Lightweight Image Crypto-Compression Using Haar Transform and Selective Encryption for Grayscale IoT Images

Joseph Azar[1], Hassan Noura[1], and Raphael Couturier[1]

[1]Univ. Franche-Comté (UFC), FEMTO-ST Institute, CNRS, Belfort, France

***Abstract*—With the advent of the Multimedia Internet of Things (MIoT), many image compression techniques have been proposed to address the network's considerable challenges related to performance and security. However, many MIoT devices, such as the nRF52832 SoC with 64Kb RAM or even less, have significant memory constraints, making conventional methods unsuitable. MIoT networks face considerable challenges related to performance and security due to limitations in the power, computation, and memory of MIoT devices. These limitations result in difficulties in handling high image volumes. Multimedia compression is a potential solution to reduce data size. As MIoT devices often rely on wireless connections, they are also vulnerable to diverse security attacks (passive and active). This work introduces a secure and efficient image crypto-compression technique dedicated to devices having limited memory. It also proposes using denoising and a super-resolution deep learning model to reduce the overhead of the compression process and a lightweight cipher scheme that requires a single round of simple operations to reduce the overhead of the encryption process. The proposed approach effectively addresses the mentioned challenges with minimal overhead on the MIoT device, especially in terms of computational and communication delays, and extensive experimentation underscores its suitability in both effectiveness and robustness.**

***Keywords*—** Multimedia Internet of Things (MIoT), Crypto-compression scheme, DWT, Selective encryption, Denoising super-resolution model.

I. Introduction

The increasing number of multimedia apps has led to an increased demand for data reduction and security measures. Numerous research studies have been undertaken to explore energy-efficient compression techniques for multimedia Internet of Things (IoT) systems to reduce power consumption[11]. These techniques aim to minimize processing operations and memory accesses inside the compression system, hence increasing battery life and reducing power consumption in the system as a whole. In terms of security, conventional encryption techniques frequently function on the entirety of the dataset, thereby incurring significant time and resource costs. Consequently, there is a need for partial encryption methodologies that selectively encrypt solely the most crucial components of the data, thus diminishing computational requirements and mitigating the effects on transmission time and bandwidth. Various studies have put forth techniques for implementing selective encryption, including zigzag permutation and encrypting a specific set of AC coefficients. These methods aim to improve security while ensuring compatibility with the JPEG standard [6], [16], [17].

MIoT faces two main challenges. The first challenge is due to the inherent characteristics of multimedia content, such as images, which have a significant size compared to scalar data. The second challenge is due to platform limitations in terms of CPU speed, memory size, and power, as most MIoT devices are battery-operated. These challenges necessitate the implementation of an effective multimedia data reduction and error correction scheme in MIoT systems, as these devices are often constrained, particularly in terms of power consumption and resource overhead. Consequently, reducing the size of transmitted/protected data, minimizing computation costs, and mitigating the impact of channel errors represent a significant research challenge in the context of MIoT, which is the main focus of this paper.

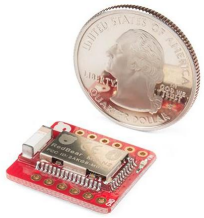


Fig. 1: The nRF52832 SoC utilized in this paper features a 32-bit ARM Cortex-M4F CPU and equipped with 512kB of flash memory and 64kB of RAM.

The work of this paper targets resource-constrained microcontrollers with limited memory ($\leq 64kB$) and processing power, such as the device shown in Figure1. This paper proposes a solution that enables application servers to recover missing or corrupted grayscale images using appropriate deep-learning denoising and super-resolution techniques. This solution addresses the intrinsic limitations of MIoT end devices. In this approach, data reduction is integrated into the MIoT devices by applying a lightweight implementation of the Discrete Wavelet Transform (DWT), namely the row-column Haar transform. The low-frequency coefficients are ciphered using a lightweight cipher scheme based on the dynamic key approach, and the high-frequency coefficients are completely discarded. With both super-resolution and visual content enhancement at the application server(s), the highly distorted images can be restored, thus resulting in a high compression ratio and enhanced image quality.

The remaining sections of this paper are organized as follows: SectionII presents the related research work to this paper. SectionIII discusses the crypto-compression scheme at MIoT devices and application server(s)/cloud. Section IV describes the proposed crypto-compression solution. Section V details the performance and security tests that are applied to demonstrate the efficiency and robustness of the proposed solution. After this, in Section VI, the experimental setup and results are presented and discussed. Finally, Section VII concludes this work.

II. Related work

Kouadria et al.[8] proposed a low complexity Discrete Cosine Transform (DCT) that can be used for image compression in wireless

visual sensor networks that combines the block discrete cosine transform (BDCT) with a pruning approach to reduce the number of arithmetic operations required. Their approach resulted in 60% time and energy savings while maintaining acceptable image quality. Campobello et al. [3] proposed an efficient encoding scheme called RAKE, which encodes positions of non-zero bits in a binary sequence. The proposed lossless technique provides highly memory-efficient compression for grayscale images with about 5% performance penalty compared to JPEG-LS[15] and about 10% compared to CALIC[18]. Lee et al.[11] propose a line-based compression system using a four-level, two-line discrete wavelet transform and adaptive line prediction. The authors also introduce a new bit rate control algorithm to improve image quality consistency in one frame. The proposed system claims to achieve visually lossless compression criteria and lower power consumption better than existing techniques. Deep learning techniques for IoT image compression have been getting more attention recently. Krishnaraj et al. [9] proposed a model based on the discrete wavelet transform (DWT) and convolutional neural network (CNN) to achieve effective compression with better-reconstructed image quality. Experimental results show that the author's approach outperforms existing methods such as super-resolution convolutional neural networks, JPEG, and JPEG2000 regarding compression performance and reconstructed image quality. However, such techniques require acceptable computing power, which is different from the case of this work. Hu et al.[7] introduced Starfish, a new design for compressing images in IoT applications. Interestingly, the authors proposed a solution that is resistant to packet loss compared to traditional techniques, and their experiments showed that their solution outperforms JPEG in terms of bandwidth efficiency and energy consumption and maintains good image quality even in the presence of packet loss. However, Starfish integrates a deep neural network to generate a loss-resilient compressed representation of images. This solution can work on microcontrollers and IoT devices but not on the minimal devices as the one targeted in this paper.

Different works in the literature tackled the crypto-compression research area. Cidjeu et al.[5] proposed two lossless crypto-compression schemes to secure medical images based on the Elliptic Curve (EC). The schemes involve grouping pixel values as finite field elements and transforming them into points on an elliptic curve for compression. They apply EC-based encryption schemes utilizing the difficulty of solving the Discrete Logarithm Problem. These schemes demonstrate better performance in terms of compression rate, image quality, and execution time compared to existing systems. However, the authors do not mention any specific details or implications regarding IoT. Puech et al.[14] applied the DCT and Advanced Encryption Standard (AES) methods and proposed using DC and some AC coefficients of the lowest or highest frequencies to construct a stream for encryption. The rationale is that encrypting more elements at the low frequency of each block in a JPEG image results in better encryption. The authors mention that the approach aims to reduce computational resources for low-power networks but does not explicitly address the limitations of limited RAM in IoT devices. Inspired by the demonstrated efficiency of Artificial Neural Networks (ANNs) in image compression compared to traditional methods, especially for noisy or incomplete images, Benlcouiri et al.[2] apply encryption based on the AES algorithm on the parameters of compression by a network of multi-layer neurons. While the experiments have shown the efficiency of the proposed hybrid approach, neural networks-based compression can only work on some kinds of microcontrollers and requires a training process.

III. Background

The MIoT systems consist of numerous edge devices that process captured multimedia data (such as images) and transmit it via multi-hop or star wireless communications (as shown in Fig. 2) to the application server(s) or data center. After acquiring the image, each MIoT device must be capable of reducing the size of the multimedia data before forwarding it to the application server, which will subsequently decompress it to retrieve the visual content. However, due to resource and computation constraints, some MIoT devices are unable to perform conventional lossy image compression techniques. Thus, to minimize the required resource overhead, an effective compression strategy with minimal memory and computation requirements should be implemented on MIoT devices to minimize the overhead of compressing and transmitting collected compressed images. As a result, this can regulate both bandwidth and energy consumption on the MIoT side.

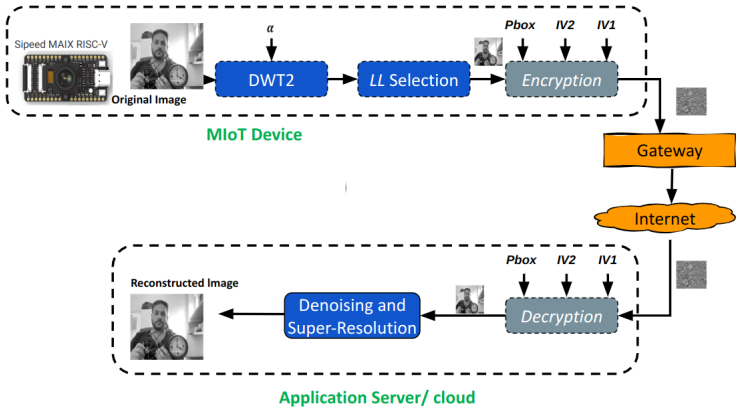


Fig. 2: The proposed crypto-compression scheme at MIoT devices and application server(s)/cloud.

As depicted in Fig. 2, the proposed compression solution involves performing one or two levels of the 2D-DWT (Discrete Wavelet Transformation) by using the Haar transformation at the MIoT devices. To accommodate the constraints of resource-limited microcontrollers, we implemented this solution using the row-column integer Haar wavelet transform. The reason for adopting the integer version is to eliminate the computational overhead and potential inaccuracies associated with floating point operations, which are often not well supported or are performance-intensive on limited microcontrollers. DWT is a signal processing technique used to extract information and is prevalently applied in multimedia compression standards like JPEG2000 or MPEG-4. By design, it is a one-dimensional transform. However, it can be extended as a two-dimensional transform in both horizontal and vertical directions. In its two-dimensional implementation, DWT produces four sub-matrices, with each sub-matrix being a quarter of the original matrix. The results for a single level of 2D-DWT are a low-resolution sub-matrix (LL) denoting the low frequencies; high vertical and low horizontal resolution (HL); low vertical and high horizontal resolution (LH); and a high-resolution sub-matrix (HH). Here, HL and LH represent the mid frequencies and HH indicates the high frequencies. Further, the second level transform is specifically applied to the LL part, termed as dyadic decomposition, as depicted in Fig. 3.

Fig. 3: Two-level DWT generates two-dimensional coarse and detailed values [4].

The proposed lossy compression approach helps reduce the size of protected transmitted multimedia data (i.e. raw data) by selecting only the last LL band, which will be encrypted using the proposed lightweight cipher scheme. Subsequently, these visual contents can be decrypted first at the application server(s) and potentially be further improved by using a trained super-resolution denoising model.

Increasing the level of 2D-DWT will decrease the communicated/protected data size but it can ensure additional visual degradation.

1) Mathematical Description of Row-Column Integer Haar Wavelet Transform: The Haar wavelet transform is based on a pair of functions, a scaling function $\phi(t)$ and a wavelet function $\psi(t)$. In the Haar wavelet case, these functions are defined as follows:

$$\phi(t) = \begin{cases} 1, & 0 \leq t < 1 \\ 0, & \text{otherwise} \end{cases}$$

$$\psi(t) = \begin{cases} 1, & 0 \leq t < 0.5 \\ -1, & 0.5 \leq t < 1 \\ 0, & \text{otherwise} \end{cases}$$

The row-column integer Haar wavelet transform is applied in two steps:

1) **Column-wise transform**: Starting with the original 2D image of size $m \times n$, compute the average of each pair of adjacent elements in each column. This results in a new image of size $m \times \frac{n}{2}$. Let $I_{i,j}$ represent the pixel intensity in the row i and the column j in the input image. Then, the transformed image $I'_{i,j}$ can be described as:

$$I'_{i,j} = \frac{I_{i,2j} + I_{i,2j+1}}{2}$$

2) **Row-wise transform**: Apply the same operation to the rows of the resulting image from the previous step. This will result in the final compressed image of size $\frac{m}{2} \times \frac{n}{2}$. Let $I''_{i,j}$ represent the pixel intensity at row i and column j in the intermediate image I'. The final compressed image $I^*_{i,j}$ can be described as:

$$I^*_{i,j} = \frac{I''_{2i,j} + I''_{2i+1,j}}{2}$$

2) Complexity Analysis: The row-column integer Haar wavelet transform is an in-place technique, meaning that it does not require any additional memory beyond the original input image. The transform is applied directly to the input image array, and the compressed image occupies the first $\frac{m}{2} \times \frac{n}{2}$ elements of the captured 1D array. The computational complexity is $O(mn)$. For each of the m rows, the algorithm computes the averages for $\frac{n}{2}$ pairs of columns, resulting in $O(m \cdot \frac{n}{2})$ complexity for the column-wise transform. Similarly, for each of the n columns, the algorithm computes the averages for the $\frac{m}{2}$ pairs of rows, resulting in $O(n \cdot \frac{m}{2})$ complexity for the row transformation. Thus, the overall complexity is $O(mn)$, which is a linear complexity concerning the number of elements in the input image.

IV. Proposed Crypto-Compression Solution

A. Image Compression Approach

The proposed image compression method described in Algorithm 1 utilizes a row-column integer Haar wavelet transform. This technique involves performing a Haar wavelet transform on an input image first column-wise and then row-wise, effectively reducing the image size and memory requirements. The last LL will only be selected as compressed data and will be transmitted after encryption to the application server. To recover the original image on the other side, decryption should be applied first. Then, we will use the trained super-resolution Deep Learning (DL) to reconstruct the original image. This model is built to reduce the effect of losing high and middle frequencies, in addition, it can help to reduce the effects of wireless channels. On the other hand, the proposed crypto scheme is designed to have minimum error propagation. These steps are illustrated in Fig. 4. The image denoising and recovery model used is an SRGAN model[10][1]. This model comprises a series of residual blocks with two convolutional layers each to aid in deep network gradient flow. The generator utilizes these blocks, post-residual layers, and upsampling to produce super-resolved images. In parallel, a discriminator with convolutional layers and leaky ReLUs classifies between real and super-resolved images. We employ a pre-trained VGG network to guide the super-resolution process to extract salient image features. Finally, an adversarial model combines the functionality of the generator and the discriminator, optimizing image quality and recovery.

The proposed solution's innovation is by using a super-resolution denoising model that can be applied at the application server or cloud to enhance the low resolution of the collected image. This model allows discarding all the high-frequency components of the DWT and some of the low frequencies, thus achieving a higher compression ratio. In addition, this model can also enhance packet loss, erroneous packets, or encryption error propagation without introducing additional overhead on the side of MIoT devices, which represents our main contribution.

Algorithm 1: One level row-column Haar transform.

Input: img (1D uint8 vector)

Set m to the number of rows in img
Set n to the number of columns in img
`// Lightweight Haar starts`
for $i \leftarrow 0$ **to** $m * n$ *with step* n **do**
  for $j \leftarrow 0$ **to** n *with step* 2 **do**
    Update img with the average of the adjacent elements in the column
for $i \leftarrow 0$ **to** $(m * n)/2$ *with step* n **do**
  for $j \leftarrow 0$ **to** $n/2$ **do**
    Update img with the average of the adjacent elements in the row
`// Lightweight Haar ends`

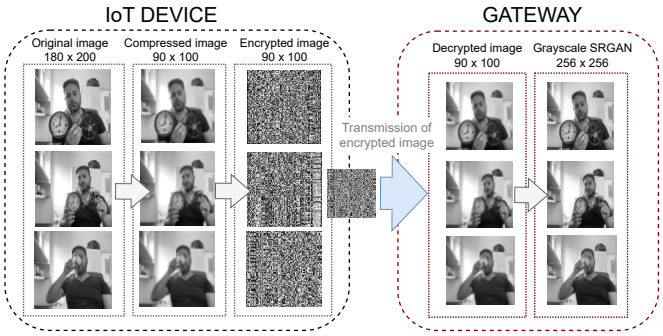


Fig. 4: Illustration of the steps: compression, encryption, transmission, decryption, and recovery in the IoT device and gateway.

B. Proposed Cipher scheme

The proposed cipher scheme is based on the dynamic key approach, where a dynamic key is generated for each new input image (or for a set of images; depending on configuration). Based on the generated dynamic key, the permutation table π in addition to two initial vectors will be generated ($IV1$,$IV2$). The encryption scheme is based on a single round consisting of a row permutation layer and a mixing layer, as illustrated in Figure 5. After this, the mixing process will be realized by mixing rows (chaining operation mode)

[1]https://github.com/idearibosome/tf-perceptual-eusr

forward (from the first rows to the last ones) with the first initial vector $IV1$ and then backward mixing with the second initial vector $IV2$ (from the last rows to the first one). The decryption scheme is similar to the encryption one; the same dynamic key should be generated, and consequently the required cryptographic primitives. Furthermore, the first decryption step will be the inverse mixing process using the same initial vectors ($IV1$, $IV2$). This can be done by applying the inverse of the backward row mixing using $IV2$ followed by the inverse of the forward row mixing using $IV1$. Finally, an inverse permutation process is performed using the inverse permutation table. This cipher follows one of the recent cryptographic approaches that proposes to use the dynamic key approach to reduce the round number and by designing lightweight round function for the encryption algorithms towards reducing the required computation and resources of the encryption process.

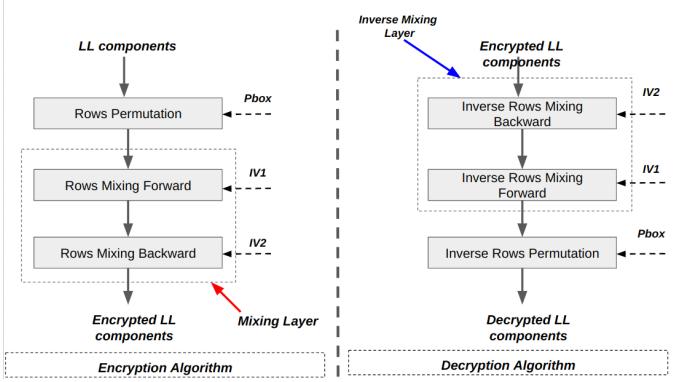


Fig. 5: The proposed cipher scheme: encryption and decryption algorithm.

V. Security & Performance Analysis

In this section, several performance and security tests are applied to demonstrate the efficiency and robustness of the proposed solution.

A. Security Analysis

Several security tests were presented to show that encrypted images achieve the desired randomness, uniformity, independence, and key avalanche effect, as discussed in the following.

1) Statistical Analysis: For a cipher scheme to be robust against statistical attacks, it must achieve some random properties [13], [12]. To this end, we conducted a statistical analysis involving the following statistical tests: (a) Histogram analysis, (b) Entropy analysis, and (c) correlation between plain and encrypted images. Original images and their corresponding cipher images and probability distribution are shown in Fig. 6.

2) Uniformity: In security, uniformity implies that every symbol has an equal probability of occurrence. Typically, this probability is equal to $\frac{1}{N}$, where N is the total number of existing symbols. One way to assess uniformity is by plotting the Probability Density Function (PDF) of the encrypted data and evaluating it visually. Unlike plaintext, which has a bell-shaped normal distribution (some symbols are more likely to occur than others) (Fig. 6c), the ciphertext resulting from the proposed scheme is uniformly distributed (Fig. 6f). This proves that the proposed scheme can produce randomized ciphertext, hence, it is immune to statistical attacks. Furthermore, Figures 6a and 6d show that the proposed cipher scheme generates encrypted images that have all the values of the symbols in comparison to the original symbols.

3) Recurrence: Another important security test that evaluates the uniformity property is the recurrence test. In particular, each encrypted symbol is compared with a delayed version of it. To attain the desired uniformity and randomness levels, the recurrence plot should be highly scattered and should cover all available regions. On the other hand, the recurrence plot of the original LL band (normally distributed data) contains values that are concentrated in one specific region, as depicted in Fig. 6b (not randomized). From Fig. 6e, it is evident that the desired recurrence plots are obtained, which validates the high level of randomness and uniformity obtained with the proposed scheme.

B. Resistance Against Key-Related and Brute Force Attacks

Two important parameters that should be taken into consideration when discussing a security scheme in the context of brute force attacks and key-related attacks are the key size and key sensitivity.

1) Key Size: The generated dynamic key consists of 512 bits, which is sufficient to resist brute-force attacks. In particular, the combination of the working key and the channel-derived nonce is hashed using the SHA-512 scheme to obtain a 512-bit key. This key is divided into several sub-keys to generate several cipher primitives.

2) Key Sensitivity: Key sensitivity mandates that at least half of the output bits are changed upon a slight change in the utilized key. In other words, a one-bit change in the key should result in at least 50% different ciphertext (bit level). As shown in Fig. 7-a), it is clear that key sensitivity is successfully achieved using the proposed scheme, where most of the sensitivity values are close to 50%. It should be noted that the cryptographic primitives that are used in the encryption/decryption process are frequently updated after a specific number of communicated image(s). Consequently, this complicates the adversary's task in acquiring the dynamic key, and thus deriving the required cryptographic primitives. By constantly shuffling the ciphering primitive, one can guard against brute force attacks and key-related attacks.

C. Resistance Against Linear and Differential Attacks

Linear and differential attacks are directly related to the independence property.

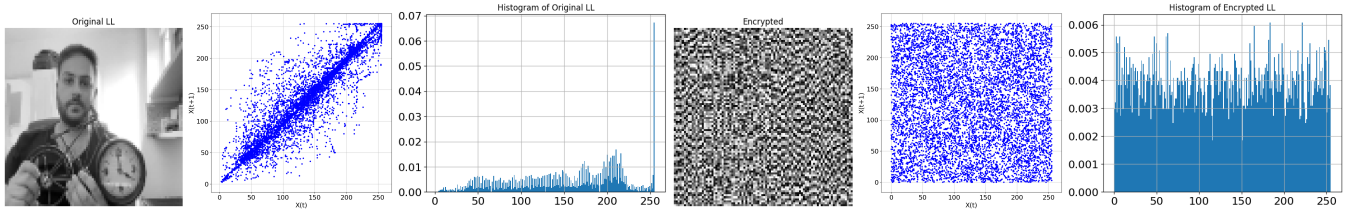
1) Independence: When the independence property is achieved, the original data should be different compared to encrypted data at the bit level (the percentage of difference should be close to 50%). To prove this property, the difference test is carried out. The results presented in Fig. 7-b) reveal that the difference values are always equal to 50% using the proposed scheme, and hence the independence property and the avalanche effect are both achieved.

VI. Performance Analysis

The proposed approach is implemented on a Redbear Nano V2 microcontroller and compared to the publicly available JPEG implementation for Arduino on GitHub [1].

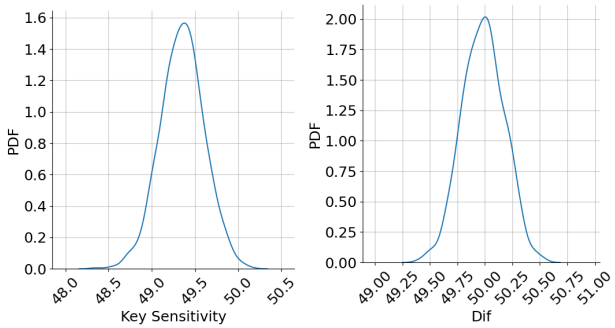
JPEG offers various compression quality settings. As we employ a super-resolution model for quality enhancement, we opted for JPEG's low-quality setting, which provides the highest compression ratio. Figure 8-(a) shows the time required (in milliseconds) to compress grayscale images of varying sizes using JPEG, Haar L1 (single-level transformation), and Haar L2 (two-level transformation). The inherent complexity of JPEG, compared to the Row-Column Haar transform, makes the differences in compression times evident. Notably, we encountered buffer overflow errors for images larger than 200×200, attributed to the memory constraints of the Redbear Nano V2 (64 kB) and JPEG's memory demands relative to our approach. In terms of compressed size, as presented in Figure 8 -b, JPEG outperformed the compression of Haar L1 and closely matched the compression of Haar L2. However, Figure 9 reveals that the image quality of Haar L2 compression is inferior to that of Haar L1 and JPEG. While JPEG allows a higher compression ratio than our Haar L1-based method, our technique allows capturing larger images due to its lower memory requirement.

Fig. 10 presents the Redbear Nano V2's power consumption during image transmission via Bluetooth Low Energy (BLE) for a 200×180 image and its Haar L1 compressed counterpart. The BLE settings



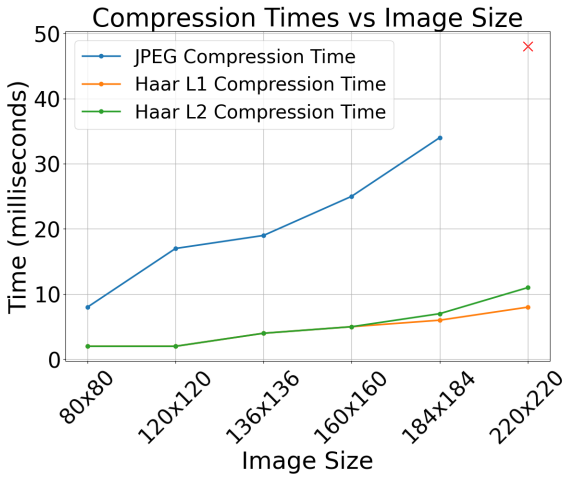
(a) Original LL (b) Recurrence of the original LL band (c) PDF of original LL band (d) Encrypted band (e) Recurrence of encrypted LL band (f) PDF of encrypted LL band

Fig. 6: Image visualization, recurrence, and PDF of the original LL band and the corresponding encrypted one using the proposed cipher scheme

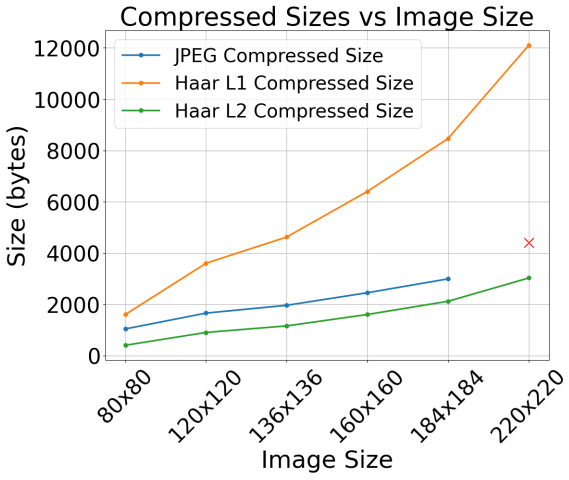


(a) (b)

Fig. 7: Key sensitivity (a) and difference Independence (b) tests



(a)



(b)

Fig. 8: Compression Times (a) and Compressed Sizes (b) vs Image Size: A comparison of JPEG and Haar level 1 & 2 compression techniques. Note: JPEG compression encounters a buffer overflow for image sizes exceeding 200x200, indicated by the red 'X'.

were: advertising interval at 1800 ms, transmit power at +4 dBm, and 15-byte data packets. Power consumption measurements employed an Arduino UNO coupled with an INA219 breakout board. In the initial signal phase, representing device advertising, there is a consistent low consumption interspersed with brief spikes every 2 seconds due to packet transmission. A notable increase to around seven mA signifies the main image transmission phase. During periods preceding and succeeding this phase, increased data exchange with the gateway is observed, as the connection interval is shorter than the advertisement interval. With our compression method that reduces the image to a quarter of its original size, the time required to exit sleep mode decreases, resulting in reduced energy consumption and transmission time.

Figure 11 displays the restoration of multiple images compressed with Haar L1. As part of our future work, we plan to train the SRGAN to restore images compressed using Haar L2. While Haar L2-compressed images exhibit lower quality than those using Haar L1 or JPEG, they offer superior size reduction, potentially leading to significant energy savings.

VII. Conclusion

In this paper, a novel lossy crypto-compression scheme tailored for resource-constrained MIoT devices is introduced, aiming for optimal performance with minimal overhead. Central to our proposal is using the lossy row-column integer Haar transform, a memory-efficient compression method particularly suited for microcontrollers with limited memory and computational capabilities. This compression is performed before encryption, to ensure data compactness. Further, our scheme exploits the super-resolution denoising model and the pre-shared working key to generate a dynamic key (a new key is generated after a specific period). The produced dynamic key is used to derive simple cryptographic primitives that change frequently depending on the configuration; they can be updated for each new input image or set of images. In particular, the cryptographic primitives consist of one/two initial vector(s), a permutation table, and two update permutation tables, each used to update one cryptographic primitive.

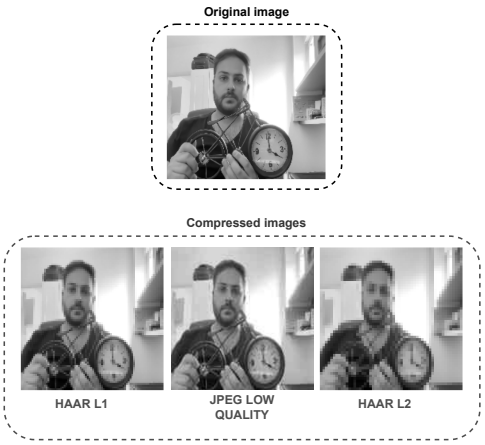


Fig. 9: Visual degradation due to Haar L1 compression, low-quality JPEG, and Haar L2 compression.

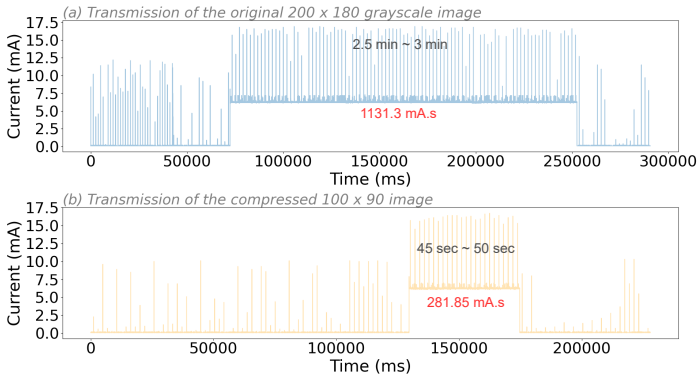


Fig. 10: Power consumption from transmitting an original 200 x 180 grayscale image using Bluetooth Low Energy, compared to a compressed image with Haar L1.

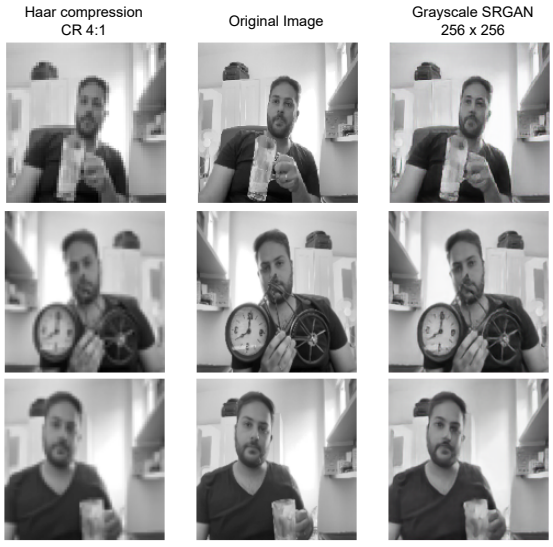


Fig. 11: Restoration and noise removal of various images compressed using Haar L1, utilizing the SRGAN.

The proposed updating process, which is based on the permutation process, significantly increases the robustness and security level of the scheme (linear computational overhead). The existing security mechanisms in MIoT are based on the multi-round scheme such as AES. In contrast, the proposed scheme is implemented and requires only one round and uses simple operations. The security and performance tests demonstrate the robustness and efficiency of the proposed scheme.

REFERENCES

[1] Bank, L.: JPEGENC github repository (2023), last visited: 12/09/2023

[2] Benlcouiri, Y., Benabdellah, M., Ismaili, M., Azizi, A.: Crypto-compression of images based on the anns and the aes algorithm. International Journal of Communications and Computer Engineering **2**(3), 1–6 (2012)

[3] Campobello, G., Segreto, A.: A low complexity image compression algorithm for iot multimedia applications. In: 2019 27th European Signal Processing Conference (EUSIPCO). pp. 1–5. IEEE (2019)

[4] Christopoulos, C., Skodras, A., Ebrahimi, T.: The jpeg2000 still image coding system: an overview. IEEE transactions on consumer electronics **46**(4), 1103–1127 (2000)

[5] Cidjeu, D., Tieudjo, D.: Some elliptic curve based crypto-compression schemes for medical images. International Journal of Advanced Research in Computer and Communication Engineering: Vol. **8**(5) (2019)

[6] Fisch, M.M., Stögner, H., Uhl, A.: Layered encryption techniques for dct-coded visual data. In: 2004 12th European Signal Processing Conference. pp. 821–824. IEEE (2004)

[7] Hu, P., Im, J., Asgar, Z., Katti, S.: Starfish: Resilient image compression for aiot cameras. In: Proceedings of the 18th Conference on Embedded Networked Sensor Systems. pp. 395–408 (2020)

[8] Kouadria, N., Doghmane, N., Messadeg, D., Harize, S.: Low complexity dct for image compression in wireless visual sensor networks. Electronics Letters **49**(24), 1531–1532 (2013)

[9] Krishnaraj, N., Elhoseny, M., Thenmozhi, M., Selim, M.M., Shankar, K.: Deep learning model for real-time image compression in internet of underwater things (iout). Journal of Real-Time Image Processing **17**, 2097–2111 (2020)

[10] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4681–4690 (2017)

[11] Lee, S.W., Kim, H.Y.: An energy-efficient low-memory image compression system for multimedia iot products. EURASIP Journal on Image and Video Processing **2018**, 1–15 (2018)

[12] Noura, H., Chehab, A., Noura, M., Couturier, R., Mansour, M.M.: Lightweight, dynamic and efficient image encryption scheme. Multimedia Tools and Applications **78**, 16527–16561 (2019)

[13] Noura, H.N., Noura, M., Chehab, A., Mansour, M.M., Couturier, R.: Efficient and secure cipher scheme for multimedia contents. Multimedia tools and applications **78**, 14837–14866 (2019)

[14] Puech, W., Rodrigues, J.M.: Crypto-compression of medical images by selective encryption of dct. In: 2005 13th European signal processing conference. pp. 1–4. IEEE (2005)

[15] Savakis, A., Piorun, M.: Benchmarking and hardware implementation of jpeg-ls. In: Proceedings. International Conference on Image Processing. vol. 2, pp. II–II. IEEE (2002)

[16] Tang, L.: Methods for encrypting and decrypting mpeg video data efficiently. In: Proceedings of the fourth ACM international conference on Multimedia. pp. 219–229 (1997)

[17] Van Droogenbroeck, M., Benedett, R.: Techniques for a selective encryption of uncompressed and compressed images. In: Advanced Concepts for Intelligent Vision Systems (ACIVS) (2002)

[18] Wu, X., Memon, N.: Calic-a context based adaptive lossless image codec. In: 1996 IEEE international conference on acoustics, speech, and signal processing conference proceedings. vol. 4, pp. 1890–1893. IEEE (1996)