BIT-STREAM-BASED ENCRYPTION FOR REGIONS OF INTEREST IN H.264/AVC VIDEOS WITH DRIFT MINIMIZATION

Andreas Unterweger[⋆]*, Jan De Cock*[†]*, Andreas Uhl*[⋆]

[⋆] University of Salzburg, Department of Computer Sciences, Salzburg, Austria
[†] Ghent University – iMinds, ELIS – Multimedia Lab, Ledeberg-Ghent, Belgium

ABSTRACT

We propose a new encryption approach for regions of interest in H.264/AVC bit streams. By encrypting at bit stream level and applying drift minimization techniques, we reduce the processing time by up to 45% compared to full re-encoding. Depending on the input video quality, our approach induces an overhead between -0.5 and 1.5% (high resolution sequences) and -0.5 and 3% (low resolution sequences), respectively, to minimize the drift outside the regions of interest. The quality degradation in these regions remains small in most cases, and moderate in a worst-case scenario with a high number of small regions of interest.

Index Terms— H.264/AVC, RoI, Encryption, Drift, Bit Stream

1. INTRODUCTION

In surveillance videos, regions of interest (RoI) like people's faces are often encrypted in order to preserve their privacy. The images are not encrypted entirely so that people's actions can still be seen unencrypted by surveillance personnel to determine whether intervention is required. In addition, decryption allows law enforcement to recover the identities later if necessary. This is not possible with other forms of de-identification like pixelation or blurring (e.g. [1, 2, 3]). An example of a surveillance system with RoI encryption is depicted in Fig. 1.

In this paper, we assume a video surveillance scenario where a typical surveillance camera (as of 2015) outputs compressed videos in the form of Motion JPEG [4] or H.264/AVC bit streams [5] and provides information on the location of the RoI through face detection to the encryption system as in [6].

Many approaches for RoI encryption have been proposed. Most of them either perform encryption format-independently in the image domain (e.g., [7, 8, 2]) or format-family-dependently in the transform domain (e.g., [9, 10, 11]). However, this is not practical since neither the captured images nor the encoder within the camera can be modified in typical surveillance equipment. The alternative of applying these methods by decoding the video stream received from the camera and re-encoding it is considered too time consuming and therefore impractical.

Approaches for bit-stream-based RoI encryption are very sparse. Although solutions for Motion JPEG exist [12, 13, 14], all of the H.264/AVC-focused approaches have severe disadvantages. Note that we only consider RoI encryption approaches, i.e., those which aim at maintaining the visual information outside of the RoI.

Dufaux et al. [15] describe an approach for MPEG-4 Part 2 which can be extended to H.264/AVC (and other DCT-based formats). Although their encryption algorithm can be performed at bit stream level, their method of preventing drift, i.e., the propagation of encrypted pixels into non-RoI areas, requires selectively re-encoding the video from the first encrypted frame onwards. As explained above, this is impractical due to its high computational complexity.

The approaches of Iqbal et al. [16] and Unterweger et al. [6] require bit streams in which all RoIs are in separate slices or slice groups to prevent spatial drift through the imposed prediction borders. This is a serious restriction since it requires the camera to reliably detect RoI and to create according slices. Furthermore, the authors do not discuss temporal drift, which is an important issue addressed in this paper.

Our work aims at minimizing the amount of drift after encryption while significantly reducing the computational complexity required for processing. This way, we contribute an alternative to the infeasible full re-encoding techniques at the cost of a small amount of drift outside the RoI.

This paper is structured as follows: In Section 2, we present our encryption approach. In Section 3, we describe how we minimize drift. Finally, we present a practical evaluation of our method in Section 4 before concluding the paper.

2. ENCRYPTION APPROACH

For all blocks inside a RoI, we perform a three-step bit-stream-level coefficient encryption as follows: First, we change the signs of all AC coefficients by xor-ing the original sign bits with the output of a one-time pad. Second, we encrypt the DC coefficient signs in the same way if the coefficients are stored directly in the bit stream, i.e., if the processed block does not use $16 \cdot 16$ intra prediction, where

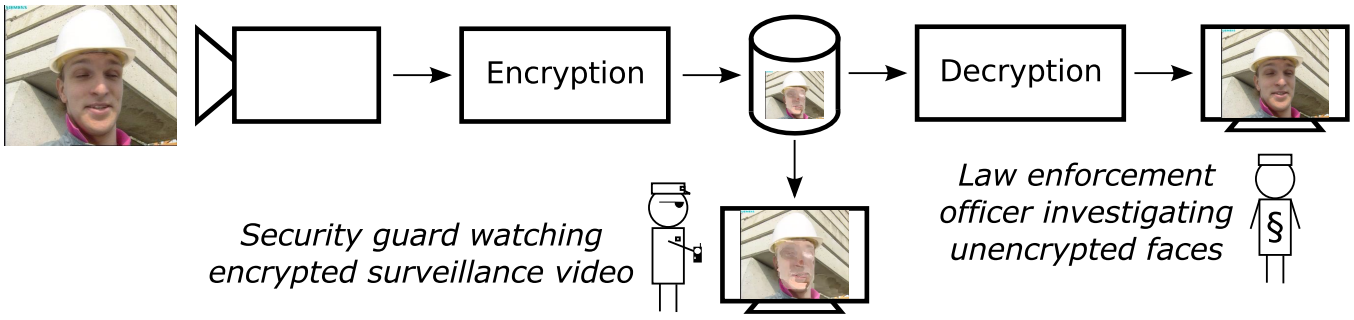

Fig. 1. Surveillance system use case: Images captured by a surveillance camera are encrypted and stored in a database. From there, the encrypted images can either be viewed directly or decrypted and used for legal investigations.

an additional Hadamard transform is used (which makes it impossible to change the DC coefficient signs at bit stream level). Third, we change the least significant bit of all AC coefficients whose absolute value is greater than one. This can be done by entropy code word replacement at bit stream level.

We also perform encryption on all blocks which are not fully, but only in parts inside of a RoI. This assures that no information at the borders of RoI leaks at the cost of encrypting a small number of non-RoI pixels as well. We limit the encryption to the luminance channel since hardly any identification-related information is contained in the chrominance channels and extending our approach would be trivial.

When applying encryption, spatial and temporal drift occurs due to the difference between the encrypted and the unencrypted pixel values which are used for prediction. As illustrated in Fig. 2 (middle-left), this affects regions around the RoI (e.g., on the top-right part of the helmet in the bottom picture), disfiguring them to an extent that they cannot be used any more for video surveillance and similar applications.

Thus, it is necessary to minimize drift. One way to do so is full re-encoding, offering perfect compensation (see Fig. 2, middle-right) at infeasible computational complexity. Therefore, we subsequently propose an approach which aims at minimizing drift at moderate computational complexity.

3. DRIFT MINIMIZATION

When encrypting macroblocks in the RoI, we have to take into account the dependencies that arise due to spatial and temporal prediction. Simply modifying the residual coefficients in a macroblock will affect the surrounding macroblocks due to (spatial) intra prediction, and due to (temporal) motion-compensated prediction.

When fully re-encoding the sequence, the original sequence is first decoded, and subsequently re-encoded using the same mode and motion information as in the input bitstream. In this case, the RoI encryption is embedded after the second (encoding) loop. The complexity of such an approach, however, will be high given the two prediction loops. We use this approach as a benchmark.

In the past, reduced-complexity transcoding approaches have been presented that were based on single-loop or open-loop architectures, mainly in the context of transrating [17]. Open-loop techniques can be used to perform modifications to coefficients with minimal complexity, but are unable to compensate for potential error drift. A *single-loop* approach, however, can compensate for the drift by storing the differences between the reconstructed coefficients before and after encryption. These differences can afterwards be used for intra prediction (IP) and/or motion-compensated prediction (MCP) in dependent blocks.

In this paper, we introduce such a compensation loop in the encryption framework, resulting in a significant reduction of the drift in the macroblocks which are dependent on the RoI area. In our framework, we make a distinction between three types of macroblocks:

- Macroblocks that are not inside the RoI or dependent on the RoI can be processed open-loop (i.e., by performing an entropy decoding and encoding step only), without further calculations.

- For the RoI macroblocks, the encryption approach introduced in Section 2 is performed. Additionally, the differences between the (inverse quantized and transformed) coefficients before and after encryption are calculated and accumulated, as illustrated in Fig. 3. The accumulation is performed along the direction of the intra prediction, or along the motion-compensated prediction direction. The resulting accumulated values are only stored within the RoI, but are not yet used for compensation. Hence, the coefficients for the RoI blocks are only modified by the encryption process, with no impact on the bit rate.

- For the macroblocks that are directly dependent on the RoI (either through intra prediction or motion-compensated prediction), single-loop compensation is applied and the differences that were accumulated in the RoI are used to compensate for the error drift. This

Fig. 2. First (top) and eleventh (bottom) frame of the *foreman* sequence (left-most) with QP 27 and $IP*$ GOP structure. The face as RoI is encrypted without drift compensation (middle-left), full compensation (middle-right) and our approach (right-most), respectively. Our approach shows only very little drift outside the RoI and is comparable to full compensation (middle-right).

will lead to a small increase in the bits required to code these blocks, as discussed in Section 4.3. The single-loop compensation loop is illustrated in Fig. 4. By applying compensation, we notice that a significant portion of the drift is eliminated for the non-RoI area.

An overview of the error accumulation and compensation process for the second and third types of macroblocks is given in Fig. 5, for the case of spatial error propagation (intra prediction). The non-colored macroblocks correspond to the first type of macroblocks and can be processed open-loop.

It has to be noted that the drift compensation will not be perfect, due to non-linear operations in the H.264/AVC/AVC encoder and decoder loops. For example, the division/shift operations during intra prediction (modes 3-8) and motion compensation, along with clipping at the boundaries of the pixel value range (for 8 bits, between 0 and 255) can lead to rounding errors in the compensation loop. As such, we can still encounter drift errors (albeit to a smaller extent) outside the encrypted RoI.

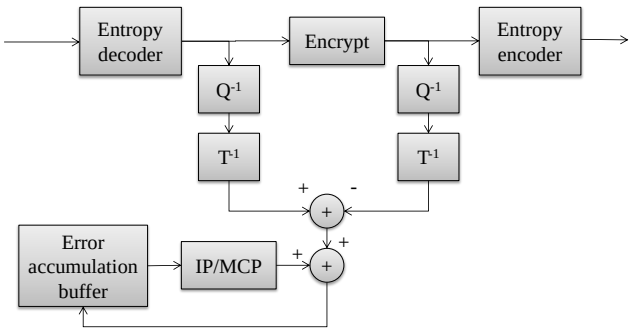

Fig. 3. Schematic transcoder architecture for RoI macroblocks (error accumulation): Differences between the encrypted and unencrypted values are accumulated in an error buffer. $Q(^{-1})$: (inv.) quantization, $T(^{-1})$: (inv.) transform, IP: intra prediction, MCP: motion-compensated prediction.

4. EVALUATION

We implemented our proposed approach as described in sections 2-3, and a full re-encoding variant, i.e., a method which fully decodes the bit stream to the pixel domain, fully compensates the drift therein and re-encodes the pictures with the same parameters, for comparison. In this section, we evaluate our approach in terms of execution time, drift and overhead.

For the evaluation, we use the *foreman* and *akiyo* sequences with one RoI each as simple test cases as well as the *crew* sequence with up to eleven RoIs as an advanced test case. All three sequences have CIF resolution ($352 \cdot 288$). In addition, we use the *Vidyo1* sequence ($1280 \cdot 720$) with three RoIs and the *Kimono* sequence ($1920 \cdot 1080$) with one RoI to show the impact of higher resolutions. The RoI are the faces of the depicted actors which were segmented manually. As explained above, this information can be easily augmented to be provided by the surveillance system, together with the bit streams.

We used the H.264/AVC reference software (JM) to create Baseline profile bit streams of the sequences listed above. We used default settings and an $IPPP$ GOP structure, i.e., one I frame followed by 3 P frames, repeatedly, to simulate a typical video surveillance camera configuration.

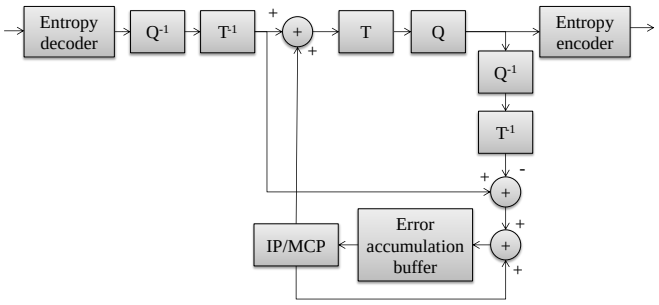

Fig. 4. Schematic transcoder architecture for macroblocks depending on the RoI (error compensation): The accumulated error is corrected approximatively by being considered in the prediction process. The abbreviations are the same as for Fig. 3.

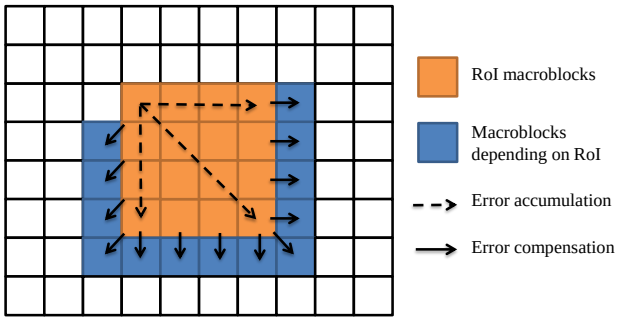

Fig. 5. Overview of error accumulation and compensation for intra prediction. Light (orange) blocks are processed as shown in Fig. 3, dark (blue) blocks as shown in Fig. 4.

4.1. Execution Time

To measure the transcoding time, i.e., the time for encryption and drift minimization, we executed our software implementation three times for cache warming and five times for measuring the time between the entry and exit of the `main` function. The five measurements were averaged; fluctuations were insignificant at around 1%.

Fig. 6 depicts the transcoding time for one sequence per tested resolution and various QP. Our approach (black) is significantly faster than the implemented full re-encoding implementation (grey), saving between 30 and 45% of the execution time, the upper bound being achieved at high resolutions and QP. Transcoding time decreases with increasing QP in both implementations evenly.

4.2. Drift

To quantify the quality decrease due to drift, we measured the Y-PSNR of all pixels outside the RoI with the respective

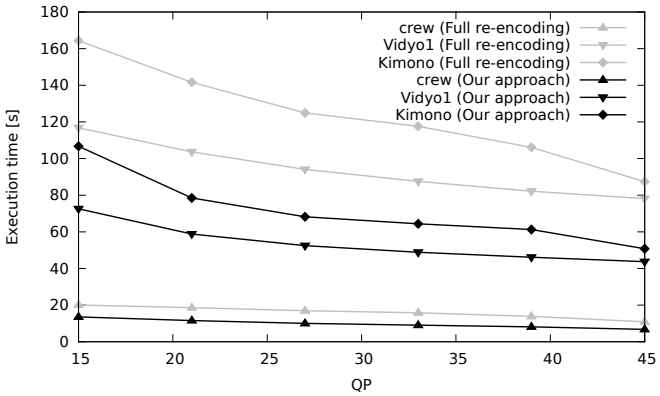

Fig. 6. Execution time of our approach (black) compared to full re-encoding (grey) for different QP: Our approach is between 30 and 45% faster.

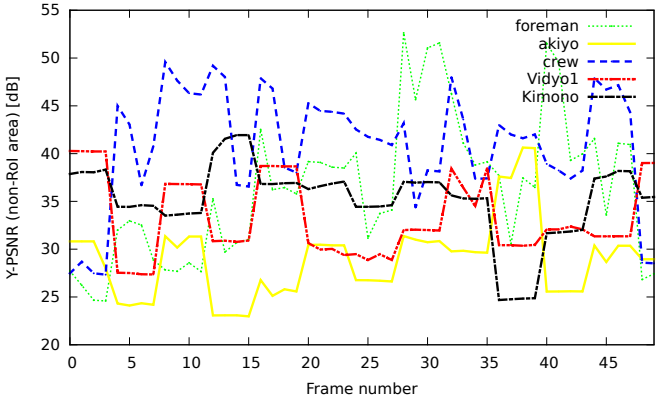

Fig. 7. Per-frame Y-PSNR outside the RoI for different sequences with QP 27 (first 50 frames): Spatial resolution has little to no effect on the quality degradation.

unencrypted (compressed) sequence as reference. We only depict the value of the first 50 frames for the sake of visualization. Fig. 7 illustrates the Y-PSNR values for all frames of the tested sequences with QP 27.

There are quasi no differences between the CIF and high resolution sequences. The amount of drift largely depends on the amount of movement and changes at GOP borders, most notably in the first few GOPs of the *Vidyo1* sequence (dash-dot-dotted line).

Used as a worst-case scenario, the *crew* sequence (dashed) with up to eleven RoI performs better than most of the other sequences in the first 50 frames, but shows some drops in quality around frame 109 (not depicted). Note, however, that the perceived quality is still relatively very high, i.e., the amount of drift is low and spatially limited at the minimum PSNR value at frame 109, as illustrated in Fig. 8 (bottom right). For the average case like in frame 45 (Fig. 8, top right), there is quasi no drift at all.

Fig. 8. Left: Frames 46 (top) and 109 (bottom) of the *crew* sequence with QP 27; right: Encoded with our proposed approach with examples of average drift (28 dB, top) and worst-case drift (14 dB, bottom), respectively, for a high number of encrypted RoIs.

4.3. Overhead

Finally, we determine the increase in file size caused by our approach due to the drift minimization. Fig. 9 illustrates this increase with respect to the original, i.e., unencrypted compressed, file size.

The bit rate increase depends on the QP. High and low QP induce little increase or even decrease, while medium QP increase the bit rate by about 1-3% for the CIF resolution sequences and significantly less for the high resolution sequences. The exact increase depends on the sequence and the number of RoIs. In general, increasing the resolution decreases the overhead.

The number of additional bits required for drift compensation decreases with increasing QP since high QP induce large quality degradations regardless of the presence of drift. Similarly, low QP yield a high number of non-zero coefficients in the transformed intra and inter prediction residuals, making the amount of bits required for drift minimization relatively small in comparison. For medium QP, the number of additional bits required for drift minimization is about the same, but the number of non-zero coefficients is smaller, therefore leading to a relative increase in bit rate.

Note that the bit rate increase of the *akiyo* sequence can be considered a worst-case scenario since there is practically no movement in the sequence outside the RoI. This allows coding this area with a small amount of bits, making every change due to drift minimization relatively large in comparison.

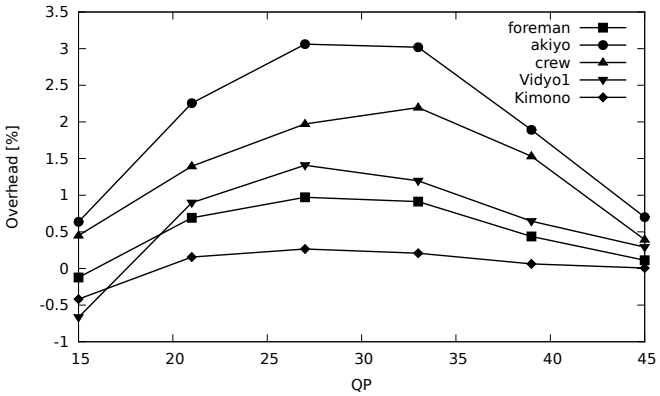

Fig. 9. Bit rate increase for different sequences and QPs: The overhead increases with the number of RoIs and peaks at medium-quality QPs.

5. FUTURE WORK

Two main aspects remain future work. First, the approach proposed in this paper can be combined with the approach of Unterweger et al. [6] which eliminates spatial, but not temporal drift. Since our approach minimizes temporal drift, a combination of the two approaches would allow for nearly drift-free bit-stream-based RoI encryption.

Second, decoding the sequences encoded with our proposed approach restores the RoI, but introduces additional drift outside the RoI due to the mismatch between the original prediction values and our drift-compensated ones. Although it is possible to copy the non-RoI areas (which are unencrypted) from the encrypted video, this does not allow for perfect reconstruction due to the remaining small amount of drift. Thus, a method to signal the RoI (as proposed, e.g., in [18]) as well as to compress and signal the difference signal between the original non-RoI areas and their counterparts with drift has to be devised so that the original video can be fully restored. Note that this may not be necessary for many use cases since, typically, only the RoI need to be fully restored, which is already the case with our approach.

6. CONCLUSION

We proposed a region of interest encryption approach for H.264/AVC bit streams. Despite being significantly faster than full re-encoding, it keeps the amount of drift outside the regions of interest at acceptable levels. The remaining amount of drift in all of the tested sequences is relatively small, apart from the *crew* sequence which exhibits some spatially limited drift in a small number of frames due to the high number of small regions of interest and intra-prediction-related dependencies. The bit rate overhead of our proposed approach is small (1.5% for high resolution sequences and 3% for low resolution sequences, tops) and depends on the quality and

motion characteristics of the sequence to be encrypted. Our proposed approach is therefore a viable alternative to full re-encoding without the drawback of high computational complexity.

Acknowledgments

This work is supported by FFG Bridge project 832082.

7. REFERENCES

[1] E.M. Newton, L. Sweeney, and B. Malin, "Preserving privacy by de-identifying face images," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 2, pp. 232–243, Feb 2005.

[2] Frederic Dufaux and Touradj Ebrahimi, "A framework for the validation of privacy protection solutions in video surveillance," in *Proceedings of the IEEE International Conference on Multimedia & Expo, ICME '10*, Singapore, July 2010, pp. 66–71, IEEE.

[3] Pavel Korshunov and Touradj Ebrahimi, "Towards Optimal Distortion-Based Visual Privacy Filters," in *21st IEEE International Conference on Image Processing (ICIP 2014)*, Paris, France, Oct. 2014, IEEE.

[4] ITU-T T.81, "Digital compression and coding of continuous-tone still images — requirements and guidelines," Sept. 1992, Also published as ISO/IEC IS 10918-1.

[5] Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard, and Ajay Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003.

[6] Andreas Unterweger and Andreas Uhl, "Slice groups for post-compression region of interest encryption in H.264/AVC and its scalable extension," *Signal Processing: Image Communication*, 2014, accepted.

[7] T. E. Boult, "PICO: Privacy through invertible cryptographic obscuration," in *IEEE/NFS Workshop on Computer Vision for Interactive and Intelligent Environments*, Lexington, KY, USA, Nov. 2005, pp. 27–38.

[8] Paula Carrillo, Hari Kalva, and Spyros Magliveras, "Compression Independent Reversible Encryption for Privacy in Video Surveillance," *EURASIP Journal on Information Security*, vol. 2009, pp. 1–13, Jan. 2009.

[9] Qi Meibing, Chen Xiaorui, Jiang Jianguo, and Zhan Shu, "Face Protection of H.264 Video Based on Detecting and Tracking," in *8th International Conference on Electronic Measurement and Instruments 2007 (ICEMI'07)*, Xi'an, China, Aug. 2007, pp. 2–172–2–177.

[10] Yeongyun Kim, Sung Jin, and Yong Ro, "Scalable Security and Conditional Access Control for Multiple Regions of Interest in Scalable Video Coding," in *International Workshop on Digital Watermarking 2007 (IWDW 2007)*, Yun Shi, Hyoung-Joong Kim, and Stefan Katzenbeisser, Eds., vol. 5041, pp. 71–86. Springer Berlin / Heidelberg, 2008.

[11] Lingling Tong, Feng Dai, Yongdong Zhang, and Jintao Li, "Restricted H.264/AVC video coding for privacy region scrambling," in *2010 17th IEEE International Conference on Image Processing (ICIP)*, Sept. 2010, pp. 2089–2092.

[12] Xiam Niu, Chongqing Zhou, Jianghua Ding, and Bian Yang, "JPEG Encryption with File Size Preservation," in *International Conference on Intelligent Information Hiding and Multimedia Signal Processing 2008 (IIHMSP '08)*, Aug. 2008, pp. 308–311.

[13] Bian Yang, Chong-Qing Zhou, C. Busch, and Xia-Mu Niu, "Transparent and perceptually enhanced JPEG image encryption," in *16th International Conference on Digital Signal Processing*, July 2009, pp. 1–6.

[14] Stefan Auer, Alexander Bliem, Dominik Engel, Andreas Uhl, and Andreas Unterweger, "Bitstream-Based JPEG Encryption in Real-time," *International Journal of Digital Crime and Forensics*, vol. 5, no. 3, pp. 1–14, 2013.

[15] Frederic Dufaux and Touradj Ebrahimi, "Scrambling for privacy protection in video surveillance systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1168–1174, 2008.

[16] R. Iqbal, S. Shahabuddin, and S. Shirmohammadi, "Compressed-domain spatial adaptation resilient perceptual encryption of live H.264 video," in *10th International Conference on Information Sciences Signal Processing and their Applications (ISSPA 2010)*, 2010, pp. 472–475.

[17] Jan De Cock, Stijn Notebaert, Peter Lambert, and Rik Van de Walle, "Requantization transcoding for H.264/AVC video coding," *Signal Processing: Image Communication*, vol. 25, no. 4, pp. 235–254, 2010.

[18] Dominik Engel, Andreas Uhl, and Andreas Unterweger, "Region of Interest Signalling for Encrypted JPEG Images," in *IH&MMSec'13: Proceedings of the 1st ACM Workshop on Information Hiding and Multimedia Security*. June 2013, pp. 165–174, ACM.