they were successfully transferred to the cloud data storage.

Table 1. Video frames recorded and memory footprint occupied by video data using conventional approach and proposed model on different samples of video data

File Name	Duration (seconds)	No. of frames			Memory footprint(megabytes)		
		Conventional	Proposed	Percentage difference (%)	Conventional	Proposed	Percentage difference (%)
Campus (Seq.1,cam.0)	80	1719	699	59.34	118	48	59.32
Passageway (Seq.1,cam.0)	100	2500	810	67.60	160	54	66.25
Laboratory (Cam 0)	156	3914	3693	5.65	322	308	4.35
Terrace (Seq.1,cam.0)	200	5010	4261	14.95	341	307	9.97
Basketball (Seq.1,cam.2)	374	9368	6550	30.08	675	456	32.44

Table 2. Time taken to encrypt video frames using Raspberry pi zero W and Raspberry pi 3B+

	Encryption Time (seconds)		
Frames	Raspberry pi zero W	Raspberry pi 3B+	Percentage difference (%)
1	0.191	0.069	63.87
10	1.566	0.287	81.67
100	16.781	2.575	84.74
1000	171.997	30.224	82.43

5. CONCLUSION

The smooth working of WMSN IoT system is hindered due to common problems such as resource limitations and security weaknesses. This study addresses these issues by providing a joint solution for an efficient resource utilization framework that serves securely in constrained video surveillance scenarios.

The results during the experimentation prove the proposed model to be useful in terms of scalability of the network, lower memory footprint, useful data extraction, lower processing needs, and secure data transmission. The results also showed that the proposed framework is best suited for video surveillance in an environment with fluctuations in activity levels, for example a classroom that has periods of peak activity followed by periods of little-to-no activity. In future work, hardware friendly encryption algorithms like Adiantum [21] will be considered to improve efficiency and reduce the memory footprint generated by encryption algorithms.

6. ACKNOWLEDMENTS

This research was funded and supported by British Telecom Innovation Ireland Centre (BTIIC). I would like to appreciate the efforts of all my supervisors specially Dr. Zhiwei for his support.

7. REFERENCES

[1] H. Saadeh, W. Almobaideen, and K. E. Sabri, "Internet of Things: A review to support IoT architecture's design," in *2017 2nd International Conference on the Applications of Information Technology in Developing Renewable Energy Processes & Systems (IT-DREPS)*, 2017: IEEE, pp. 1-7.

[2] S. K. Lee, M. Bae, and H. Kim, "Future of IoT networks: A survey," *Applied Sciences,* vol. 7, no. 10, p. 1072, 2017.

[3] S. U. Jan, Y.-D. Lee, J. Shin, and I. Koo, "Sensor fault classification based on support vector machine and statistical time-domain features," *IEEE Access,* vol. 5, pp. 8682-8690, 2017.

[4] M. H. Mazaheri, S. Ameli, A. Abedi, and O. Abari, "A millimeter wave network for billions of things," in *Proceedings of the ACM Special Interest Group on Data Communication*, 2019, pp. 174-186.

[5] B. S. I. Association, "The Picture is Not Clear: How many CCTV surveillance cameras in the UK," *no,* vol. 195, pp. 1-7, 2013.

[6] M. Di Francesco, N. Li, M. Raj, and S. K. Das, "A storage infrastructure for heterogeneous and multimedia data in the internet of things," in *2012 IEEE International Conference on Green Computing and Communications*, 2012: IEEE, pp. 26-33.

[7] T. Li, Y. Liu, Y. Tian, S. Shen, and W. Mao, "A storage solution for massive iot data based on nosql," in *2012 IEEE International Conference on Green Computing and Communications*, 2012: IEEE, pp. 50-57.

[8] H.-H. Lee, J.-H. Kwon, J.-J. Jung, and E.-J. Kim, "Virtual Storage System based on Multiple Embedded Devices in IoT Environments," 2017.

[9] Y. Mo, "A Data Security Storage Method for IoT Under Hadoop Cloud Computing Platform," *International Journal of Wireless Information Networks,* vol. 26, no. 3, pp. 152-157, 2019.

[10] J. Yang, S. He, Y. Lin, and Z. Lv, "Multimedia cloud transmission and storage system based on internet of things," *Multimedia Tools and Applications,* vol. 76, no. 17, pp. 17735-17750, 2017.

[11] H. Yan, X. Li, Y. Wang, and C. Jia, "Centralized duplicate removal video storage system with privacy preservation in IoT," *Sensors,* vol. 18, no. 6, p. 1814, 2018.

[12] H.-Y. Chang, K.-B. Chen, and H.-C. Lu, "A novel resource allocation mechanism for live cloud-based video streaming service," *Multimedia Tools and Applications,* vol. 76, no. 19, pp. 19689-19706, 2017.

[13] L. P. Van, J. De Praeter, G. Van Wallendael, J. De Cock, and R. Van de Walle, "Machine learning for arbitrary downsizing of pre-encoded video in HEVC," in *2015 IEEE International Conference on Consumer Electronics (ICCE)*, 2015: IEEE, pp. 406-407.

[14] A. M. Elshamy, M. Abdelghany, A. Q. Alhamad, H. F. Hamed, H. M. Kelash, and A. I. Hussein, "Secure Implementation for Video Streams Based on Fully and Permutation Encryption Techniques," in *2017 International Conference on Computer and Applications (ICCA)*, 2017: IEEE, pp. 50-55.

[15] N. Surantha and W. R. Wicaksono, "Design of smart home security system using object recognition and PIR sensor," *Procedia Computer Science,* vol. 135, pp. 465-472, 2018.

[16] N. A. Othman and I. Aydin, "A new IoT combined body detection of people by using computer vision for security application," in *2017 9th International Conference on Computational Intelligence and Communication Networks (CICN)*, 2017: IEEE, pp. 108-112.

[17] M. Conti, N. Dragoni, and V. Lesyk, "A survey of man in the middle attacks," *IEEE Communications Surveys & Tutorials,* vol. 18, no. 3, pp. 2027-2051, 2016.

[18] A. N. Ansari, M. Sedky, N. Sharma, and A. Tyagi, "An Internet of things approach for motion detection using Raspberry Pi," in *Proceedings of 2015 International Conference on Intelligent Computing and Internet of Things*, 2015: IEEE, pp. 131-134.

[19] T. Popkin, A. Cavallaro, and D. Hands, "Accurate and efficient method for smoothly space-variant Gaussian blurring," *IEEE Transactions on image processing,* vol. 19, no. 5, pp. 1362-1370, 2010.

[20] H. Shitrit, "Multi-camera pedestrians video,'EPFL'data set: multi-camera pedestrian videos," *Internet web page cvlab. epfl. ch/data/pom/, dated Feb,* vol. 18, 2013.

[21] P. Crowley and E. Biggers, "Adiantum: length-preserving encryption for entry-level processors," *IACR Transactions on Symmetric Cryptology,* pp. 39-61, 2018.