ASePPI: Robust Privacy Protection against De-Anonymization attacks

Natacha Ruchaud and Jean-Luc Dugelay
Eurecom
450 Route des Chappes, Sophia Antipolis, FRANCE
{ruchaud, dugelay}@eurecom.fr

Abstract

The evolution of the video surveillance systems generates questions concerning protection of individual privacy. In this paper, we design ASePPI, an Adaptive Scrambling enabling Privacy Protection and Intelligibility method operating in the H.264/AVC stream with the aim to be robust against de-anonymization attacks targeting the restoration of the original image and the re-identification of people. The proposed approach automatically adapts the level of protection according to the resolution of the region of interest. Compared to existing methods, our framework provides a better trade-off between the privacy protection and the visibility of the scene with robustness against de-anonymization attacks. Moreover, the impact on the source coding stream is negligible.

1. Introduction

The use of video surveillance continues to grow. Moreover, the resolution of visual sensors (*e.g*., 4k, HD) and the performance of video processing algorithms (*e.g*., identity recognition) are continuously increasing. This allows efficient automatic scene understanding (*e.g*. recognition of people, vehicles or animals) in CCTV (Closed-Circuit TeleVision) systems. Detection and recognition systems combined with pervasive networks of dense cameras highlight issues in privacy policy. Indeed, a lot of private information could become accessible.

Basic black masking methods exist to protect private data in surveillance cameras, *e.g*. to hide a PIN number entry for ATM security cameras, or to protect private property from outdoor security cameras. However, protecting the privacy of people is more complex given that the monitoring of their actions should not be hampered. The current solutions to anonymize people are the blurring or the pixelation (*e.g*., Google Street View) but they are not reversible.

To be reversible, authors in [13, 12] encrypt and insert the Most Significant Bits (MSBs) of the pixels of the original region of interest (RoI) in the Least Significant Bits (LSBs) of a resulting image. To keep the scene understandable, they put the bits of the edge image of the RoI in the MSBs of the resulting image. This method, operating in the spatial domain, is not robust against some manipulations, particularly against compression. Nowadays, almost all videos are compressed, therefore, image processing algorithms must be compliant with the source coding.

Someone may easily attack privacy with some knowledge about the privacy protection algorithm and the localisation of the region of interest. Unfortunately, few authors of privacy protection methods have provided an evaluation of de-anonymization attacks on their approaches.

The challenge raised in this article is to manage the trade-off between privacy protection and the intelligibility (i.e. keeping a fair visualization of the scene) while preventing re-identification after common de-anonymization attacks. In addition, the approach must be robust against compression and be reversible for authorized people.

The rest of the paper is organized as follows: in the next section, we summarize the current state-of-the-art of privacy protection techniques using in-compression encryption. In Section 3, we describe the proposed approach. We present and discuss the results in Section 4. Finally, we draw some conclusions and give an outlook for possible future work in Section 5.

2. Related Works

H.264/AVC is the most popular current standard for video compression. The baseline profile supports Intra (I) and Predicted frames (P) and entropy coding with context-adaptive variable-length codes (CAVLC). I frames contain only intra blocks (intra prediction) are predicted from previously coded data within the same frame. P frames contain intra prediction, but also inter prediction, where inter blocks are predicted from blocks of a previous reference frame. The residual blocks are the differences between the predicted blocks and the correct ones. The 4*4 integer transform, which is an approximate DCT (Discrete cosine transform), is applied to the residual blocks before being quantized and encoded.

In the privacy protection domain, encryption methods are often used to allow the reversibility of the process, only for authorized people. Applying encryption methods before compression may compromise the reversibility of the process due to the lossy compression. Doing encryption methods after the compression requires an additional step to make sure that the generated bitstream is still decodable by a conventional decoder, but it is too complex and has a little added value. Therefore, the following methods, including the one proposed in this paper, operate during the compression in the H.264/AVC codec and predict each block from unencrypted blocks during the encoding.

To prevent drift error in H.264/AVC produced in the non-privacy region due to the predictions from the private encrypted regions, Tong, Dai *et al*. [4] propose two main methods: Mode Restricted Intra Prediction (MRIP) and Search Window Restricted Motion Estimation (SWRME). Not all the existing approaches include a mechanism that prevents this degradation of non-private areas.

Dufaux and Ebrahimi [5] propose to scramble the signs of the nonzero coefficients of each residual block of the private regions within the MPEG-4 framework. Although the coefficient sign scrambling is a common encryption technique when working in DCT-based compression formats, it produces a relatively weak scrambling effect (if applied alone) especially for high resolution images.

To enhance the scrambling effect for privacy protection, Wang *et al*. [16] encrypt the intra prediction modes (IPM) in addition to the signs of the nonzero coefficients (SNC) within the private areas. Su *et al*. [15] directly modify the intra prediction modes (IPM) as well as the motion vector differences (MVD) while embedding their original information in the AC coefficients. Khlif *et al*. [7] scramble the signs of motion vectors using a chaotic cryptography algorithm. Unlike [5], these three previous methods produce a strong scrambling effect yielding noisy pictures which may hamper the monitoring. Encryption and scrambling algorithms have issues to manage the trade-off between the intelligibility and the privacy.

Ruchaud and Dugelay [14] handle this trade-off by applying a bitwise XOR operation between each DCT (DC+AC) coefficient and pseudo-random numbers within the JPEG framework (operating on still images, not on videos). Thus, they shift down the encrypted coefficients from one position which allows the insertion of a value of their choice into the DC of each block enabling the control of the appearance of the decompressed privacy-protected images.

We follow up the idea of [14]: choosing the DCs values to better control the content of the final image (i.e. what will remain visible to all viewers) while encrypting the original coefficients to protect privacy. In addition, we make this compliant with H.264/AVC and automatically adapt the level of the privacy protection depending on the size/resolution of the region of interest. Indeed, the sizes of the privacy regions vary in a video. Thus, the higher these sizes, the stronger should be the privacy protection.

The authors of [5], identified two essential types of de-anonymization attacks when encrypting the coefficients of the residual coefficients generated by the H.264/AVC codec. The first one is the brute force attack (i.e. testing all combinations to reverse the process) and the second one is the suppression attack (i.e. removing the encrypted coefficients). Thus, we design the proposed approach to be robust on these de-anonymization attacks where their purpose is to restore the initial image and then re-identify people.

3. ASePPI, an Adaptive Scrambling enabling Privacy Protection and Intelligibility

To work in the DCT domain, our process operates during the compression, after the transformation and the quantization of the residual blocks in the H.264/AVC framework. We only apply our approach on the residual blocks of the regions of interest of the luminance channel (Y). We use MRIP and SWRME to avoid drift error produced by our process. The code of our proposed process is available on a Github website [1].

3.1. The region of interest (RoI)

For each I frame, region of interests (*e.g*., people's faces and bodies), denoted RoIs, are annotated either manually or automatically using standard tools. The position of the RoI is described by its upper left point and its size (four numbers). We compute a bitwise XOR operation among each of the four number and a random number (RN) generated by a pseudo-random sequence controlled by a secret key. We did not encode the encrypted RoI position into the scrambled video stream because non authorized people do not need it, and without it the system becomes more difficult to attack. Therefore, we store it independently from the privacy protected video. The number of bits needed to store the RoI position is negligible. For instance, for a 4K resolution (i.e., 4096*2160 pixels), we use 12 bits to store each encrypted number, thus 48 bits every 10 frames (i.e., 4.8 each frame).

3.2. Encrypting the residual of I frames blocks

Each residual block of an I frame inside the RoI follows additional steps illustrated in bold in Figure 1. We encrypt the DCT coefficients with a pseudo random number generator (PRNG) controlled by a secret key, in order to protect data information and to be reversible only by authorized people. Then, we shift the encrypted coefficients from one position towards the higher frequencies to make

[1] `https://github.com/NatachaRuchaud/ASePPI`

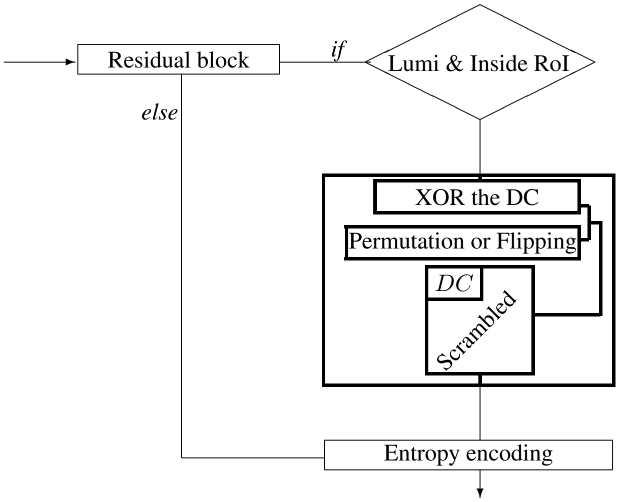


Figure 1: Scheme for residual blocks of I frame.

available the DC (i.e. the lowest frequency representing the average of the pixel values) position. This position will be later used to insert the minimum information required for surveillance. We intentionally lost the least significant coefficients to store the encrypted DC into the AC coefficients (i.e. the remaining frequencies).

3.2.1 Encrypting the DC

```
if ($|DC| < 16$) then
  | $X$ = 16;
else
  | $X = 2^n$;
if ($DC \neq 0$) & (($|DC| \neq (RN \mod X)$)) then
  | $DC_e$ = ($|DC| \oplus (RN \mod X)$)*sign(DC);
else
  | $DC_e$ = DC;
with $n = \lfloor \log_2 |DC| \rfloor$ an integer
```

Algorithm 1: DC encryption

Doing an XOR between the DC and a random number generated from an infinite range may lead to bigger encrypted DC than the original one. This size difference produces noise in the decompressed privacy-protected images. Therefore, to minimize this noise, we design an encryption algorithm where the encrypted values will remain in the same range than their original one. According to the algorithm (1), a $|DC| \in [2^n, 2^{n+1}]$ produces $(RN \mod 2^n) \in [0, 2^n[$ (i.e. $[0, 2^{n-1}]$), thus, $DC_e \in [2^n, 2^{n+1}]$ with RN a random number generated by the PRNG, $sign(DC)$ equal to -1 if the DC sign is negative and +1 otherwise, and DC_e the encrypted DC. For values lower than 16, we increase the range of values (i.e. $[0, 15]$) which creates more possibilities. For instance, if $DC = 1, n = 0$ and $\mod 2^n$ = $\mod 1 = 0$ thus the DC_e remains the same as the original one.

This encryption algorithm (1) leaves the DC as it is, in two cases: (i) if the DC is null and, (ii) if the DC is equal to RN in order to (i) avoid too much degradation in the decompressed privacy-protected images and, (ii) to obtain the DC_e in the same range than the DC.

3.2.2 Scrambling the coefficients (the encrypted DC + the original AC)

The last AC coefficient is voluntary lost so we set it to 0. For each block, we select the scrambling method (between RP and SNC, explained in the following) that creates the highest number of combinations to recover the original data. The encrypted DC is included in the scrambling. We use the PRNG to generate a random sequence.

RP: Note p_1, the number of coefficients before EOB (End-of-Block, the remaining coefficients are zero). To scramble them, we randomly permute the $p_1 - 1$ coefficients using the Knuth shuffle algorithm [3] that re-arranges their order. The last non-zero coefficient (i.e. the i_{p_1} th coefficient) is used to mark the end of the permutation (i.e. the AC coefficients before the last non-zero coefficient are randomly permuted). Thus, there are $(p_1 - 1)!$ combinations.

SNC: Note p_2, the number of non-zeros coefficients. We flip randomly the sign of these p_2 coefficients. Therefore, there are 2^{p_2} combinations.

3.2.3 Shifting the scrambled coefficients to the AC ones

As an example, we suppose that the original extracted coefficients are [31 (DC), 0, -2, -1, -1, -1, 0, 0, -1, EOB]. We encrypt the DC which becomes 24: [24 (DC_e), 0, -2, -1, -1, -1, 0, 0, -1, EOB]. There are $8! = 40320$ combinations with the RP method and $2^6 = 64$ with the SNC one. Thus, we select the RP method to scramble the coefficients which becomes [-1, 0, 24 (DC_e), -2, -1, 0, 0, -1, -1, EOB] and we shift them one position towards the higher frequencies which leads to [DC_{new}, -1, 0, 24 (DC_e), -2, -1, 0, 0, -1, -1, EOB]. Then, we re-insert the scrambled coefficients into a block according to the zigzag code and choose the DC value with the formula defined in Section 3.2.4.

3.2.4 Choice of the DC_{new} value

We dedicate the DC_{new} value to restitute some of the original information (*e.g.* the average luminance of a block).

Keeping the original DC (i.e. the mean) of each residual block of the luminance channel produces 4*4 coloured blocks in the decompressed image (similar to a pixelation). To get stronger privacy protection, we keep the DC of a bigger block and insert it in the DCs of its 4*4 sub-blocks. For

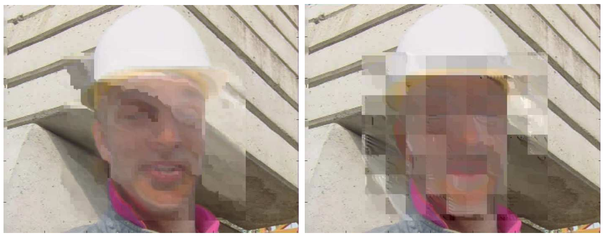


Figure 2: Keeping only the DC of each block of the luminance channel with $h = 204$ and $w = 220$ (on the RoI). Blocks size: 4*4 for the right image and 24*24 for the left one.

example, in the right picture in Figure 2, we kept the original DC of each 4*4 residual block and in the left one, we inserted the DC of each block of size 24*24 inside the DCs of its corresponding 4*4 sub-blocks.

The equation (1) represents the relation between the size of these average blocks, denoted S, and the number of blocks, denoted Nb, depending on the number of pixels (h x w) inside the RoI. For instance, if S is equal to 24, the residual blocks inside the 24*24 block have the same DC coefficient, which is the DC of the 24*24 block (i.e. the mean of the 24*24 block).

$$Nb = \frac{h * w}{S * S} \tag{1}$$

The higher Nb (i.e. the higher is the image quality), the better the recognition is in general. Our goal is to find the maximum Nb to preserve as intelligibility as possible while minimizing the performance of face recognition. Therefore, to fulfil this purpose, we did the following empirical study by fixing several values.

We have selected as a baseline the face recognition algorithm Eigen described in [8] and based on the Euclidean distance because of its robustness to pixelated face images (compared to other descriptors). We trained Eigen, using a subset of Feret [11] and ScfaceData [6] databases and tested it on another subset from the same database and on the pixelated versions of them with different parameter values (i.e. S, h and w).

According to Table 1, we have selected the S values associated with the highlighted boxes representing the biggest reduction in recognition performance at each resolution. From these results and the equation (1), we deduce the maximum Nb which is 99 (e.g. $\frac{176*144}{16*16} = 99$). Face recognition performance significantly drop if Nb is equal to 99 or less.

Looking for maximizing Nb, the relation can be rewritten as in equation 2. S is rounded to its nearest multiple of 4 as in equation (3) because the size of each residual block is 4*4. Therefore, the equation (3) automatically defines S, a multiple of 4, maximizing the number of blocks such as we protect the privacy. However, we can change the value of $max(Nb)$ to have stronger or weaker protection.

S \ *h*w*	**128 x 96**	**176 x 144**	**352 x 288**	**704 x 576**
Original	95.6	96	96.4	96.4
8	68.4	76.41	85.4	86
12	22.9	66.5	76.4	85.5
16	20.7	21.3	75.9	84.4
28	5.5	18.8	58.02	77.9
32	4.2	12.7	20.8	75.4
36	3.6	8.5	20.4	73.7
60	0	0	9.6	50.47
64	0	0	8.1	20.2
68	0	0	5.5	19.5

Table 1: Accuracy of identity recognition (%) from faces.

$$\begin{aligned} max(Nb) &\geq Nb \\ \Leftrightarrow max(Nb) &\geq \frac{h * w}{S * S} \\ \Leftrightarrow S &\geq \sqrt{\frac{h * w}{max(Nb)}} \end{aligned} \tag{2}$$

$$\begin{aligned} S \approx \left\lceil \frac{\sqrt{\frac{h*w}{max(Nb)}}}{4} \right\rceil * 4 &\geq \sqrt{\frac{h * w}{max(Nb)}} \\ S &= \left\lceil \frac{\sqrt{\frac{h*w}{99}}}{4} \right\rceil * 4 \end{aligned} \tag{3}$$

3.3. Encrypting the residual of P frames blocks

In H.264/AVC framework, the blocks inside the RoI may be predicted from unscrambled blocks or become closer to the original one if the reference scrambled blocks are already close to the original one. Therefore, we encrypt the DCT coefficients of each residual block of P frames as in Section 3.2.1 and Section 3.2.2. Contrary to I frame blocks, there is no necessity to insert a new DC, thus, no information is lost.

3.4. Reverse process: Decryption

Since we predict each block from unencrypted blocks during the encoding, we only need to decrypt the scrambled residual blocks (i.e. the ones inside the RoI). The correct secret key generates the same random numbers than the ones in the encoding part allowing to recover the original data.

For I frames, we extract the AC coefficients (not the DC) of the blocks inside the RoI and for P frames all the coefficients (DC+AC) of these blocks. As in Section 3.2.2, we select the scrambling method (RP or SNC) that produces the highest number of combinations to recover the original data and, then, apply its reverse process. Finally, we decrypt the DC (i.e. the first decrypted coefficient) by applying the algorithm 1.

4. Experimental Results

We compare the proposed method (ASePPI), with the encryption of the signs of the non-zero coefficients (SNC) and with the addition of the encryption of the intra prediction modes (SNC+IPM) within the privacy area. We apply all methods only on the luminance channel for a fair comparison with our proposed approach. Three video examples of the application of these methods are available [2].

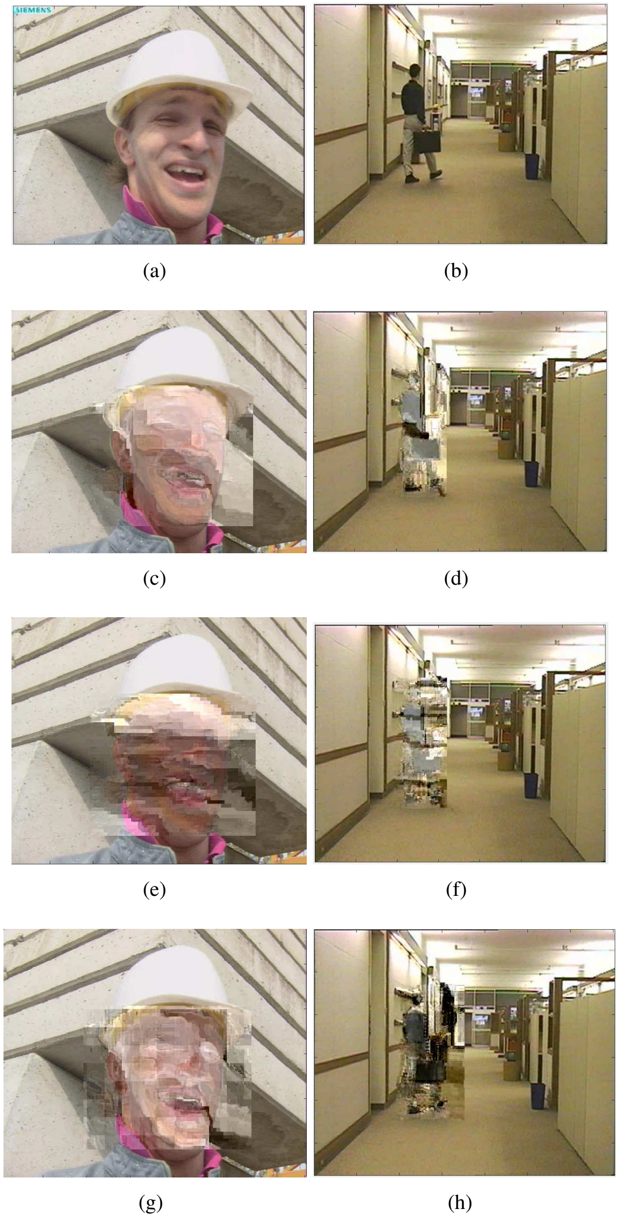
For the evaluation, we have selected the following sequences: ’hall’, ’foreman’, ’suzie’, ’akiyo’, ’carphone’, ’claire’ and ’miss-america’ all available on the web[3]. We use different values of QP and IP in our evaluations. QP is the quantization parameter and IP the intra period that defines the frames number between two I frames.

In Section 4.1, we evaluate the intelligibility for the decompressed privacy-protected frames of the sequences. Next, in Section 4.2, we assess the privacy protection without and with de-anonymization attacks, assuming that the attacker knows the RoI (detectable from the image) and the process.

In Section 4.3, we also evaluate the bits overhead and the quality loss in terms of PSNR for the decompressed decrypted (with the correct key) sequences.

4.1. Intelligibility

Using both SNC and IPM hampers the global understanding of the scene or human actions. For example, in 3(f) it is not obvious that the protected area contains a person carrying her bag whereas in 3(h) the shape of the head and feet are clearly distinguishable as well as the bag. We evaluate the intelligibility with two metrics, the peak signal-to-noise ratio (PSNR) to measure the amount of the degradation and the edge similarity score (ESS) [9] to assess the degree of resemblance of the edge and contour information between two images. We apply these metrics between the original RoI and the scrambled RoI of the seven sequences for each IP=1, 5, 10, 30 with QP=24. We report in Table 2, the additional degradation compared to ASePPI, on average. For instance, for ’Carphone’, compared to ASePPI, SNC+IPM degrades, on average, 8.61% more and its degree of resemblance of the edge is 22.93% less important.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 3: With CIF size, QP= 24 and IP= 5: (a) The 15th original frame of the ’foreman’ sequence (I), (b) the 39th original frame of the sequence ’hall’ (P), (c) and (d) encrypted by SNC, (e) and (f) encrypted by SNC+IPM, (g) and (h) encrypted by ASePPI.

	Suzie	Foreman	Hall	Akiyo	Carphone	Claire	Miss America
PSNR	3.13	17.94	4.33	2.26	8.61	3.02	5.48
ESS	16.22	8.31	13.58	21.19	22.93	17.88	13.15

Table 2: Amount of additional degradation (%) applying SNC+IPM compared to ASePPI

[2] www.dropbox.com/s/r0hbc8n48ocu4uk/Video_Examples.zip?dl=0

[3] http://trace.eas.asu.edu/yuv/

According to the results in Table 2, we conclude that the SNC + IPM method degrades the visibility of the scene more than ASePPI. Indeed, the encryption of the Intra Prediction Mode (IPM) leads to predict blocks from wrong ones (not the same ones as in the encoding). This produces disturbances. Moreover, in ASePPI, we design the encryption of the DC so that it produces limited noise.

4.2. Privacy Protection

We trained the OpenFace CNN algorithm [2] with 40 aligned face images of each following sequence in CIF size: 'foreman', 'suzie', 'akiyo', 'carphone', 'claire' and 'miss-america', and tested on 150 other faces from each sequence. For original faces, we get 95.7% of accuracy with 91% of confidence rate in average for correct classifications. For faces protected by SNC (see 3(c)), SNC+IPM (see 3(e)) or ASePPI (see 3(g)) less than 20% of faces are well identified and among those well identified their associated confidence score is lower than 40%. Therefore, all hamper the face recognition algorithm. However, from a subjective point of view, it is not obvious that SNC method always protects the privacy of the face. For example, in the picture 3(c) (SNC), we can identify the details of the face compared to the two other methods (pictures 3(e) and 3(g)).

4.2.1 Robust privacy protection against suppression attack (SA)

An attacker may try to suppress encrypted data. This attack consists of extrapolating the scrambled data by motion compensation from the previous frame using the motion vectors which are available to the attacker. Thus, to implement this attack in our method, the RoI AC coefficient for the I frames and the RoI DCT coefficients (DC+AC) for the P frames are set to 0. For SNC method, we simply set to 0 all RoI AC coefficients, and for SNC+IPM we set, in addition, all intra prediction modes to 2 (the mean). We produced video examples [4], with 'foreman' and 'carphone' sequences, on which the suppression attack is applied on the different methods.

We report, in Table 3, the accuracy of identity recognition with OpenFace CNN tool (the same than in Section 4.2) with their confidence associated, when the suppression attack is applied on the different methods. For instance, for the SNC method with IP=10, 50.61% of faces are well identified and among those well identified their confidence score is 52%. According to the results, the SNC method becomes weaker in terms of privacy protection because the algorithm re-identifies people with more than 50% of good recognition, whereas with our method the identity recognition rate is still under 25% with a lower confidence (under 38%).

The suppression attack on the SNC method makes the DC of each block available, whereas the AC coefficients are set to 0, therefore pictures look like pixelated images with identical sizes of squares that are not enough to protect privacy especially on high resolution images. For ASePPI method, we automatically adapt the size of these squares by using the same DC for multiple blocks in order to protect the privacy at any resolution which makes it stronger against the suppression attack.

According to Table 3, the SNC+IPM method is also robust against suppression attack. However, from a human point of view, the details of the face or the shape of the body can be much more visible than in the case of ASePPI as is illustrated in Figure 4 and in the video examples [4].

IP	SNC	SNC+IPM	ASePPI
10	50.61 / 52	14.3 / 39.26	12.11 / 38
30	51.33 / 49	21.34 / 41	23.43 / 35

Table 3: Accuracy (%) / Confidence (%) of face recognition with suppression attack

4.2.2 Robust privacy protection against brute force attack

We consider an exhaustive search of all combinations. The number of combinations per block with ASePPI method is always greater than or equal to the one of SNC. Indeed, as explained in Section 3.2.2 we select the method (between SNC or RP) which performs the higher number of combinations.

The total number of combinations to recover an original RoI with $NbBlocks = \frac{h*w}{4*4}$, denoted $NbComb$, might be defined by the average number of combinations for one block, denoted $avCombi1Block$ powers $NbBlocks$. According to this, we deduce the average number of combinations for one block as detailed in equations 4. Thus, we compute this formula, for several frames and different value of QP (12, 18, 24, 30). We obtain the average results for each QP shown in Table 4.

$$\begin{aligned} NbComb &= avCombi1Block^{NbBlock} \\ \Leftrightarrow \log_{10} NbComb &= \log_{10}(avCombi1Block^{NbBlock}) \\ \Leftrightarrow \frac{\log_{10}(NbComb)}{NbBlock} &= \log_{10}(avCombi1Block) \\ \Leftrightarrow avCombi1Block &= 10^{\frac{\log_{10}(NbComb)}{NbBlock}} \end{aligned} \tag{4}$$

Generally, a minimum size of the image is required for an identification. As an example, to be eligible to recog-

[4] www.dropbox.com/s/39ke5wy6mgezq4k/Video_Examples_SA.zip?dl=0

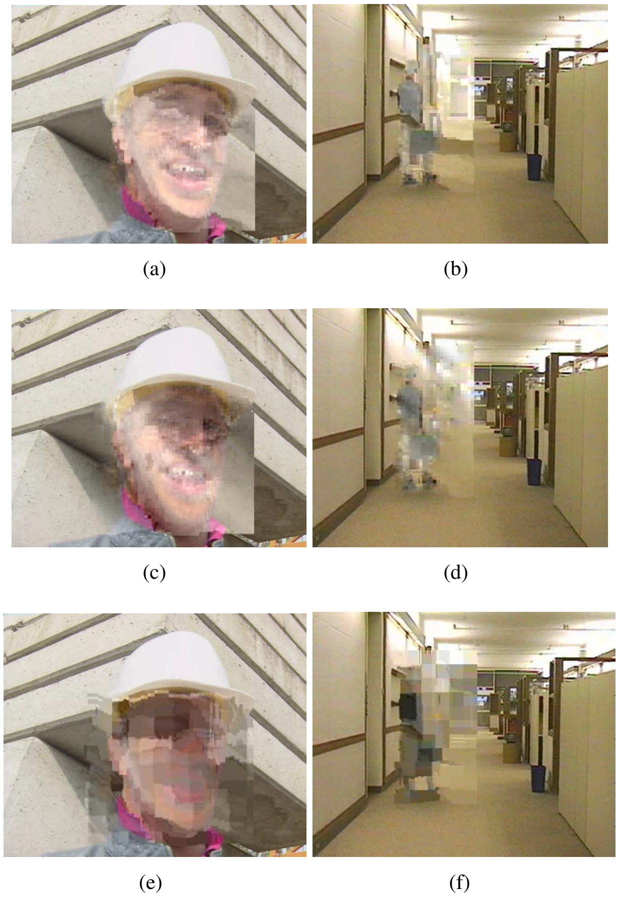

(a) (b)
(c) (d)
(e) (f)

Figure 4: With CIF size, QP= 24 and IP= 5: After the suppression attack on the SNC privacy protection (a) and (b), on the SNC+IPM privacy protection (c) and (d), on the ASePPI one (e) and (f).

QP	**12**	**18**	**24**	**30**
I frames	$10^{8.76}$	$10^{5.87}$	$10^{3.16}$	$10^{1.36}$
P frames	$10^{2.97}$	$10^{2.5}$	$10^{1.76}$	$10^{0.86}$

Table 4: Average number of combinations to recover one block encrypted by ASePPI method.

nize the identity of a person, the laws in France impose the face region to have at least 90 pixels between the bottom of the chin and the top of the skull or hair, and 60 pixels between the two ears (included) [1]. Thus, 90 x 60 pixels should be the minimum size allowed to identify someone, and an image of this size contains 337.5 4*4 blocks. To recover an I frame, with QP = 30, the number of combinations is $10^{1.36*337.5} = 10^{459} > 2^{1048}$, and for P frame $10^{0.86*337.5} = 10^{290} > 2^{963}$. Therefore, the method provides a good level of security.

4.3. Impact on source coding stream

The bits overhead is the percentage of bits added by our process compared to the baseline profile (H.264 without privacy-protection). For example, for the 'foreman' sequence, with QP = 24 and IP = 10, the number of bits are 83289 for the baseline profile and 85600 with the integration of our process which produces $100 - 100 * 83289/85600\%$ of bits overhead, i.e. 2.7%. As is shown in Table 5, the I frames produce the most significant increase in the number of bits in the stream, this is due to the insertion of an important coefficient (the DC). We generate bits overhead also because the DC encryption loses some efficiency in quantification.

IP	Suzie	Foreman	Hall	Akiyo	Carphone	Claire	Miss America
1	9.56	2.82	3.43	4.75	2.49	9.46	4.33
5	8.4	2.74	3.3	4.3	1.88	9.25	2.87
10	8.22	2.7	3.26	4.21	1.52	8.08	2.48
30	7.67	2.63	3.15	3.74	1.43	5.98	2.34

Table 5: Bits overhead (%) with QP set to 24

The drop of PSNR performance in percentage for the decompressed decrypted images (using the correct secret key) compared to the original ones, is computed in the same way as the bits overhead and reported in Table 6. For example, for the 'suzie' sequence, with QP = 24 and IP = 10, PSNR of RGB channel for the baseline profile is 44.28 and 43.9 with the integration of our process which produces $100 - 100 * 43.9/44.28$ of percentage of decline, i.e. 0.86%. We lost some coefficients only for the I frames, but since blocks in P frames are predicted from blocks of I frames we also lost some information for P frames.

IP	Suzie	Foreman	Hall	Akiyo	Carphone	Claire	Miss America
1	0	0	0.79	0	0.12	0.47	0.07
5	0.41	0.63	0.78	0.47	0.46	0.94	0.23
10	0.86	1.26	1.28	1.18	1.76	1.11	0.41
30	1.44	1.95	1.96	1.6	1.98	1.93	0.98

Table 6: PSNR decrease (%) for RGB channels with QP set to 24.

According to the results, the impact on the source coding stream is negligible.

5. Conclusions

We design a privacy protection method robust against common de-anonymization attacks. Indeed, for the scrambling step we select the method that produces the highest number of combinations to recover the original data. Moreover, our approach automatically adapts the level of the pixelization effect to be optimal in terms of privacy protection.

Therefore, in this paper, we prove that the application of ASePPI provides a better trade-off, compared to the existing methods, between the privacy protection and the visibility of the scene with a robustness against de-anonymization attacks. Furthermore, we perform evaluations on the impact to the efficiency of the codec with the integration of our application. We conclude that the quality of the reconstructed videos is close to the original ones ($< 2\%$) and the process produces a small percentage of bits overhead ($< 10\%$).

As future work, we plan to subjectively evaluate the efficiency of the privacy protection and the intelligibility by doing a survey using images that are protected by our method and asking the identity of famous people, the level of pleasantness of the image and the activity of the persons.

In the same way as the authors did for pixelation, blurring and P3 in [10], we will train an identity recognition algorithm on face images protected by our tool to evaluate the robustness against this parrot attack.

6. Acknowledgements

This work has been partially performed under the Identity project[5].

References

[1] Legislation française en matière de vidéo surveillance. http://www.telecoute.re/livre-blanc-conformite-v31.pdf, 2010. Visited on 2017-04-19.

[2] B. Amos, B. Ludwiczuk, and M. Satyanarayanan. Openface: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science, 2016.

[3] D. Chafaï and F. Malrieu. Permutations, partitions, et graphes. In *Recueil de Modèles Aléatoires*, pages 57–68. Springer, 2016.

[4] F. Dai, L. Tong, Y. Zhang, and J. Li. Restricted h. 264/avc video coding for privacy protected video scrambling. *Journal of Visual Communication and Image Representation*, 22(6):479–490, 2011.

[5] F. Dufaux and T. Ebrahimi. Scrambling for privacy protection in video surveillance systems. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(8):1168–1174, 2008.

[6] M. Grgic, K. Delac, and S. Grgic. Scface–surveillance cameras face database. *Multimedia tools and applications*, 51(3):863–879, 2011.

[7] N. Khlif, T. Damak, F. Kammoun, and N. Masmoudi. Motion vectors signs encryption for h. 264/avc. In *Advanced Technologies for Signal and Image Processing (ATSIP), 2014 1st International Conference on*, pages 1–6. IEEE, 2014.

[8] V. Kshirsagar, M. Baviskar, and M. Gaikwad. Face recognition using eigenfaces. In *Computer Research and Development (ICCRD), 2011 3rd International Conference on*, volume 2, pages 302–306. IEEE, 2011.

[9] Y. Mao and M. Wu. A joint signal processing and cryptographic approach to multimedia encryption. *IEEE Transactions on Image Processing*, 15(7):2061–2075, 2006.

[10] R. McPherson, R. Shokri, and V. Shmatikov. Defeating image obfuscation with deep learning. *arXiv preprint arXiv:1609.00408*, 2016.

[11] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Transactions on pattern analysis and machine intelligence*, 22(10):1090–1104, 2000.

[12] N. Ruchaud and J. L. Dugelay. Efficient privacy protection in video surveillance by stegoscrambling. In *WIFS 7th IEEE International Workshop on Information Forensics and Security*, 2015.

[13] N. Ruchaud and J. L. Dugelay. Privacy protection filter using stegoscrambling in video surveillance. In *MediaEval*, 2015.

[14] N. Ruchaud and J.-L. Dugelay. Privacy protecting, intelligibility preserving video surveillance. In *Multimedia & Expo Workshops (ICMEW), 2016 IEEE International Conference on*, pages 1–6. IEEE, 2016.

[15] P.-C. Su, W.-Y. Chen, S.-Y. Shiau, C.-Y. Wu, and A. Y. Su. A privacy protection scheme in h. 264/avc by data hiding. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific*, pages 1–7. IEEE, 2013.

[16] Y. Wang, F. Kurugollu, et al. Privacy region protection for h. 264/avc with enhanced scrambling effect and a low bitrate overhead. *Signal Processing: Image Communication*, 35:71–84, 2015.

[5] http://www2.warwick.ac.uk/fac/sci/dcs/research/df/identity/