Serdica J. Computing **2** (2008), 163–180

Serdica
Journal of Computing

Bulgarian Academy of Sciences
Institute of Mathematics and Informatics

IMPROVING THE WATERMARKING PROCESS WITH USAGE OF BLOCK ERROR-CORRECTING CODES

Thierry Berger, Todor Todorov

Abstract. The emergence of digital imaging and of digital networks has made duplication of original artwork easier. Watermarking techniques, also referred to as digital signature, sign images by introducing changes that are imperceptible to the human eye but easily recoverable by a computer program.

Usage of error correcting codes is one of the good choices in order to correct possible errors when extracting the signature.

In this paper, we present a scheme of error correction based on a combination of Reed-Solomon codes and another optimal linear code as inner code. We have investigated the strength of the noise that this scheme is steady to for a fixed capacity of the image and various lengths of the signature. Finally, we compare our results with other error correcting techniques that are used in watermarking. We have also created a computer program for image watermarking that uses the newly presented scheme for error correction.

ACM Computing Classification System (1998): D.1.7 – E.4 – I.4.0.

Key words: Watermarking, error-correcting codes, Reed-Solomon codes, software for watermarking.

1. Introduction. The proliferation of digitized media is creating a pressing need for copyright enforcement schemes that protect copyright ownership. Conventional cryptographic systems permit only valid key holders to access encrypted data, but once such data is decrypted there is no way to track its reproduction or retransmission. A digital watermark is intended to complement cryptographic processes. It is a visible, or preferably invisible, identification code that is permanently embedded in the data, that is, it remains present within the data after any decryption process [5].

In order to be effective, a watermark should be:

- **Unobtrusive**

  The watermark should be perceptually invisible, or its presence should not interfere with the work being protected.

- **Robust**

  The watermark must be difficult to remove. In particular it should be robust to:

  - Common signal processing

    These include digital-to-analog and analog-to-digital conversion, resembling, re-quantization, and common signal enhancement.
  - Common geometric distortions

    These include operations such as rotation, translation, cropping and scaling.
  - Subterfuge Attacks: Collusion and Forgery

    That is, watermark should be robust to combining copies of the same data set to destroy the watermarks.

- **Unambiguous**

  Retrieval of the watermark should unambiguously identify the owner.

Numerous papers [2, 6, 7, 8, 9, 17] mention the possibility of using error-correcting codes in order to improve the basic algorithms in terms of watermark robustness. This approach appears natural if one compares the watermarking problem with the transmission of a signal over a noisy channel. This model considers the image as a channel and the different attacks as a noise signal.

Error-correcting codes are widely used in channel coding, which makes them relevant for watermarking issues. Since the signal to noise ratio under which this particular communication system operates is very low due to imperceptibility constraints it is natural to envision the use of error-correcting codes. In this work we adopt a binary symmetric channel representing the watermarking process. Such a channel is completely defined by the probability of error (denoted p_{bsc}). A message transmitted through this channel may have some of its bits altered. We consider the signature to be received in error if one or more of its bits are in error. Also we are bounded with the capacity of the image. Capacity is the maximum amount of bits we can hide into an image without visual deterioration in image quality. To find the watermarking capacity of an image, one can apply the classical Shannon model for channel capacity [15]. A "watermarking channel" is modeled as a binary symmetric channel when the embedded information is demodulated with hard-decision to a binary code word. Expected values for the channel error probability in case of watermarking are in a wide range. They depend on the possible attacks and on the watermarking method. Values starts from 5% and can reach more than 40%.

As opposed to the "classical" channel coding applications where the noise signal can generally be efficiently modeled as a Gaussian noise, watermarking applications must take into account several attacks representing a wide range of noises of different natures. The Gaussian assumption is then no longer valid. In this context, it is very difficult to design a unique code that could meet the different requirements coming from different attacks.

That is why the use of error-correcting codes for watermarking is still a very open problem. It requires the design of error-correcting codes which are very compact and able to take account many different kinds of noise [11].

In this article we investigate the effectiveness of error-correcting codes in protecting watermark message.

Section 2 presents how Reed-Solomon can be used for the creation of a new technique for error protection in watermarking process.

Section 3 contains the results of computations of error probabilities for different coding strategies. There we compare the results of the proposed error-correcting scheme with other existing techniques.

Section 4 present a computer realization of a spatial domain watermarking algorithm that uses the proposed error-correcting scheme and is used for verification of theoretical results.

2. Error-correcting scheme for watermarking. RS codes are often used as "outer codes" in a system that uses a simpler "inner code". The inner code gets the error rate down and the RS code is then applied to correct the rest of the errors. We denote this encoding scheme with RS/Inn.

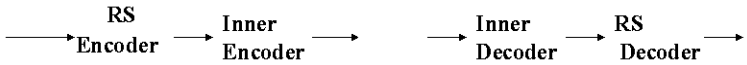

Fig. 1. RS/Inn encoder

In this note we apply a similar error-correcting scheme by using RS code with proper parameters as an outer code and other optimal linear code as an inner code.

When we have fixed capacity and payload we have to get the two codes in a way that they are with a good dimension according to the given payload and in the same way to fit the given capacity after the whole encoding is done.

In most of the cases the watermarking signature consists of binary symbols. This is why we could use so called "binary" RS codes. Let $RS(n,k)$ be a code over $GF(2^m)$. Every element in this field can be represented uniquely by a binary m-tuple, called m-bit byte. To encode binary data which such a code a message of km bits is first divided into k m-bit bytes. Each m-bit byte is regarded as a symbol in $GF(2^m)$. The k-byte message is then encoded into n-byte codewords based on the RS encoding rule. By doing this, we actually expand the RS code with symbols from $GF(2^m)$ into a binary (nm, km) linear code, called a "binary" RS code. Such a code is very effective in correcting bursts of bit errors, which the inner code can produce, as long as no more than t bytes are affected.

After the RS code is selected for the given case we proceed with the selection of the "inner code". From a practical point of view (computer encoding and decoding) it is more convenient to select the dimension of the inner code to be equal to m. This code will correct errors on bit level in each of the m-bit bytes. Also the length of the inner code depends on the parameters of the RS code because the final length of the encoded sequence should be less than the overall available capacity. So with fixed dimension and bounded length of the inner code we could search for the largest possible minimum distance. This could be done either in Brouwer's table [4] or in other sources.

In all considered cases the payload is a multiple of 8, because in practice we are working on byte level (8 bits). If we want to use a code which doesn't fit the payload we assume that necessary zero bits are added to the real signature.

Example: Let the capacity be fixed to 400 bits and the payload be 64 bits. We could choose $RS(17, 13, 5)$ code over $GF(2^5)$ for outer code and $(23, 5, 11)$ optimal code for inner code. We divide the payload into thirteen 5-bit bytes and encode them into seventeen 5-bit bytes. Each of these seventeen 5-bit bytes is encoded to a 23-bit byte. So we have the overall encoding length of $17 \times 23 = 391 < 400$ bits.

The $RS(17, 13, 5)$ is not a full length RS code. This is the better choice in this case because we have a limited small capacity. Because RS codes are MDS codes we could shorten $RS(31, 27, 5)$ code to $RS(17, 13, 5)$ code retaining the minimum distance to 5 and respectively the number of errors we can correct to 2.

What is the exact best choice for the parameter m, the length of the inner and outer code and the number of errors that they can correct depends on the every given case and we discuss this more in the next section.

3. A comparison of performances.

3.1. Computation of error probabilities. Here we will give formulas for the computation of the signature error probability of different error correcting strategies that we are comparing. For Repetition coding, BCH coding and Hybrid coding we are using formulas deducted in [2].

Repetition coding. Let us have a signature of length w repeated r times. The bit error probability after r repetitions is given by:

$$P_{\text{rep}} = \sum_{i=\frac{r}{2}+1}^{r} \binom{r}{i} p_{bsc}^{i} (1 - p_{bsc})^{r-i}$$

where p_{bsc} is the bit error probability in the binary symmetric channel, and C_r^i is the combinatorial expression. Consequently, the signature error probability, that is the probability of having at least one bit in error in the w bits of the watermark message is:

$$P_{\text{sig,rep}} = 1 - (1 - P_{\text{rep}})^w.$$

BCH coding. Let us consider $BCH(n, k, t)$ code where t is the number of errors it can correct.

An upper bound on the signature error probability can be calculated by computing the probability that t or more errors occur in the received word:

$$P_{\text{sig,code}} = \sum_{i=t+1}^{n} \binom{n}{i} p_{bsc}^{i} (1 - p_{bsc})^{n-i}.$$

Hybrid coding. As we have already explained it is more applicable to use repetition code as inner code and BCH code as outer in hybrid error correcting scheme. In this case the signature error probability is given with the next formula:

$$P_{\text{sig,hybrid}} = \sum_{i=t+1}^{n} \binom{n}{i} P_{\text{rep}}^{i} (1 - P_{\text{rep}})^{n-i}$$

where P_{rep} is defined in the section of computation of the error probability for repetition code.

RS coding. Here we compute the signature error probability for the newly presented scheme. It is quite similar to the formula for hybrid coding, but here we use different optimal linear codes as inner code and RS code as outer code:

$$P_{\text{sig,rs}} = \sum_{i=t+1}^{n} \binom{n}{i} P_{\text{sig,inn}}^{i} (1 - P_{\text{sig,inn}})^{n-i}$$

where $P_{\text{sig,inn}}$ is the signature error probability, which can be computed in a way that it is done in BCH coding section.

3.2. Results. First we should say that the capacity and the length of the mark values are very important for the experiments. The good results depend not only on the choice of the payload and the capacity that we can use to encode this payload in the possible way, but also what is the ratio between them. That is why the next presented results should be still precise to find what are the best choices for the inner and outer code and for what values for capacity and payload they performed best.

Next we will present the results for two specific channel error rates: 5% and 15%. On the following graphics one can see the performance for the known watermarking error correcting schemes that we present here. The results on the graphics are with averaged results for every capacity between 200 and 500 bits.

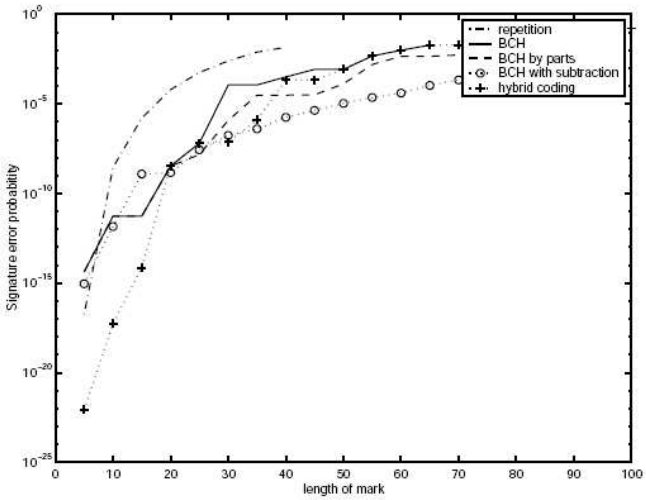

Fig. 2. Channel error rate 5% [17]

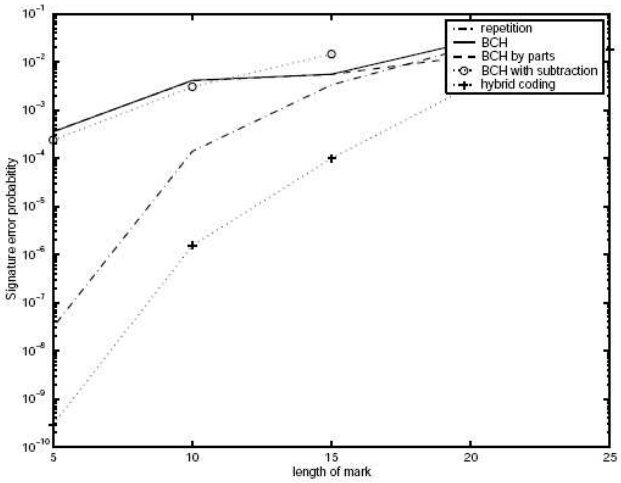

Fig. 3. Channel error rate 15% [17]

In the next table we give the results for the signature error probabilities for the same channel error rates but using the newly proposed technique. Up to now the results are only for the fixed capacity of 400 bits.

Channel error rate

Payload	5 %	15 %
8 bits	2.10^{-27}	2.10^{-8}
16 bits	3.10^{-19}	5.10^{-5}
32 bits	4.10^{-14}	3.10^{-2}
40 bits	6.10^{-14}	4.10^{-2}
56 bits	1.10^{-12}	7.10^{-2}
64 bits	6.10^{-11}	14.10^{-2}
128 bits	4.10^{-4}	—
256 bits	32.10^{-3}	—

Table 1. Performance of RS/Inn. encode scheme

It is clear that the RS code is a good choice when the payload is not too small or too near to the capacity. In the first case, when the payload is too small, we cannot fully use the ability of the RS code to correct block errors because we cannot split the short signature to blocks in effective way. In these cases BCH codes have near or even better performance. When the payload increases (32 and more bits) the RS coding has the best performance among all the examined coding schemes. When the signature's length becomes close to the capacity we again cannot fully use the RS coding abilities because if we do so there will be no capacity left for the inner code. The new scheme performs better than others in these cases but doesn't have a low enough error probability. When the channel error rate increases the performance of the new technique drops down but it still performs better than others for higher payloads. The last two values in that column of the table are omitted because they are too big and so useless in practice.

Finally we made a comparison of the different techniques to see which stands to much noise for different capacity, fixed 400 bits capacity and $P_{\text{sig}} \leqslant 0,01$.

Again the same tendency can be noticed, that the RS/Inn.code technique performs better for mid range payload values. Also important is the fact that this scheme gives relatively good results for big payloads like 128, 256 bits where other techniques are useless.

4. Watermarking with amplitude modulation. Embedding and retrieving information from other information is of basic importance in watermarking and is done by the stegosystems principles. The basic elements of a typical stegosystem for digital watermark are:

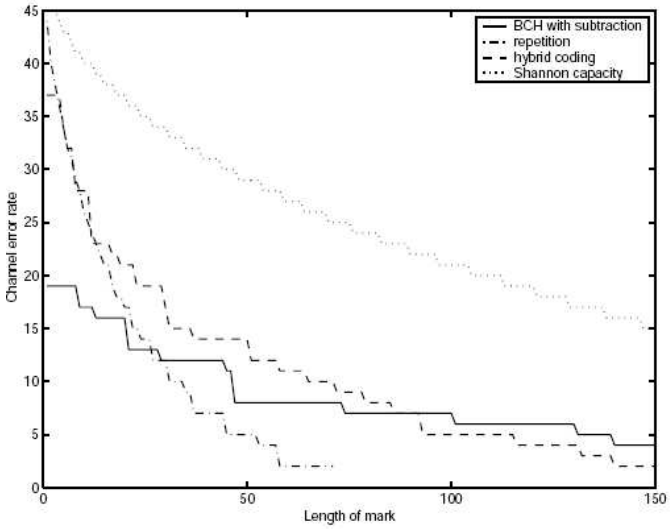

Fig. 4. Length to noise performance for other codings [17]

— precursory coder – structure for proper transforming of the secret message in order to embed it in the signal container (information sequence in which the message is put);

— stegocoder – structure for embedding the secret message in other data and reading its specialties -structure for watermark retrieving;

— stegodetector – structure for stegomessages presence determination;

— decoder – structure for secret message s restoring.

Before watermark embedding appropriate transformations are necessary so that it corresponds to the container. For example, if the container is an image, then the watermark must be a two-dimensional array of bits. All transformations are done by the precursory encoder. Calculation of the general Fourier transformations for the message and the container are done in it. That enables embedding in the spectral area and increases the stability of the watermark. An embedding key (K) is often used to increase the secrecy. Embedding and transforming messages in the container are done by the encoder. There are different methods for that, which depend on the containers character and will be referred

Payload	
8 bits	28 %
16 bits	21 %
32 bits	14 %
40 bits	14 %
56 bits	13 %
64 bits	12 %
128 bits	6 %
256 bits	4 %

Table 2. Length to noise performance for RS/Inn.code scheme

later. There are detectors for finding an existing watermark and for selecting it. In the first case detectors are possible with either hard or soft resolve. Metrics as Hamming distance and mutual correlation between the initial and delivered signal are used for choosing the proper resolve. When the initial signal is unknown, statistic methods are used.

Next we will describe single bit embedding and retrieving using watermarking with amplitude modulation. This could be easily generalized for multiple bits [13].

Let s be a single bit to be embedded in an image $I = (R, G, B)$, and $p = (i, j)$ a pseudo-random position within I. This position depends on a secret key K, which is used as a seed to the pseudo-random number generator. The bit s is embedded by modifying the blue channel B at position p by a fraction of the luminance $L = 0{,}299R + 0{,}587G + 0{,}114B$ as: $B_{ij} = B_{ij} + (2s - 1)L_{ijq}$ where q is a constant determining the signature strength. The value q is selected such as to offer best trade-off between robustness and invisibility. In order to recover the embedded bit, a prediction of the original value of the pixel containing the information is needed. This prediction is based on a linear combination of pixel values in a neighborhood around p. The sign of the difference between the prediction and the actual value of the pixel determines the value of the embedded bit.

$$B'_{ij} = \frac{1}{4c} \sum_{k=-c}^{c} B_{i+k,j} + \sum_{k=-c}^{c} B_{i,j+k} - 2B_{ij}$$

where c is the size of the cross-shaped neighborhood. To retrieve the embedded bit the difference δ between the prediction and the actual value of the pixel is

taken: $\delta = B_{ij} - B'_{ij}$ The sign of the difference δ determines the value of the embedded bit.

Also, robustness could be improved with the use of optimal error correcting codes. The method is steady to filtering, JPEG compression, geometrical transforms [13].

We create software realization of this watermarking algorithm by improving it with our error-correcting scheme. For development we use VC++ 6.0 and OpenSource library CxImage.

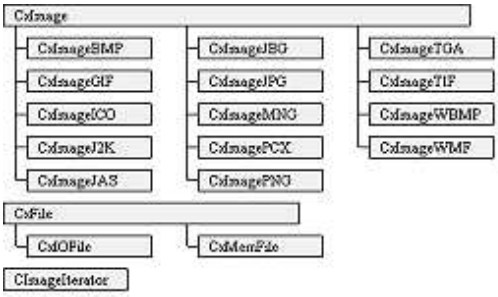

Fig. 5. Structure of the CxImage library

Next is the source code snippet for single bit embedding and retrieving:

int EmbedSingleBit(CxImage *img, int x, int y, unsigned char bitToEmbed, double signatureStrenght)

{

int bitResult = 0; double nB = 0; int r, g, b; COLORREF clr; RGBQUAD rq;

rq = img->GetPixelColor(x,y);

clr = img->RGBQUADtoRGB(rq);

b = GetBValue(clr);

g = GetGValue(clr);

r = GetRValue(clr);

nB = b + (2*bitToEmbed - 1) * signatureStrenght * GetLuminance(img, x, y);

b = (BYTE)nB;

img->SetPixelColor(x, y, RGB(r, g, (BYTE)b));

return bitResult;

}

```
int RetrieveSingleBit(CxImage *img, int x, int y, int crossSize){

int bitResult = 0; int i; int bik, bjk, bij; RGBQUAD rgq; COLORREF
pixel; double nBij; double delta = 0;
bik = bjk = 0;
for (i = -crossSize; i <= crossSize; i++)
{
      rgq = img->GetPixelColor(x + i, y);
      pixel = img->RGBQUADtoRGB(rgq);
      bik += GetBValue(pixel);
      rgq = img->GetPixelColor(x, y + i);
      pixel = img->RGBQUADtoRGB(rgq);
      bjk += GetBValue(pixel);
}
rgq = img->GetPixelColor(x, y);
pixel = img->RGBQUADtoRGB(rgq);
bij = GetBValue(pixel);
nBij = (bik + bjk - 2*bij)/(4*crossSize);
delta = bij - nBij;
return (delta > 0) ? 1: -1; }
```

Before embedding we should perform encoding according to the error-correcting scheme described in Section 2. After retrieving the data the decoding procedure will output the signature that is embedded in the image. This software has been used for securing the data in an information system [3].

5. Conclusion. We present a new error-correcting scheme that can be used in conditions of watermarking systems – short payloads in small available capacity. The technique combines Reed-Solomon codes as outer code and optimal linear code as inner code. We are still conducting experiments and comparisons with other error-correcting schemes but up to now we can conclude that the RS/Inn.code scheme performs better then others when the payload is not too small and the channel error-rate is not too high. We also create a watermarking software that uses amplitude modulation signing and the newly created error-correcting scheme. Experimental results produced by this computer program confirm the theoretical results.

Acknowledgment. This paper has been done partially during the stay of the second author in L'Equipe Arithmetique, Cryptographie, Codage at the Universite de Limoges. The author would like to thanks to the whole equipe of Arithmetique, Cryptographie, Codage for the productive environment and helpful discussions.

Appendix
Error-correcting codes

Basics

The object of an error-correcting code is to encode the data, by adding a certain amount of redundancy to the message, so that the original message can be recovered if not too many errors have occurred.

Definition 1. *A q-ary code is a given set of sequences of symbols where each symbol is chosen from a set $F_q = \{\lambda_1, \lambda_2, \ldots, \lambda_q\}$ of q distinct elements.*

The set F_q is called the alphabet and is often taken to be the set $Z_q = \{0, 1, 2, \ldots, q-1\}$. If q is a prime power we often take the alphabet F_q to be the finite field of order q.

Definition 2. *A binary code is a given set of sequences of 0s and 1s which are called codewords.*

Definition 3. *The (Hamming) distance between two vectors x and y of $(F_q)^n$ is the number of places in which they differ. It is denoted by $d(x, y)$.*

Definition 4. *Let F_q be the Galois field $GF(q)$, where q is a prime power, and let $(F_q)^n$ be the vector space $V(n, q)$. A linear code C over $GF(q)$ is a subspace of $V(n, q)$, for some positive integer n.*

If C is a k-dimensional subspace of $V(n, q)$, then we call it $[n, k, d]$-code, where n is length, k is dimension and d is the minimum distance of the code. Sometimes we denote it just $[n, k]$ code.

Definition 5. *We call an $[n, k, d]$-code optimal if it has the largest possible d for fixed n and k.*

Definition 6. *A communication channel is called q-ary symmetric channel if following assumptions are made about it:*

- *Each transmitted symbol has the same probability* $p\left(<\frac{1}{2}\right)$ *of being received in error.*
- *If a symbol is received in error, then each of the* $q-1$ *possible errors is equally likely.*

The binary symmetric channel is shown on the next figure:

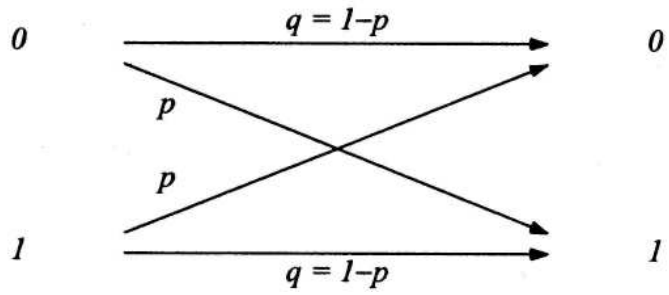

Fig. 6. Formal model for a binary symmetric channel [12]

Theorem 1. *A code* C *can detect up to* s *errors in any codeword if* $d(C) \geq s+1$ *and can correct up to* t *errors in any codeword if* $d(C) \geq 2t+1$ [8].

Repetition Coding

The simplest way to prevent errors is to repeat the watermark signature which is tantamount to spatial diversity reception. The signature of length w is repeated r times such that $r \times w \leqslant c$ is satisfied, where c is the embedding capacity of the image. Every bit is decided for separately using majority rule.

Repetition code is $[r, 1, r]$ code, so according to Theorem 1 it can correct up to $\left\lfloor \dfrac{r-1}{2} \right\rfloor$ errors.

BCH codes

Standard encoding with BCH codes. BCH codes are a large class of cyclic codes that include both binary and nonbinary alphabets. Binary BCH codes can be constructed with parameters (n, k, t), where n is the length of the codeword, k is the length of the data bits and t is the number of bit errors this BCH code can correct. Obviously one has $d = 2t + 1$, where $n = 2^m - 1$, $n - k \leqslant mt$, m and t being arbitrary integers.

If the whole w bit message will be transmitted via one BCH code, than one must satisfy the constrains $w \leqslant k$ and $n \leqslant c$.

Encoding by parts with BCH codes. To obtain more flexibility in embedding codewords in order to use all the available capacity the signature can be split into smaller parts and a separate BCH code can be used for each part. For example if we have 32 bits of payload and 500 bits of capacity we can use $BCH(255, 37, 45)$ and waste 245 of capacity. But if we divide the encoding process to three parts we can use $BCH(127, 15, 27)$ code for each part which also exceeds the number of correctable errors from 45 to $3 \times 27 = 81$.

Encoding with shortened BCH codes. Let $GF(2^m)$ be the finite field with 2^m elements, $0, 1, \ldots, n = 2^m - 1$. A t-bit error-correcting BCH code (n, k, t) is defined by a generating polynomial. The generating polynomial of any BCH code is only constrained by t and m. So for a BCH code (n, k, t), it is equivalent to $(n - b, k - b, t)$ defined by the same generating polynomial, where $b < k$ is any positive integer. In this way we can create a cross-section of the original code in order to shorten the code.

Hybrid coding. This refers to using a combination of repetition and BCH coding. There are two possibilities: BCH after repetition or repetition after BCH.

In practice, the first case is not useful, because the BCH decoder can only correct up to t errors. If the received codeword has more than t errors the BCH decoder fails and if it has less than t errors it corrects them all and there is no need of repetition.

The second method can be useful because the bit error rate of the received code is decreased by repetition and then BCH decoding can be applied [1, 14, 17].

Reed–Solomon codes

Reed–Solomon (RS) codes are nonbinary cyclic codes with symbols made up of m-bit sequences, where m is any positive integer having a value greater than 2.

Let α be a primitive element in $GF(2^m)$. This means that α is an element of $GF(2^m)$ such that each nonzero element of the field can be represented by a power of α. In these conditions for any positive integer $t \leqslant 2^m - 1$, there exists a

t-symbol error-correcting RS code with symbols from $GF(2^m)$ and the following parameters:

$$n = 2^m - 1$$
$$n - k = 2t$$
$$k = 2^m - 1 - 2t$$
$$d = 2t + 1 = n - k + 1$$

The generating polynomial for an RS code takes the following form:

$$g(X) = g_0 + g_1 X + g_2 X^2 + \ldots + g_{2t-1} X^{2t-1} + X^{2t}$$

where $g_i \in GF(2^m)$ and $g(x)$ has $\alpha, \alpha^2, \ldots, \alpha^{2t}$ as roots.

One of the most important features of RS codes is that the minimum distance of an RS(n, k) is $n - k + 1$. Codes of this kind are called "maximum distance separable codes" (MDS). RS codes achieve the largest possible code minimum distance for any linear code with the same encoder input and output block lengths.

Also Reed-Solomon codes have an erasure-correcting capability, ρ, which is:

$$\rho = d - 1 = n - k.$$

Simultaneous error-correction capability can be expressed as follows:

$$2\alpha + \gamma < d < n - k$$

where α in the number of symbol-error patterns that can be corrected and γ is the number of symbol erasure patterns that can be corrected.

There are many proposed algorithms for efficient encoding and decoding of RS codes [16].

REFERENCES

[1] Bastug A. Watermarking capacity improvement by low density parity check codes. Master of Science Thesis, 1999.

[2] Baudry S., J.-F. Delaigle, B. Sankur, B. Macq, H. Maitre. Analyses of error correction strategies for typical communication channels in watermarking. *Signal Processing*, 2001, 1239–1250.

[3] Bogdanova G., T. Todorov, Ts. Georgieva. New approaches for development, analyzing and security of multimedia archive of folklore objects. *CSJMol* **2**, 2008 (to appear).

[4] `http://www.win.tue.nl/ aeb/voorlincod.html.`

[5] Cox I. et al. Secure Spread Spectrum Watermarking for Multimedia. *IEEE Transactions on Image Processing*, 1995.

[6] Darmstaedter V. et al. A block based watermarking technique for MPEG-2 Signals: Optimization and validation on real digital TV distribution links. In: Proceedings of the European Conference on Multimedia Applications, Services and Techniques, 1998.

[7] Delaigle J.-F. et al. Digital images protection techniques in a broadcast framework: Overview. In: Proceedings of European Conference on Multimedia Applications, Services and Techniques, Louvain-la-Neuve, Belgium, 1996, 711–728.

[8] Delaigle J.-F., C. De Vleeschouwer, B. Macq. Watermarking Using a Matching Model Based on Human Visual System. Ecole thematique CNRS GDR-PRC ISIS: Information Signal Images, Marly le Roi, 1997.

[9] Hernandez J. R. et al. The impact of the channel coding on the performance of spatial watermarking for copyright protection. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, vol. **5**, 1998, 2973–2976.

[10] Hill R. A first course in coding theory. Calendar Press, Oxford, 1986.

[11] Katzenbeisser S., F. Peticolas. Information hiding techniques for steganography and digital watermarking. Artech House, 2000.

[12] Koucheryavy E. Error control: Lecture, 2005.

[13] Kutter M. Digital Signature of Color Images using Amplitude Modulation. *Journal of Electronic Imaging*, April 1998, 326–332.

[14] MacWilliams F. J., N. A. Sloane. The theory of error-correcting codes. North-Holland publishing company, Amsterdam, New York, Oxford, 1977.

[15] Ramkumar M., A. N. Akansu. Information Theoretic Bounds for Data Hiding in Compressed Images. In: IEEE Second Workshop on Multimedia Signal Processing, 1998, 267–272.

[16] Skallar B. Digital Communications: Fundamentals and Applications. Prentice-Hall, 2001.

[17] Zinger S. et al. Optimization of watermarking performances using error correcting codes and repetition. In: Proceedings of Communications and Multimedia Security Conference, 2001.

Thierry Berger
L'Equipe Arithmetique, Cryptographie, Codage
Universite de Limoges
XLIM DMI, UMR CNRS 6172
123 avenue Albert Thomas
87060 LIMOGES CEDEX, France
e-mail: `thierry.berger@unilim.fr`

Todor Todorov
Institute of Mathematics and Informatics
Bulgarian Academy of Sciences
P.O. Box 323, 5000 Veliko Tarnovo, Bulgaria
e-mail: `todor@moi.math.bas.bg.`

Received April 10, 2008
Final Accepted June 6, 2008