Spot encryption: An area-of-interest photographic encryption technique

Malek Ayesh
Department of Computer Science and Engineering
Qatar University
ma1509162@qu.edu.qa

Abdulhady Fezooni
Department of Computer Science and Engineering
Qatar University
af1403357@qu.edu.qa

Layth Hamad
Department of Computer Science and Engineering
Qatar University
lh2009770@qu.edu.qa

***Abstract*— With cameras being used everywhere, modern techniques for computation and memory-usage must be implemented for efficient and optimal performance. For instance, real-time surveillance requires little delay therefore, the encryption technique applied should be light weight to maintain good Quality of Experience (QOE) and relatively strong security. This project proposes the use of traditional block and stream cipher techniques, AES and RC4 respectively, to heavily encrypted a detected region of interest in a photograph while quickly encrypting the background region. To accomplish this, the region of interest will be encrypted by either one of the following AES modes; Electronic Code Book (ECB), Cipher Block Chaining (CBC), and Counter (CTR) while the background encryption will be done using RC4.**

Keywords—object detection, region-of-interest, cryptography, hybrid encryption, AES, RC4, ECB, CBC, CTR, Python.

I. Introduction

Security is the single largest bottle neck for network services [1]. Despite security techniques ensuring the safety of individuals and resources over networks, it also may negatively affect the Quality of Experience perceived by clients [2]. Moreover, with media applications' global popularity on the rise, security techniques must satisfy user requirements and meet their desired needs [3]. On the other hand, attackers are constantly evolving their arsenals; therefore, it is necessary to decrease their probability of a successful attack [4]. With the aforementioned in mind, the goal of this study is to construct an encryption technique which detects regions of interest and 'heavily' encrypt them, using AES-CBC, and 'lightly' encrypting the background using RC4. Furthermore, the encrypted regions of interest will use lossless compression while the background region will use lossy compression. By doing this, the aim is to construct a technique which is closely comparable to RC4's speed while being relatively more secure than it.

By conducting this study, we aim to answer the following scientific questions. (1) How does the proposed hybrid technique compare, to using only using RC4 or only AES, in terms of security? (2) Is the proposed technique faster or slower than AES and RC4 respectively? Is this performance difference significant? (3) In terms of bandwidth consumption, how does the proposed technique compare to AES and RC4? (4) What are some possible practical applications for this proposed technique?

Primarily this study will utilize Python for object detection, compression, encryption, decryption, and performance evaluation. This is due to the plentiful availability of libraries and resources.

The outline of this study will be as follows. Section 2 discusses related work. section 3 provides some background on vital concepts related to this study. Section 4 details the experimental setup and methodology while section 5 describes the results, followed by section 6 which discusses the outcome of the simulation. Finally, section 7 mentions future work and concludes the study.

II. Related work

When it comes to object detection, YOLO is a relatively excellent candidate. It provides real-time object detection and can be run on both CPU and GPU [5][6].

In [7], a light YOLO object detector than run on GPU was proposed. Their training was first done on PASCAL VOC, and achieved 33.81% mean average precision, followed by the COCO dataset 12.26%. [8] proposed a lossless image compression technique which outperformed L3C, PNG, WebP, and JPEG 2000. This was done by using a Convolutional Neural Network based model.

Multiple studies investigated the performance of AES with RC4. These studies had different approaches and metrics that determined the better performer. Study [9] compared the performance of one block cipher technique and two stream cipher techniques namely, AES, RC4, and XOR respectively. The study looked into real-time MPEG-1 video streaming and evaluated the chosen ciphers in-terms of time to send packets and packet delay. The study concluded that AES could provide satisfactory performance for MPEG-1 applications and that its overhead is less than that of RC4 and XOR. An optimized implementation of AES modes was conducted in [10]. The study looked at the AES modes implementation on wireless network sensors. Furthermore, they considered ECB, CBC, CFB, and CTR modes and considered execution time and energy efficiency as their evaluation metrics. Their research concluded that if encryption is needed and is the at most then CFB and CTR are the best candidates. Otherwise, CBC is satisfactory. In [11], only AES-128 ECB was

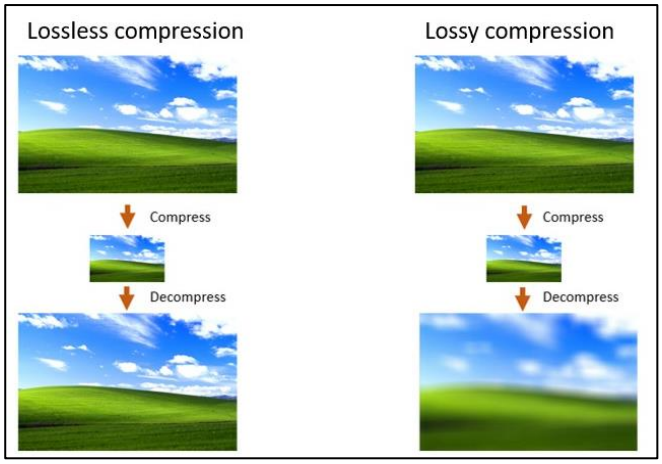

Figure 1. Lossless versus Lossy compression.

evaluated on three different GPU architectures namely, Kepler, Maxwell, and Pascal. Calculating speed is the only metric considered and the results show that 207 Gbps was achieved in the Maxwell architecture (NVIDIA GTX TITAN X) on the other and, 280 Gbps on the Pascal architecture (NVIDIA GTX 1080). A hardware implementation of RC4 was carried out in [12]. Different optimization strategies were placed such as hardware pipelining and loop unrolling. VHDL was used to the implementation and the study achieved of speeds up to 30.72 Gbps for a clock speed of 1.92GHz. [13] aims to make a more effective RC4 cipher by combining the work of Jian Xie et al and T.D.B Weerasinghe. The analysis was done using Shannon's theory of secrecy. The purpose of the proposed method is to balance between RC4's high speed and its relatively lower security, when compared to other encryption techniques. To achieve this, randomness was implemented in RC4 to achieve higher secrecy along with parallelism to maintain RC4's notorious speed.

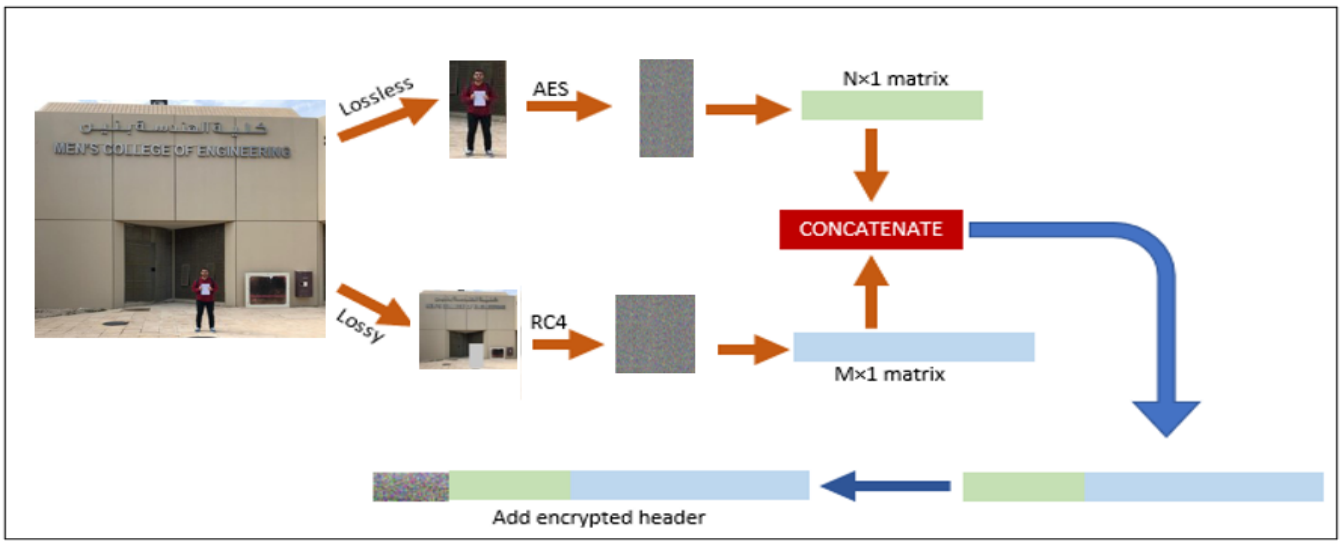

Figure 2. Sender's side.

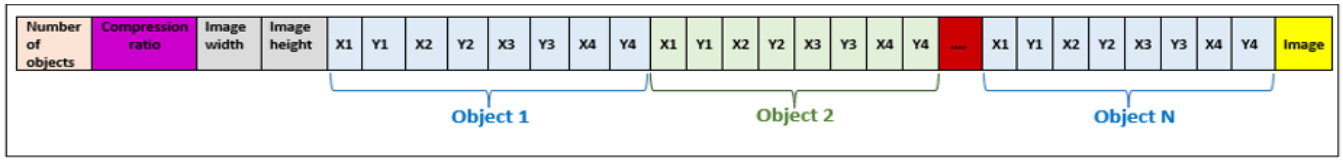

Figure 3. Header format.

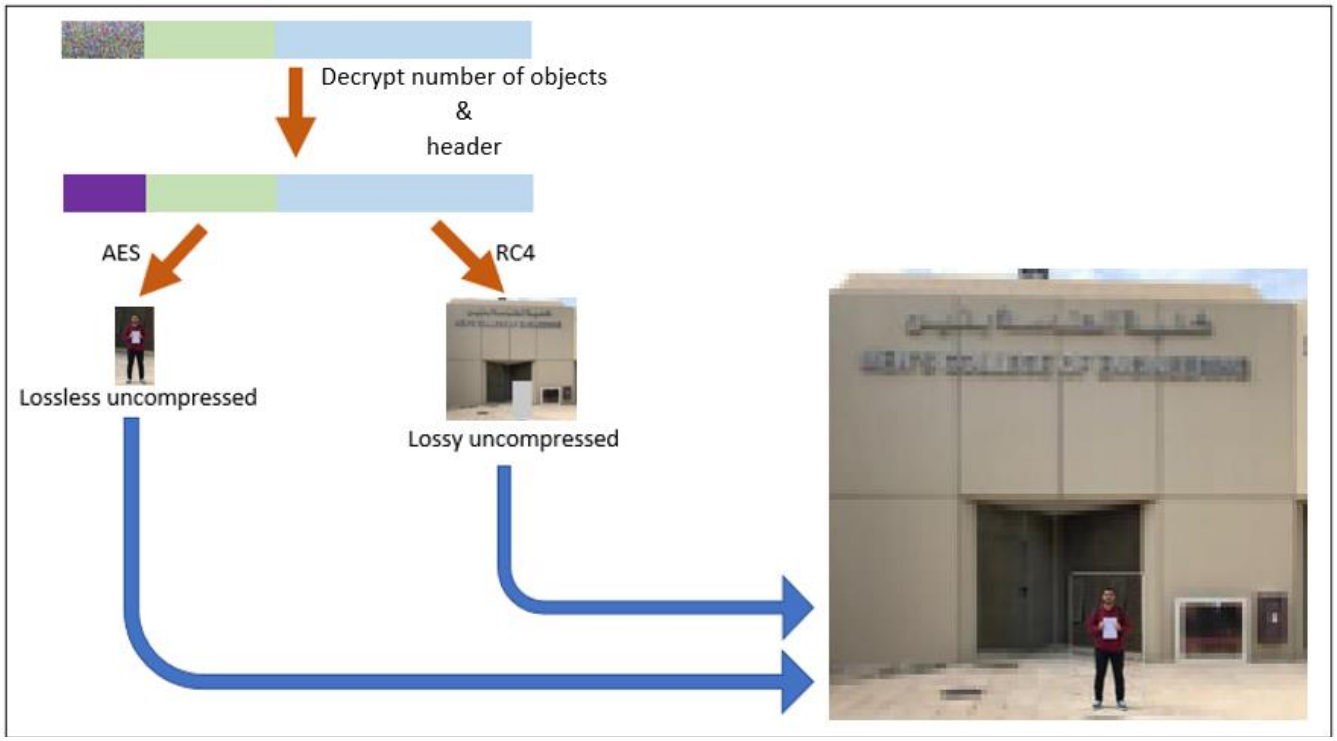

Figure 4. Receiver's side.

III. Background

In order to grasp the significance of this work, some vital concepts related to compression and encryption need to be discussed. Compression is an encoding technique which aims to represent an image with a smaller number of bits [14]. This proposed technique focuses on lossy and lossless compression. Lossy compression loses some of its information during processing [15]. On the other hand, lossless compression maintains all data, without any loss, after processing [16]. A comparison between the two compression techniques can be seen in figure 1. Encryption is a method of concealing data in a way such that only authorized people may access that information. Data is converted from a format comprehendible by humans, plaintext, to one that is not, ciphered text. Despite the ciphered text appearing random, a plaintext can be retrieved if the correct algorithm and key(s) are used. There are two main categories of ciphers:

A. Block ciphers:

Ciphers that segment data into equal size blocks to be encrypted. They are usually utilized to encrypt large data. The encryption process uses a variable sized key for the fixed size blocks on which the operational modes can be integrated with.

B. Stream ciphers:

Stream ciphers encrypt the entire data at once. Stream ciphers usually heavily rely on the XOR operation. RC4 is an example of stream cipher.

Two ciphers will be used in this study. AES, which is a block cipher, and RC4, a stream cipher:

A. Advanced Encryption Standard (AES):

Is a cipher that uses private key encryption. Furthermore, it is an example of block ciphers. AES uses the substitution and permutation techniques to cipher information.

Operation modes are integrated with block ciphers. Their purpose is to deal with larger amounts of data by allowing the encryption/decryption of more than one block at a time. Furthermore, operating modes enhance security by usually ensuring that no two encrypted blocks have the same ciphered text. In this paper, we will look at three different operational modes:

- Electronic Code Book (ECB): It segments the large data into smaller and more manageable blocks. ECB supports parallelism therefore it is fast, but two blocks may have the same ciphered text as future outputs do not depend on past inputs. This provides lower security.
- Cipher Block Chaining (CBC): Blocks are dependent of one another therefore; better security is provided. Each block is XORed the previous ciphered block before encrypting it. As the first block does not have a previous block, a random value named Initialization Vector (IV) is employed. The secrecy of the IV is not important furthermore, the use of an IV makes CBC probabilistic.
- Counter (CTR): ECB is fast but does not provide strong protection on the other hand, CBC provides strong protection but is slow. One mode that mediates between ECB and CBC is CTR. CTR's speed is fast as it supports parallel computation furthermore, the computation of the ciphered text relies on a counter value chosen.

B. Rivest Cipher 4 (RC4):

RC4 is a stream cipher technique. RC4 is a fast cipher as it encrypts byte-by-byte. It is widely used and is known for the simplicity of its operation. Randomly generated pseudo random bits are generated which are then combined with the plaintext, using the XOR operation, to obtain a ciphered text.

IV. Experimental setup and methodology

The proposed technique will be implemented as follows:

The sender detects the object-of-interest, using the YOLOv4, which bounds that targeted object in a box. The bounded object will be removed from the image and compressed using lossless compression, encrypted using one of the test AES modes and then reshaped into a N×1 matrix. The 'Background', which is the original image but without the object of interest is compressed using lossy compression, encrypted using RC4, and converted into a M×1 matrix. A header, which contains crucial information about the image is encrypted using AES along with the number of objects of interest. This header is what allows the receiver to retrieve and reconstruct the sender's image. Finally, the encrypted header, the N×1 matrix, and the M×1 matrix are concatenated together and send to the receiver. Further clarification can be found in figures 2.

The image sent must be accompanied by the header, shown in figure 3. The header contains information vital to the reconstruction of the image at the receiver's side. It consists of number of objects-of interest, compression ratio, image width and height, and the objects-of-interest bounding box coordinates.

At the receiver, the encrypted header is first decrypted. The N×1 matrix is decrypted, uncompressed, and reshaped to produce a clear image with no pixel loss. The M×1 matrix is also decrypted, uncompressed, and reshaped to produce an image with some information loss. Using the header, the object of interest is reinserted back into the background image. More information is provided in figure 4.

The proposed technique can detect multiple objects of interest. The object of interest must be bounded by a generic shape and not one that borders the encrypted object-of-interest. This is so not to reveal its silhouette, preventing an attacker from obtaining any information by just looking at the encrypted image. Furthermore, the proposed approach will use AES-CBC for the area of interest and RC4 for the background. AES-CBC is used over ECB as it is more secure [17] and CBC is used over CTR due to CTR being susceptible to bit-flip errors [18].

To observe and quantify the performance of the proposed approach, comparisons with existing techniques need to conduct. This experimental setup is as follows:

1. Test RC4 on the entire image using lossless compression.
2. Test AES, with ECB, CBC, and CTR modes, along with lossless compression.
3. Test the proposed approach.

The performance of the proposed technique will be evaluated based on multiple factors. These include:

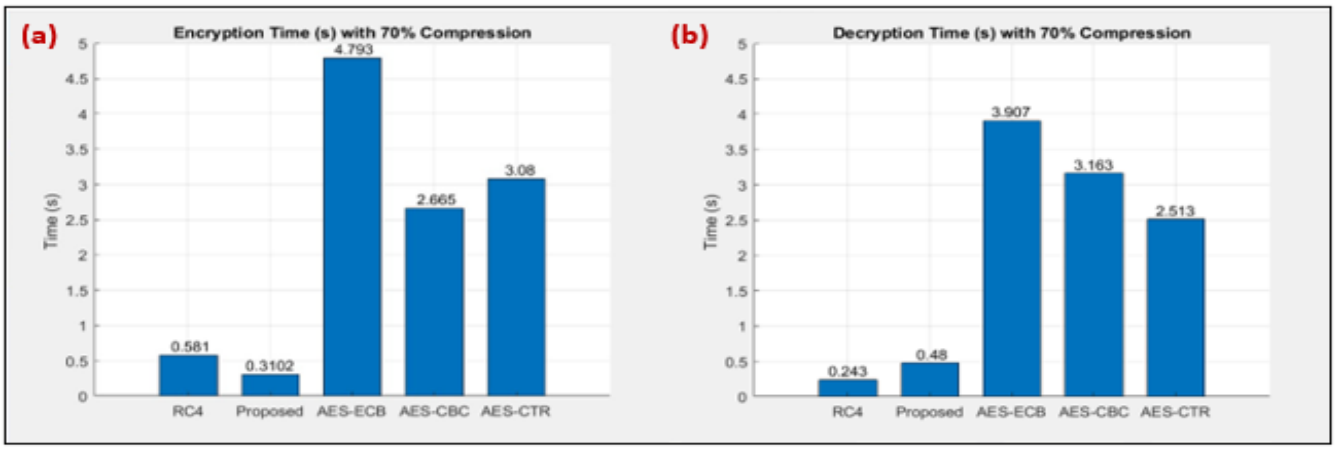

Figure 5. (a) illustrates the encryption times of the different competitors, with 70% compression while (b) shows the decryption times with 70% compression.

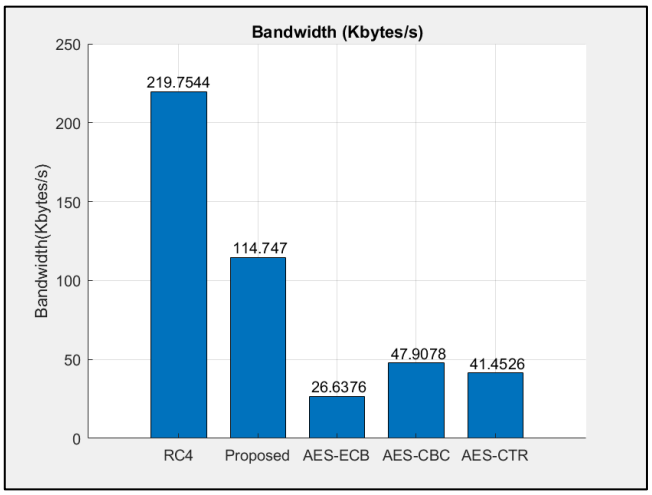

Figure 6. Competitors' bandwidth comparison.

1. Encryption and decryption times: This aims to compare the performance of the proposed technique with its competitors with respect to encryption and decryption computational time.
2. Bandwidth consumption; Proposed versus competitors: This test aims to illustrate how the proposed method competes with its competitors in terms of bandwidth.
3. Compression effects on proposed method: This test demonstrates how different compression factors affects the size of output image at the sender's side.
4. Bandwidth versus compression percentage on proposed technique: Aims to show how bandwidth can drastically vary as the compression factor changes.

V. Results

This section displays how the proposed method compares to preexisting ones. This will be done by calculating the previously mentioned performance metrics.

A. Encryption and decryption times:

Measured in seconds, figure 5a and figure 5b show how different methods compare. All the methods in this test were compressed by 70%. RC4 and the AES modes all used lossless compression while the proposed method uses a combination of lossy and lossless compression.

B. Bandwidth consumption; Proposed versus competitors:

Shown in figure 6, the comparison between the bandwidth, in Kbytes/s, of each of the competitors and the proposed method. This test illustrates the speed each method is capable of.

C. Compression effects on proposed method:

This test solely investigates the proposed method's final packet size variation with respect to the compression percentage. This test is shown in figure 7a.

D. Bandwidth versus compression percentage on proposed technique:

The aim of this test, shown in figure 7b is to clarify how bandwidth increases as the compression percentage increases.

VI. Results' Discussion

This section aims analyze, compare, and criticize the results produced in the previous section. Figure 5a and figure 5b compare RC4, the proposed, and AES with its three modes with respect to encryption and decryption time. The slowest in terms on encryption time is AES-ECB. The proposed method proved to be the fastest out of all its competitors with respect to encryption time. The proposed method, which took 0.31 seconds, was around 47% faster than the second-best performing method, RC4, which took 0.58 seconds. When decrypting, RC4 managed to beat the proposed method as it only took 0.24 seconds while the proposed took 0.48 seconds. This is an approximate difference of 49%. AES-ECB was once again the slowest out of all the competitors. The fastest AES mode is CTR and even then, it is more than five times slower than the proposed approach.

Figure 6 shows how the different competitors compare to one another. At 219.75 Kbytes/s, RC4 is the best performer. The second-best contestant is the proposed method which comes in at 114.75 Kbytes/s. This is an approximate performance difference of 48%. Despite this, the proposed approach has a bandwidth of around 2.4 times better than the third-place competitor, AES-CBC. AES-CBC was followed by AES-CTR and then finally by AES-ECB which had a bandwidth of only 26.64 Kbytes/s, which is approximately 4.4 times lower than the proposed method.

Figure 7a and figure 7b only study the proposed approach. First, shown in figure 7a, is how when the compression percentage increases, the transfer image size decreases. The

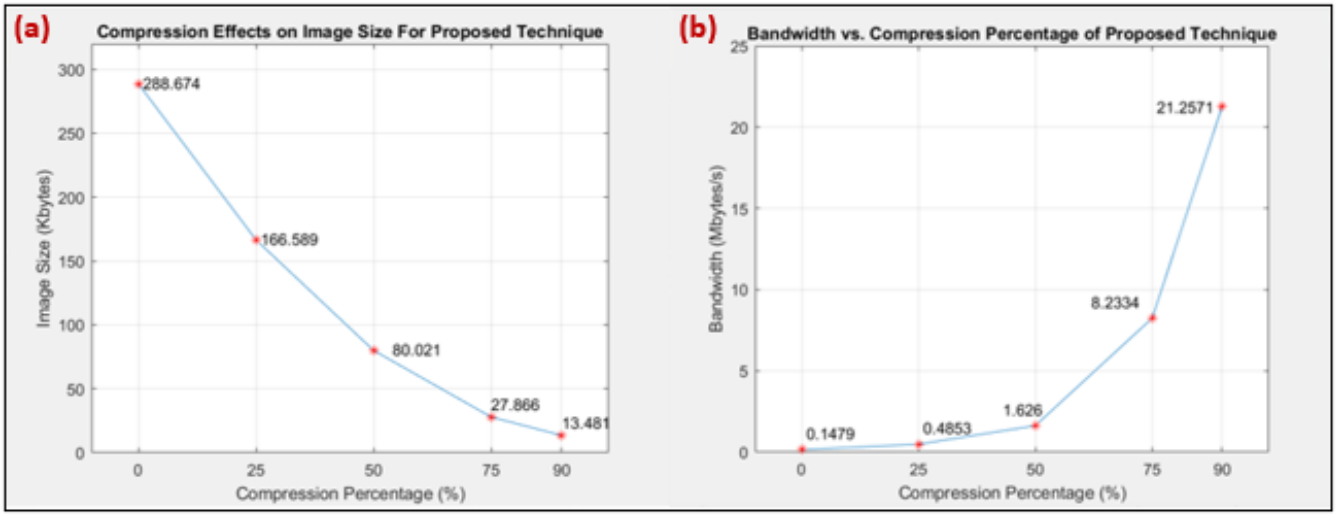

Figure 7. (a) Shows the compression effects on an image using the proposed technique. (b) illustrates the bandwidth variation with compression percentage.

Figure 8. Visual effect of compression on the background using the proposed method. (a) 0% compression, (b) 25% compression, (c) 50% compression, (d) 75% compression, and (e) 90% compression.

decrease in images size is exponentially decreasing. When the original image is not compressed, at 0%, the image size remains the same, 288.67 Kbytes. When looking at 90% compression, the final image size is reduced to 13.48 Kbytes, which is around 21 times less than the original image. Despite this drastic decrease in the output image size, the background is nearly indistinguishable. The variation of the output images, at the sender, as compression percentages vary are shown in figure 8.

Secondly, figure 7b shows how as compression percentage increases, the bandwidth also increases. In this case, the increase is exponential. No compression of the image at the sender's side leads to a small bandwidth, of only 0.15 Mbytes/s. This bandwidth value increases to 0.49, 1.63, 8.23, and 21.26 Mbytes/s for 25, 50, 75, and 90 percent compression. This means that 90% compression has a larger bandwidth than 0% compression by a factor of 145.

With the results in mind, many critical outcomes can be deduced. Fist, the proposed technique compares very well to its rivals, with respect to encryption and decryption. In fact, the proposed approach's performance was overall slightly more superior than RC4's, when considering the combined encryption and decryption times. Second, the bandwidth of the proposed method was less than RC4. Despite this, the proposed approach was significantly better than the different AES modes. Third, depending on the user's application, a higher compression percentage can be used to provide a smaller sent packet size and a larger output bandwidth. With this in mind, the user needs to determine how important the background is important to them. If the background is

significant for the user, then a lower compression percentage should be used. And finally, the demonstrated proposed method has slightly lower performance than RC4 but due to the hybrid encryption technique, it should provide better security.

VII. CONCLUSION AND FUTURE WORK

This study investigates a proposed hybrid encryption method which detects an area of interest with an object detection program, such as YOLO. This area of interest is compressed using lossless compression and encrypted using AES-CBC, while the background is encrypted lossy compression and encrypted using RC4. The aim of this study is to find an encryption technique that closely compares to the fast speed of RC4 but is relatively more secure. Different performance tests were conducted namely, encryption and decryption times, bandwidth consumption; proposed versus competitors, compression effects on proposed method, bandwidth versus compression percentage on proposed technique. It was concluded that the proposed approach is slightly outperformed by RC4, in terms of speed, but not security. On the other hand, the proposed approach performs significantly better than the different AES modes, in terms of speed, with comparable security level within the area-of-interest.

Future work includes testing more ciphers against the proposed approach along with a greater focus on security metrics. Furthermore, work will be done to adopt a dynamic bandwidth sensing approach where the compression level and the cipher technique will be autonomously adjusted based on the channel quality.

REFERENCES

[1] N. R. Potlapally, S. Ravi, A. Raghunathan, and G. Lakshminarayana,"Optimizing public-key encryption for wireless clients," in2002 IEEEInternational Conference on Communications. Conference Proceedings.ICC 2002 (Cat. No. 02CH37333), vol. 2. IEEE, 2002, pp. 1050–1056.

[2] C. Lorentzen, M. Fiedler, H. Johnson, J. Shaikh, and I. Jørstad, "On userperception of web login—a study on qoe in the context of security,"in2010 Australasian Telecommunication Networks and ApplicationsConference. IEEE, 2010, pp. 84–89.

[3] J.-B. Wang, H. Yang, M. Cheng, J.-Y. Wang, M. Lin, and J. Wang, "Jointoptimization of offloading and resources allocation in secure mobileedge computing systems,"IEEE Transactions on Vehicular Technology,vol. 69, no. 8, pp. 8843–8854, 2020.

[4] C. Xie, Y. Wu, L. v. d. Maaten, A. L. Yuille, and K. He, "Featuredenoising for improving adversarial robustness," inProceedings of theIEEE/CVF Conference on Computer Vision and Pattern Recognition,2019, pp. 501–509.

[5] M. B. Ullah, "Cpu based yolo: A real time object detection algorithm,"in2020 IEEE Region 10 Symposium (TENSYMP).IEEE, 2020, pp.552–555.

[6] N. Artamonov and P. Yakimov, "Towards real-time traffic sign recog-nition via yolo on a mobile gpu," inJournal of Physics: ConferenceSeries, vol. 1096, no. 1. IOP Publishing, 2018, p. 012086.

[7] R. Huang, J. Pedoeem, and C. Chen, "Yolo-lite: a real-time objectdetection algorithm optimized for non-gpu computers," in2018 IEEEInternational Conference on Big Data (Big Data).IEEE, 2018, pp.2503–2510.

[8] F. Mentzer, L. V. Gool, and M. Tschannen, "Learning better losslesscompression using lossy compression," inProceedings of the IEEE/CVFConference on Computer Vision and Pattern Recognition, 2020, pp.6638–6647.

[9] W. S. Elkilani and H. M. Abdul-Kader, "Performance of encryption tech-niques for real time video streaming," in2009 International Conferenceon Networking and Media Convergence. IEEE, 2009, pp. 130–134.

[10] C. Panait and D. Dragomir, "Measuring the performance and energyconsumption of aes in wireless sensor networks," in2015 FederatedConference on Computer Science and Information Systems (FedCSIS).IEEE, 2015, pp. 1261–1266.

[11] A. A. Abdelrahman, M. M. Fouad, H. Dahshan, and A. M. Mousa, "Highperformance cuda aes implementation: A quantitative performance anal-ysis approach," in2017 Computing Conference. IEEE, 2017, pp. 1077–1085.

[12] S. S. Gupta, A. Chattopadhyay, K. Sinha, S. Maitra, and B. P. Sinha,"High-performance hardware implementation for rc4 stream cipher,"IEEE Transactions on Computers, vol. 62, no. 4, pp. 730–743, 2012.

[13] T. Weerasinghe, "An effective rc4 stream cipher," in2013 IEEE 8thinternational conference on industrial and information systems. IEEE,2013, pp. 69–74.

[14] J. Ziv and A. Lempel, "A universal algorithm for sequential datacompression,"IEEE Transactions on information theory, vol. 23, no. 3,pp. 337–343, 1977.

[15] A. Said and W. A. Pearlman, "An image multiresolution representationfor lossless and lossy compression,"IEEE Transactions on imageprocessing, vol. 5, no. 9, pp. 1303–1310, 1996.

[16] K. Sayood,Lossless compression handbook. Elsevier, 2002.

[17] D. Blazhevski, A. Bozhinovski, B. Stojchevska, and V. Pachovski,"Modes of operation of the aes algorithm," 2013.

[18] H. Lipmaa, P. Rogaway, and D. Wagner, "Ctr-mode encryption," inFirstNIST Workshop on Modes of Operation, vol. 39, 2000.