Hindawi Publishing Corporation
EURASIP Journal on Information Security
Volume 2009, Article ID 429581, 13 pages
doi:10.1155/2009/429581

Research Article

Compression Independent Reversible Encryption for Privacy in Video Surveillance

Paula Carrillo,[1] Hari Kalva,[1] and Spyros Magliveras[2]

[1] *Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL 33431, USA*
[2] *Department of Mathematical Sciences, Florida Atlantic University, Boca Raton, FL 33431, USA*

Correspondence should be addressed to Author please provide, Author please provide

Received 16 April 2009; Revised 21 September 2009; Accepted 13 December 2009

Recommended by Andrew Senior

One of the main concerns of the wide use of video surveillance is the loss of individual privacy. Individuals who are not suspects need not be identified on camera recordings. Mechanisms that protect the identity while ensuring legitimate security needs are necessary. Selectively encrypting regions that reveal identity (e.g., faces or vehicle tags) are necessary to preserve individuals' right to privacy while recognizing the legitimate needs for video surveillance. The video used in surveillance applications usually needs to be transcoded or recoded for distribution and archival. Transcoding a traditionally encrypted video is not possible without decrypting the video first. This paper presents a compression algorithm independent solution that provides privacy in video surveillance applications. The proposed approach uses permutation-based encryption in the pixel domain to hide identity revealing features. The permutation-based encryption tolerates lossy compression and transcoding and allows decryption of the transcoded video at a later time. The use of permutation-based encryption makes the proposed solution independent of the compression algorithms used and robust to transcoding. The cost of providing this privacy is an increase in bitrate that depends on the percentage of blocks encrypted.

1. Introduction

With video surveillance becoming an integral part of our security infrastructure, privacy rights are beginning to gain importance. The key concern is the fact that private citizens, who are not suspects, are being recorded and recordings archived through the use of video surveillance systems. Such a record-everything-and-process-later approach has serious privacy implications. The same privacy issues arise when surveillance cameras routinely record highway traffic as vehicle tags are recorded. The solution of removing the identities by blurring/blackening the portions of video is not acceptable to security personnel as they may have legitimate need to review the videos. On the contrary, leaving the videos with identities of people and vehicles public is a breach of privacy.

A solution to the problem is selective encryption of portions of the video that reveal identity (e.g., faces, vehicle tags) in surveillance applications. Regions of a video can be encrypted to ensure privacy and still allow decryption for legitimate security needs at anytime in the future. The goals of the video surveillance are still met as selective encryption allows monitoring the activities without knowing the identities of those being monitored. When a suspicious activity needs to be investigated, the identities can be uncovered with proper authorization. The few existing solutions are specific to video and image compression algorithms used and require modification to the video encoders [1, 2]. These approaches limit the flexibility of surveillance systems. This paper presents an innovative solution that meets the needs of individuals' privacy and legitimate security needs. Preliminary results from this work were reported in [3]. The proposed solution is independent of the image and video compression algorithms used. This allows the use of standard video encoders and decoders and also enables smart-cameras that output encrypted video. The proposed solution also survives video transcoding and recoding allowing a normal video distribution chain with multiple video encoding and decoding operations. Another innovation in the proposed approach is the use of permutation based encryption that can

survive lossy compression. Yet another feature is the ability of the system to detect encrypted regions automatically and allow for automatic decryption without any additional information to identify the encrypted blocks sent to the decoding terminals. The proposed system is considered to support selective encryption as it can encrypt only regions of a video that reveal identity. However, the task of detecting regions of a video that reveal identity is outside the scope of the work and is not addressed by the proposed system.

The rest of the paper is organized as follows: Section 2 presents the background work and summarizes the characteristics of privacy preserving surveillance systems. Section 3 presents the proposed solution. Section 4 presents experimental results and performance evaluation and conclusions are drawn in Section 5.

2. Background

There are many challenges in ensuring privacy in video surveillance. One of them is the complexity of encrypting huge amount of video data. Other challenges appear when applications require real time performance, total recovery of the hidden objects, recompression, and/or transcoding for distribution and archiving. Balancing privacy implies some form of video encryption. Depending on stage at which encryption is performed the process can be classified as pixel domain, transform domain, or bitstream domain. Figure 1 highlights the opportunities for encryption in a general video encoder.

The main techniques used in video privacy systems are summarized below.

Obfuscation. The system presented in [1] describes a privacy preserving video console that uses a rendering face images technique in the pixel domain and leaves the face unrecognizable by identification software. Based on computer vision techniques, the video console determines the interesting components of a video and then obscures that piece of information, or its components, such that face recognition software cannot recognize the faces. With this method the privacy is maintained but the surveillance and security needs are not met due to the irreversibility of the obfuscation process. In [4] a medical application for automatic patient detection, tracking, labeling and obscuring (the obscuring option in the case the patient does not want to be involved in the research) in real time has been developed. In this particular case, reversibility is not required or desired. Martin and Plataniotis present an interesting solution of shape and texture encryption using Secure Shape and Texture Set Partitioning in Hierarchical Trees (SecST-SPIHT) [5]. This method encrypts about 5% of the bitstream using a secure key to protect the video. Since the encryption is in the bitstream domain, any transcoding requires decoding the frames first. This method has a desirable attribute of obfuscating the shape and thereby providing a layer of security.

Transform-Domain Coefficient Scrambling. This technique is applied in the transform domain for motion JPEG or MPEG video and was presented in [2]. The region of interest is detected and then the signs of selected transform coefficients are scrambled. More specifically for JPEG2000 Discrete Wavelet Transform (DWT) and for MPEG the Discrete Cosine Transform (DCT) coefficients, corresponding to the regions of interest (ROIs), are scrambled by pseudo-randomly inverting their signs. Consequently, the scene remains understandable, but the ROIs are unidentifiable. The decoded video will have blocky regions unless a proper key is used for descrambling. This process is reversible but it is specific to video compression used and cannot survive operations such as transcoding and recoding that may be necessary to distribute video.

Invertible Cryptographic Obfuscation. Another technique proposed in [6] is privacy through a cryptographic obfuscation; it uses Data Encryption Standard (DES) and Advanced Encryption Standard (AES) to encrypt regions of JPEG images during the compression stage, after Huffman encoding, in the bitstream domain. This is similar to the transform coefficient sign scrambling. This method also suffers from the same drawbacks: it is compression algorithm specific and cannot survive transcoding.

Skin Tone Detection and Replacement. In [7] the approach to privacy protection is based on detecting skin tones in images and replacing it with other colors, hence making it impossible to determine the race of the individual. This process works in the pixel domain (see Figure 1). Cameras systems based on this method have been developed; the idea is to detect the face and then overlay this information with a dark patch or a mosaic or any other obfuscation technique before the video is recorded. At the end no copies of the original faces will exist. This method is compression and transcoding independent. However, specifically in the case of just color replacement, it does not hide the identity completely and since the cameras perform the replacement before recording the video, the method is not reversible. Another issue is that the skin replacement method is applicable only for privacy involving human identity and cannot be used in applications where identity of nonhuman objects has to be protected, for example, a car's license tag.

Low Quality ROI Coding. In the privacy system proposed in [8], the authors propose to decrease the ROI quality in JPEG2000, locating this information in the lowest quality layer of the codestream. This ensures poor visual quality in lossy compression, up to invisibility if required. This proposal is in the bitstream domain and hence specific to compression standards used and it is not reversible. Hence, when a suspicious activity needs to be investigated, the identities cannot be uncovered to meet the security needs.

An ideal surveillance system should ensure individual privacy while meeting law enforcement/security needs. Key characteristics of such surveillance systems are briefly discussed below.

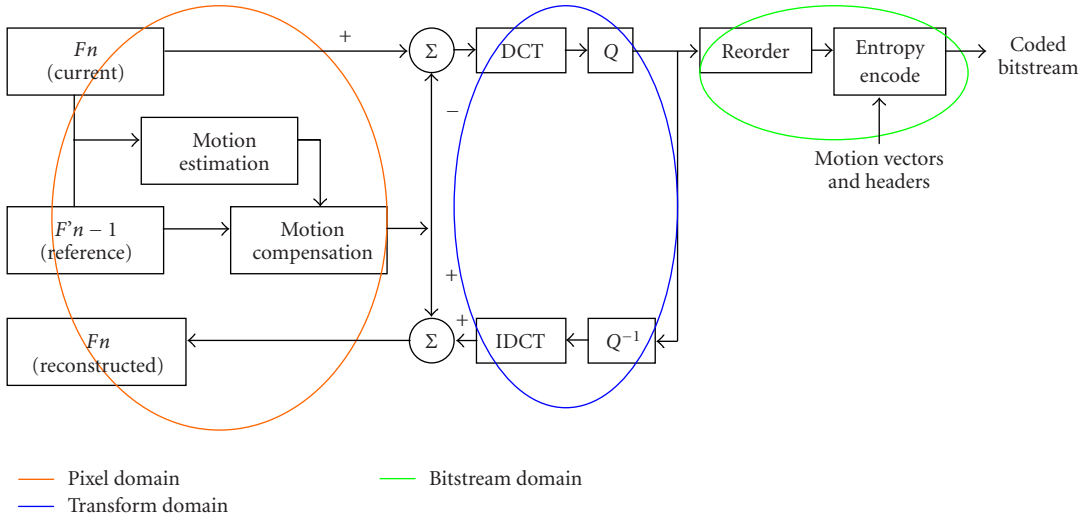


Figure 1: Video encoder highlighting opportunities for encryption.

(1) *Provides Complete Privacy*. A surveillance system should provide complete privacy by hiding portions of the video that reveal the identity of individuals. These features include faces, license tags on cars, and textual information/markings. Assuming the identifying features in a video can be detected, a surveillance system should then hide these features. A few ways of hiding such features are: (1) removing/replacing the corresponding pixels from the frame and (2) encrypt the corresponding pixels. Completely encrypting the video streams will not serve the purpose of surveillance as the monitors cannot understand the context without decrypting the video first.

(2) *Balance Security Needs*. It is an important requirement to balance the security needs. Meeting the needs of the law enforcement personnel implies that a means of revealing the hidden identity shall be provided. If the identities in a video are hidden by removing the pixels corresponding to the identity revealing features, then these features can never be recovered. On the other hand, if the identify revealing features are hidden by encrypting, the hidden areas can be uncovered by decrypting the relevant regions. Rights and privileges to access hidden identities can be managed through well designed corporate security policies. For example, hidden regions in a video cannot be decrypted unless explicitly authorized by the chief of security and/or a court order.

(3) *Compression Independent*. Compression independence is an important requirement that is essential to keep surveillance systems independent of the compression algorithms used. If a privacy solution depends on the compression algorithms, the system has to be redesigned for each compression algorithm used in the surveillance system. A system designed for MPEG-2 video will not work when MPEG-4 video compression is used. Privacy solutions that use coefficient scrambling [2] are compression algorithm specific and decrypting and decoding have to be integrated. Furthermore, the system cannot easily evolve to use new compression algorithms. Another drawback of this solution is that it cannot survive re-encoding or transcoding.

(4) *Survive Recoding and Transcoding*. Video surveillance can span large areas and videos captured are typically distributed over networks. Network distribution may require using a different video formats or changing the video bitrate to meet the network and receiver constraints. Surveillance videos may have to be compressed for archival purposes. More importantly, as surveillance systems evolve, there are bound to be receivers, players, or systems that require conversion to a specific format and robustness to recoding and transcoding becomes a key requirement of such surveillance systems. Privacy solutions that use coefficient scrambling [2] or solutions that are compression dependent cannot survive any recoding or transcoding or even bitrate changes. A secure system for transcoding video without decrypting was proposed in [9]. This approach uses scalable video and truncation of enhancement layers to reduce the bitrate or resolution. Since the encryption is done in the bitstream layer, video will have to be decrypted and transcoded if the bitrate or resolution has to be reduced below that of base layer or coding format has to be changed.

3. Compression Independent Reversible Encryption

This section presents a surveillance system designed to meet the key requirements discussed in Section 2. Figure 2 shows the system diagram with key components. A surveillance

camera captures video and before video encoding takes place, regions that reveal identity are encrypted. Detecting such regions is not addressed in this paper. The video frames with encrypted regions are then passed to a standard video encoder. The encryption keys are dynamically selected using a key management system. The encoded video is then distributed through a standard video communication system. The encoded video may be recoded at a lower bitrate or transcoded to a different format as needed. The video is also fed to monitoring stations that play the video in realtime. The monitors use a standard video decoder to match the encoder used. The security personnel will be able to see the video and observe but with all the identifying features hidden. This allows the security personnel to monitor the activities while the identity remains hidden.

The system can be configured to automatically decrypt and display the live video while keeping the identifying features encrypted in all recorded videos. When recorded video is played, all the identifying features are obscured through encryption. Since the proposed system is compression independent, this encrypted video can be played back on any standard video player such as a standard Media Player. However, when there is a legitimate need to decrypt and reveal the identifying features, for example to aid a criminal investigation, the video has to be played in a special player/security console that has the ability to decrypt the regions. This solution provides additional security when access to surveillance consoles with decrypting ability is further restricted.

3.1. Compression Independent Encryption. Encryption before encoding makes this system compression independent. The regions detected as containing identifying features are encrypted before the encoding stage. With this approach a standard encoder can be used for encoding and a standard decoder can be used for decoding. The encoder and decoder are not aware of the encryption used. Since video compression is a lossy process, the decoded video is not identical to the video input to the encoder. This means that the input to the decryption stage is not identical to the output of the encryption stage. This "corruption" of the encrypted data caused by lossy video encoding rules out traditional encryption algorithms such as AES.

The encryption used in the proposed solutions is based on permuting pixel values using pseudo-random permutations. The generating process for these pseudorandom permutations is based on "logarithmic signatures" as described in [10–12] and uses a secret pass phrase as a key. This pass phrase can also be automatically generated and managed by the surveillance system.

The identity revealing regions in a video are encrypted on a block basis. The region of interest is expanded to an area with width and height that are integral multiples of 16. This region expansion is done to allow encryption of 16×16 blocks as this is the standard unit of coding in most compression algorithms. The 16×16 blocks that cover the selected regions are determined and then a block based encryption is applied. The size of the block or the number of pixels in a block affects the strength of encryption. A small block size is easy to attack because of small set of permutations possible. A larger block size significantly increases the encryption strength. Larger block sizes also increase the number of pixels that are randomized because of permutations and result in a higher bitrate because of the loss in correlation. A 16×16 block is selected as this balances the encryption strength and penalty due to loss in correlation. Video codecs typically encode/decode video one 16×16 block known as a macro block (MB) at a time. The block size of 16×16 is also a unit of rate control and allows one to adjust the quantization parameter (QP) per MB and perform better rate control in order to reduce the increase in bitrate resulting from the encrypted regions. Keeping the encryption block size fixed is also necessary to support automatic recognition and decryption of encrypted regions without transmitting additional information to the decoders.

For each of the 16×16 blocks to be encrypted in a frame, a sequence of pseudorandom permutations (α_t) are applied to the cleartext sequence of blocks to yield the encrypted block sequence. Each key choice yields a sequence of random permutations (α_t) of periodicity $(16^2)! \cong 8.6 \times 10^{506}$. For the sake of economy we do not present the method of generating pseudorandom permutations here, a complete discussion of the method can be found in [10–12]. The size of the theoretical key-space (the number of logarithmic signatures, each providing a different pseudorandom permutation generator) by far exceeds $(2! \times 3! \times \cdots \times 256!)$ making a brute force attack impossible. The encryption key can be generated dynamically based on the frame number and block number. The encryption key can be varied on a per-block or per-frame basis if desired.

Figure 3 shows an example of applying permutations to a 4×4 block. The block data is rearranged using the given permutation resulting in an encrypted block.

3.2. Security Analysis. The cryptographic robustness of the technique we use has been discussed in length and theoretically established in [10–12]. Here we only make some simple observations. Once a logarithmic signature α has been selected for the symmetric group of degree 256, by means of the secret pass phrase key, a seed s_0 in the range $[1, 256!]$ is also selected by means of the secret key, and the sequence of random permutations $\alpha(s_0)$, $\alpha(s_0 + 1)$, $\alpha(s_0 + 2), \ldots$ is generated and applied to the blocks to be encrypted. The periodicity of the random sequence of permutations is $256! \cong 8.6 \times 10^{506}$, significantly larger than what is considered adequate by modern standards. Because the number of logarithmic signatures is gigantic (much larger than $(2!) \times (3!) \times \cdots \times (256!)$, a brute force attack is out of the question. If a fixed permutation were to be used for all blocks, within all frames, then one might consider the possibility of a cryptanalytic attack based on the constancy of the permutation. However in our scheme, different blocks within frames are encrypted with different and distinct elements of the random sequence of permutations. Finally, it is well established that knowledge of the statistical distribution of pixel values in no way allows for the reconstruction of

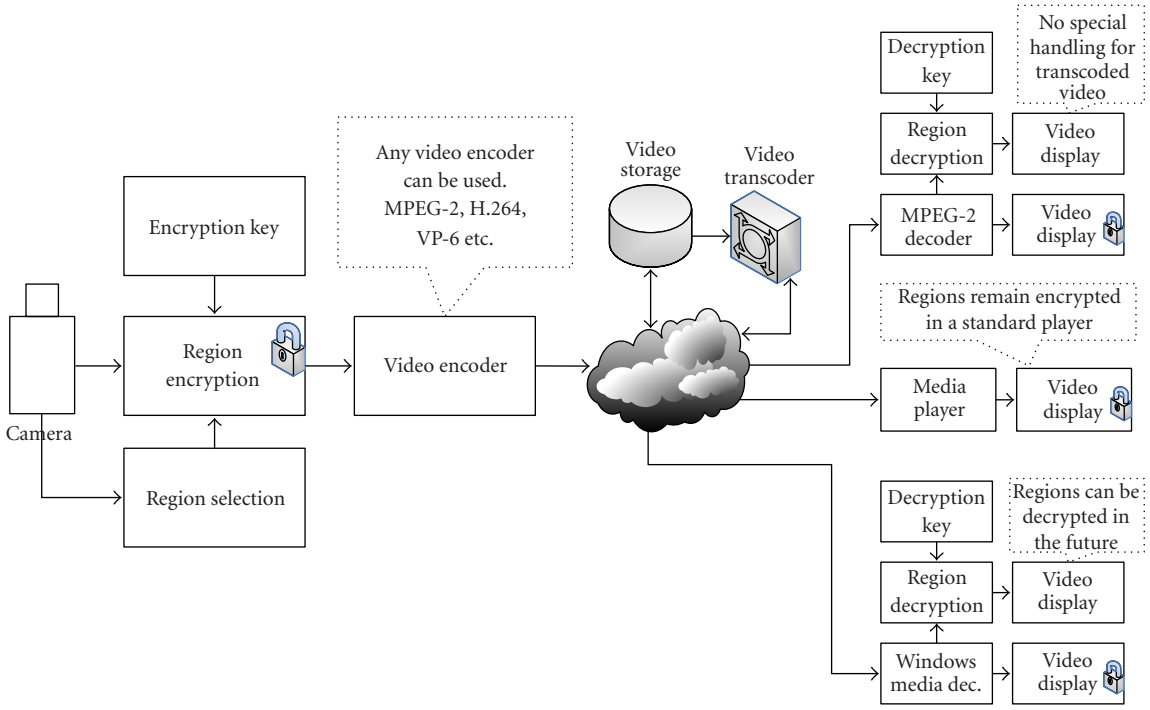


Figure 2: Surveillance System with Privacy Protection.

$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} \xrightarrow{\text{Random permutation } \pi} \begin{pmatrix} a & b & c & d & e & f & g & h & i & j & k & l & m & n & o & p \\ l & i & d & g & j & h & m & k & e & a & p & o & f & b & n & c \end{pmatrix} \longrightarrow \begin{bmatrix} l & i & d & g \\ j & h & m & k \\ e & a & p & o \\ f & b & n & c \end{bmatrix}$$

Figure 3: 4×4 block encrypted using a random permutation.

the encrypted image. For instance in [13] by applying an appropriate permutation to the pixels of a grey scale image of Marilyn Monroe produces an image of John Wayne.

The encrypted macro blocks are then encoded using a standard encoding process. Since encryption rearranges the pixels in a block, the correlation is decreased and the compression rate for the block decreases. Figure 4 shows an example of original and encrypted video before the encoding operation. The application of the proposed encryption thus leads to increase in bitrate and the amount of increase depends on the content and the number of blocks encrypted. The video can be decoded on any standard decoder for the compression format used and the video remains encrypted after decoding and a key is necessary for decrypting the video.

The proposed system satisfies the key requirements of the privacy protecting surveillance systems set out in Section 2. Table 1 summarizes how each of the features are satisfied by the system.

Table 1: Tools used in the proposed surveillance system.

Feature	Tools Used To Support the Feature
Compression Independence	Pixel domain encryption
Privacy hiding	Selective encryption of regions of interest
Balance Security Needs	Allows decryption of just the selected regions/frames
Survive Recoding/Transcoding	Permutation based encryption

4. Performance Evaluation

The proposed encryption is evaluated for compression independence and robustness to transcoding. The experimental conditions are summarized in Table 2.

4.1. Compression Independence. The system is evaluated with H.264, MPEG-2, MPEG-4, and H.263 video encoders using the Crew sequences at 352×288 resolution. Face detection for the experiments was done manually and face regions are input to the system. The videos are encrypted and Figure 4 shows the video with encrypted regions that is input video

(a)

(b)

Figure 4: Original and encrypted frames from a test sequence.

Table 2: Experimental Setup.

Processor	Intel Core 2 CPU 6600 @ 2.4 GHz
Videos	Crew.yuv CIF (352 × 288), 237 frames CarTags.yuv SD (720 × 480), 240 frames
Encoder Software	Intel IPP® codec suit
GOP	I and P frames only; Number of frames between I = 15
Entropy coding	CABAC
QP (No used CBR)	28,32,36,40
Percentage of encrypted ROI MBs	Crew.yuv: 6.5% CarTag.yuv: 1%

encoders. When video with encrypted regions is encoded, the encrypted blocks also suffer distortion due to lossy coding. The loss of correlation in the encrypted blocks leads to larger non-zero coefficients and quantization of these large coefficients increases the distortion in these blocks.

(a)

(b)

Figure 5: (a) Decoded and decrypted H.264 video with Quantization Parameter (QP) of 35; (b) decoded and decrypted H.264 video with QP of 26.

Figure 5(a) shows H.264 coded video with some visible distortion, but with largely acceptable video quality for faces. As quantization parameter (QP) increases, the quality of decrypted faces deteriorates. Figure 5(a) shows that the face quality degrades when QP increases to 35.

Our experiments show that the quality of decrypted regions is good for H.264 encoded videos with QP of up to 26. The same set of experiments were repeated and the upper bounds for QP to ensure acceptable quality for decrypted regions is QP of 6 for H.263 and MPEG-4 and 3 Mbps for MPEG-2. The experiments show that the video has to be recorded with good quality in order to preserve the quality of decrypted regions.

The proposed approach increases the bitrate of the encrypted and encoded video compared to the video encoded without encryption. This increase in bitrate is the tradeoff for the desired encryption features. Figure 6 shows the plot of encrypted video bitrate versus standard video bitrates for the Crew and CarTag video sequences. The X-axis shows the bitrate without encryption and the Y-axis shows

the bitrate when portions of the video are encrypted and then encoded using the same encoding parameters. In the case of standard video encoding without encryption, the bitrate is the same and the slope of the curve is 45°. With encrypted video, the bitrate of the video increases because of loss of correlation due to encryption and the amount of increase depends on content and the number of blocks encrypted. The bitrate increase because of encryption is around 23% on average for the Crew sequence with 6.5% of encrypted blocks. For the CarTag video with 1% of blocks encrypted, the max bitrate increase is 10.2%. The total bitrate can be decreased if the encrypted regions are treated as regions of interest (ROI) and coded using a fixed QP that results in a good reconstructed object while increasing the QP for the sequence. This ROI based approach is described in the next section.

4.1.1. Fixed Region of Interest QP for Low Bitrate Surveillance. Limiting the distortion of the encrypted regions, referred to as Region of Interest (ROI) here, allows surveillance systems to record video at lower bitrates. The relatively higher quality of ROI maintains the quality of decrypted regions at an acceptable level. The upper bound on QP, however, increases the bitrate. This increase in bitrate is, as in the previous case, the cost of providing privacy in video surveillance systems.

Figure 7(a) shows the quality of video at low bitrates, encoded with QP of 40. The high QP value distorts the encrypted blocks and the decrypted areas are essentially lost. Figure 7(b) shows the video with encoded with QP of 40 but with ROI QP set to 26. With minimal quality maintained for the ROI, the faces can be clearly seen in the decoded video. As in the previous case the bitrate of the video increases. Figures 7(c) and 7(d) show the effect of fixing ROI QP for CarTag video capturing license plates.

Figure 8 shows the relative increase in bitrate with the QP value used for the ROI. The increase depends on the ROI QP. As expected the bitrate increases are higher for lower QP and lower for higher QP. Keeping ROI at 26 gives the lowest increase in bitrate over standard video. As the ROI QP decreases, the output bitrate increases. Surveillance systems should select the right QP to meet the bitrate requirements.

With a QP of 26, the encrypted video takes 23% more bits as explained in Section 4. However, if the ROI QP is fixed at 26 and if a higher QP is used for the video sequences, the video bitrate can be reduced without affecting the quality of encrypted regions. Table 3 shows the reduction in overall bitrate compared to the encrypted video coded with QP of 26. The encrypted video at QP 26 was taken as the base bitrate for percentage comparison.

Figure 9 shows the PSNR of the ROI (faces) with standard encoding and the encoding of encrypted regions proposed in this paper. The figure shows that the minimum expected ROI PSNR when the ROI QP is fixed at 20 is approximately 39 dB. Another important observation is that the ROI PSNR, when the ROI QP is fixed, is better or equal to that of non-ROI decryption and decoding process. However, the key metrics for evaluating this system are the ability to encrypt selected regions, ability to support multiple video compression algorithms, and the resulting increase in bitrate. Based on these metrics we conclude that the proposed system meets all the requirements of surveillance systems that can protect individual privacy rights. The tradeoff of bitrate for ensuring a minimum quality for the encrypted regions is a reasonable tradeoff and is within bounds of practical systems.

Table 3: Comparison between a video encoded with a QP of 26 and videos with QP > 26 and a fixed QP ROI of 26.

General video QP	Bitrate (Kbps) ROI QP = 26	% bitrate reduction compared to video coded with QP of 26
26	1571	0%
27	1438	−8%
28	1311	−17%
29	1184	−25%
30	1085	−31%
35	734	−53%
40	561	−64%
45	478	−70%

4.2. Robustness to Transcoding. Video surveillance systems can use different formats and there is a need to convert the video from one format to the other. If regions of video are encrypted, any transcoding or recoding would "corrupt" the encryption and decryption would not be possible. The proposed permutation based encryption, however, survives such transcoding and recoding operation. The key benefit of the proposed systems is that only the endpoints—capture end and authorized playback end—have to be aware of the encryption.

The system is evaluated with H.264 to MPEG-2 and H.264 to MPEG-4 video transcoders, using the Crew sequence at 352 × 288 resolution. The experiments were based on the video encoders available in the Intel Integrated Performance Primitives (IPP) SDK. Face detection for the experiments was done manually and face regions are input to the system. As in the compression independence experiments, regions of interest are identified, encrypted and encoded using H.264 video encoding. The H.264 video is then transcoded to MPEG-2 and MPEG-4—simulating a scenario for legacy codec support in video surveillance system.

When video with encrypted regions is encoded or it goes through a transcoding process, the encrypted blocks also suffer distortion due to lossy coding. The loss of correlation in the encrypted blocks leads to larger non-zero coefficients and quantization of these large coefficients increases the distortion in these blocks. Figure 10 shows the screenshot of decoded and MPEG-2 video followed by decryption. The MPEG-2 video was created by transcoding an encrypted H.264 video encoded with QP of 26.

Experiments show that the video has to be recorded with good quality in order to preserve the quality of decrypted regions. When lower bitrate surveillance is necessary, the encoder can enforce an upper bound on the QP used for the

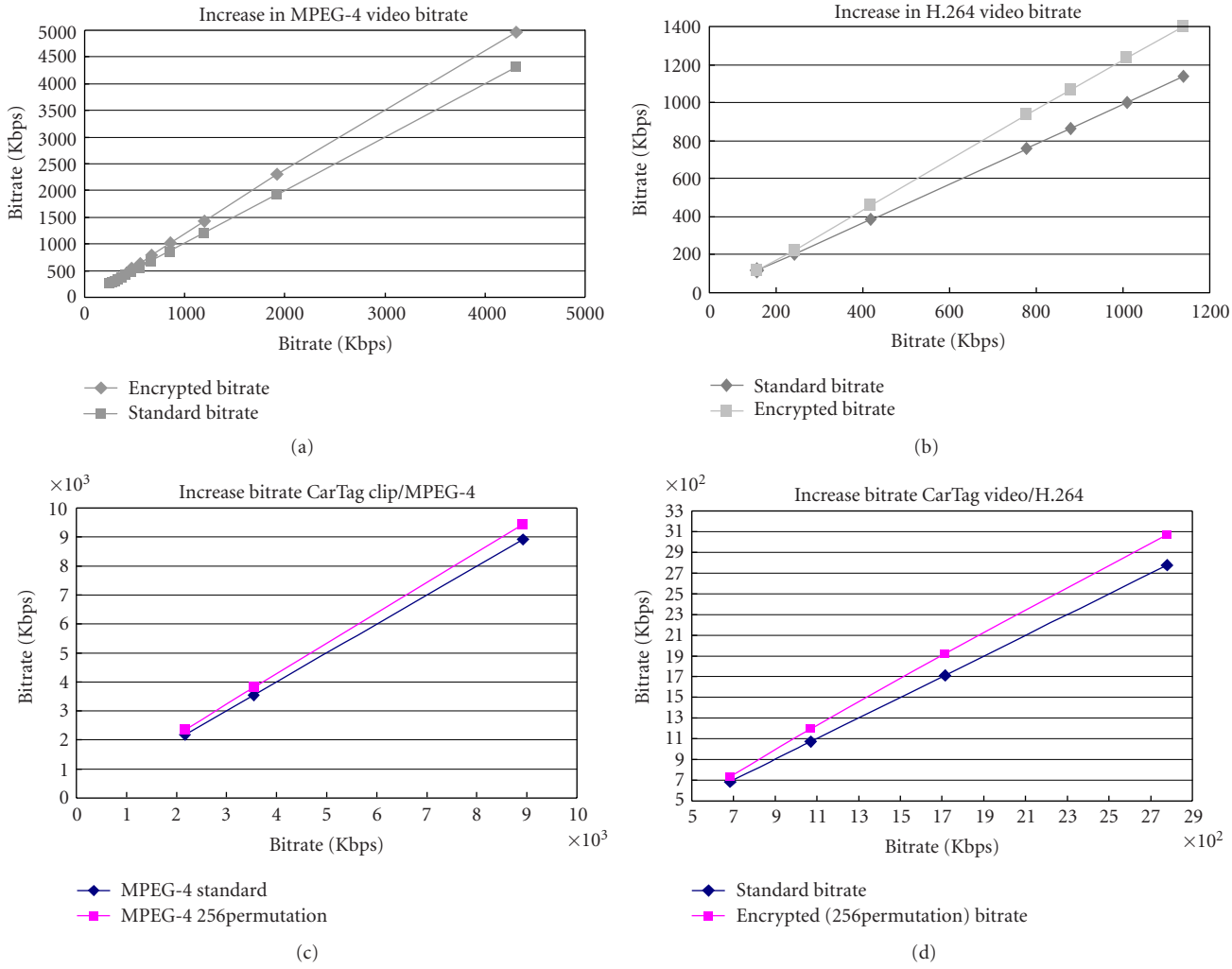


Figure 6: Bitrate increase because of encryption. (a) MPEG-4 Crew Video, (b) H.264 Crew Video, (c) MPEG-4 Cat Tag Video, (d) H.264 Car Tag Video.

encrypted blocks. In the encoder independence experiments, we showed that for H.264 video a QP of 26 is necessary to maintain the quality of decrypted video and this is used as a basis for comparisons. It was chosen as a base because it gives a good tradeoff between quality and bitrate when H.264 is used.

In order to find the upper bounds in a surveillance environment with transcoders (H.264 to MPEG-2, H.264 to MPEG-4 and H.264 to H.263), the input video was encrypted and coded with H.264 with a QP of 26 and different QPs and bitrates were evaluated in H.263, MPEG-2 and MPEG-4 encoders. Table 4 shows the upper bound results. To maintain video quality, the transcoded video has to be encoded at a bitrate of 3 Mbps for MPEG-2, and use a QP of 4 for H.263 and MPEG-4. This high bitrate requirement can be reduced by encoding the encrypted regions in the input H.264 with a higher quality (QP < 26).

4.2.1. Transcoding with a Fixed ROI QP. Limiting the distortion of the encrypted regions, referred to as Region of Interest (ROI) here, allows surveillance systems to record video at lower bitrates. The relatively higher quality of ROI maintains the quality of decrypted regions at an acceptable level. Lowering the upper bound on the ROI QP increases the video bitrate. However, for transcoding purposes, lowering the ROI QP below the upper bound (26) can decrease the overall bitrate of the transcoded video. With higher quality for the ROI (lower QP), transcoders can use a lower bitrate and still preserve the quality of the decrypted regions (ROI) at an acceptable level as the impact of lower transcoder bitrate on high quality ROI would be small.

Table 4: Upper bounds for transcoding.

Transcoders	Upper bound
H264/MPEG-2	MPEG-2 bitrate of 3 Mbps
H264/MPEG-4	MPEG-4 QP of 4
H264/H263	H263 QP of 4

Figure 7: (a), (c) Decoded and decrypted H.264 video with QP of 40; (b), (d) decoded and decrypted H.264 video with QP of 40 and with ROI QP fixed to 26.

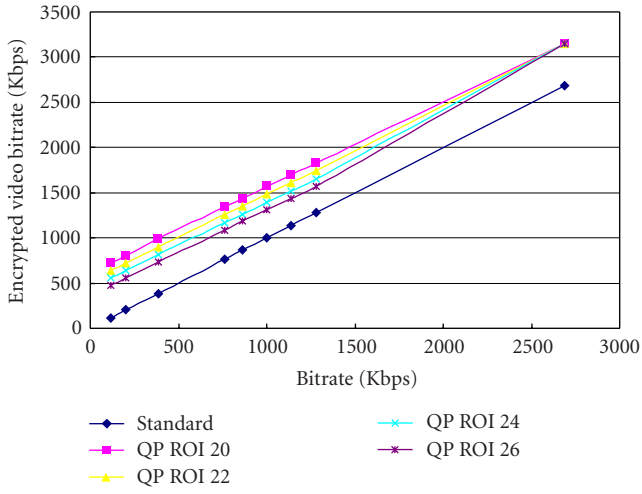


Figure 8: Bitrate increase because of constant QP for the ROI.

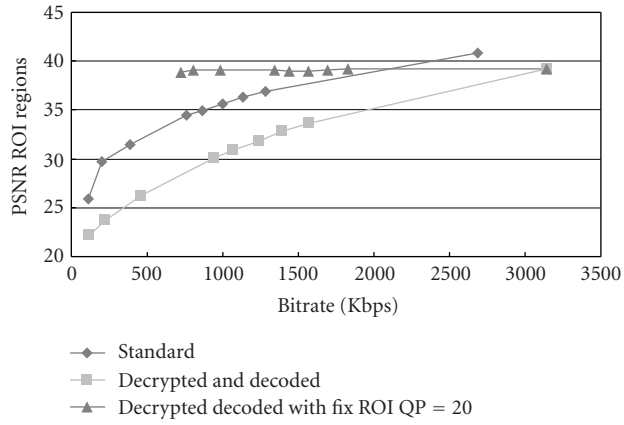


Figure 9: Bitrate versus ROI PSNR.

Figure 11(a) shows the quality of video at low bitrates; H.264 encoded with QP of 35 and transcoded to MPEG-4 with QP of 5. The high H.264 QP value distorts the encrypted blocks in the transcoding process and the decrypted area is essentially lost. Figure 11(b) shows the video encoded with QP of 35 but with ROI QP set to 20. With quality maintained for the ROI and video transcoded to MPEG-4 with the same QP of 5, the faces can be clearly seen in the transcoded video. Compared to the case shown in Figure 6, when higher quality ROI is used, the bitrate of the transcoded video increases by 10%. However, a high quality is maintained for the faces. Using a lower QP for the ROI thus increases the H.264 bitrate but allows lower bitrates for the transcoded video.

4.3. Automatic Detection of Encrypted Regions. The proposed system allows playback of videos with encrypted regions

Figure 10: Transcoded H264 to MPEG-2 video and decrypted. H264 QP video of 26, and MPEG-2 Bitrate of 3 Mbps.

on standard video players. The encrypted regions can be decrypted, with appropriate authorization, on special decrypting consoles. Decrypting a video requires identification of encrypted blocks. The encrypted regions can be signaled to decrypting decoders using an additional data stream. The additional data stream increases the amount of data to be transmitted and managed. We have developed a solution to identify the encrypted regions automatically. The proposed solution is based on measuring the randomness in a decoded block. Two approaches are evaluated (1) measuring high frequency coefficients of DCT and (2) measuring the number of row-wise and column-wise changes. Figure 12 shows a block diagram with key components of automatic detection.

4.3.1. Encrypted Block Detection Using DCT. With this approach, the randomness of a block is measured by examining the high frequency coefficients. A 16×16 DCT is applied to all macro blocks in the decoded video. A block is marked as a candidate for decryption when non-zero coefficients are present in the bottom-right 3×3 block of the 16×16 DCT block (high frequency coefficients). A block is marked as encrypted if the sum of the absolute value of the high frequency coefficients in the bottom-right 3×3 block of the 16×16 DCT block is greater than 5. The threshold is determined experimentally after evaluating encoding videos at various bitrates.

4.3.2. Encrypted Block Detection Using Row-Column Differences. This approach is similar to edge detection in a block; the pixel values are compared with the neighbors, first along rows and then along columns. In our case, if the difference between neighboring pixels is greater than 11, the big-pixel-difference count is incremented by 1. If the total number of big-pixel-differences is greater than 115, the block is marked as encrypted. Thresholds for this method are also determined experimentally. We use Crew.yuv video as the basic video for tuning.

(a)

(b)

Figure 11: Transcoded H264 to MPEG-4 with QP of 5; (a) H264 general QP of 35 (b) H264 general QP of 35 and ROI QP of 20. MPEG-4 QP of 5. Video bitrate 1220 Kbps.

Table 5: Auto Detection Performance Summary for Crew Video.

	Crew		CarTag	
	Ro-Col Diff	DCT	Ro-Col Diff	DCT
Correctly Classified MBs	91132	93780	3482	3525
Incorrectly Classified as Encrypted	2722	74	9151	9108
False Negatives (marked as not encrypted)	547	60	43	0
False Positives	2175	14	5669	5583
Precision	97.1%	99.92%	27.9%	27.6%
Recall	90.92	99.00	98.7%	100%

4.3.3. Performance of Automatic Encrypted Block Detection. Experiments were conducted to detect the encrypted blocks in Crew and CarTag videos. The Crew video has 237 frames with 6023 encrypted 16×16 blocks out of a total of 93, 852 16×16 blocks. The CarTag video has 240 frames with 3525 encrypted 16×16 blocks out of a total of 324,000 16×16 blocks.

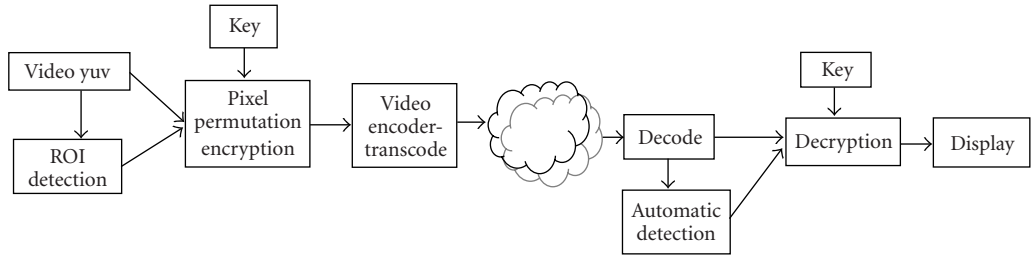


Figure 12: Automatic detection of encrypted regions.

The DCT based method clearly outperforms the Row-col method but is computationally more expensive. When a block is incorrectly marked as encrypted, the decryption performed on the block (i.e., inverse permutation) essentially encrypts the block. Since the false positive rate is very low, the effect of incorrect classification is minimal. To overcome this, users can interactively undo the decryption when necessary. It is also important to mention the importance of increase ROI quality with a higher QP, not only because that permit us compress the rest of the image even more without fear of losing ROI quality, but also because ROI enhancement quality also increase the auto-detection accuracy.

The detection performance of these methods will drop when video includes regions that have high frequencies naturally. We discovered this in scenes with grass and waves in an ocean. The CarTag video has grass and leaves in the background and result in a large number of false positives. The DCT method has a recall of 100% and is able to detect all the encrypted blocks. The decreased performance is in the form of increased false positives and these can be interactively addressed when the surveillance videos are reviewed.

4.4. Comparative Evaluation. The proposed system is compared against other known approaches. The performance is summarized in Table 6. The proposed solution meets all the requirements for a surveillance system that balances privacy and law enforcement needs.

5. Conclusion

This paper presents a system for encrypting selected regions in videos. The system can be used for ensuring privacy in video surveillance by hiding the identity revealing regions in the video. The encrypted video can be transcoded and or decrypted at a later time with the right decryption keys. The proposed system is independent of the compression algorithms used. The system was tested using H.264 to MPEG-2, H.264 to MPEG-4, and H.264 to H.263 video transcoders in order to verify video quality. The bitrate increases with the number of encrypted blocks. The proposed reversible encryption increases the video bitrate and experiments with H264 to MPEG-4 show that the increase is up to 23% for high bitrate videos with about 7% of blocks encrypted. This bitrate can be reduced by keeping the ROI QP constant and increasing the frame QP. The increase in bitrates depends on the type of video and the size of encrypted regions. The increase in bitrate is a reasonable cost to pay for protecting individual privacy. The proposed solution does not require any additional information to detect and decrypt encrypted regions. A DCT based method was developed to automatically detect the encrypted regions thus making the system truly independent of the compression algorithms used.

Appendices

A. Random Permutations from Logarithmic Signatures

In this section we briefly present notation, definitions and some basic facts about logarithmic signatures, covers for finite groups and their induced mappings. For more details the reader is referred to [11, 12, 14].

Let $\mathcal{G}$ be a finite group. A sequence $\alpha = [A1, A2, \ldots, As]$ of ordered subsets of $\mathcal{G}$ is said to be a logarithmic signature (LS) for $\mathcal{G}$ if each element $g \in \mathcal{G}$ can be expressed uniquely as a product of the form

$$g = q_1 \cdot q_2 \cdots q_{s-1} \cdot q_s \tag{A.1}$$

for $q_i \in A_i$. The *length* of α is defined by $l(\alpha) = \sum_{i=1}^{s} |A_i|$.

In addition to the fundamental defining property above, we require that $s > 1$, $|A_i| \geq 2$ for each $i \in \{1, \ldots, s\}$, and that $l(\alpha)$ is bounded by a polynomial in $\log |\mathcal{G}|$.

Let $\alpha = [A_1, \ldots, A_s]$ be an LS for ζ with $r_i = |A_i|$, then the A_i are called the *blocks* of α and the vector $(r_1, \ldots, r_s)$ of block lengths r_i the *type* of α.

Thus, the *length* of α is the integer $\mathcal{G}_l$:

$$l(\alpha) = \sum_{i=1}^{s} r_i \tag{A.2}$$

Let $\Gamma = \{(\mathcal{G}_l, \alpha_l)\}_l \in \mathbb{N}$ be a family of pairs, indexed by the security parameter l, where $\mathcal{G}_l$ are groups in a common representation, and where α_l is a specific LS for $\mathcal{G}_l$ of length polynomial in l. We say that Γ is *tame* if there exists a probabilistic polynomial time algorithm **A** such that for each $g \in \mathcal{G}_l$, **A** accepts (α_l, g) as input, and outputs a factorization $\varphi(g)$ of g with respect to α_l (as in (A.1)) with overwhelming probability of success. We say that Γ is *wild* if for any probabilistic polynomial time algorithm **A**, the probability that **A** succeeds in factorizing a random element g of $\mathcal{G}$ is

Table 6: Comparative evaluation of the proposed solution.

Video selective encryption method	Privacy completeness	Compression Independence	Reversibility	Robustness to Transcoding	Domain	General bit-rate increases
Obfuscation [1]	Yes	Yes	No	No	Pixel	No
Transform-domain scrambling coefficients [2]	Yes	No	Yes	No	Transform	Yes
Invertible cryptographic obscuration [6]	Yes	No	Yes	No	Bit-stream	Yes
Skin tone replacement [7]	Not always	Yes	No	Yes	Pixel	No
Lower quality ROI [8]	Yes	No	No	No	Bit-stream	No
Proposed Solution	Yes	Yes	Yes	Yes	Pixel	Yes

negligible. For finite groups there are instances $\{(\mathcal{G}_l, \alpha_l)\}_l$ where the factorization in (A.1) is believed to be hard, for instance, such an example can be constructed by considering hard instances of the traditional discrete logarithm problem [12].

Suppose that $\mathcal{G}_0 = \{1\} < \mathcal{G}_1 < \cdots < \mathcal{G}_{s-1} < \mathcal{G}_s = \mathcal{G}$ is a strictly ascending chain of subgroups of $\mathcal{G}$, and for each $i \in \{1, \ldots, s\}$ suppose that $\mathbf{A}_i$ is a complete set of closest representatives of the subgroup $\mathcal{G}_{i-1}$ in $\mathcal{G}_i$. Then, $\alpha = [A_1, A_2, \ldots, A_s]$ is a logarithmic signature for $\mathcal{G}$. Logarithmic signatures obtained this way, and certain transforms of such signatures, are said to be transversal and are generally known to be tame [11]. If $|A_i| = r_i$, it is shown [11] that the total number of transversal log signatures that can be produced from a single chain $\mathcal{G}_0 = \{1\} < \mathcal{G}_1 < \cdots < \mathcal{G}_s$ is an astonishing:

$$\prod_{i=1}^{s} \left(\prod_{j=1}^{i-1} r_j \right)^{r_i} r_i!. \tag{A.3}$$

For example for the well known Mathieu group M24 there are at least 10^{600} transversal logarithm signatures of type (24,23,22,21,20,48).

Let $\alpha = [A_1, A_2, \ldots, A_s]$ be a logarithmic signature of type $(r_1, r_2, \ldots, r_s)$ for $\mathcal{G}$ with $A_i = [a_{i,1}, a_{i,2}, \ldots, a_{i,ri}]$ and let $\prod_{i=1}^{s} r_i$. Let $m_1 = 1$ and $\prod_{j=1}^{i-1} r_j$.

Let τ denote the canonical bijection from $\mathbb{Z}_{r_1} \oplus \mathbb{Z}_{r_2} \cdots \oplus \mathbb{Z}_{r_s}$ on $\mathbb{Z}_m$, that is,

$$\begin{aligned} \tau &: \mathbb{Z}_{r_1} \oplus \mathbb{Z}_{r_2} \cdots \oplus \mathbb{Z}_{r_s} \longrightarrow \mathbb{Z}_m, \\ \tau(j_1, j_2, \ldots, j_s) &:= \sum_{i=1}^{s} j_i m_i. \end{aligned} \tag{A.4}$$

Using τ we now define the surjective mapping $\breve{\alpha}$ induced by α.

$$\begin{aligned} \breve{\alpha} &: \mathbb{Z}_m \longrightarrow \mathcal{G}, \\ \breve{\alpha}(x) &:= a_1, j_1 \cdot a_2, j_2 \cdots a_s, j_s, \end{aligned} \tag{A.5}$$

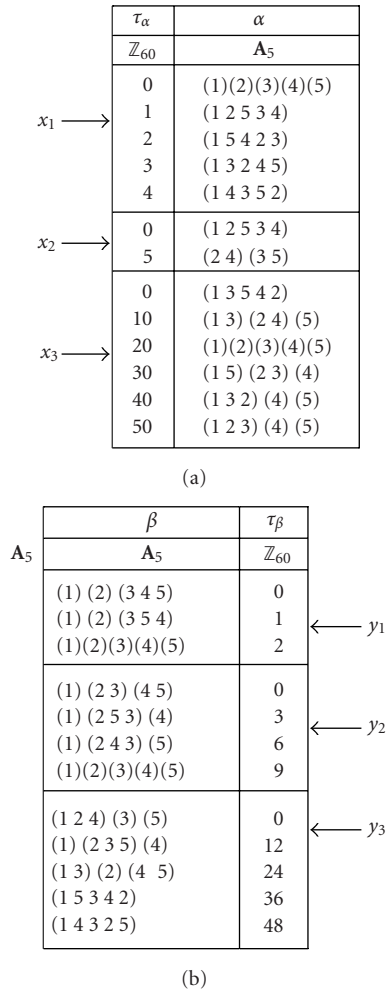
where $(j_1, j_2, \ldots, j_s) = \tau^{-1}(x)$. Since τ and τ^{-1} are efficiently computable, the mapping $\breve{\alpha}(x)$ is efficiently computable.

In [11, 14] the authors show how a pair of logarithmic signatures (α, β) can be selected as a secret key for a symmetric cryptographic system. Here we only need one randomly selected logarithmic signature, say α, for the full symmetric group S_{256} on 256 letters, and we proceed to compute a sequence of pseudo-random permutations by using the sequence $\breve{\alpha}(x_0), \breve{\alpha}(x_0+1), \breve{\alpha}(x_0+2), \ldots$. Provably, the sequence has periodicity the order of the group, here 256!. Thus we see that to generate securely random permutations it suffices to use a single logarithmic signature α. It is known that a plaintext attack against this method of generating permutations requires $O(l(\alpha))$ tests. But in practice even a single permutation will be very hard to obtain as the permutations shuffling successive images constantly change from image to image. If extreme security is required, then the system can be altered to computing a sequence of the form $\sigma(x0), \sigma(x1), \ldots$ where $\sigma(x) = \breve{\alpha}\breve{\beta}^{-1}\breve{\gamma}$, and where α, β and γ are logarithmic signatures of our group. Note that the part $\breve{\alpha}\breve{\beta}^{-1}$ constitutes PGM symmetric key encryption which remains unbroken since 1977.

B. An Example

We present a small example involving two logarithmic signatures α and β for the alternating group $\mathbf{A_5}$. The types of α and β are (5, 2, 6) and (3, 4, 5) respectively, and $|A_5| = 5 \cdot 2 \cdot 6 = 3 \cdot 4 \cdot 5 = 60$. In Figure 13, the blocks of α and β are listed vertically. To compute τ^{-1} and τ efficiently we attach canonical logarithmic signatures τ_α and τ_β of the additive group $\mathbb{Z}_{60}$ to the left of α and to the right of β. The respective types of τ_α and τ_β are (5, 2, 6) and (3, 4, 5), just as for α and β.

We now illustrate how $\breve{\alpha} : \mathbb{Z}_{60} \rightarrow \mathbf{A}_5$ is computed in practice. Any element $x \in \mathbb{Z}_{60}$ can be written uniquely as the sum of elements of τ_α, using exactly one element from each block. Determining this decomposition of x involves a greedy selection of components, one from each block, sequentially from the bottom block upwards, and essentially determines $\tau^{-1}(x) = (j_1, j_2, j_3)$. If x_i are the elements of $\mathbf{A_5}$ corresponding to the j_i, we then compute: $\breve{\alpha}(x) = x_1 x_2 x_3$. In particular, if $x = 47$, we have $47 = 40 + 5 + 2$ and the



	τ_α	α
	$\mathbb{Z}_{60}$	$\mathbf{A}_5$
x_1 →	0	(1)(2)(3)(4)(5)
	1	(1 2 5 3 4)
	2	(1 5 4 2 3)
	3	(1 3 2 4 5)
	4	(1 4 3 5 2)
x_2 →	0	(1 2 5 3 4)
	5	(2 4) (3 5)
x_3 →	0	(1 3 5 4 2)
	10	(1 3) (2 4) (5)
	20	(1)(2)(3)(4)(5)
	30	(1 5) (2 3) (4)
	40	(1 3 2) (4) (5)
	50	(1 2 3) (4) (5)

(a)

$\mathbf{A}_5$	β	τ_β	
	$\mathbf{A}_5$	$\mathbb{Z}_{60}$	
	(1) (2) (3 4 5)	0	
	(1) (2) (3 5 4)	1	← y_1
	(1)(2)(3)(4)(5)	2	
	(1) (2 3) (4 5)	0	
	(1) (2 5 3) (4)	3	← y_2
	(1) (2 4 3) (5)	6	
	(1)(2)(3)(4)(5)	9	
	(1 2 4) (3) (5)	0	
	(1) (2 3 5) (4)	12	← y_3
	(1 3) (2) (4 5)	24	
	(1 5 3 4 2)	36	
	(1 4 3 2 5)	48	

(b)

Figure 13: Two logarithmic signatures of $\mathbf{A_5}$.

components $j_1 = 2$, $j_2 = 5$, $j_3 = 40$ point to elements $x_1 = (1\ 5\ 4\ 2\ 3)$, $x_2 = (2\ 4)(3\ 5)$, and $x_3 = (1\ 3\ 2)$ of $\mathbf{A_5}$. We then compute $\breve{\alpha}(47) = x_1x_2x_3 = (1\ 5\ 4\ 2\ 3)\cdot(2\ 4)(3\ 5)\cdot(1\ 3\ 2) = (1\ 2\ 5)$.

If we now factorize $y = \breve{\alpha}(x)$ with respect to the second logarithmic signature β, we obtain $y = y_1y_2y_3$. From the elements y_i, the corresponding elements of the additive τ_β are obtained and their sum is formed. In our particular case, $y = (1\ 2\ 5) = y_1y_2y_3 = (3\ 5\ 4)\cdot(2\ 5\ 3)\cdot(1\ 2\ 4)$, corresponding to the τ_β components 1, 3, 0, respectively. Thus, $\breve{\beta}^{-1}((1\ 2\ 5)) = 1 + 3 + 0 = 4$. We would like to mention that in this example, α and β belong to the class of tame log signatures, in fact β is supertame. Here, we do not explain further how the factorization $y = y_1y_2y_3$ is obtained efficiently. For further details please see [11].

When the underlying group is chosen appropriately the bijections $\breve{\alpha}\breve{\beta}^{-1}$ can be used as cryptographic transformations with key (α, β) in symmetric cryptosystem PGM [11], or as cryptographic primitives in other systems.

References

[1] A. Senior, S. Pankanti, A. Hampapur, et al., "Enabling video privacy through computer vision," *IEEE Security and Privacy*, vol. 3, no. 3, pp. 50–57, 2005.

[2] F. Dufaux and T. 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.

[3] P. Carrillo, H. Kalva, and S. Magliveras, "Compression independent object encryption for ensuring privacy in video surveillance," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '08)*, pp. 273–276, Hannover, Germany, June 2008.

[4] I. Martínez-Ponte, X. Desurmont, J. Meessen, and J. Delaigle, "Robust human face hiding ensuring privacy," in *Proceedings of the Workshop on the Integration of Knowledge, Semantics and Digital Media Technology (WIAMIS '05)*, Montreux, Switzerland, April 2005.

[5] K. Martin and K. N. Plataniotis, "Privacy protected surveillance using secure visual object coding," *IEEE Transactions of Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1152–1162, 2008.

[6] T. E. Boult, "PICO: privacy through invertible cryptographic obscuration," in *Proceedings of the Computer Vision for Interactive and Intelligent Environment*, pp. 27–38, November 2005.

[7] M. Berger, "Privacy mode for acquisition cameras and camcorders," US patent no. 6,067,399, May 2000.

[8] D. Chen, Y. Chang, R. Yan, and J. Yang, "Tools for protecting the privacy of specific individuals in video," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, Article ID 75427, 2007.

[9] S. J. Wee and J. G. Apostolopoulos, "Secure scalable streaming enabling transcoding without decryption," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '01)*, vol. 1, pp. 437–440, 2001.

[10] S. S. Magliveras and N. D. Memon, "Random permutations from logarithmic signatures," in *Proceedings of the 1st Great Lakes Computer Science Conference Computing in the 90's*, vol. 507 of *Lecture Notes in Computer Science*, pp. 91–97, Springer, 1989.

[11] S. S. Magliveras and N. D. Memon, "Algebraic properties of cryptosystem PGM," *Journal of Cryptology*, vol. 5, no. 3, pp. 167–183, 1992.

[12] S. S. Magliveras, T. van Trung, and D. R. Stinson, "New approaches to designing public key cryptosystems using one-way functions and trap-doors in finite groups," *Journal of Cryptology*, vol. 15, pp. 285–297, 2002.

[13] D. Socek, H. Kalva, S. S. Magliveras, O. Marques, D. Culibrk, and B. Furht, "New approaches to encryption and steganography for digital videos," *Multimedia Systems*, vol. 13, no. 3, pp. 191–204, 2007.

[14] S. S. Magliveras, "A cryptosystem from logarithmic signatures of finite groups," in *Proceedings of the 29th Midwest Symposium on Circuits and Systems*, pp. 972–975, Elsevier, 1986.

Composition Comments

1. Please specify one corresponding author and his/her academic email address.

2. We made the highlighted change as per journal style and to avoid repetition of references. Please check.

3. We renamed "Figure 1A" to "Figure 13" as per journal style. Please check the changes due to this amendment.

Author(s) Name(s)

It is very important to confirm the author(s) first and last names in order to be displayed correctly on our website as well as in the indexing databases:

Author 1

Last Name: Carrillo

First Name: Paula

Author 2

Last Name: Kalva

First Name: Hari

Author 3

Last Name: Magliveras

First Name: Spyros