Hindawi Publishing Corporation
EURASIP Journal on Information Security
Volume 2007, Article ID 48179, 16 pages
doi:10.1155/2007/48179

Research Article

Transmission Error and Compression Robustness of 2D Chaotic Map Image Encryption Schemes

Michael Gschwandtner, Andreas Uhl, and Peter Wild

Department of Computer Sciences, Salzburg University, Jakob-Haringerstr. 2, 5020 Salzburg, Austria

Correspondence should be addressed to Andreas Uhl, uhl@cosy.sbg.ac.at

Received 30 March 2007; Revised 10 July 2007; Accepted 3 September 2007

Recommended by Stefan Katzenbeisser

This paper analyzes the robustness properties of 2D chaotic map image encryption schemes. We investigate the behavior of such block ciphers under different channel error types and find the transmission error robustness to be highly dependent on the type of error occurring and to be very different as compared to the effects when using traditional block ciphers like AES. Additionally, chaotic-mixing-based encryption schemes 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 transmission errors and 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 [9, 20] for extensive overviews). The most prominent reasons not to stick to classical full encryption employing traditional ciphers like AES [6] for such applications are the following:

(i) to reduce the computational effort (which is usually achieved by trading off security as it is the case in partial or soft encryption schemes);

(ii) to maintain bitstream compliance and associated functionalities like scalability (which is usually achieved by expensive parsing operations and marker avoidance strategies);

(iii) to achieve higher robustness against channel or storage errors.

Using invertible two-dimensional chaotic maps (CMs) on a square to create symmetric block encryption schemes for visual data has been proposed [4, 8] mainly to serve the first purpose, that is, to create encryption schemes with low computational demand. CMs operate in the image domain which means that in some sense bitstream compliance is not an issue, however, they cannot be combined in a straightforward manner with traditional compression techniques.

Compensating errors in transmission and/or storage 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. On the one hand, effective error concealment techniques already exist for most current file formats, but when image data needs to be encrypted, these techniques only partly apply since they usually depend on the data format which is not accessible in encrypted form. On the other hand, error correction codes may be applied at the network protocol level or directly to the data but these techniques exhibit several drawbacks which may be not acceptable in certain application scenarios.

(i) Processing overhead: applying error correction codes before transmission causes additional computational demand which is not desired if the acquiring and sending device has limited processing capability (like any mobile device).

(ii) Data rate increase: error correction codes add redundancy to data; although this is done in a fairly efficient manner, data rate increase is inevitable. In case of low-bandwidth network links (like any wireless network) this may not be desired.

One famous example for an application scenario 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. If further error correction can be avoided, the remaining processing capacity (if any) can be used for image enhancement and higher network capacity allows better quality images to be transmitted. In this work we investigate a scenario where neither error concealment nor error correction techniques are applied, the encrypted visual data is transmitted as it is due to the reasons outlined above.

Due to intrinsic properties (e.g., the avalanche effect) of cryptographically strong block 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. For example, it is well known that a single bit failure of AES-encrypted ciphertext destroys at least one whole block plus further damage caused by the encryption mode architecture. 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 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 [21, 22] or wavelet-based [14, 23] 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 occur [19].

Bearing in mind that CM crypto systems mainly rely 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. While the expected conclusion that the higher security level of cryptographically strong ciphers implies higher sensitivity to errors compared to CM crypto systems is nothing new, we investigate the impact of different error models on image quality to obtain a quantifiable tradeoff between security and transmission error robustness. The rise of wireless local area networks and its diversity of errors enforce the development of new transmission methods to achieve good quality of transmitted image data at a certain protection level.

Accepting the drawback of a possibly weaker protection mechanism, it may be possible to achieve better quality results in the decrypted image after transmission over noisy channels as compared to classical ciphers. In this work we compare the impact of different types of distortions of transmission links (i.e., channel errors) on the transmission of images using block cipher encryption with CM encryption (see Figure 1, part A).

Additionally (see Figure 1, part B), we focus on an issue different to those discussed so far at first sight, however, this topic is related to the CMs' robustness against a specific type of errors (value errors): we investigate the lossy compression of encrypted visual material [10]. Clearly, data encrypted with classical ciphers cannot be compressed well: due to the statistical properties of encrypted data no data reduction may be expected using lossless compression schemes, lossy compression schemes cannot be employed since the reconstructed material cannot 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 lossy compression after encryption (i.e., in the encrypted domain). 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 is given in Section 2. Experimental setup and used image quality assessment methods are presented in Section 3. Section 4 discusses the robustness properties of CM block ciphers with respect to different types of network errors and compares the results to the respective behavior of a classical block cipher (AES) in these environments. Section 5 discusses possible application scenarios requiring compression to be performed after encryption and provides experimental results evaluating a JPEG compression, a JPEG 2000 compression and finally JPEG 2000 with wavelet packets, all with varying quality applied to CM encrypted data. Section 6 concludes the paper.

2. CHAOTIC MAP ENCRYPTION SCHEMES

Using CMs as a (mainly) permutation-based symmetric block cipher for visual data was introduced by Scharinger [17] and Fridrich [8]. CM encryption relies on the use of discrete versions of chaotic maps. The good diffusion properties of chaotic maps, such as the *baker map* or the *cat map*, soon attracted cryptographers. Turning a chaotic map into a symmetric block cipher requires three steps, as [8] 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.

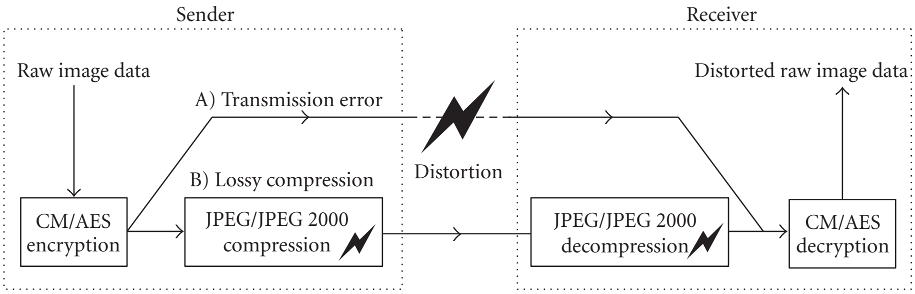


Figure 1: Experimental setup examining (A) transmission error resistance and (B) lossy compression robustness of CM and AES encryption schemes.

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

$$B\colon [0,1]^2 \longrightarrow [0,1]^2,$$
$$B(x,y) = \begin{cases} \left(2x, \dfrac{y}{2}\right) & \text{if } 0 \le x < \dfrac{1}{2}, \\ \left(2x-1, \dfrac{y+1}{2}\right) & \text{if } \dfrac{1}{2} \le x \le 1. \end{cases} \tag{1}$$

This corresponds geometrically to a division of the unit square into two rectangles $[0,1/2[\times[0,1]$ and $[1/2,1]\times[0,1]$ 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, introduced parameters. Another choice of a chaotic map is the *Arnold Cat map*:

$$C\colon [0,1]^2 \longrightarrow [0,1]^2,$$
$$C(x,y) = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \bmod 1, \tag{2}$$

where $x \bmod 1$ 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:

$$C_{\text{gen}}(x,y) = A \begin{pmatrix} x \\ y \end{pmatrix} \bmod 1, \quad A = \begin{pmatrix} 1 & a \\ b & ab+1 \end{pmatrix}. \tag{3}$$

Now each generalized chaotic map needs to be modified to turn into a bijective map on a square lattice of pixels. Let $\mathcal{N} := \{0,\dots,N-1\}$, 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

$$\lim_{N\to\infty} \max_{0\le i,j<N} \left| f(i/N, j/N) - F(i,j) \right| = 0. \tag{4}$$

Discretizing a chaotic *Cat map* is fairy simple and introduced in [4]. Instead of using the fractional part of a real number, the integer modulo arithmetic is adopted:

$$C_{\text{disc}} : \mathcal{N}^2 \longrightarrow \mathcal{N}^2,$$
$$C_{\text{disc}}(x,y) = A \begin{pmatrix} x \\ y \end{pmatrix} \bmod N, \quad A = \begin{pmatrix} 1 & a \\ b & ab+1 \end{pmatrix}. \tag{5}$$

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 grey-level shift (assuming L grey levels $\mathcal{L} := \{0,\dots,L-1\}$) at each level of iteration:

$$F_{3D} : \mathcal{N}^2 \times \mathcal{L} \longrightarrow \mathcal{N}^2 \times \mathcal{L}$$
$$F_{3D}(i,j,g_{ij}) = \begin{pmatrix} i' \\ j' \\ h(i,j,g_{ij}) \end{pmatrix}, \qquad \begin{pmatrix} i' \\ j' \end{pmatrix} = F(i,j). \tag{6}$$

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

Since 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 extremely vulnerable to *known plaintext attacks*, since each pixel in the encrypted image corresponds exactly to one pixel in the original. The diffusion step is often realized as a linewise process, for example,

$$v(i,j)^* = v(i,j) + G\big(v(i,j-1)^*\big) \bmod L, \tag{7}$$

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 an arbitrarily chosen random lookup table.

Concerning robustness against transmission errors, CMs of course are expected to be more robust when diffusion steps are avoided (compare results). If local information is spread

Table 1: Cardinality of key spaces $K(N)$.

	$N = 20$	$N = 25$	$N = 128$	$N = 512$
Baker map keyset1	83343	571	10^{31}	10^{126}
Baker map keyset2	524288	16777216	10^{38}	10^{153}
Cat map	400	625	16384	262144
AES128	10^{38}	10^{38}	10^{38}	10^{38}
AES256	10^{77}	10^{77}	10^{77}	10^{77}

during encryption, that is, 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.

It should be clear that chaotic maps have different properties when compared to conventional block ciphers. Typically, conventional block encryption schemes like AES work on block sizes of 128, 256, or 512 bit. key space contains 2^n elements, where n is the number of key bits, which is usually often 1 : 1 to block size.

As the main property of CM is permutation, it operates on larger units, that are full (square) images. Their smallest element to be permuted is a pixel. To encrypt an $N \times N$ image, $N^2!$ permutations exist. However, the key space available to parameterize the chaotic map is often orders of magnitude smaller. Another drawback is dependency on image size. There are configurations where a small change in image size causes key space to shrink dramatically (see keyset1 and keyset2 in Table 1). In Table 1, cardinalities of key spaces $K(N)$ for *Baker map*, *Cat map*, and *AES* are compared choosing a representative $N \times N$ grey-scale image. While the number of iterations and parameters for the diffusion step is usually part of the key for chaotic encryption algorithms they have been neglected for this comparison. It is evident that key space, especially for smaller image sizes, is insufficient. In this case or for problematic image sizes, padding should be used to prevent a guessing of all possible key combinations. At this point a main drawback of the *Cat map* becomes evident: its parameters offer little combinations compared to other chaotic maps.

Chaotic maps are generally sensitive to initial conditions and parameters. But some discrete versions bear unexpected behavior when using similar keys. While classical encryption algorithms are sensitive to keys, chaotic maps such as the *Baker map* exhibit a set of keys $S(K)$ for each key K, such that the image encrypted with K and decrypted using $k \in S(K), k \neq K$ is *close to its original*. We get similar results when using keys that are derived from the original by replacing a large parameter by two smaller ones or merging two small parameters into a larger one. This has been observed by [8]. Accepting the drawback of a further limitation of key space (the intruder may be content to find a key that produces acceptable approximations of original images and continues with refinement), this may also be seen as a feature of the encryption system. Transmission errors destroying single bits of the key do not necessarily lead to fully destroyed decryption. Heuristics could produce a similar key, that allows decryption at a low but probably sufficient quality.

Table 2: Tested image encryption algorithms for part A.

Name	Description
2DCatMap	Cat map
2DBMap	Baker map
3DCatMap	Cat map with 3D extension
2DCatDiff	Cat map with diffusion step
AES128ECB	AES using ECB on 128 bit blocks
AES128CBC	Same as AES128ECB, using CBC

Table 3: Tested image encryption algorithms for part B.

Name	Description
2DCatMap5/7/10	Cat map with 5/7/10 iterations
2DCatDiff5	Cat map with diffusion step and five iterations
3DCatMap5	Cat map with 3D extension and five iterations
2DBMap5/17	Baker map with 5/17 iterations

Table 4: Employed keys/parameters for experiments.

Name	Value
BakerMapKey1	192,32,32
BakerMapKey2	32,64,32,16,32,32,16,8,8,8,8
AES_IV	10111213141516171819202122232425
AESKey	000102030405060708090A0B0C0D0E0F
CatMapKey	2,3,1,1

3. EXPERIMENTAL SETUP

We analyze both transmission error resistence (part A) and compression robustness (part B) of three different flavors of the chaotic *Cat map* algorithm, a simple 2D version of the *Baker map* and AES using different block encryption modes (see Tables 2, 3). All chaotic ciphers use 10 iteration rounds, if not specified differently.

Since the number of iterations used in CM algorithms largely affects the distribution of distortions caused by lossy compression, we examine the impact of this parameter on image quality. The diffusion step has been excluded from all chaotic maps, except *CatDiff*. All algorithms are applied to a set of 10 natural and 6 synthetic 256×256 images with 256 grey levels referenced in Figure 2 (only 13 of 16 pictures are shown due to copyright restrictions) using two sets of representative encryption keys (keyset2 represents a strong key whereas keyset1 exhibits certain weaknesses with respect to security). Key parameters for the visual quality experiment are given in Table 4.

3.1. Setup

A flow chart to illustrate the test procedure for both part A and part B is depicted in Figure 1. Recapitulating, the test procedure is as follows.

(i) *Part A: transmission error robustness.* After encryption, a specific type of error as introduced in Section 4.1 is applied to the encrypted image data. Finally, the image is decrypted and the result is compared to the original.

(ii) *Part B: compression robustness*. After encryption, three different compression algorithms (JPEG, JPEG 2000, and JPEG 2000 with wavelet packets) are applied to the encrypted image data. To assess the behavior 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 (using the encrypted image as reference) is recorded.

3.2. Image quality assessment

It is difficult to find reliable tools to measure quality of distorted images. This is especially true 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 [3, 7]). Mao and Wu [11] propose a measure specifically tailored to encrypted imagery that separates evaluation of luminance and edge information into a *luminance similarity score* (LSS) and an *edge similarity score* (ESS), reflecting properties of the human visual system. According to the authors, this measure is well suited for assessing distortion of low-quality images. LSS behaves in a way very similar to PSNR. ESS is the more interesting part in the context of the survey presented here, as it reflects the extent for structural distortion. ESS is computed by block-based gradient comparison and ranges, with increasing similarity, between 0 and 1. However, reliable assessment of low-quality images should be made by human observers in a subjective rating as this cannot be accomplished in a sensible way using the metrics above. Subjective visual assessment of transmissions yields a mean opinion score (MOS) [1] evaluating gradings of human observers according to strictly specified testing conditions. Such conditions are specified in, for example, [2] for the subjective assessment of the quality of television pictures. These methods can be extended to the assessment of images in general and are frequently adopted, such as in [5]. Recommendation ITU-R-BT500-11 [2] introduces both double stimulus (with reference picture) and single stimulus (without reference picture) assessment methods with a strictly defined testing environment, that is, quality and impairment scales, lighting conditions and also restrictions regarding selection of observers. We have decided to adopt only a subset of features, in particular,

(i) we adopt to a simultaneous double stimulus method (SDSCE) with reference and test pictures being shown at the same time;

(ii) we employ the specified five-graded quality scale (see Table 5).

Additionally, we conform the specified condition, that at least fifteen subjects, nonexperts, should be employed.

Since [2] specifies subjective video quality assessment methods, it should be noticed that observers evaluate the average quality of the frames displayed. In our case still images are evaluated. Therefore, we let the observer vote for the average quality of three different test pictures (encrypted using the same algorithm, but different keys) with respective originals being shown at the same time, that is, in one assessment step, using the quality levels introduced in Table 5.

Table 5: ITU-R-BT500-11 subjective quality rating scales.

Quality	Description
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

In the following section we give a short description of the observed results with respect to distortions. In order to complement the subjective ratings, we also report the reference PSNR value. While it is clear, that in some cases further error correction by means of denoising might be useful and thus better results can be achieved, we do not concentrate on postprocessing techniques at this point.

4. TRANSMISSION ERROR ROBUSTNESS

In this section, our goal is to provide a comparison of two completely different block ciphers with respect to their behavior in the transmission of encrypted visual data over noisy channels. Therefore, this section introduces a set of distortion models we believe are practical and illustrative for applications.

4.1. Classification of used error models

Much work has already been done to classify transmission errors occurring at wireless data transmission and a variety of sophisticated network simulators already exist. To focus on a generally applicable comparison of the two encryption mechanisms CM and AES, we arrange simulations that can be described by the following model: a sender S transmits a sequence $s_0, s_1, s_2, \dots, s_n$ of $n + 1$ bytes over a lossy channel. Receiver R receives a sequence $r_0, r_1, r_2, \dots, r_m$ of bytes, that is possibly different to $s_0, s_1, s_2, \dots, s_n$. There are situations where $n \neq m$. We identify two categories of observable errors.

(i) *Value errors,* where $n = m$ and $r_0, r_1, \dots, r_n$ are derived from the original sequence alternating selected bytes. More formally, there exists a set $A \subset \{0, \dots, n\}$ and error function f such that for all $i \in \{0, \dots, n\}$

$$r_i = \begin{cases} f(s_i) & \text{if } i \in A; \\ s_i & \text{else.} \end{cases} \qquad (8)$$

Note that f may depend on additional random variables.

(ii) *Buffer errors,* where bytes are changed, inserted, removed, and possibly resorted. There exists a set $A \subset \{0, \dots, m\}$ and error function f such that a received stream may be described as

$$\forall j \leq m \quad \exists i \leq n : r_j = \begin{cases} f(s_i) & \text{if } j \in A; \\ s_i & \text{else.} \end{cases} \tag{9}$$

Various combinations of such errors can occur. However, to extend the observations to existing network behavior, it is inevitable to model characteristics of transmission packets and network protocols. We believe at this point that the introduced classes are sufficient to show the main differences between the two algorithms CM and AES. Another reason why further modeling is not adequate at this point is the following: if we get close to an error saturation, the category of error should be negligible, as many small buffer errors behave similar to many value errors.

4.2. Value errors

Proceeding with the notion of an incoming distorted sequence $r_0, r_1, \dots, r_n$, one can identify several different subsets A and functions f to model a value error.

(i) Static error

In this model every single byte will be changed, that is, $A = \{0, \dots, n\}$. The change for all bytes is quite simple: each byte gets logically *OR*ed with a static byte $b \in \{0, \dots, 255\}$. For our experiments we have assigned to b the value 85. Thus, we have for all $i \in \{0, \dots, n\} : r_i = s_i \, OR \, b$. This can be used to simulate defect bus lines, which are permanently at a high error level.

(ii) Random error and random Gaussian error

The most general error assumption may be the selection of A using distribution functions. Having to transmit n bytes, for each byte s_i a specifically distributed random variable decides whether $i \in A$ or $i \notin A$, that is, whether it is transmitted correctly or not. The classes *random error* and *random Gaussian error* use the uniform distribution and normal distribution for selection, respectively. Let $X \sim U(0,1)$ be a (standard, continuous) uniformly distributed random variable and let $E \sim UD(0,255)$ denote a discrete uniformly distributed random variable, then a *random error* is defined for all $i \in \{0, \dots, n\}$ by

$$r_i = \begin{cases} E_i & \text{if } X_i < p; \\ s_i & \text{else.} \end{cases} \tag{10}$$

The choice of $p \in [0,1]$ influences error rate and was selected to be $p = 0.01$ for our experiments. For *random Gaussian error* the random variable X is chosen to be normally distributed, that is, $X \sim \mathcal{N}(\mu, \sigma^2)$ and we define $\forall i \in \{0, \dots, n\}$:

$$r_i = \begin{cases} E_i & \text{if } |X_i| > p; \\ s_i & \text{else.} \end{cases} \tag{11}$$

The assignments for our experiments are as follows: $\mu = 0$, $\sigma = 1, p = 2.5$. This error model is often used to simulate distortions in RF transmissions. Moderate rain causes pixels in satellite TV transmissions to be distorted using specific distribution functions.

Table 6: State transitions in Two-State Model.

Probability	State transition
p	Stay in normal
$(1-p)$	Change to error
q	Stay in error
$(1-q)$	Change to normal

(iii) Random Markov chain

Similarly to the error model introduced before this model assumes that a byte is overwritten by a random value if it is selected to contain an error. But the decision if a byte has an error is made conforming to a 2-state Markov chain.

Given two states (1 = error and 0 = normal), there are transition properties to stay or change the current state. Transitions are handled as shown in Table 6. Especially for modeling errors in wireless transmission, this model has frequently been adopted (see, e.g., [13]). Let $X \sim U(0,1), Y \sim U(0,1)$ be uniformly distributed random variables and $p, q \in [0,1]$ denote state-transition probabilities as introduced before, then we formulate a state function returning the current state at time t_i with starting state $I_0 \in \{0,1\}$ as follows:

$$I(t_0) := I_0$$

$$I(t_{i+1}) := \begin{cases} 1 & \text{if } I(t_i) = 0 \wedge X_i > p \\ & \text{or } I(t_i) = 1 \wedge Y_i \leq q; \\ 0 & \text{else.} \end{cases} \tag{12}$$

Thus, if we use again $E \sim UD(0,255)$, we have $\forall i \in \{0, \dots, n\}$:

$$r_i = \begin{cases} E_i & \text{if } I(t_i) = 1; \\ s_i & \text{else.} \end{cases} \tag{13}$$

For the implemented error model we make the following assignments: $p = 0.98, q = 0.03, I_0 = 0$.

4.3. Buffer errors

In contrast to value-errors representatives of the following type of errors correspond to distortions in packet-switched data networks. Being able to restore single damaged bytes, for example, by the employment of error-correcting codes, the major problem here is a possible perturbation, replaying and loss of packets consisting of one or multiple bytes.

These errors are often simulated with special network simulators like ns2 (see at http://www.isi.edu/nsnam/ns). Reference [12] shows that these errors happen in bursts

```
def random_buffer()
{
    for (i = 0; i < Image.Length; i++)
    {
        if (randomDouble(0.0,1.0) < p)
        {
            switch(mode)
            {
                case InsertBytes
                {
                    Image.InsertByte(i, randomInt(255)) i++
                }
                case RemoveBytes
                {
                    Image.RemoveByte(i)
                }
            }
        }
    }
}
```

Algorithm 1: Pseudocode representation of the random buffer error algorithm with an error probability of p.

(subsequently). We do not consider the error in bursts as this makes an assumption on the transmission channel, and in the encryption context "real random" errors are the worst case scenario. As the error may occur inside the destroyed buffer and on the "error edges" (for blockciphers in chaining mode only), we can see that the impact with bursts is less severe as there are fewer "error edges."

(i) Random buffer error

The most simple case is when packet size is a single byte. To model a behavior where each sent byte may be lost, replicated, or finally perturbated in the final sequence the corresponding actions are modeled as random variables. In our current implementation, only one type of error (add or remove of a selected byte) per transmission is possible. The described simulation models errors appearing on serial transmission links, where the sender and the receiver are slightly out of synchronization. Algorithm 1 is a simplified pseudocode representation of the implemented algorithm.

(ii) Random packet error

Compared to the random buffer error, the random packet error represents an error which is more likely in current systems. As practically any modern computer networks (wired and wireless) are packet switched, packet loss errors, duplicated packets, or out-of-order packets of any common size can occur during transmissions. Simulation of packet loss (the most common error) is done by cutting out parts (consisting of an arbitrary number of bytes) of the encrypted image or overwriting them with a specified byte. The implemented algorithm is sketched in Algorithm 2.

```
def random_packet()
{
    for (i = 0; i < Image.Length/64; i++)
    {
        if (randomDouble(0.0,1.0) < p)
        {
            switch(mode)
            {
                case LooseBytes {
                    Image.RemoveRange(i*64, 64)
                }
                case ConceilBytes {
                    Image.SetRange(i*64, 64, 0)
                }
            }
        }
    }
}
```

Algorithm 2: Pseudocode representation of the random packet error algorithm with an error probability of p.

4.4. Experiments

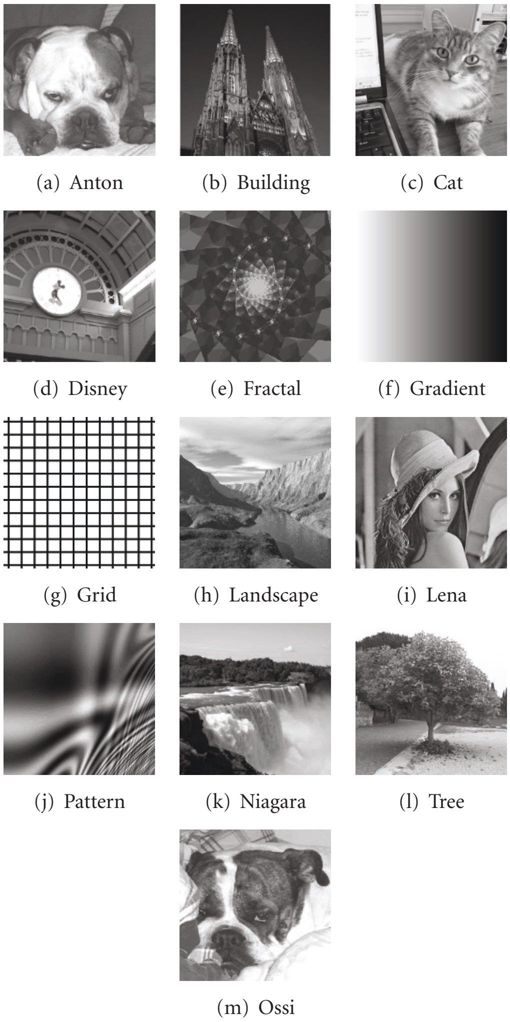
We show the mean opinion scores of 107 (90 male, 17 female) human observers for the test pictures Lena, Landscape, and Ossi together with the reference mean PSNR values in Table 7. The maximum absolute MOS distance between male and female observers is 0.26 and 0.19 for image-quality experts versus nonexperts. Especially for random packet errors, experts tend to grade AES and CM diffusion results better, while finding CM random Gaussian errors to be more bothersome.

As can be seen in Table 7, mean PSNR is a good indicator for MOS. Since subjective image assessments are time consuming (they cannot be automated), we analyze the complete test picture set in Figure 2 with respect to this quality metric.

It is clear that comparison results largely depend on the parameters of the error model, such as the error byte b for static error or the error rate r. Figure 3 depicts exactly this relationship comparing CM and AES error resilience performance against different error rates (the plots display average PSNR values of the images displayed in Figure 2). Inspecting the mean PSNR curves, we can see that for all different types of errors, 2DCatMap and 2DBMap do not differ much, as well as do not differ AES encryption modes. It also illustrates CMs superiority in transmission error robustness for random errors. Interestingly, also 3DCatMap performs equivalently to the pure 2D case for value errors (compare also Table 6). The results for random buffer errors also indicate superiority of CMs, but the low overall PSNR range obtained does not really lead to visually better results. For random buffer errors, 3DCatMap gives equal results to the 2DCatDiff variant contrasting to the value error cases. For random packet errors, AES exhibits 1.5–2 dB higher mean PSNR values than standard 2D CM crypto systems. It is

Table 7: Comparing AES and CM with respect to objective and subjective image quality using Landscape, Lena, and Ossi test images.

Algorithm	Static error		Random error		R. Gaussian error		R. buffer error		R. Packet error	
	Mean PSNR	MOS	Mean PSNR	MOS	Mean PSNR	MOS	Mean PSNR	MOS	Mean PSNR	MOS
Original	13.87	3.10	28.36	4.61	27.53	4.57	10.54	1.39	11.25	2.12
2DCatMap	13.87	3.06	28.34	4.50	27.52	4.56	9.56	1.02	9.73	1.43
2DBMap	13.87	3.07	28.47	4.57	27.37	4.58	9.60	1.00	10.13	1.13
3DCatMap	14.74	2.78	28.43	4.53	27.59	4.56	8.47	1.00	8.92	1.17
2DCatDiff	8.47	1.00	14.24	3.03	13.30	2.75	8.47	1.00	8.46	1.00
AES128ECB	8.52	1.00	16.56	3.21	15.77	3.00	8.58	1.02	10.93	2.40
AES128CBC	8.46	1.00	16.47	3.12	15.63	2.92	8.55	1.04	11.48	2.23



(a) Anton (b) Building (c) Cat

(d) Disney (e) Fractal (f) Gradient

(g) Grid (h) Landscape (i) Lena

(j) Pattern (k) Niagara (l) Tree

(m) Ossi

Figure 2: Test pictures for transmission errors and compression robustness.

also interesting to see that for AES even at very low error rates starting at 4-5 percent random errors cause at least as much damage to image quality than random packet errors. However, when error rates become very high, there is not much difference between any of the introduced error models.

4.4.1. Static error

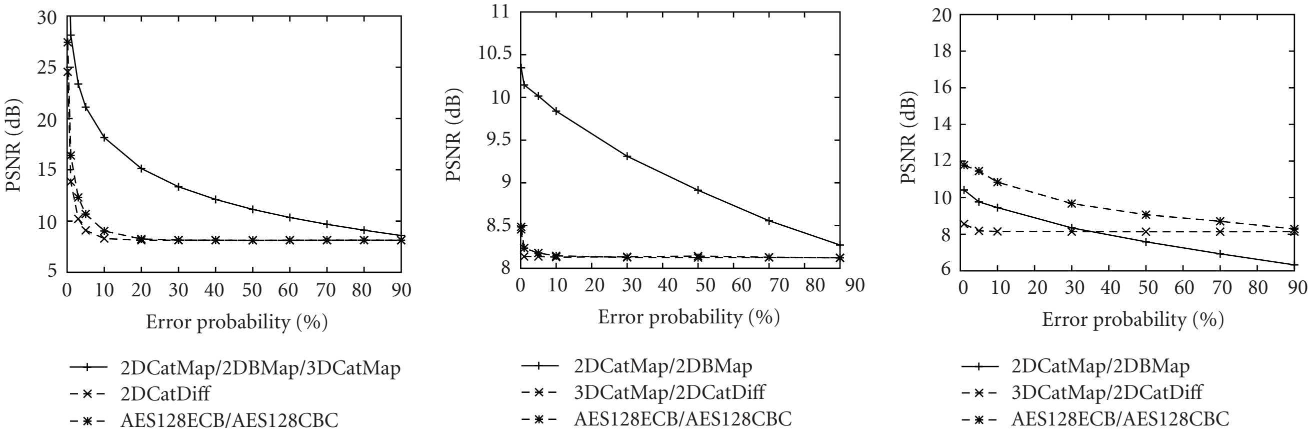
For simulating the *static error* case, all bytes are *OR*ed with $b = 85$ (Figures 4(a) and 4(b)). It is evident that results for AES are unsatisfactory. As every byte of the encrypted image is changed, the decrypted image is entirely destroyed resulting in a noise-type pattern. The distortion of the CM-encrypted image is exactly as significant as if the image had not been encrypted. The cause for the observable preservation of the original image is the fact that simple 2D CM is solely a permutation. In contrast, 3D CM consists of an additional color shift depending on pixel positions. Also the 3D CM handles this type of distortion well whereas the diffusion step added destroys the result. The number of alternately dependent bits can be controlled with the number r of iteration rounds. If just a few rounds are used, an error does not spread over large parts of the image. Using many rounds, a single flipping bit causes the scrambling of the entire image.

4.4.2. Random error and random Gaussian error

As we have expected, *random error* and *random Gaussian error* show very similar results. When considering properties of block ciphers, we can see that the alternation of a single byte destroys the encrypted block in ECB mode (including a byte of the following block in CBC/CFB mode). This causes every error to destroy b_s bytes (b_s+1 in CBC/CFB) in the decrypted image, where b_s is the used block size (see Figure 5(b)). Further errors occurring in already destroyed blocks have no effect. This leads to stronger impact on block ciphers when parameters for error probability are small. When the error rate is high, this drawback is reduced as more and more errors lie within the same damaged block. The CMs cope very well with this distortion type since errors are not expanded and the result is again identical as if the image had not been encrypted (see Figure 5(a)). Again, applying diffusion is the exception where degradation may become even more severe as compared to the AES cases.

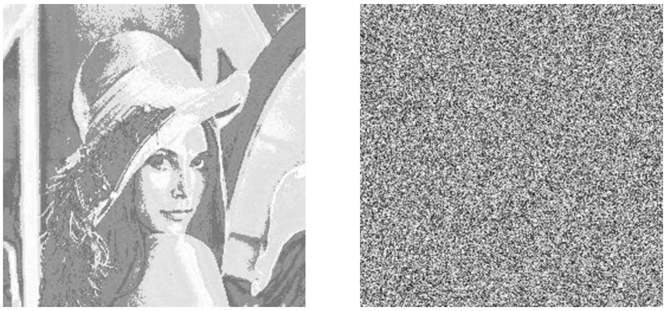
4.4.3. Random buffer error

Using random buffer error in the AES case, we observe the following phenomenon. Each time the encrypted blocks get *synchronized* with their respective original counterparts, the following blocks are decrypted correctly until the next error



(a) Random error (b) R. buffer error (c) R. packet error

Figure 3: Comparing AES and CM transmission error robustness against error rate.



(a) 2DCatMap (b) AES128ECB

Figure 4: Effect of static byte errors on Lena image.

occurs (see Figure 6(b)). If we use CBC or CFB, the block directly after the *synchronization point SP* is additionally destroyed. Of course, this analysis is only correct in case identical keys are employed for each block.

As we model only insertion or deletion of bytes, we reach *SP*s every blocksize (*bs*) errors. Each time an error occurs we step either into an *error phase*, where every pixel is decrypted incorrectly, or a *normal phase* (where pixels get decrypted correctly). Let us assume that for the number of errors *e*, the blocksize *bs*, and the image size *is* the relation

$$bs \ll e \ll \frac{is}{bs} \tag{14}$$

holds. Then we get approximately $(bs - 1)$ times more *error phases* than *normal phases*. If the error rate exceeds the upper bound, the entire image is destroyed.

The reason why CM-encrypted images are completely destroyed with *random buffer error* (Figure 6(a)) is the inherent sensitivity with respect to initial conditions. In most cases, neighboring pixels in the encrypted image are far apart in the decrypted image. Every time an error occurs, the pixels are shifted by one and the decrypted pixels are completely out of place. In CM we cannot identify *SP*s.

4.4.4. Random packet error

For *random packet error* we distinguish two different versions:

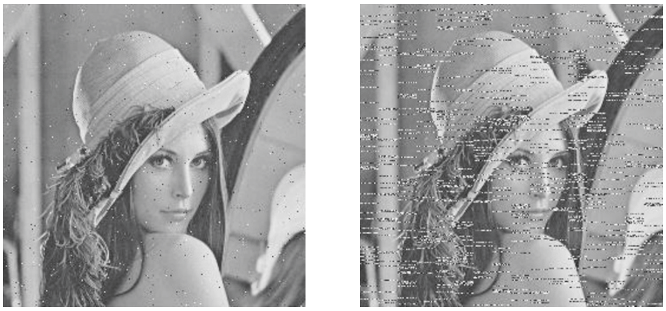
(1) the packet loss gets detected and the space is padded with bytes;
(2) no detection of the packet loss is done.

As to the first version we observe, when using *AES*, that the lost part plus *bs* (respective 2 × *bs*) bytes are destroyed. With *2DCatMap* and *3DCatMap* only the amount of lost pixels is destroyed. This case corresponds to a value error occurring in bursts or a local static error, the results obtained show the respective properties.

In the second case (which is covered in Table 7) CM has the same synchronization problems as in *random buffer error* which causes the image to be entirely degraded (Figure 7(a)). The impact on block ciphers depends on the size of the packet *ps*. If the equation

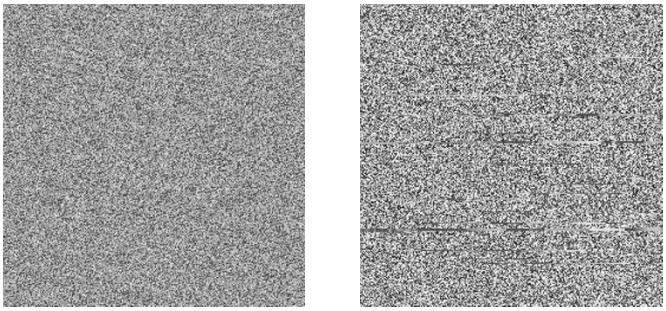
$$ps \bmod bs = 0 \tag{15}$$

holds, the error gets compensated very well (shown in Figure 7(b); this block-type shift can be inverted very easily). Scrambled parts after the cut points come to *bs* (respective 2 × *bs*). If the packet size is different, only the



(a) 2DCatMap (b) AES128ECB

Figure 5: Effect of random byte errors on Lena image.



(a) 2DCatMap (b) AES128CBC

Figure 6: Effect of buffer errors on Lena image.

parts of the image lying between *synchronization points* and the next error are decrypted correctly.

In normal packet switched networks, the packets need identification numbers and therefore lost packets can be detected. That is why the first case of *random packet errors* is most likely to occur.

Overall we have found excellent robustness of CM with respect to value errors which results in significantly better behavior as compared to classical block ciphers in such scenarios. However, CM cannot be said to be robust against transmission errors in general, since the robustness against buffer errors is extremely low due to the high sensitivity towards initial conditions of these schemes. Depending on the target scenario, either CM or classical block ciphers may provide better robustness properties.

5. COMPRESSION ROBUSTNESS

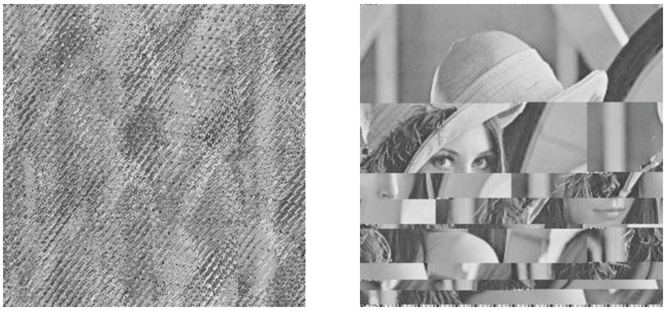
As already outlined in the introduction, classically encrypted images cannot be compressed well, because of the typical properties encryption algorithms have. In particular it is not possible to employ lossy compression schemes since in this case potentially each byte of the encrypted image is changed (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 offense has taken place.

To assure that data cannot 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 8, 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 block 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 undesired additional computational demands for the camera system, this has further disadvantages, as transmission errors in compressed images have usually an even bigger impact without error concealment



(a) 2DCatMap

(b) AES128CBC

Figure 7: Effect of packet errors on Lena image.

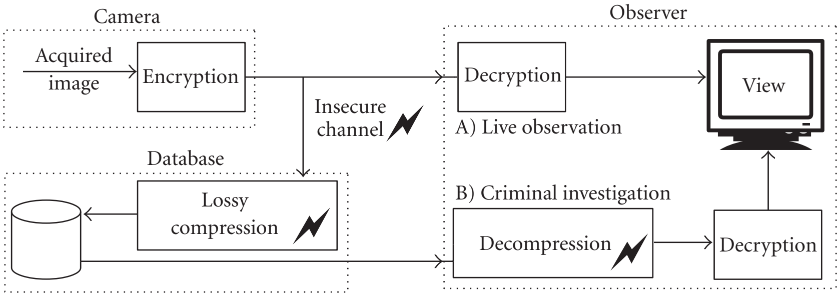


Figure 8: Privacy solution for surveillance systems.

strategies enabled. This strategy increases the error rate as induced by decrypting partially incorrect data even further. 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. Figuure 8 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 heterogeneous 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 as occurring in telemedicine or teleradiology, for example, 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. The classical approach to tackle this second scenario is to apply format compliant encryption to a scalable or embedded bitstream like JPEG2000. While this approach solves the question of transcoding in the encrypted domain in the most elegant manner, the transmission error robustness problem as discussed for the surveillance scenario remains unsolved.

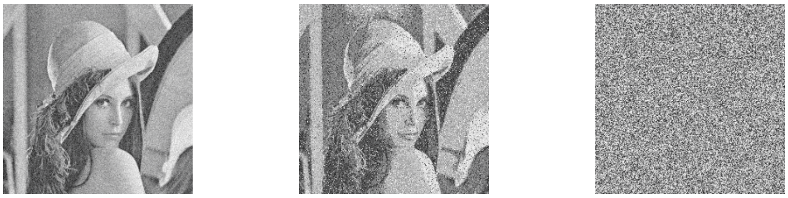
5.1. Experiments

Based on the observation of the excellent robustness of CM against value errors, these encryption schemes seem to be natural candidates to tolerate the application of compression directly in the encrypted domain without the need for decryption and re-encryption. The reason is that compression artifacts caused by most lossy compression schemes may be modeled as random value errors (e.g., errors caused by quantization of single coefficients in JPEG are propagated into the entire block due to the nature of the DCT). In the following, we experiment with applying lossy compression to the encrypted domain of CM.

5.1.1. JPEG-compression of CM encrypted images

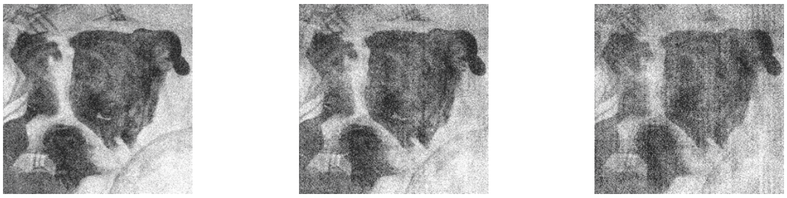
Figures 9–14 show images where the encrypted data got lossy JPEG compressed [15], decompressed, and finally decrypted again. In these figures, we provide the quality factor q of the JPEG compression, the data size of the compressed image in percent % of the original image size, and the PSNR of the decompressed 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



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

Figure 9: Cat map with 5 iterations (without extensions and using 3D and diffusion extensions, resp.), keyset2.



(a) $q = 30$: 29%, 18.9 dB (b) $q = 20$: 21%, 16.4 dB (c) $q = 10$: 13%, 14.5 dB

Figure 10: Cat Map with 5 iterations using different compression ratios on the Ossi image, keyset1.

quite well (depending on the compression rate of course). As already mentioned, this is never the case if classical encryption is applied.

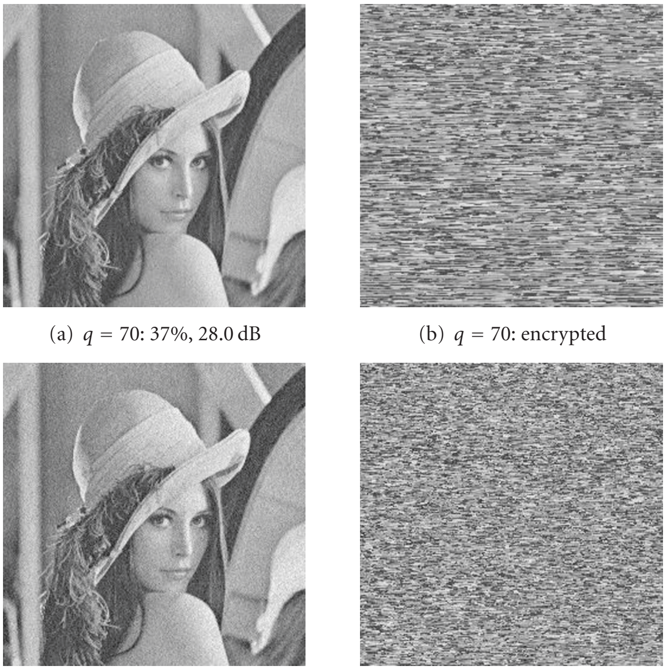
Figure 9 compares the application of the standard 2D *Cat map* without and with additional extensions to increase security (i.e., 3D or diffusion extensions are employed additionally). At a fixed compression rate (slightly lower than 3), we obtain a somewhat noisy but clearly recognizable image in case of no further extensions are used (Figure 9(a)). Applying the 3D extension to the standard *Cat map* (Figure 9(b)), we observe significant degradation of the decrypted image as compared to the standard *Cat map* 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 Figure 9(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 a cryptographically strong cipher like AES had been used instead of *Catdiff*.

The effect when compression ratio is steadily increased is shown in Figure 10 on the Ossi test image. Lower data rates in compression increase the amount of noise in the decrypted images, however, still with a compression ratio of 5 (21%) the image is clearly recognizable and the quality would be sufficient for a handhold phone or PDA display, for example (Figure 10(b)). Of course, higher compression ratios lead to even more severe degradations which are hardly acceptable for any application (e.g., compression ratio 7.5 in Figure 10(c)). However, higher compression ratios could be achieved with sensible quality using more advanced lossy compression schemes like JPEG2000 [18] for example.

Increasing the number of iterations to more than 5 does not affect the results of the *Cat map* for a sensible keyset (as used, e.g., in Figure 9). This is not true for the *Baker map* as shown in Figure 11. When using 5 iterations, the compression result is significantly better as compared to the *Cat map* case with the same data rate (compare Figure 11(a) to Figure 9(a)). The reason is displayed in Figure 11(b); using the *Baker map* with 5 iterations, we still recognize structures (horizontal areas of smoothly varying gray values in a single line) 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 derive key data used in the encryption process.

Increasing the number of iterations (e.g., to 17 as shown in Figures 11(c) and 11(d)) significantly reduces the amount of visible structures. As it is expected, the compression results are similar now to the *Cat map* case using 5 iterations. Using 20 iterations and more, no structures are visible any more and the compression results are identical to the *Cat map* case.

In Figure 12 we give examples of the effects in case pathological key material is used for encryption. When using keyset 1 for encryption with the *Baker map* (Figures 12(a) and 12(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 (Figure 11). With these setting, an even higher number of iterations are required to achieve reasonable security (which again destroys the advantage with respect to compression). Also for the *Cat map*, weak keys exist. In Figure 12(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 Figure 9(a)). These parameters (weak keys) and corresponding effects (reduced security) have been described in the literature on CM and have to be avoided for any application of course.



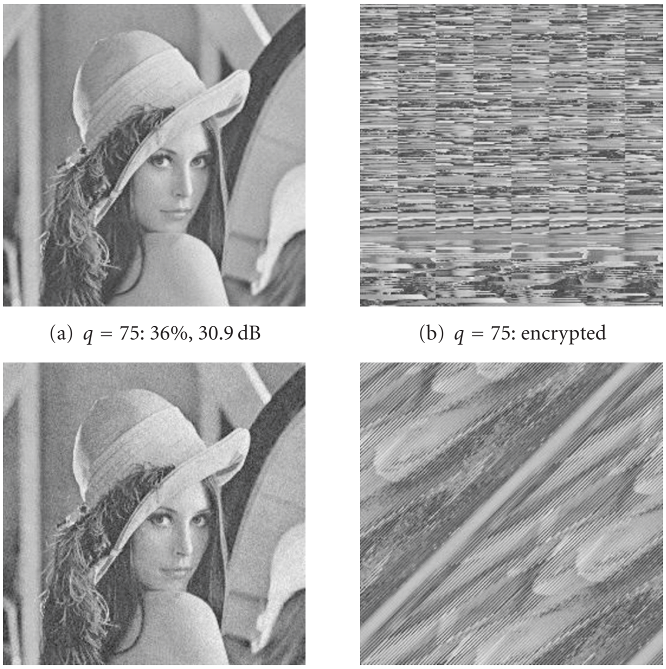
(a) $q = 70$: 37%, 28.0 dB

(b) $q = 70$: encrypted

(c) $q = 60$: 36%, 24.9 dB

(d) $q = 60$: encrypted

FIGURE 11: Baker map with varying number of iterations (5 and 17 iterations), keyset2.



(a) $q = 75$: 36%, 30.9 dB

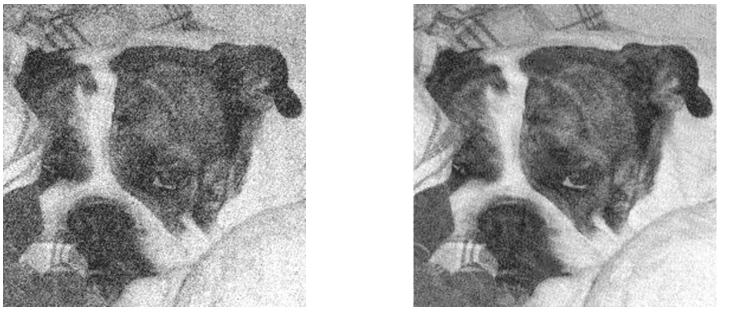
(b) $q = 75$: encrypted

(c) $q = 70$: 36%, 27.3 dB

(d) $q = 70$: encrypted

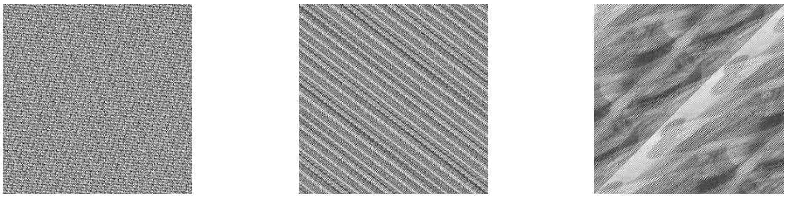
FIGURE 12: Baker map and Cat map with pathological keyset1 (5 and 10 iterations).

Applying the *Cat map* with poor quality keys shows another unique property. While increasing the number of iterations increases the security of the *Baker map* as we have observed, the opposite can occur for the *Cat map* for specific keysets. Accordingly, also compression results are better in this case for a higher number of iterations. Figure 13 shows the Ossi image when applying 7 and 10 iterations using keyset1, while Figure 10(a) shows the case of 5 iterations. Fixing the data rate, the higher the number of iterations is, the better the quality gets.



(a) $q = 30$: 28%, 19.3 dB, 7 iterations (b) $q = 50$: 29%, 23.4 dB, 10 iterations

FIGURE 13: Cat map with 7–10 iterations on the Ossi image, keyset1.



(a) $q = 30$, 5 iterations (b) $q = 30$, 7 iterations (c) $q = 30$, 10 iterations

FIGURE 14: Cat map with 5–10 iterations on the Ossi image, keyset1, encrypted domain.

The reason for this effect is shown in Figure 14. The more iterations are applied, the more structural information is visible and key information may be derived. As shown before for the Lena image, with 10 iterations in use already image content is revealed. Of course, due to the higher amount of coherent structures present in the encrypted domain (especially exhibited in Figure 14(c)), corresponding compression can achieve better results.

5.1.2. JPEG 2000-compression of CM encrypted images

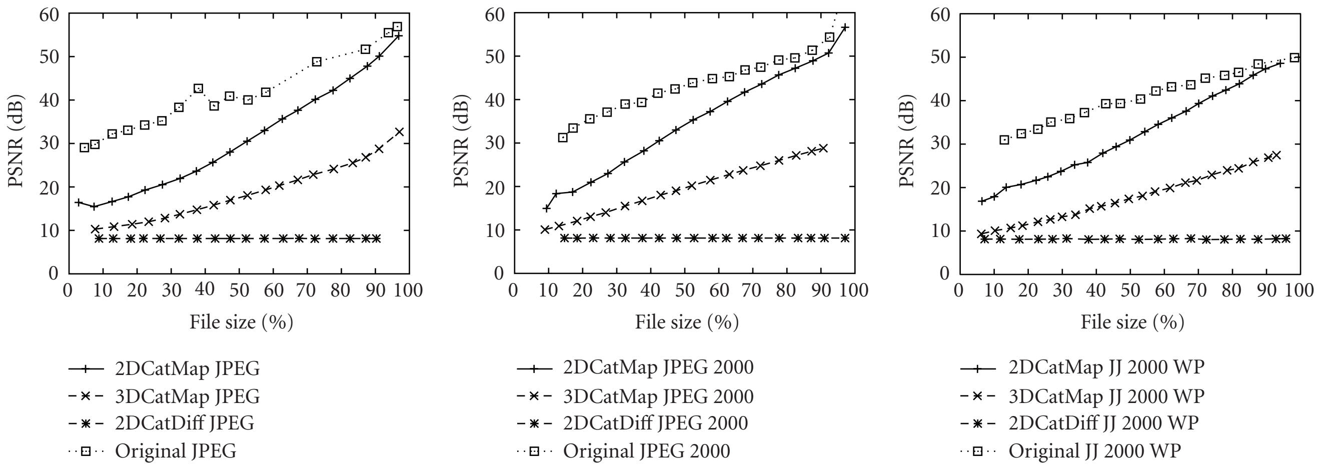
We have not only evaluated lossy compression using the JPEG algorithm but also with JPEG 2000 [18] and JPEG 2000 with wavelet packet decomposition [16] and best basis selection using log energy as cost function and full decomposition. Apart from providing visual evidence as shown in the preceeding subsection, we have also conducted large scale experimentation using the images shown in Figure 2. Figure 15 shows averaged PSNR results for a decreasing amount of compression comparing PSNR quality of original images to three variants of CMs. The results show that the choice of the algorithm has very little impact on the overall trend of our results. While diffusion entirely destroys robustness to lossy compression, 2D (as well as 3D variants to some extent) CMs exhibit a certain amount of robustness against all sorts of compression. While JPEG2000 with classical pyramidal decomposition outperforms the JPEG results by up to 2 dB, the wavelet-packet-based technique performs similar to JPEG only. It seems that the deep decomposition structures produced by the best basis search caused by the noise in the subbands tend to detoriate the results.

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 *Baker map* 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.

6. CONCLUSION

CM behaves differently with respect to robustness against transmission errors depending on the nature of errors. Whereas CM has turned out to be extremely robust in case of value errors, the opposite is true for buffer errors. If pixel values change, the errors remain restricted to the affected pixels even after decryption whereas missing or added pixels entirely destroy the synchronization of the CM schemes. The observed robustness against value errors also explains the unique property to tolerate a medium amount of lossy compression which is an exceptional property not found in other ciphers. Applying the *Cat map* with 5 iterations or the *Baker map* with 20 iterations provides a certain degree of security and decrypted images show acceptable image quality even after significant JPEG compression.



(a) JPEG

(b) JPEG 2000

(c) JJ 2000 WP

Figure 15: Mean PSNR versus file size of 16 different test images under varying using JPEG, JPEG 2000, and JPEG 2000 compression with wavelet packets.

However, the statements about robustness only apply if CM is used without diffusion step (i.e., in a less secure mode). If diffusion is added, robustness against transmission value errors and compression is entirely lost. Even in case only the 3D extension technique is used, robustness is significantly reduced.

As long as a lower security level is acceptable (i.e., diffusion is omitted), classical block ciphers like AES may be complemented by CM block ciphers in case of value errors in an efficient manner (computational demand is much lower and robustness to transmission value errors is higher). Also, lossy compression may be applied in the encrypted domain to a certain extent which is not at all possible with classical ciphers. If high security is required, it is better to stick to classical block ciphers in any environment.

ACKNOWLEDGMENTS

This work has been partially supported by the Austrian Science Fund, Projects nos. 15170 and 19159. The following pictures are licensed under Creative Commons: Figure 2(b) by Emmanuel SalÃ, Figure 2(c) by Michael Jastremski, Figure 2(d) by Natthawut Kulnirundorn, Figure 2(h) by Vinu Thomas, and Figure 2(k) by Scott Kinmartin.

REFERENCES

[1] "Methods for subjective determination of transmission quality," ITU-R Recommendation P.800, 1996.

[2] "Methodology for the subjective assessment of the quality of television pictures," ITU-R Recommendation BT.500-11, 2002.

[3] I. Avcibas, B. Sankur, and K. Sayood, "Statistical evaluation of image quality measures," *Journal of Electronic Imaging*, vol. 11, no. 2, pp. 206–223, 2002.

[4] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons and Fractals*, vol. 21, no. 3, pp. 749–761, 2004.

[5] S.-G. Cho, Z. Bojkovic, D. Milovanovic, J. Lee, and J.-J. Hwang, "Image quality evaluation: Jpeg 2000 versus intraonly h.264/avc high profile," *Facta Universitatis, Nis, Series: Electronics and Energetics*, vol. 20, no. 1, pp. 71–83, 2007.

[6] J. Daemen and V. Rijmen, *The Design of Rijndael: AES—The Advanced Encryption Standard*, Springer, New York, NY, USA, 2002.

[7] A. M. Eskicioglu, "Quality measurement for monochrome compressed images in the past 25 years," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '00)*, vol. 4, pp. 1907–1910, Istanbul, Turkey, June 2000.

[8] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, vol. 8, no. 6, pp. 1259–1284, 1998.

[9] B. Furht and D. Kirovski, Eds., *Multimedia Security Handbook*, CRC Press, Boca Raton, Fla, USA, 2005.

[10] M. Gschwandtner, A. Uhl, and P. Wild, "Compression of encrypted visual data," in *Proceedings of the 10th IFIP International Conference on Communications and Multimedia Security (CMS '06)*, H. Leitold and E. Markatos, Eds., vol. 4237 of *Lecture Notes on Computer Science*, pp. 141–150, Springer, Crete, Greece, October 2006.

[11] Y. Mao and M. Wu, "Security evaluation for communication-friendly encryption of multimedia," in *Proceedings of International Conference on Image Processing (ICIP '04)*, vol. 1, pp. 569–572, Singapore, October 2004.

[12] V. Markovski, F. Xue, and L. Trajković, "Simulation and analysis of packet loss in user datagram protocol transfers," *Journal of Supercomputing*, vol. 20, no. 2, pp. 175–196, 2001.

[13] G. T. Nguyen, R. H. Katy, B. Noble, and M. Satyanaryanan, "Trace-based approach for modeling wireless channel behavior," in *Proceedings of the Winter Simulation Conference (WSC '96)*, pp. 597–604, Coronado, Calif, USA, December 1996.

[14] R. Norcen and A. Uhl, "Encryption of wavelet-coded imagery using random permutations," in *Proceedings of International Conference on Image Processing (ICIP '04)*, vol. 2, pp. 3431–3434, Singapore, October 2004.

[15] W. B. Pennebaker and J. L. Mitchell, *JPEG—Still Image Compression Standard*, Van Nostrand Reinhold, New York, NY, USA, 1993.

[16] M. Reisecker and A. Uhl, "Wavelet-packet subband structures in the evolution of the JPEG 2000 standard," in *Proceedings of the 6th Nordic Signal Processing Symposium (NORSIG '04)*, vol. 46, pp. 97–100, Espoo, Finland, June 2004.

[17] J. Scharinger, "Fast encryption of image data using chaotic Kolmogorov flows," *Journal of Electronic Imaging*, vol. 7, no. 2, pp. 318–325, 1998.

[18] D. Taubman and M. W. Marcellin, *JPEG2000—Image Compression Fundamentals, Standards and Practice*, Kluwer Academic, Boston, Mass, USA, 2002.

[19] A. S. Tosun and W. Feng, "On error preserving encryption algorithms for wireless video transmission," in *Proceedings of the ACM International Multimedia Conference and Exhibition*, no. 4, pp. 302–308, Ottawa, Ontario, Canada, September-October 2001.

[20] A. Uhl and A. Pommer, *Image and Video Encryption. From Digital Rights Management to Secured Personal Communication*, vol. 15 of *Advances in Information Security*, Springer, New York, NY, USA, 2005.

[21] J. G. Wen, M. Severa, W. Zeng, M. H. Luttrell, and W. Jin, "A format-compliant configurable encryption framework for access control of video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 545–557, 2002.

[22] W. Zeng, J. Wen, and M. Severa, "Fast self-synchronous content scrambling by spatially shuffling codewords of compressed bitstreams," in *Proceedings of International Conference on Image Processing (ICIP '02)*, vol. 3, pp. 169–172, Rochester, NY, USA, September 2002.

[23] W. Zeng and S. Lei, "Efficient frequency domain selective scrambling of digital video," *IEEE Transactions on Multimedia*, vol. 5, no. 1, pp. 118–129, 2003.