# Compression of Encrypted Visual Data

Michael Gschwandtner, Andreas Uhl, and Peter Wild*

Department of Computer Sciences
Salzburg University, Austria
Email: {mgschwan,uhl,pwild}@cosy.sbg.ac.at

**Abstract.** Chaotic mixing based encryption schemes for visual data are shown to be robust to lossy compression as long as the security requirements are not too high. This property facilitates the application of these ciphers in scenarios where lossy compression is applied to encrypted material – which is impossible in case traditional ciphers should be employed. If high security is required chaotic mixing loses its robustness to compression, still the lower computational demand may be an argument in favor of chaotic mixing as compared to traditional ciphers when visual data is to be encrypted.

## 1 Introduction

A significant amount of encryption schemes specifically tailored to visual data types has been proposed in literature during the last years (see [6, 10] for extensive overviews). The most prominent reasons not to stick to classical full encryption employing traditional ciphers like AES [3] for such applications are

- to reduce the computational effort (which is usually achieved by trading off security as it is the case in partial or soft encryption schemes),
- to maintain bitstream compliance and associated fuctionalities like scalability (which is usually achieved by expensive parsing operations and marker avoidance strategies), and
- to achieve higher robustness against channel or storage errors.

Compensating errors in transmission of data, especially images, is fundamental to many applications. One example is digital video broadcast or RF transmissions which are also prone to distortions from atmosphere or interfering objects. One famous example for an application scenario requiring security of that type are RF surveillance cameras with their embedded processors, which are used to digitize the signal and encrypt it using state of the art ciphers.

Due to intrinsic properties (e.g. the avalanche effect) of cryptographically strong ciphers (like AES) such techniques are very sensitive to channel errors. Single bits lost or destroyed in encrypted form cause large chunks of data to be lost. Permutations have been suggested to be used in time critical applications since they exhibit significantly lower computational cost as compared to other ciphers, however, this comes at a significantly reduced security level (this is the reason why applying permutations is said be

* This work has partially been funded by the Austrian Science Fund (FWF), project no. 15170.

a type of "soft encryption"). Hybrid pay-TV technology has extensively used line permutations (e.g. in the Nagravision / Syster systems), many other suggestions have been made to employ permutations in securing DCT-based [11, 12, 11] or wavelet-based [7, 13] data formats. In addition to being very fast, permutations have been identified to be a class of cryptographic techniques exhibiting extreme robustness in case transmission errors occor [9].

The idea of using invertible two-dimensional chaotic maps (CMs) on a square to create symmetric block encryption schemes for visual data is not new and is described in detail in [5] or [2]. Bearing in mind that this type of crypto systems mainly relies on permutations makes them interesting candidates for the use in error-prone environments. Taken this fact together with the very low computational complexity of these schemes, wireless and mobile environments could be potential application fields. In related work we have shown that indeed CMs can cope well with static and random value errors, however, no robustness could be observed with respect to buffer errors since CMs are sensitive to changes in initial conditions.

In this work we focus on an issue different to those discussed so far at first sight, however, this topic is related to the CMs' robustness against value errors: we will investigate the compression of encrypted visual material. Clearly, data encrypted with classical ciphers can not be compressed well: due to the statistical properties of encrypted data no data reduction may be expected using lossless compression schemes, lossy compression schemes can not be employed since the reconstructed material can not be decrypted any more due to compression artifacts. For these reasons, compression is always required to be performed prior to encryption when classical ciphers are used. However, for certain types of application scenarios it may be desirable to perform compression after encryption. CMs are shown to be able to provide this functionality to a certain extent due to their robustness to random value errors. We will experimentally evaluate different CM configurations with respect to the achievable compression rates and quality of the decompressed and decrypted visual data.

A brief introduction to chaotic maps and their respective advantages and disadvantages as compared to classical ciphers will be given in Section 2. Section 3 discusses possible application scenarios requiring compression to be performed after encryption. Experimental results evaluating a JPEG compression with varying quality applied to CM encrypted data are provided in Section 4. Section 5 concludes the paper.

## 2 Chaotic Map Encryption Schemes

To achieve fast and error-robust encryption of visual data we use CM in form of a permutation based symmetric cipher. This approach was originally introduced by the work of F. Pichler and J. Scharinger [8] and has been extended by J. Fridrich [5]. CM encryption relies on the use of discrete versions of chaotic maps. The good diffusion properties of chaotic maps, such as the *Bakermap* or *Catmap* soon attracted cryptographer. Turning a chaotic map into a symmetric block cipher requires three steps, as [5] points out.

1. **Generalization**: Once the chaotic map is chosen, it is desirable to vary its behavior through parameters. These are part of the *key* of the cipher.

2. **Discretization**: Since chaotic maps usually are not discrete, a way must be found to apply the map onto a finite square lattice of points that represent pixels in an invertible manner.
3. **Extension to 3D**: As the resulting map after step two is a parameterized permutation, an additional mechanism is added to achieve substitution ciphers. This is usually done by introducing a position-dependent gray level alteration.

In most cases a final **diffusion step** is performed, often achieved by combining the data line or column wise with the output of a random number generator.

The most famous example of a chaotic map is the standard *Bakermap*:

$$B : [0,1]^2 \to [0,1]^2.$$

$$B(x,y) = \begin{cases} (2x, \frac{y}{2}) & \text{if } 0 \leq x < \frac{1}{2}; \\ (2x-1, \frac{y+1}{2}) & \text{if } \frac{1}{2} \leq x \leq 1. \end{cases}$$

This geometrically corresponds to a division of the unit square into two rectangles $[0, \frac{1}{2}[\times[0,1]$ and $[\frac{1}{2}, 1] \times [0,1]$ that are stretched horizontally and contracted vertically. Such a scheme may easily be generalized using $k$ vertical rectangles $[F_{i-1}F_i[\times[0,1[$ each having an individual width $p_i$ such that $F_i = \sum_{j=1}^{i} p_j$, $F_0 = 0, F_k = 1$. The corresponding vertical rectangle sizes $p_i$, as well as the number of iterations, are introduced as parameters. Another choice of a chaotic map is the *Arnold Catmap*:

$$C : [0,1]^2 \to [0,1]^2.$$

$$C(x,y) = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} mod\ 1$$

where $x\ mod\ 1$ denotes the fractional part of a real number $x$ by subtracting or adding an appropriate integer. This chaotic map can be generalized using a matrix $A$ introducing two integers $a, b$ such that $det(A) = 1$ as follows:

$$C_{gen}(x,y) = A \begin{pmatrix} x \\ y \end{pmatrix} mod\ 1,\ A = \begin{pmatrix} 1 & a \\ b & ab+1 \end{pmatrix}.$$

Now each generalized chaotic map needs to be modified to turn into a bijective map on a square lattice of pixels. Let $\mathcal{N} := \{0, \ldots, N-1\}$, the modification is to transform domain and codomain to $\mathcal{N}^2$. Discretized versions should avoid floating point arithmetics in order to prevent an accumulation of errors. At the same time they need to preserve sensitivity and mixing properties of their continuous counterparts. This challenge is quite ambitious and many questions arise, whether discrete chaotic maps really inherit all important aspects of chaos by their continuous versions. An important property of a discrete version $F$ of a chaotic map $f$ is:

$$\lim_{N \to \infty} \max_{0 \leq i,j < N} |f(i/N, j/N) - F(i,j)| = 0.$$

To give an example, discretizing a chaotic *Catmap* is fairy simple and introduced in [2]. Instead of using the fractional part of a real number, the integer modulo arithmetic is adopted:

$$C_{disc} : \mathcal{N}^2 \to \mathcal{N}^2.$$

$$C_{disc}(x,y) = A \begin{pmatrix} x \\ y \end{pmatrix} mod\ N,\ A = \begin{pmatrix} 1 & a \\ b & ab+1 \end{pmatrix}$$

Finally, an *extension to 3D* is inserted, that may be applied to any two-dimensional chaotic map. As all chaotic maps preserve the image histogram (and with it all corresponding statistical moments) a procedure to result in a uniform histogram after encryption is desired. The extension of a two dimensional discrete chaotic map $F : \mathcal{N}^2 \to \mathcal{N}^2$ to three dimensions consists of a position dependent gray-level shift (assuming $L$ gray-levels $\mathcal{L} := \{0, \ldots, L-1\}$) at each level of iteration:

$$F_{3D} : \mathcal{N}^2 \times \mathcal{L} \to \mathcal{N}^2 \times \mathcal{L}$$

$$F_{3D}(i,j,g_{ij}) = \begin{pmatrix} i' \\ j' \\ h(i,j,g_{ij}) \end{pmatrix},\ \begin{pmatrix} i' \\ j' \end{pmatrix} = F(i,j).$$

The map $h$ modifies the gray-level of a pixel and is a function of the initial position and color of the pixel, that is $h(i,j,g_{ij}) = g_{ij} + \overline{h}(i,j)\ mod\ L$. There are various possible choices of $\overline{h}$, we use $\overline{h}(i,j) = i \cdot j$.

Chaotic maps after step two or three are bijections of a square lattice of pixels. An additional spreading of local information over the whole image is desirable. Otherwise the cipher is vulnerable to *Known Plaintext Attacks*, since each pixel in the encrypted image corresponds to exactly one pixel in the original. The diffusion step is often realized as a line-wise process, e.g.

$$v(i,j)^* = v(i,j) + G(v(i,j-1)^*)\ mod\ L$$

where $v(i,j)$ is the not-yet modified pixel at position $(i,j)$, $v(i,j)^*$ is the modified pixel at that position, and $G$ is a random look-up table.

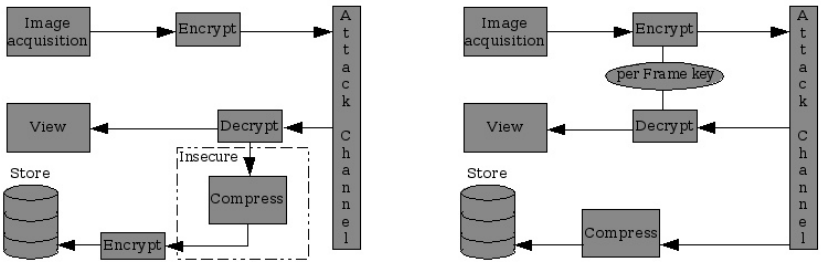
Concerning robustness against transmission or storage errors, it is of course better to avoid diffusion steps. If local information is spread during encryption, i.e. in diffusion steps, a single pixel error in the encrypted image causes several pixel errors in the original image. For this reason we investigate both settings, with and without diffusion.

## 3 Application Scenarios: Compression and Encryption

As already outlined in the introduction, classically encrypted images normally can not be compressed very well (actually these data should not be compressible at all), because of the typical properties encryption algorithms have. In particular it is not possible to employ lossy compression schemes since in this case potentially each byte of the encrypted image is changed (and most bytes in fact are), which leads to the fact that the decrypted image is entirely destroyed resulting in a noise-type pattern. Therefore, in all applications involving compression and encryption, compression is performed prior to encryption.

On the other hand, application scenarios exist where a compression of encrypted material is desirable. In such a scenario classical block or stream ciphers cannot be employed. For example, dealing with video surveillance systems, often concerns about

protecting the privacy of the recorded persons arise. People are afraid what happens with recorded data allowing to track a persons daily itineraries. A compromise to minimize impact on personal privacy would be to continuously record and store the data but only view it, if some criminal offence has taken place.




(a) Conventional Solution (b) Proposed Solution

**Fig. 1.** Privacy Solutions for Surveillance-Systems

To assure, that data can not be reviewed unauthorized, it is transmitted and stored in encrypted form and only few people have the authorization (i.e. the key material) to decrypt it.

The problem, as depicted in Figure 1.a , is the amount of memory needed to store the encrypted frames (due to hardware restrictions of the involved cameras, the data is transmitted in uncompressed form in many cases). For this reason, frames should be stored in a compressed form only. When using classical ciphers the only way to do this would be the decryption, compression and re-encryption of frames. This would allow the administrator of the storage device to view and extract the video signal which obviously threatens privacy. There are two practical solutions to this problem:

1. Before the image is encrypted and transmitted, it is compressed. Beside the above-mentioned computational demands for the camera system, this has further disadvantages, as transmission errors in compressed images have usually an even bigger impact. This is prohibitive in environments where the radio signal is easily distorted.
2. The encrypted frames are compressed directly. In this manner, the key material does not have to be revealed when storing the visual data thereby maintaining the privacy of the recorded persons. Figure 1.b shows such a system. Clearly, in this scenario classical encryption cannot be applied. In the following we will investigate whether CM can be applied and which results in terms of quality and compression are to be expected.

A second example where compression of encrypted visual data is desirable is data transmission over heterogenous networks, for example a transition from wired to wireless networks with corresponding decreasing bandwidth. Consider the transmission of

uncompressed encrypted visual data in such an environment – when changing from the wired network part to the wireless one, the data rate of the visual material has to be reduced to cope with the lower bandwidth available. Employing a classical encryption scheme, the data has to be decrypted, compressed, and re-encrypted, similar to the surveillance scenario described before. In the network scenario these operations put significant computation load onto the network node in charge for the rate adaptation **and** the key material needs to be provided to that network node, which is demanding in terms of key management. A solution where the encrypted material may be compressed directly is much more efficient of course.

# 4 Compressing CM Encrypted Images

## 4.1 Experimental Setup

We present results of four different flavours of the chaotic *CatMap* algorithm (the results concerning the *Bakermap* are very similar, therefore we only provide results for two variants) (see Table 1). The diffusion step has been excluded from all chaotic maps, except *CatDiff*. All algorithms are applied to a $256 \times 256$ version of the *Lena* test image with 256 gray levels using two sets of representative encryption keys.

| Name | Description |
|---|---|
| 2DCatMap5 | Catmap with five iterations. |
| 2DCatMap10 | Catmap with ten iterations. |
| 2DCatDiff5 | Catmap with diffusion step and five iterations. |
| 3DCatMap5 | Catmap with 3D extension and five iterations. |
| 2DBMap5 | Bakermap with five iterations. |
| 2DBMap17 | Bakermap with seventeen iterations. |

**Table 1.** Tested image encryption algorithms

After encryption, JPEG compression is applied to the encrypted image data. To assess the behaviour of the described processing pipeline, the image is finally decompressed, decrypted and the result is compared to the original image and the achieved compression ratio is recorded. Note that it is difficult to find reliable tools to measure quality of distorted images, especially in a low-quality scenario. Several metrics exist, such as the Signal to Noise Ratio (SNR), Peak SNR (PSNR) or Mean Square Error (MSE), which are frequently used in quantifying distortions (see [4, 1]). However, reliable assessment of low quality images should be made by human observers in a subjective rating as this can not be accomplished in a sensile way using the metrics above. It is clear that these measurements are time consuming, as they can not be automated. In order to complement the visual examples, we also report the reference PSNR value.

### 4.2 Experimental Results

Figs. 2 – 5 show images where the encrypted data got lossy (JPEG) compressed, decompressed and finally decrypted again. In these figures, we provide the quality factor q of the JPEG compression, the data size of the compressed image in percent % of the original image size, and the PSNR of the decompressed and decrypted image given in dB.

In general, we observe quite unusual behavior of the CM encryption technique. The interesting fact is that despite the lossy compression a CM encrypted image can be decrypted quite well (depending on the compression rate of course). As already mentioned, this is never the case if classical encryption is applied.

Fig. 2 compares the application of the standard 2D Catmap without and with additional extensions to increase security (i.e. 3D or diffusion extensions are employed additionally). At a fixed compression rate (slightly lower than 3) we obtain a somewhat noisy but clearly recognizable image in case of no further extensions are used (Fig. 2.a). Applying the 3D extension to the standard Catmap (Fig. 2.b), we observe significant degradation of the decrypted image as compared to the standard Catmap with identical number of iterations. However, the image content is still recognizable which is no longer true in case the diffusion extension is used – see Fig.2.c. It is worthwhile noticing that we obtain the same result – noise – no matter which compression rate or image quality is used in case the diffusion step is performed. Actually this result is identical to a result if AES had been used instead of *Catdiff*.

(a) q=55: 36%, 23.4dB (b) q=45: 37%, 15.9dB (c) q=45: 37%, 9.2dB

**Fig. 2.** Catmap with 5 iterations (without extensions and using 3D and diffusion extensions, respectively), keyset 1.

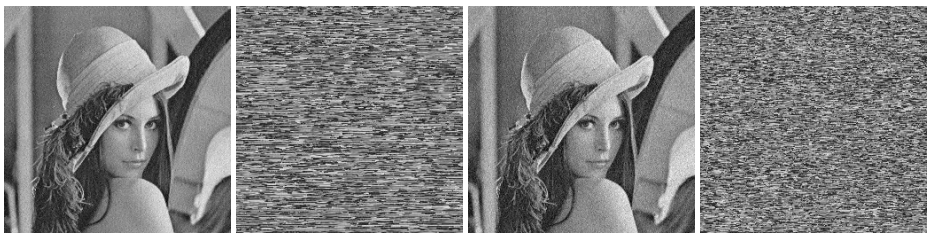
The effect when compression ratio is steadily increased is shown in Fig. 3. Lower data rates in compression increase the amount of noise in the decrypted images, however, still with a compression ratio of 5 (20%) the image is clearly recognizable and the quality would be sufficient for a handheld phone or PDA display for example (Fig. 3.b). Of course, higher compression ratios lead to even more severe degradations which are hardly acceptable for any application (e.g. compression ratio 8 in Fig. 3.c).

Increasing the number of iterations to more than 5 does not affect the results of the Catmap for a sensible keyset (as used for example in Fig. 3). This is not true for the

(a) q=35: 27%, 19.9dB (b) q=25: 20%, 17.9dB (c) q=15: 12%, 16.4dB

**Fig. 3.** Catmap with 5 iterations using different compression ratios, keyset 2.

Bakermap as shown in Fig. 4. When using 5 iterations, the compression result is significantly better as compared to the Catmap case with the same data rate (compare Fig. 4.a to Fig. 2.a). The reason is displayed in Fig. 4.b – using the Bakermap with 5 iterations, we still recognize structures in the encrypted data which means that mixing has not yet fulfilled its aim to a sufficient degree. On the one hand, this is good for compression since errors are not propagated to a large extent, on the other hand this threatens security since the structures visible in the encrypted data can be used to compute key data used in the encryption process.



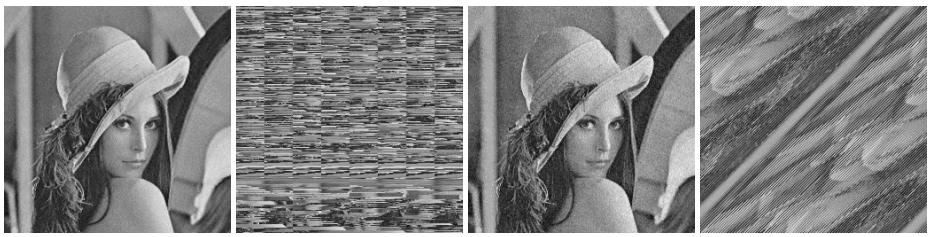
(a) q=70: 37%, 28.0dB (b) q=70: encrypted (c) q=60: 36%, 24.9dB (d) q=60: encrypted

**Fig. 4.** Bakermap with varying number of iterations (5 and 17 iterations), keyset 2.

Increasing the number of iterations (e.g. to 17 as shown in Figs. 4.c and 4.d) significantly reduces the amount of visible structures. As it is expected, the compression results are similar now to the Capmap case using 5 iterations. Using 20 iterations and more, no structures are visible any more and the compression results are identical to the Catmap case.

In Fig. 5 we give examples of the effects in case pathological key material is used for encryption. When using keyset 1 for encryption with the Bakermap (Figs. 5.a and 5.b), the structures visible in the encrypted material are even clearer and in perfect correspondence also the compression result is superior to that of keyset 2 (Fig. 4). With these setting, an even higher number of iterations is required to achieve reasonable security (which again destroys the advantage with respect to compression). Also for the Catmap, weak keys exist. In Fig. 5.d the encrypted data is shown in case 10 iterations

are performed using keyset 1. In this case, even image content is revealed and the key parameters are reconstructed easily with a ciphertext only attack. Correspondingly, also the compression results are much better as compared to the case when 5 iterations are applied (see Fig. 2.a). These parameters (weak keys) and effects (reduced security) have been described in the literature on CM and have to be avoided for any application of course.



(a) q=75: 36%, 30.9dB (b) q=75: encrypted (c) q=70: 36%, 27.3dB (d) q=70: encrypted

**Fig. 5.** Bakermap and Catmap with pathological keys (5 and 10 iterations).

In general, we observe a significant tradeoff between security and visual quality of compressed data when comparing the different settings as investigated. Increasing the number of iterations up to a certain level increases security but decreases compression performance (this is especially true for the Bakermap which requires a higher number of iterations in general to achieve reasonable security). However, of course the computational effort increases as well.

We face an even more significant tradeoff when increasing security further – the 3D extensions already strongly decrease image quality whereas diffusion entirely destroys the capability of compressing encrypted visual data. When the security level approaches the security of cryptographically strong ciphers like AES, also CMs do not offer robustness against lossy compression any longer.

## 5 Conclusion

Chaotic mixing based encryption techniques are shown to tolerate a medium amount of lossy compression which is an exceptional property not found in other ciphers. Applying the Catmap with 5 iterations or the Bakermap with 20 iterations provides reasonable security and decrypted images show acceptable image quality even after significant JPEG compression. However, if techniques enhancing CMs security like the 3D extension technique or diffusion are used, the robustness against compression is reduced or entirely lost.

As long as a lower security level is desired or acceptable (i.e. 3D extension or diffusion is omitted), CM may be employed in application scenarios where lossy compression is applied to the encrypted data. This type of application scenarios cannot be operated with traditional ciphers. If high security is required (and the lower computational demand of CM is not an issue) it is better to stick to classical block ciphers in any environment since CM loses its robustness to compression anyhow.

## References


1. I. Avcibas, B. Sankur, and K. Sayood. Statistical evaluation of image quality measures. *Journal of Electronic Imaging*, 11(2):206–223, April 2002.
2. G. Chen, Y. Mao, and C.K. Chui. A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos, Solitons and Fractals*, 21:749–761, 2004.
3. J. Daemen and V. Rijmen. *The Design of Rijndael: AES — The Advanced Encryption Standard*. Springer Verlag, 2002.
4. A. Eskicioglu. Quality measurement for monochrome compressed images in the past 25 years. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 1907–1910, 2000.
5. J. Fridrich. Symmetric ciphers based om two-dimensional chaotic maps. *Int. J. of Bifurcation and Chaos*, 8(6):1259–1284, 1998.
6. B. Furht and D. Kirovski, editors. *Multimedia Security Handbook*. CRC Press, Boca Raton, Florida, 2005.
7. R. Norcen and A. Uhl. Encryption of wavelet-coded imagery using random permutations. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'04)*, Singapore, October 2004. IEEE Signal Processing Society.
8. J. Scharinger. Fast encryption of image data using chaotic Kolmogorov flows. *Journal of Electorinic Imaging*, 7(2):318–325, 1998.
9. Ali Saman Tosun and Wu chi Feng. On error preserving encryption algorithms for wireless video transmission. In *Proceedings of the ninth ACM Multimedia Conference 2001*, pages 302–307, Ottawa, Canada, October 2001.
10. A. Uhl and A. Pommer. *Image and Video Encryption. From Digital Rights Management to Secured Personal Communication*, volume 15 of *Advances in Information Security*. Springer-Verlag, 2005.
11. Jiangtao Wen, Mike Severa, Wenjun Zeng, Max Luttrell, and Weiyin Jin. A format-compliant configurable encryption framework for access control of video. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(6):545–557, June 2002.
12. W. Zeng, J. Wen, and M. Severa. Fast self-synchronous content scrambling by spatially shuffling codewords of compressed bitstreams. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'02)*, September 2002.
13. Wenjun Zeng and Shawmin Lei. Efficient frequency domain selective scrambling of digital video. *IEEE Transactions on Multimedia*, 5(1):118–129, March 2003.