Application of rank metric codes in digital image watermarking

Pascal Lefèvre, Philippe Carré, Philippe Gaborit

Application of rank metric codes in digital image watermarking

Pascal Lefèvre[1]

Laboratoire XLIM CNRS UMR 7252, Université de Poitiers

Philippe Carré

Laboratoire XLIM CNRS UMR 7252, Université de Poitiers

Philippe Gaborit

Laboratoire XLIM CNRS UMR 7252, Université de Limoges

Abstract

In this paper, we propose a new digital image watermarking algorithm where the resistance against attacks is studied using error correcting codes. Using the well known Lattice QIM in the spatial domain, we propose to use a different kind of error correcting codes called *rank metric codes*. These codes are already well used in cryptography and communications for network coding but not used yet in the context of watermarking.

In this article, we show how this metric permits to correct errors with a specific structure and is adapted to specific image attacks when combined with a watermarking technique. In particular, we describe a rank metric code family called *Gabidulin codes* analogous to the well known Reed-Solomon codes. If one considers a rank code over a finite field extension, then any codeword has a matrix representation. One can decode the original message if the matrix rank of the detected codeword is small enough.

We propose a study to validate the concept of rank metric in watermarking applications. First, we introduce a theoretically invariant method to luminance additive constant change. After combining the Lattice QIM method and rank metric codes, we add a multi-detection strategy on the damaged images with controlled luminance distortions. Then, using a block-based watermarking approach, we show how the proposed association can also be robust to an image distortion we called *content erasure* or *copy-paste*. The proposed approach completes other watermarking strategies against attacks with random errors such as JPEG compression.

Keywords: Watermarking, error correcting codes, rank metric, luminance modification, content erasure, copy-paste.

1. Introduction

Image watermarking is an important area of research. An example of motivation is the strong need to protect online multimedia contents. To ensure copyright and intellectual property over massive online distribution, we need efficient protection to control the distribution, stop manipulations and duplications by pirates or unaware normal users.

To be efficient, a watermark needs to be imperceptible, needs to embed high capacity payloads and has be to robust [1] against the most common image processings (malicious or not) while ensuring a secure transmission of the payload [2].

A well-known and powerful tool to enchance the robustness of a watermark is the use of error-correcting codes which permit to correct errors induced by a given attack. To embed a message, it is first encoded into a codeword that is used as the watermarking payload. For detection, an estimation of the original codeword is computed and then decoded by the same correcting code to recover the original message.

Now, depending on the embedding strategy and the error structure induced by an image modification (also called an attack or a distortion), the type of codes used to encode the payload can be more or less efficient. For instance, if the error induced on the watermark is random (e.g. JPEG compression), the best results are obtained with binary codes, like BCH codes for instance (in the case of small lengths).

For other attacks, it may happen that the error comes in packet. In that case, it is better to use more structured codes over a larger alphabet (say $GF(2^m)$), like Reed-Solomon codes where decoding is done by packets [3].

One does not decode error independently on each bit, but on packet of m bits, so that an error on each bit of the packet or only one error on one bit of the packet is corrected the same way. Therefore, based on the attack, *i.e.* based on the error type, we can choose an adapted error correcting code. This point of view on the error type is rather well known and led to numerous industrial applications of these Hamming codes.

In this paper, we consider a new type of metric called *rank metric* which will allow us to be robust against attacks that are not handled by Hamming codes. Error correcting codes using this metric are already often used in network coding [4] and

[1]Corresponding authors

cryptography [5]. They permit to correct errors with a specific structure. If one considers a code over $GF(2^m)$ of length m, each coordinate of a codeword over $GF(2^m)$ is encoded by m bits, and since the code has length m, any codeword can be seen as a $m \times m$ matrix. Now, it is possible to correct errors for $m \times m$ error matrix of low rank.

For instance, consider an attack on the image which flips every bit of the payload (or codeword). If one consider the usual Hamming metric, it is not possible to correct this error since all bits are false. Meanwhile, in terms of rank metric, the associated error matrix has rank 1 (because it is filled with ones only) and hence, the received modified matrix can be decoded and the original message retrieved.

In this paper, we propose to use this original concept of rank metric with *Lattice QIM* watermarking strategy to obtain new robustness properties against a new family of attacks.

Our contribution: we introduce the concept of particular error structure and how rank metric codes (section 2) can be useful for watermarking. Then, we propose a watermarking process which combines rank codes with Lattice QIM method to deal with structured errors produced by luminance modifications and copy-paste modifications. In the case of both attacks, conventional Hamming codes are not as efficient as rank metric codes.

We explain why such structure exists and provide an *enhanced Lattice QIM detector*. It is based on a LQIM multi-detection strategy on attacked images with controlled luminance distortions. Theoretically, we obtain error free detections against luminance modifications. Finally, we propose a block-based watermarking scheme to resist *content erasure/copy-paste attack*.

2. Rank metric codes

In this section, we introduce rank metric codes definitions, properties and their applications in practice. Considering existing approaches, these codes allows us to propose a complementary watermarking strategy against particular attacks such as *content erasure* or *copy-paste* attack.

2.1. Linear codes

We consider a linear code C of length n over the alphabet $GF(q^m)$. Codewords from C are row vectors from the vector space $GF(q^m)^n$ usually denoted by c. Every components of c can be written as a vector of $GF(q)^m$. Then, it is possible to write every component of c by a column vector and represent the codeword x as a matrix of $GF(q)_{m\times n}$.

Now, let us consider $\mathcal{B} = (\beta_1, ..., \beta_m)$ a basis of $GF(q^m)$ over $GF(q)$ and a codeword $x = (x_1, ..., x_n) \in GF(q^m)^n$. The matrix representation of x denoted by $Mat(x) = (x_{ij})_{i,j}$ (or denoted by X when there is not ambiguity) is defined as :

$$X = \begin{pmatrix} x_{11} & \dots & x_{1n} \\ \vdots & & \vdots \\ x_{m1} & \dots & x_{mn} \end{pmatrix}$$

such that for all $1 \leq j \leq n$:

$$x_j = \sum_{i=1}^{m} x_{ij}\beta_i$$

This matrix representation allows to define a new metric over $GF(q^m)^n$ using the matrix rank.

2.2. Rank distance

Let $x = (x_1, \dots, x_n) \in GF(q^m)^n$. The rank weight of x is written $w_r(x)$ and is equal to :

$$w_r(x) = rk(X) \tag{1}$$

with $rk(X)$ the matrix rank of X (the number of independent matrix rows or columns).

Let $y = (y_1, \dots, y_n) \in GF(q^m)^n$. The rank distance from x to y, noted $d_r(x, y)$ is equal to :

$$d_r(x, y) = rk(X - Y) \tag{2}$$

Of course, one can check that d_r has the distance properties. Compared to Hamming distance d_h, we have the following property :

$$w_r(x) \leq w_h(x) \tag{3}$$

with x a codeword and $w_h(x)$ the Hamming weight of x :

$$w_h(x) = |\{(i, j) \mid x_{ij} \neq 0\}| \tag{4}$$

with x_{ij} the matrix components of $Mat(x)$.

We also deduce that :

$$d_r(x, y) \leq d_h(x, y) \tag{5}$$

with $d_h(x, y)$ the Hamming distance between x and y :

$$d_h(x, y) = w_h(x - y) \tag{6}$$

Briefly, this property is true because the number of non zero linearly independent matrix rows or columns (the rank) is always lower than the number of base field symbol differences (Hamming weight). In other words, as the rank of a vector is independant of the basis, the rank metric is less precise than the Hamming metric as two vectors with different Hamming distance could have the same rank.

2.3. Minimal distance

Rank metric codes were first studied by Delsarte [6] in 1978. Many properties of Hamming codes are adapted to rank metric codes. A linear code C defined of the finite field $GF(q^m)$ can be seen as a subspace of $GF(q^m)^n$ but also as a metric space when equipped with the rank distance.

Moreover, C is a linear code if C is a vector subspace of $GF(q^m)^n$. Code linearity is an interesting property because it

is easier to manipulate codewords for example and allows easier decoding. As in Hamming codes, we define the minimal distance d_{min} of a rank code such that :

$$d_{min} = \min_{x \neq y \in C} d_r(x, y) \tag{7}$$

and with code linearity property, we have :

$$d_{min} = \min_{x \in C^*} w_r(x) \tag{8}$$

If a linear rank code C has length n, dimension k and minimal distance d, we denote it by its parameters $[n, k, d]_r$ or $[n, k]_r$ if we don't need to know the minimal distance of the code.

2.4. Decoding Gabidulin codes

Decoding bounds for rank codes (Singleton and Gilbert-Varshanov bounds) are similar to Hamming codes decoding bounds. They are very useful to design decoding algorithms. Unlike classical Hamming codes, only few families of codes with easy rank metric decoding algorithms are known.

Gabidulin codes [7] is one code family among them and has parameters $[n, k, n-k+1]_r$ over $GF(q^n)$. They are called Maximum Rank Distance codes (MRD) because they can decode errors with rank at most :

$$t = \left\lfloor \frac{n-k}{2} \right\rfloor \tag{9}$$

They can be seen as an analogous family in rank metric of the well-known Reed-Solomon codes family which are Maximum Distance Separable (MDS). Many algorithms to decode Gabidulin codes has been proposed in the literature such as [8, 9].

Reed-Solomon code decoding algorithm can be generalized for Gabidulin codes. Using the Welch-Berlekamp algorithm on linear polynomials, one can decode faster with quadratic complexity.

2.5. Rank metric codes in practice

In practice, we use Gabidulin codes in an extension $GF(q^m)$ of $GF(2)$, and one associates a binary vector of length m to any coordinate of the codeword, so that a codeword c can be seen as a $m \times m$ binary matrix.

After an attack on the watermark, the codeword c is modified with error e, which also is a $m \times m$ binary matrix. To evaluate if the rank metric is better than the classical Hamming metric, we compare the embedded watermark (a codeword c) with the modified watermark ($y = c + e$). Suppose $m = 4$. Let c be a codeword and $y = c + e$ a modified codeword such that :

$$c = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}, y = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Then, the error matrix is:

$$e = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

The error matrix e has rank 2, hence, if the code can correct up to 2 rank errors, then it is possible to decode y into c. In terms of Hamming metric, if we had started from a length 16 binary code, it would correspond to an error of weight 4.

In that particular case, it is possible to find both Hamming or rank metric codes which can decode this type of errors, for reasonable dimensions k. Suppose we now have an error matrix such that:

$$e = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

The Hamming weight of e is 9 and the rank of e is 4. We see that it is not possible to decode with both metrics with this error matrix. In fact, rank metric is more interesting when the error has a particular structure such as:

$$e = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

With Hamming metric, there are still 9 errors out of 16 bits transmitted, no binary code of length 16 is able to decode properly while with the rank metric, the rank of e is only 1. We can easily decode with such error (for instance, with a Gabidulin code of parameters $[4, 2, 3]$).

Of course, such a structured error does not happen necessarily often, especially in the case of binary flipping. Unlike classical Hamming codes, rank metric codes are very efficient when dealing with this error structure. In this contribution, we use LQIM method to take advantage of the lattice construction and capture the error structure. In the next section, we briefly describe this method used in combination with rank codes.

3. Lattice QIM (LQIM)

The vector quantization method called *Lattice Quantization Index Modulation* (LQIM) was introduced by B. Chen and Gregory W. Wornell ([10, 11]).

In our work, we use two cosets to embed binary information using the lattice $\Delta\mathbb{Z}^L$ of dimension L and a quantizer Q_m defined such that :

$$\begin{aligned} \Lambda_0 = -\frac{\Delta}{4} + \Delta\mathbb{Z}^L, \Lambda_1 = \frac{\Delta}{4} + \Delta\mathbb{Z}^L \\ y = Q_m(x, \Delta) = \left\lfloor \frac{x}{\Delta} \right\rfloor \Delta + (-1)^{m+1} \frac{\Delta}{4}, \end{aligned} \tag{10}$$

with x a host sample, y the quantized sample and a bit $m = 0, 1$.

Figure 1 illustrates an example of the quantization space. For any circle or cross (say a quantized vector y), the diamond delimited by the dotted lines will be denoted by "quantization

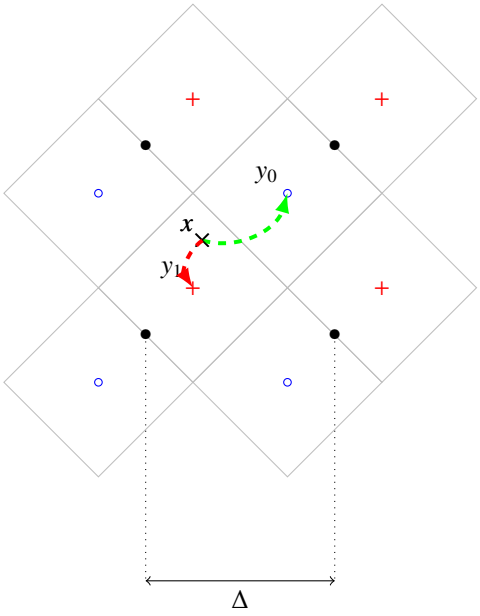


Figure 1: Representation of the quantization space of dimension $L = 2$. + symbols are the result of quantization carrying bit 1 (coset Λ_1) and ∘ symbols are associated with bit 0 (coset Λ_0).

cell". To embed information in x, one can quantize x to the nearest quantization cell center y.

For detection, we compute which coset is closer to the received vector z:

$$\begin{aligned} \hat{m} &= \arg\min_{m\in\{0,1\}} dist(z, \Lambda_m), \\ dist(z, \Lambda) &= \min_{y\in\Lambda} \|z - y\|_2 \end{aligned} \tag{11}$$

Vector z is transformed into y which is the quantization cell center where z is located. Quantization is an interesting concept when dealing with an image processing that applies the same distortion everywhere on an image.

Since the quantization space of LQIM method is divided into equally-sized cells, the lattice structure preserves the potentially particular form of errors (with respect to rank codes) at the detection step. Hence, these codes' properties would allow the receiver to recover the embedded message. However, the amount of embbedable information is reduced. Given an error correcting code of ratio k/n combined with LQIM method of dimension L, the maximum payload r is :

$$r = \frac{khw}{nL} \tag{12}$$

with (h, w) the image size. In practice, these parameters are chosen such that image quality is satisfying.

In the next section, we study a particular image processing called luminance additive constant change. The error type produced by this attack are partially structured and very well handled by rank metric codes. A partial research is presented in the conference ICASSP 2018 (see [12]).

4. Study of the luminance modification

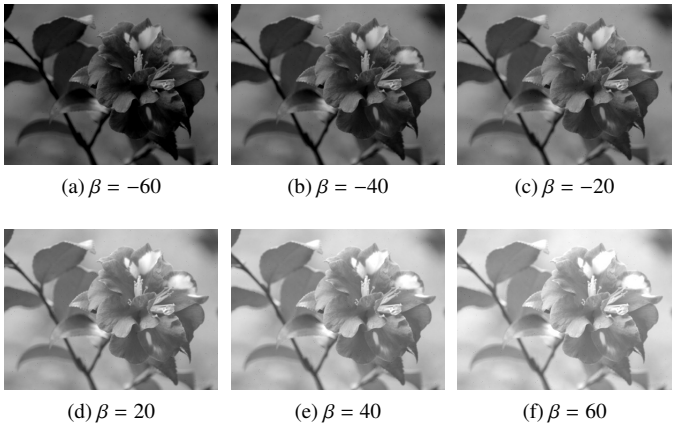
4.1. Attack definition

A luminance modification is parametrized by a parameter $\beta \in \mathbb{R}$. A modified image by this attack darkens the original image if $\beta < 0$ and lightens the original image otherwise as shown in figure 2 with pixel values in $[0, \ldots, 255]$. If $|\beta|$ is small, it is difficult to see the difference between the original and the modified images.

Let y a host vector sample of an image $\mathcal{I}$ and z the corresponding modified sample, the luminance modification is given by the formula :

$$z = y + \beta \times u \tag{13}$$

with $u = (1, \ldots, 1) \in \mathbb{R}^L$. In fact, every pixel values of $\mathcal{I}$ suffer the same distortion.



(a) $\beta = -60$ (b) $\beta = -40$ (c) $\beta = -20$

(d) $\beta = 20$ (e) $\beta = 40$ (f) $\beta = 60$

Figure 2: Image examples modified by a luminance attack. They are either very dark (pixel values close to 0), either very bright (pixel values close to 255).

4.2. Binary error rates analysis and error structure

Results were made using the Corel image database where 1000 images where randomly chosen for our tests among the 10000 images available. Since watermark invisibility is very important, we used a quality measure called *Document to Watermark Ratio* (DWR) defined as a signal to noise ratio such as :

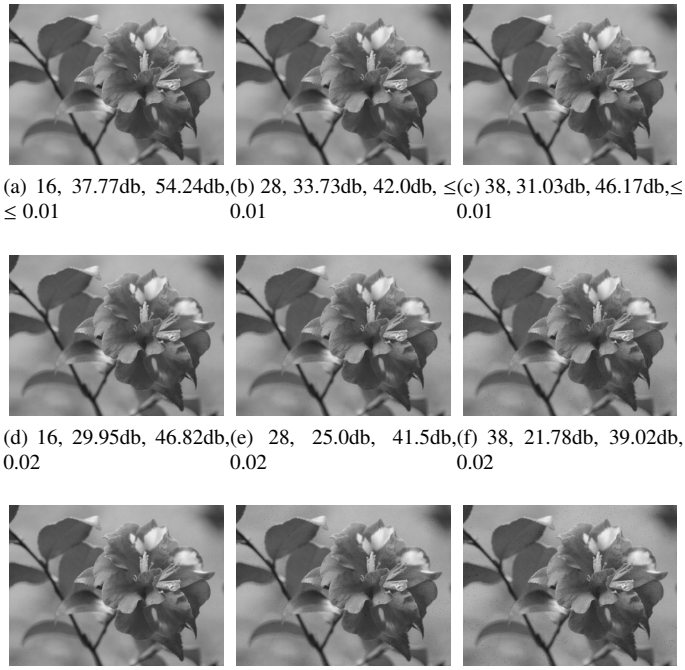
$$\mathrm{DWR}(I_h, I_w) = 10 \log_{10} \frac{\sigma_{I_h}}{\sigma_{I_w} - \sigma_{I_h}} \tag{14}$$

with I_h the host image, I_w the watermarked image and σ_I the variance of image I. In all our experiments, a DWR of 35db produce an invisible watermark. Moreover, we also chose to add Peak Signal to Noise Ratio (PSNR) and Structural SIMilarity (SSIM) quality measures to improve comparability with others contributions. If SSIM measures are close to 1 between host and watermarked images, then the quantization noise can be considered as very weak, i.e. the watermark is invisible enough to maintain a good image quality.

Embedded messages are randomly generated binary sequences of 49 bits with $L = 6$ and $\Delta = 28$ so that we obtain the desired

DWR in the spatial domain. For every measures, maintaining a DWR of at least $35db$ always gives PSNR = 48.2db SSIM = 1.

In figure 3, we illustrated examples of marked images. High values of embedding rate (ER) and quantization step Δ produce a low image quality (DWR decreases) and, is, hence, more visible to the human eye. We can see a salt and pepper noise because coefficients are randomly chosen for the embedding.



(a) 16, 37.77db, 54.24db, ≤ 0.01 (b) 28, 33.73db, 42.0db, ≤ 0.01 (c) 38, 31.03db, 46.17db, ≤ 0.01

(d) 16, 29.95db, 46.82db, 0.02 (e) 28, 25.0db, 41.5db, 0.02 (f) 38, 21.78db, 39.02db, 0.02

(g) 16, 27.75db, 43.31db, 0.03 (h) 28, 22.64db, 38.28db, 0.03 (i) 38, 19.79db, 36.42db, 0.03

Figure 3: Image examples marked with different embedding parameters (Δ, DWR, PSNR, ER). For $\Delta = 38$, we have $SSIM = 0.99$.

The first experiment shows the average binary error rates (BERs) between the original codeword and the estimated one in function of β (figure 4) over the 1000 test images.

We can see that the red curve looks like a square waveform (the curve periodically alternates between 0 and 1) and we can distinguish three cases: BER $= 0, 0.5, 1$. In the first case, there is no error at detection at regular intervals (such as $\beta \in [22, 34[$). Then, the third case is similar to the first case; BER = 1 also happens at regular intervals (such as $\beta \in [8, 20[$). This curve clearly shows the existence of a partially structured error form.

This third case represents situations when every bits of the payload are flipped at the same time because there is 100% errors. In other words, every quantized vectors have suffered from the same distortion (equation 13). In the second case (BER = 0.5), detected payloads are random sequences.

When β increases, z saturates (pixel values are moved closer to 0 if $\beta < 0$ or close to 255 if $\beta > 0$). From a geometrical point of view (in 2D for figure 5), every z travel from one quantization cell to another flipping the embedded bit at every quantization cell change.

For $\beta \leq 6$, the LQIM detector is in the first state (illustrated in subfigure 5a). Then, we have a transitional state for $\beta = 7$

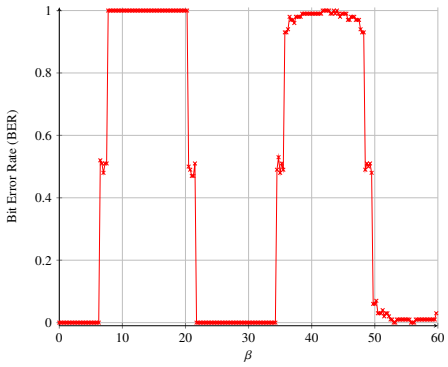
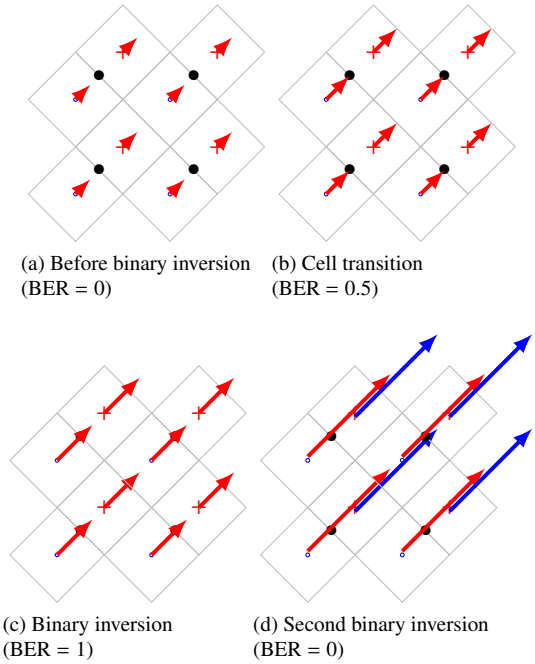


Figure 4: Binary error rate of LQIM method in function of β. This curve is similar when β is negative.

(subfigure 5b). Detected vectors z are located at the boundaries of quantization cells and the detector has a probability of 0.5 to guess which side of the boundary z is located.

Finally, every z crossed their cell boundary in the third state and the detected binary payload is entirely reversed with $8 \leq \beta \leq 20$ (subfigure 5c). At last (subfigure 5d), every z has traveled through another transitional state to another quantization cell and allow the detector to correctly retrieve the embedded payload.



(a) Before binary inversion (BER = 0)

(b) Cell transition (BER = 0.5)

(c) Binary inversion (BER = 1)

(d) Second binary inversion (BER = 0)

Figure 5: Representation of the quantization space of dimension $L = 2$ and the three cases related to a luminance modification.

For every β, the decoding step is described for three cases (BER $= 0, 0.5, 1$). Even though, only few values of β lead to the second case, it is possible to use a rank metric code to remove the partially structured errors. In the next subsection, we show how the LQIM method combined with a rank code can remove the majority of errors.

4.2.1. Rank metric codes application

As a second experiment, we used a rank metric code of parameters $[7, 3, 5]$ (corrects at most errors of rank 2) and measured Image Error Rates (IERs) and embedded a codeword as the watermark payload. IERs are the ratio of images where the message was not decoded by the rank metric, *i.e.*, the error rank $rk(e) \geq 2$.

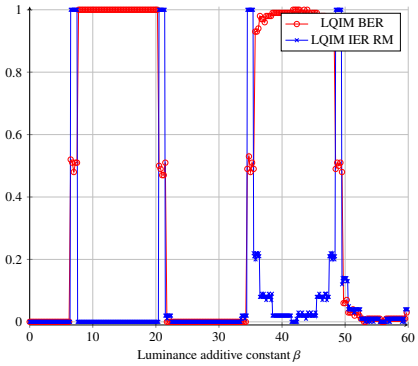


Figure 6: Red curve : Binary error rate of LQIM method in function of β. This curve is similar when β is negative. Blue curve : Image error rate of LQIM method combined with a rank metric code in function of β. Each point of the blue curve represents the ratio of images where the error rank $Rank(e) \geq 2$ (failed decoding).

In figure 6, this code is very efficient because IERs are 0 for every β except at four values. When z makes the transition from one cell to another, LQIM detector estimates BER = 0.5 (which means we have errors with full rank).

On the other hand, a similar parameters Hamming code is $[47, 23, 11]$ and corrects at most 5 binary errors over 47 bits. In that case, the IER curve obtained with this code is identical to the red BER curve because either we have no binary error either every bits of the payload is flipped, *i.e.*, the Hamming metric is inefficient against this attack.

In practice, the detector cannot guess β and the probability to find β such that errors are not structured for rank metric codes depends on Δ : a small value means a higher probability to detect with errors.

Moreover, the red curve in figure 6 does not look like a square waveform (the curve brutally alternates periodically between 0 and 1) for some images due to the random nature of the pixel values they contain. Some BER values might be slightly under 1 or slightly above 0.

It sometimes happens that luminance distortions are strong enough to completely erase the embedded information (pixel values range between 0 and 255). We show some image examples where this decoding problem occurs in figure 9. In our experiments, we chose $\beta > 0$ which explains why those attacked images look very bright. With $\beta < 0$, the same decoding problem happens with very dark images.

For a start, this justify the use of a rank metric code of parameters $(7, 3, 5)$ correcting at most errors of rank 2 to correct more errors. Theoretically, a code correcting errors of rank at most 1 is enough. In the context of a luminance image processing, these codes provide almost perfect error correction. In the next subsection, we introduce a multi-decoding strategy on images with controlled luminance distortions in order to ignore failed decoding cases when BER = 0.5.

4.2.2. Enhanced LQIM rank metric detector

The luminance channel is parametrized by the additive constant β. Suppose a watermarked image is damaged by this channel. At decoding, we have the modified versions $z = y + \beta$. Equation 13 shows how to improve the detector performances by adding a controlled luminance modifications. Periodically, one can notice that we cannot properly detect for $\beta = \sqrt{2}\Delta/4$ with $k \in \mathbb{Z}$. This case represents transition states traveling from one quantization cell to another, *i.e.*, vectors z are located at the boundaries of quantization cells.

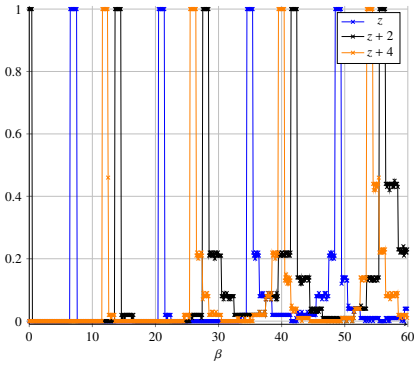


Figure 7: Image error rates of LQIM method combined with a rank metric code in function of β with controlled distortions using $\delta = 0, 2, 4$ to shift error rates curves from one to another (spike shift). Quantization step $\Delta = 28$.

Let $\delta_i \leq \sqrt{2}\Delta/4, \delta_i \in \mathbb{N}, 1 \leq i \leq n$. Then, from figure 7, we deduce the following property: there is a unique i such that the corrupted image with $z + \delta_i$ cannot be well detected with LQIM rank metric detector and for every $j \neq i$, corrupted image with $z + \delta_j$ is correctly detected with LQIM rank metric detector. By modifying z with δ_i, we guarantee to have the majority of $z + \delta_j$ perfectly detected.

A majority vote strategy on the decoding of multiple attacked image can get rid of the spikes at the only cost of time decoding. Taking $n = 3$ suffices to have good results with this decoding strategy. We have $d = \sqrt{2}\Delta/4$, $\delta_1 = 0$, $\delta_2 = d/3$ and $\delta_3 = 2d/3$. In the experiments, $d \simeq 6$ and we used $\delta_1 = 0$, $\delta_2 = 2$ and $\delta = 4$ and they represent modified versions of transmitted z.

Then, we extract 3 estimations of the original payload. Using the proposed property, two out of the three payloads are correct given fixed β. In figure 8, error rates are 0 for almost every β. For example, with $55 \leq \beta \leq 60$, we have non-zero error rates (quick variations with visible spikes). Again, the previously described problem on the random nature of images can be observed (9) since those error rates are computed from image error rates of figure 7.

As a summary, we proposed a strategy to improve the LQIM detector combined with a rank metric code against a particular attack we denoted by luminance modifications. Cases where BER = 0.5 can be avoided by taking an average estimation of

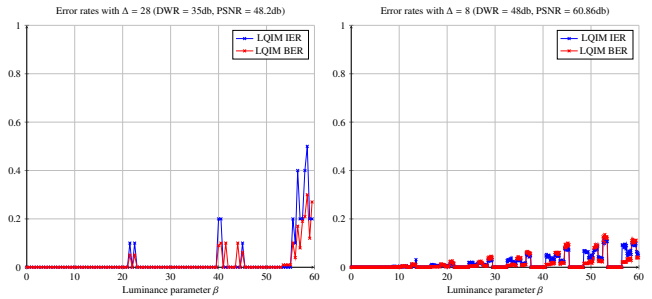
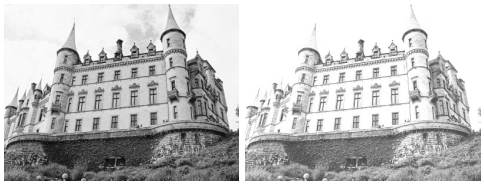


Figure 8: Red and blue curves respectively represents BER and IER of LQIM embedding with enhanced LQIM rank metric detector in function of β. Experiments with two different Δ are showed.



(a) 706.jpg, $\beta = 59$

(b) 5371.jpg, $\beta = 58$

(c) 6211.jpg, $\beta = 57$

Figure 9: Examples of marked/attacked images pairs where decoding failed after a luminance attack.

multiple codewords where the luminance parameter was modified on purpose. With this enhanced detector, it is possible to use smaller quantization steps for more invisible watermark (see figure 8 with $\Delta = 8$).

Indeed, the correction ability of rank metric codes eliminates all errors except at some particular β values. Using the proposed property, we can shift curves *i.e.* IER spikes and compute the majority vote of payloads.

Compared to other approaches against luminance image processing (such as embedding in the low frequency coefficients in the DCT domain), we innovate with a theoretically perfect resistance with weak quantization noise at the cost of some capacity (code rate k/n).

Results are very interesting for luminance because of the error structure but less interesting for others attacks (such as JPEG compression and additive white gaussian noise) where the error type is not structured until we can imagine a particular design. In the next section, we focus on the content erasure attack and how the use of rank metric codes can be useful against this image processing.

5. A watermarking framework robust to copy-paste modifications

5.1. Literature

In this section, we study an image distortion we can call *content erasure*, *image slicing* or *copy-paste* and is very close to image cropping. After a brief description of the literature on this last attack, we explain our approach based on rank metric codes. As described in [13], one of the oldest contributions on image cropping are from Swanson et al. [14]. They proposed a robust watermarking method which a LSB embedding on DCT coefficients to embed a watermark. Copy-paste can also be associated with *collage attack* (variation of the Holliman-Memon counterfeiting attack [15]) studied in the context of digital image authentication by Fridrich et al. [16].

Moreover, this image processing has more often been studied in the context of image authentication and tamper detection.

Later, others contributions were proposed such as [17]. The authors proposed a *self-embedding* mechanism that allows the recovery of cropped out, replaced and tampered image portions. Their method consists in embedding a compressed version of the host image into itself with LSB method on DCT coefficients. They can achieve a recovery of about 50% JPEG compression quality by quantizing the two least significant bits.

More recently, inspired by the mathematics of Sudoku [18, 19], research has been done such as [20, 21, 22] to solve image cropping and collage attack problems. Their embedding strategy using Sudoku follows the same principle of self-embedding previsously described except the authors use a resizing function to reduce the watermark size. A host image is divided into $N \times N$ cells identified by numbers $(1, \ldots, N)$. A new image is generated with each cell using a Sudoku grid solution and then downsized for LSB embedding in DCT coefficients.

However, those contributions mostly talk about image authentication and tamper detection as *fragile watermarking* methods. The idea of this data hiding paradigm is to embed a pattern such that analyzing it would tell if the host content could, whether or not, be considered as modified, tampered, authentic, etc. Compared to robust watermarking, goals are different since the embedded content is related to the host image and must be retrieved without errors (copyright protection for example).

The image cropping is a more complicated problem than copy-paste because of the difference of images sizes. At the detection step, one must synchronize the watermark before attempting to retrieve it. Kutter [23], in 1999, proposes a solution that involves the embedding of several watermarks in an image and was used by every contributions of the literature presented above.

5.2. Description of the proposed method and discussions

In general, it is rather difficult to design a method resistent to this type of attacks. Fundamentally, every proposed work is

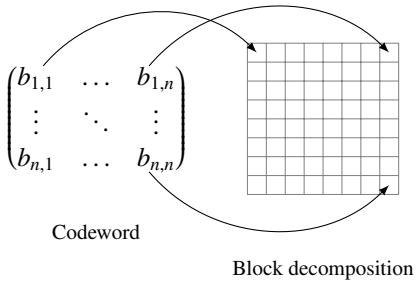


Figure 10: Rank metric codeword embedding strategy into an image decomposed in blocks. Each bit $b_{i,j}$ is associated to one block.

based on information redundancy. Hence, every locally embedded information must rely on other regions of the image containing information in order to decode the embedded codeword without errors.

The only way to retrieve the initial message without errors is to spread the embedding everywhere on the image. If a region is erased and is small enough then the watermark can be correctly retrieved.

In the literature, most approaches are very close to using a repetition code. For example, a simple strategy to embed a watermark resistant to image cropping is to repeat it several times in the image.

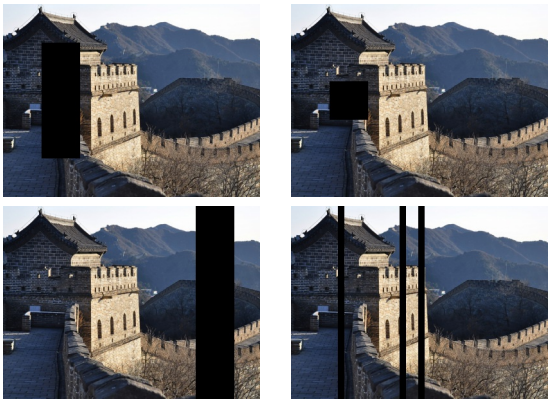


Figure 11: Attacked images with same error rank.

We propose a different approach to handle the copy-paste attack with the help of rank metric codes. An image to be marked is decomposed in n^2 blocks with n the rank metric code length (block decomposition is illustrated in figure 10). Then, every binary symbol of a rank codeword is embedded into a block using the LQIM method. For each block, the associated binary symbol is embedded using $L = 2$ coefficients. We obtain an image which looks like a matrix rank codeword. Hence, distortions produced by the studied attack are directly reproduced on the error matrix e.

For instance, a square region (of size l) of a marked image is erased. Then, affected blocks are directly represented as errors in e. Now, consider a row or a column of width l sliced out. In matrix e, rows or columns with bit 1 appear. As we saw in subsection 2.5, e has a particular structure perfectly handled by rank metric codes. Indeed, we have $rk(e) = r'$ with r' the number of blocks (on the width) affected by the content erasure operation.

Both previously described examples have the same error rank. In the second case, distortion is maximized compared to the first case with the square erased region. As an illustration, we give in figure 11 an example of attacked images which corresponding error matrix has the same rank.

Another interesting fact about rank metric is : select two rows made of blocks and swap them, even though the image is modified, the associated error rank is 0. Then swap two columns made of blocks, the error rank remains constant. These two operations can be repeated undefinitely without increasing the error rank while having a highly damaged image. The only condition to take advantage of the mathematical properties of the matrix rank is to crop entire blocks only. In this paper, this fact is not experimentally studied because of not being realistic enough although this swapping property is theoretically curious when looking at the damaged image.

In the next subsection, we describe and analyze our robustness experimental results against the content erasure attack.

5.3. Robustness experiments and analysis

In our study, we distinguish two types of content erasure : the first type gathers fully columns/rows sliced regions (subfigure 12a) and the second gathers rectangle erased regions (subfigure 12b) even though the error ranks are the same. For the same error rank, distortions are maximized with the first type compared to the second type.

For our test measures, we consider the first type of content erasure for practical reasons. With the second type, the average error ranks are exactly the same, except the maximum size of attacked regions is smaller.

Distortions are computed using the percentage of distorted regions denoted by cr. For the first type, we can define cr such that :

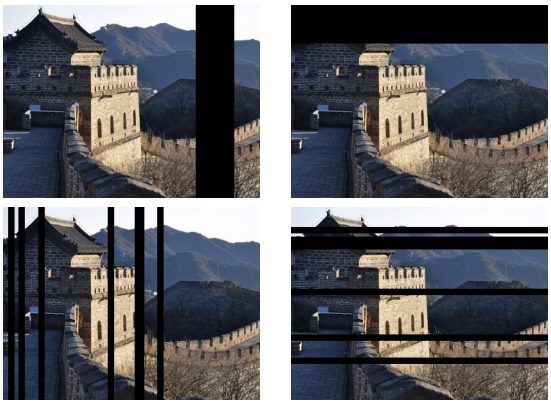
$$cr = \frac{100l}{h} \tag{15}$$

with l the number of pixel columns and h the image height. In our experiments, we consider content erasures where the left part of the image is sliced out of l columns.

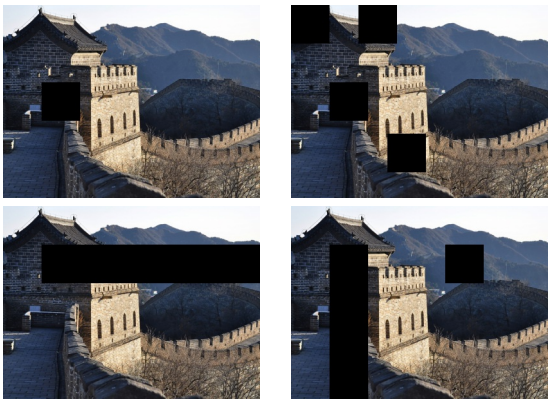
We randomly chose 1000 images from the Corel image database with image sizes of 300×400 or 400×300 and compute the averages of binary error rates, image error rates and error ranks. We chose different Gabidulin codes to measure the robustness of the proposed method against content erasure.

First of all, the proposed method show some robustness to content erasure distortions by construction. In figure 14, we can see that binary error rates and error ranks are linearly increasing as cr increases for every n.

When the average error rank (± 1.6) is greater than the maximum number of allowed errors, we can see that IERs immediately change their values from 0 to 1. It is then not possible to recover the payload. The maximum value of cr allowing an error free decoding is denoted cr_{max}.

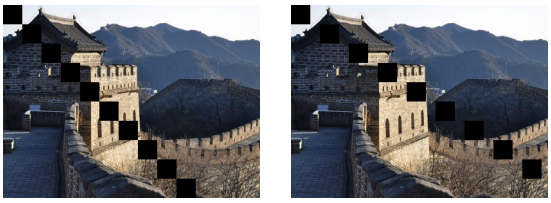


(a) Content erasure type 1



(b) Content erasure type 2

Figure 12: Content erasure types



$cr' = 7.5\%$ $cr' = 6.75\%$

Figure 13: Worst case example of attacked images where the error rank is too high ($\geq$) for a small percentage of erased region (noted cr').

This first observation experimentally shows the applicability of rank metric codes in watermarking against content erasure. Of course, there are worst case scenarios. The underlying concept of the proposed method is to take advantage of the mathematical properties of the matrix rank. Hence, it is easy to make an example of attacked image where decoding is not possible (see figure 13).

If we discuss parameters in detail, when code rates k/n are decreasing, the correction power t increases which allows us to get higher values or cr_{max}, i.e., the watermark is more robust. Secondly, when k is fixed and n increases, the correction power is higher.

However, one must carefully evaluate the image quality which drastically decreases. Indeed, a higher value of n implies n^2

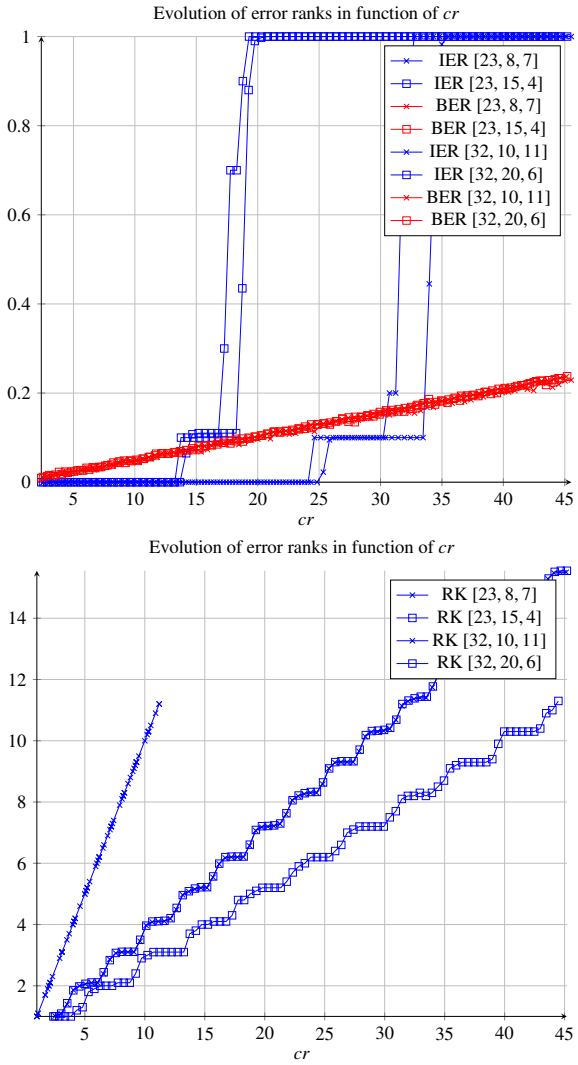


Figure 14: Error rates and ranks in function of the erased region percentage cr.

blocks with smaller size are created at the embedding and more pixel values are modified. In addition, the lower k is, the lower is the available information capacity.

Compared to BCH codes, rank metric codes are more efficient than BCH codes against the first distortion type. Indeed, we can see that cr_{max} values achieved by rank metric codes correspond to higher binary error rates than correction rate given by code rate corresponding BCH codes. For instance, with $(n, k) = (23, 8)$ and $cr_{max} = 37\%$, we have a BER $= 0.12$ which higher than the corresponding BCH codes correction ratio $t'/n' = 0.09$ (see table 15 for other examples).

Nevertheless, rank metric codes is less efficient than BCH codes when dealing against the second type because the maximum binary error rate obtained with cr_{max} is lower than the correction rate t'/n' associated to an equivalent BCH code. Plus, BCH codes are designed to handle random errors and hence, are robust against any type of content erasure distortions as long as the binary error rate is smaller than t'/n'.

Yet we are convinced rank metric codes remains a better choice in terms of error correcting codes especially for the case

Gabidulin			BCH	
$[n,k,t]$	cr_{max}	BER at cr_{max}	$[n',k',t']$	t'/n'
$16,5,5$	41%	0.11	$255,87,26$	0.10
$16,8,4$	42%	0.1	$255,131,18$	0.07
$16,11,2$	41%	0.04	$255,171,11$	0.04
$23,8,7$	37%	0.12	$511,175,46$	0.09
$23,12,5$	36%	0.10	$511,250,31$	0.06
$23,15,4$	39%	0.07	$511,340,20$	0.04
$32,10,11$	35%	0.13	$1023,348,87$	0.09
$32,16,8$	35%	0.10	$1023,513,57$	0.06
$32,21,5$	35%	0.07	$1023,688,36$	0.04

Figure 15: Error correcting code parameters. Each row are approximatively equivalent parameters between BCH and Gabidulin codes in terms of code length and dimension. n,k,t are respectively the code length, dimension and maximum error rank allowed. We have the same for BCH parameters n',k',t' (generated with SageMath).

of Gabidulin codes which are MRD. Even though BCH codes are optimal for random errors, they have parameter constraints because one may only choose the code length n such that $n = 2^m - 1$.

In this section, we have demonstrated that rank metric codes can be used to be resistant to content erasures by construction. Combined with the LQIM method on a block-based watermarking strategy in the spatial domain, we showed that the proposed method can be robust to several types of distortions under some parameters constraints and tradeoffs. Moreover, these new codes achieve slightly better robustness performances than BCH codes in some cases.

As explained in the last paragraph of section 4, rank codes are efficient against a particular error structure. The same conclusion is drawn from the previous section when adding the block decomposition against the second attack we proposed to study.

Some attacks have a random error structure which makes rank codes inefficient. For example, robustness results obtained by studying of JPEG compression and additive white gaussian noise showed that using these codes did not improve the watermark robustness at all. In this case, we have to use classical codes. However, a further investigation on the construction of these attacks may allow one to find an embedding to capture a good error structure for rank codes.

6. Conclusion

In this article, we introduced the concept of error structure. When errors have a random structure, usual Hamming codes such as BCH codes are well suited. However, it is no longer the case when the structure is particular. Then, we introduced a new type of error correcting codes for digital image watermarking which uses the rank distance instead of the usual Hamming distance. Gabidulin codes are one family of rank codes we chose to combine with the classical LQIM method. Rank metric offer many advantages due to the mathematical properties of the matrix rank such as invariances to binary flipping and columns/rows swapping. Moreover, if an error matrix e associated to a rank metric codeword has a particular form, $rk(e)$ is low.

First, we studied its robustness against luminance modifications. The application of rank codes provided good results but some errors still remain (denoted as spikes). Using a multi-decoding strategy, we enhanced the LQIM detector to obtain theoretical invariance against luminance modifications. To our knowledge, rank metric codes are optimal against this attack whereas classical Hamming codes are completely inefficient.

As a second study, we added to the previous method a block decomposition of the image instead of embedding information at random pixel locations. The image is divided into blocks where each block is associated to one bit of information. This embedding strategy takes advantage of the rank metric structure when dealing against various content erasure/slicing/copy-paste situations.

After studying the robustness of the proposed method, we showed that using Gabidulin codes allows us to handle errors more efficiently than BCH codes when distortions are maximized for a minimum error rank.

We are convinced these codes have a great potential in digital watermarking. The use of the rank metric is original and allows one to be robust against some image processings that are not handled by usual Hamming codes.

Even though combining a watermarking method with an error correcting code is not a new concept, there are new perspectives for rank metric. Theoretically, one has to study the error structure produced by the noisy channel in order to add robustness to the embedding. Nevertheless, we believe that the embedding method (and also the synchronization step) must be taken into account as we saw with the choice of LQIM method for the luminance attack and the block decomposition. A work perspective is to study other embedding techniques such as trellis coded quantization watermarking. Lastly, we also consider studying syndrome coding (already used in steganography) with rank metric.

References

[1] M. L. Miller, I. J. Cox, J.-P. M. Linnartz, T. Kalker, A review of watermarking principles and practices, Digital signal processing in multimedia systems (1999) 461–485.

[2] F. Cayre, C. Fontaine, T. Furon, Watermarking security: theory and practice, IEEE Transactions on Signal Processing 53 (10) (2005) 3976–3987. doi:10.1109/TSP.2005.855418.

[3] W. Abdul, P. Carré, P. Gaborit, Error correcting codes for robust color wavelet watermarking, EURASIP Journal on Information Security 2013 (1) (2013) 1. doi:10.1186/1687-417X-2013-1.

[4] D. Silva, F. R. Kschischang, On metrics for error correction in network coding, IEEE Transactions on Information Theory 55 (12) (2009) 5479–5490.

[5] P. Gaborit, O. Ruatta, J. Schrek, G. Zémor, New results for rank-based cryptography, in: International Conference on Cryptology in Africa, Springer, 2014, pp. 1–12.

[6] P. Delsarte, Bilinear forms over a finite field, with applications to coding theory, Journal of Combinatorial Theory, Series A 25 (3) (1978) 226–241.

[7] E. M. Gabidulin, Theory of codes with maximum rank distance, Problemy Peredachi Informatsii 21 (1) (1985) 3–16.

[8] E. M. Gabidulin, A fast matrix decoding algorithm for rank-error-correcting codes, Springer Berlin Heidelberg, Berlin, Heidelberg, 1992, pp. 126–133. doi:10.1007/BFb0034349.

[9] P. Loidreau, A Welch–Berlekamp Like Algorithm for Decoding Gabidulin Codes, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 36–45. doi:10.1007/11779360_4.

[10] B. Chen, G. W. Wornell, Quantization index modulation: A class of provably good methods for digital watermarking and information embedding, IEEE Transactions on Information Theory 47 (4) (1999) 1423–1443.

[11] P. Moulin, R. Koetter, Data-hiding codes, Proceedings of the IEEE 93 (12) (2005) 2083–2126. doi:10.1109/JPROC.2005.859599.

[12] P. Lefèvre, P. Carré, P. Gaborit, WATERMARKING AND RANK METRIC CODES, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2018), Calgary, Canada, 2018.
URL https://hal.archives-ouvertes.fr/hal-01771871

[13] F. Hartung, M. Kutter, Multimedia watermarking techniques, Proceedings of the IEEE 87 (7) (1999) 1079–1107.

[14] M. D. Swanson, B. Zhu, A. H. Tewfik, Transparent robust image watermarking, in: Image Processing, 1996. Proceedings., International Conference on, Vol. 3, IEEE, 1996, pp. 211–214.

[15] M. Holliman, N. Memon, Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes, IEEE Transactions on image processing 9 (3) (2000) 432–441.

[16] N. D. M. Jessica Fridrich, Miroslav Goljan, Further attacks on Yeung-Mintzer fragile watermarking scheme (2000). doi:10.1117/12.384997.

[17] B. Fridrich, M. Goljan, Protection of digital images using self embedding.

[18] B. Felgenhauer, F. Jarvis, Mathematics of sudoku i, Mathematical Spectrum 39 (1) (2006) 15–22.

[19] E. Russell, F. Jarvis, Mathematics of sudoku ii, Mathematical Spectrum 39 (2) (2006) 54–58.

[20] A. Aggarwal, M. Singla, Robust watermarking of color images under noise and cropping attacks in spatial domain, image 6 (9) (2011) 11.

[21] S. K. A. Khalid, M. M. Deris, K. M. Mohamad, Anti-cropping digital image watermarking using sudoku, International Journal of Grid and Utility Computing 4 (2-3) (2013) 169–177.

[22] M. S. Goli, A. Naghsh, Introducing a new method robust against crop attack in digital image watermarking using two-step sudoku, in: Pattern Recognition and Image Analysis (IPRIA), 2017 3rd International Conference on, IEEE, 2017, pp. 237–242.

[23] M. Kutter, Watermarking resistance to translation, rotation, and scaling, in: Multimedia Systems and Applications, Vol. 3528, International Society for Optics and Photonics, 1999, pp. 423–432.