Serdica J. Computing **8** (2014), No 4, 309–326

Serdica
Journal of Computing

Bulgarian Academy of Sciences
Institute of Mathematics and Informatics

MODULAR DIGITAL WATERMARKING FOR IMAGE VERIFICATION AND SECURE DATA STORAGE IN WEB APPLICATIONS

Svetozar Ilchev, Zlatoliliya Ilcheva

Abstract. Our modular approach to data hiding is an innovative concept in the data hiding research field. It enables the creation of modular digital watermarking methods that have extendable features and are designed for use in web applications. The methods consist of two types of modules – a basic module and an application-specific module. The basic module mainly provides features which are connected with the specific image format. As JPEG is a preferred image format on the Internet, we have put a focus on the achievement of a robust and error-free embedding and retrieval of the embedded data in JPEG images. The application-specific modules are adaptable to user requirements in the concrete web application. The experimental results of the modular data watermarking are very promising. They indicate excellent image quality, satisfactory size of the embedded data and perfect robustness against JPEG transformations with pre-specified compression ratios.

1. Introduction. During the last years the World Wide Web became one of the most powerful media for the search and interactive exchange of infor-

ACM Computing Classification System (1998): C.2.0.
Key words: modular digital watermarking, steganography, data hiding, JPEG, web applications, image verification, secure data storage.

mation. Besides, more and more often the World Wide Web is referred to as a "medium with multiple content forms", i.e. as multimedia.

Modern web browsers already have the capability of displaying the basic multimedia elements: text, still images, animation, audio, video and virtual reality. Combining these elements in web applications brings web pages to life, thus expanding the potential uses of the Web and making it a more exciting place to explore.

Until recently, the control of the access to Internet data and the protection of this data against illegal appropriation and alteration fell within the domain of cryptography [18]. With the advance of multimedia information on the Internet, cryptographic methods cannot always offer optimal protection and cannot guarantee the security of web applications.

Unlike other digital data, multimedia data contains high redundancy and irrelevancy. Some multimedia data processing operations like compression, filtering, etc. are often applied without affecting the semantic content, the security level and the authenticity of the data. Classic cryptographic algorithms treat the result of such operations as data which has been illegally manipulated. In fact, cryptography can only protect and authenticate the semantically irrelevant binary representation of the multimedia data instead of its perceptual semantic content.

The change of the World Wide Web in the last decade – towards the integration of diverse multimedia content – has led to a new research direction with regard to the protection of multimedia: Digital data hiding (shortly data hiding). Nowadays, data hiding encompasses two major research areas: digital steganography and digital watermarking) [4], [13].

Digital steganography studies the encoding and the detection of secret messages transmitted over digital communication platforms. Steganographic methods hide the presence of an arbitrary digital message by embedding it into other digital media making its discovery by potential attackers very difficult.

Digital watermarking, on the other hand, focuses mainly on the protection of intellectual property rights and the authentication of digital media. Digital watermarking methods hide information in digital media like steganographic methods but the hidden information pertains to the digital medium itself and contains information about its author, its creating circumstances, its buyer, the integrity of its content, etc.

With regard to web-based scenarios, there are various data hiding application areas which benefit from digital steganography or digital watermarking (Fig. 1).

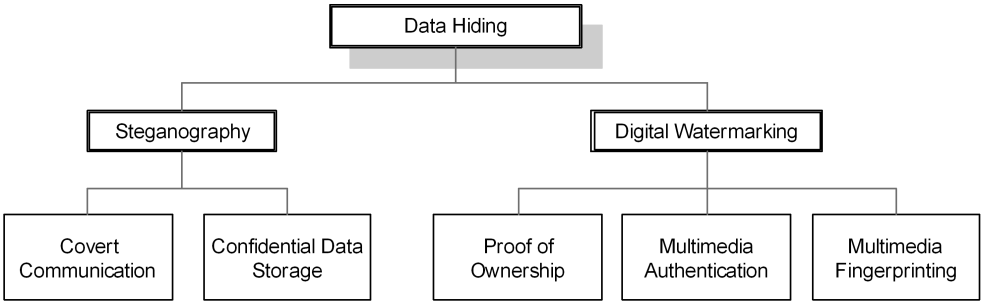

Fig. 1. Data hiding application areas

The inherent openness and volatility of the World Wide Web in combination with the rapid changes in the contemporary social, business and technological environment lead to frequent modifications in user requirements. For this reason, data hiding methods in any of the application areas should be adaptable to new user requirements and they should be capable of incorporating new features, while still providing certain basic functionality expected by the end user. These circumstances have led the authors to the development of modular data hiding methods.

Data hiding methods can work on different types of multimedia data. Our research is focused mainly on digital images as the host medium. The term "host medium" refers to any digital medium, in which steganographic and digital watermarking methods hide information. The main reason for our focus lies in the immense popularity of images in the World Wide Web. The obtained results can be easily generalized for video streams as well.

The potential benefits of the modular data hiding approach and the corresponding modular data hiding methods proposed by us lies in the flexibility and adaptability of the methods to changing user requirements. The subdivision of the methods into relatively independent modules allows the creation of new methods in a quick and easier manner in comparison with traditional data hiding methods which have a monolithic structure.

The next section of the paper presents an overview of some data hiding methods which may be relevant for applications on the Internet. Next, we introduce our modular data hiding approach and a new modular method for digital watermarking purposes. The last two chapters describe some potential applications and some experimental results and evaluations.

2. Data hiding methods for JPEG images. A typical requirement for data hiding methods is that they must ensure the preservation of the data

embedded into the image after compression. This requirement is not met by all data hiding methods. Data hiding methods are divided into two general groups: spatial domain and transform domain methods [13], [16], [17].

Spatial domain methods work directly on image pixels. Most often they fall into the group of the so-called least significant bit (LSB) methods, which modify the LSBs of image pixels. Traditionally, these methods do not ensure the robustness of the hidden information against compression.

Transform domain methods, on the other hand, work on the output of various mathematical transforms performed on the host image. As lossy image compression is often based on such mathematical transforms, these methods perform well when the hidden information has to withstand image compression. Three widely used transforms are the discrete Fourier transform (DFT), the discrete cosine transform (DCT) and the discrete wavelet transform (DWT). DCT data hiding methods are based on the same mathematical transform used by JPEG standard. They can be further divided into two groups: DCT methods which are not based on the JPEG specification and DCT methods which follow the JPEG specification in detail.

JPEG is one of the most important image file formats on the Web. Its universality and good compression ratio have made it a preferred choice for storing color and grey-scale images. For this reason, it is important for the hidden information not to be destroyed by JPEG transformations. There are three basic types of transformations: compression, decompression and recompression. The robustness against all three transformation types is important for the flexible use of data hiding algorithms in Web. Compression is used to reduce the size of newly created images and to make them readable by browsers. Decompression is used to extract the image content, so that it can be displayed, modified or saved in a different image format. Recompression is used mainly to reduce the image size. It is often applied to existing JPEG images prior to their distribution through web-based channels (sending by e-mail or uploading to a web site).

One of the first practical JPEG-based data hiding methods was JSTEG, developed in 1993 by Derek Upham [22], [23]. It is a steganographic method which hides arbitrary binary data by replacing the LSBs of the DCT coefficients of JPEG images. Another data hiding method for digital watermarking was proposed by Zhao and Koch, Fraunhofer institute, Darmstadt in 1995 [27]. It hides one bit per DCT block by creating a special relationship among the elements of a set of three DCT coefficients. In the literature, we can further trace methods developed by Cox, et al. [3], Wu and Liu, Princeton University [**?**], Lin and Chang, Columbia University [14], Niels Provos, University of Michigan [20], An-

dreas Westfeld, University of Dresden [24], Chang, et al. [1] and others. Jessica Fridrich, Binghamton University, USA and her research group proposed several digital watermarking methods based on DCT transforms [5], [6], [7], [8]. Some recent data hiding algorithms rely on a technique called Quantization Index Modulation (QIM). More information about them can be found in [26], [15], [12].

It is important to underline that none of the cited methods satisfy the requirements for extensibility and adaptability to users' needs imposed by the modern web reality. They are monolithic solutions designed for a concrete application area with specific requirements. In addition, they are usually not fully robust against all three types of JPEG transformations.

3. Modular Data Hiding and Modular Digital Watermarking Method.

3.1. General scheme. The modular data hiding enables the development of extendable data hiding methods. Each method consists two building modules – a basic module and an application-specific module. The basic module provides method features which are closely related to the host image file format. In this case, the image file format is JPEG and the basic module provides the robustness against JPEG transformations. The application-specific module is adaptable to user requirements and can be tailored to different application areas. The relationship between the basic and the application-specific modules is illustrated in Fig. 2.

Modular data hiding methods are specifically designed to answer the need for flexibility and adaptability to varying user requirements on the modern Web. They are based on combinations of modules from a pool of basic and application-specific modules designed to offer end users a variety of data hiding methods with different features. In this way, every end user (or automated client application) may assemble, via a combination of different modules, its own data hiding method that has the exact desired features. An important point to consider is the compatibility of the method interfaces with current web-related technologies such as web services. For this purpose, we need an easy to use dedicated interface for the assembling of a concrete data hiding method from existing basic and application-specific modules.

3.2. JPEG Standard overview. Before we describe the functioning of our JPEG-based basic module, we need to explain briefly some details about the popular JPEG still image standard. For more information, refer to [9], [19]. The JPEG image standard involves mathematical transformations, a lossy compression

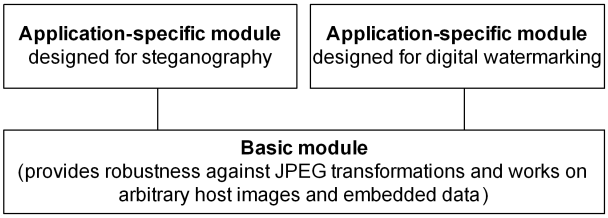

Fig. 2. General scheme of the modular data hiding

step and several lossless compression stages. It is capable of achieving compression ratios of 10:1 and even more, depending on the degree of lossy compression.

The image before compression is always represented as a matrix of pixels. The first step is to convert each pixel value from the RGB color space to the YCbCr color space and then subtract 128 from each value in order to shift the values to the interval $[-128; 127]$ and thus ensure a slightly better performance of the DCT transform. This is achieved by means of the following linear transformation:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} -128 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.1687 & -0.3313 & 0.5 \\ 0.5 & -0.4187 & -0.0813 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Then, the image is divided into 8×8 blocks and each image component (Y, Cb or Cr) of each block is processed separately by the following two-dimensional DCT transform:

$$C_{k,l} = \frac{p_k p_l}{4} \sum_{n=0}^{7} \sum_{m=0}^{7} P_{m,n} \times \cos\left[\frac{(2m+1)k\pi}{16}\right] \times \cos\left[\frac{(2n+1)l\pi}{16}\right],$$

where

$$p_k = \begin{cases} \frac{1}{\sqrt{2}}, & \text{for } k = 0 \\ 1, & \text{for } k \neq 0 \end{cases}, \quad p_l = \begin{cases} \frac{1}{\sqrt{2}}, & \text{for } l = 0 \\ 1, & \text{for } l \neq 0 \end{cases}, \quad k, l \in [0; 7]$$

and $C_{k,l}$ denotes the DCT coefficients and $P_{m,n}$ denotes the image pixels of the block.

The JPEG standard takes advantage of the properties of the human eye, which is more sensitive to low frequencies than to high ones. Therefore, high frequencies can be compressed to a very high degree or discarded altogether without significant perceptual loss of quality while low frequencies undergo very little

lossy compression. This selective compression is achieved by the employment of a special quantization table, the values of which determine the compression level. Finally, a combination of differential, run-length and Huffman coding of the DCT coefficients is performed to reduce further the size of the JPEG file.

The popularity of the JPEG image format is due mainly to its excellent compression ratios and its universal applicability in the domain of photos and multi-purpose scanned images.

3.3. Basic module robust against JPEG transformation. The basic module described in this section has the major objective of providing robustness against JPEG compression, decompression and recompression. The module itself performs two main activities – a preliminary analysis of the image and the actual embedding of the binary information into the image.

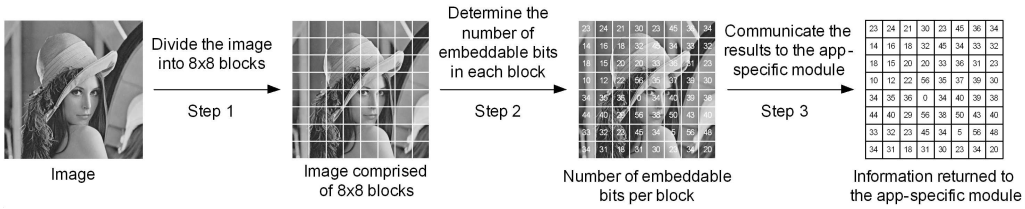

Fig. 3. Basic module – image analysis

The image analysis (Fig. 3) divides the image into blocks (of size 8x8 according to the JPEG specification), determines the *maximum* amount of data embeddable into each block and forwards this information to the application-specific module. Then, the application-specific module, according to the particular application, specifies the *actual* binary information, which the basic module will encode into each image block.

The encoding of the binary information into the host image is depicted in Fig. 4. First, the binary information is encoded (embedded) into the image. Then, stepwise, the robustness against JPEG decompression, JPEG compression and JPEG recompression is guaranteed by means of a special algorithm working in close connection with the JPEG specification. Finally, some image post-processing is performed to improve the quality of the resulting host image.

Achieving accurate data embedding for JPEG images is not trivial because the JPEG specification provides room for discretion and there are limitations in the integer implementations of color spaces and transformations. As the image processing libraries used to edit the image after the embedding has taken place are not known in advance, a good knowledge of the popular variations of the

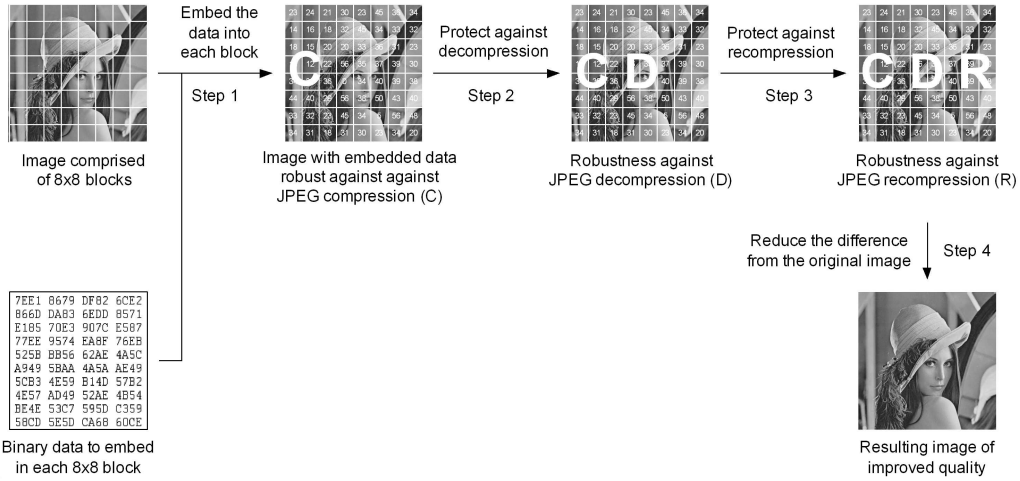

Fig. 4. Basic module – data encoding

standard is required [8], [19].

We consider the compression, decompression and recompression as separate JPEG transformations and ensure robustness against each one sequentially. We embed the data into the least significant bits of the DCT coefficients and this guarantees implicitly the robustness against compression as well as a relatively good image quality for a given size of the embedded data [3]. The other two JPEG transformations may cause loss of data. This problem can be alleviated to a certain extent by the use of error-correcting codes but they cannot ensure a perfect accuracy of the data extraction. In the case of the decompression, there are two causes for data loss:

- the integer implementations of the transformations
- the limited integer representation of the color spaces – often using subsets of the set of natural numbers $S = \{0, 1, 2, \ldots, 255\}$.

The integer implementation of the transformations may in some rare cases cause the flipping of some of the embedded data bits. A calculation of a chain of forward and inverse image transformations (stages 1 to 3 of the algorithmic description below) resolves this problem.

The limited integer representation of the color spaces poses another problem related to the lossy compression, which achieves information reduction by mapping multiple similar variations of a pixel block onto the same DCT block. If

some of the original pixel values before the compression are close to the boundaries of the set S, some pixel values obtained after the decompression could drop out of the boundaries of the set. Such values are usually changed to either 0 or 255. Because of this change, the new pixel block may now map onto a different DCT block thus losing the embedded data. We pre-scale the original pixel values so that the decompression always yields valid values (stages 4 and 5 of the algorithmic description). The adjustment coefficient α is set empirically at stage 0. It may be changed dynamically by small amounts, which was not necessary so far in our tests.

Considering the image recompression, we want to make the data robust against recompression with $\forall Q^{(j)} | Q^{(j)}_{k,l} \leq Q_{k,l}$ for $\forall k, l \in \{0, 1, 2, \ldots, 7\}$, where j is an iteration index and Q is the quantization table corresponding to the user-specified JPEG quality ratio.

Initial tests show a good but not perfect intrinsic robustness of the coefficients against JPEG recompression with smaller quantization coefficients depending on Q, the amount of the data and the image textures. To improve the robustness of the embedding, we employ a modified Quantization Index Modulation technique [2]. By varying the quantization step $q_{k,l}$, a coefficient $C_{k,l}$ may be mapped onto multiple new values $C^{(i)}_{k,l}$ which represent the same embedded data (stages 7 and 8 of the algorithmic description). The coefficient β is initialized to 1. If for a given $Q^{(j)}$ and $\forall (k, l) \in Z_{data}$, $C^{(i)}_{k,l} = E^{(i,j)}_{k,l}$, then the JPEG block is considered robust against recompression with $Q^{(j)}$. The quantization tables $Q^{(j)}$ depend on the concrete software implementations (e.g. the open-source IJG library, Adobe Photoshop, Paint Shop Pro, etc.).

3.4. Algorithmic description of the proposed method. We assume that the coefficients $C_{k,l}$ have already been calculated and data has been embedded into some of them. We define the set $Z_{data} = \{(k, l) | C_{k,l}$ contains embedded data$\}$, $k, l \in \{0, 1, 2, \ldots, 7\}$. Then, the algorithmic description can be described in the following way:

Stage 0. Perform initialization: $i = 1$, $u^{(0)} = 0$, $v^{(0)} = 255$, $\alpha = 1$.

Stage 1. Calculate $P^{(1)}_{m,n}$ from $C_{k,l}$ by performing the inverse JPEG DCT transform. [5], [6].

Stage 2. Calculate $C^{(1)}_{k,l}$ from $P^{(1)}_{m,n}$ following the JPEG standard.

Stage 3. If $\exists (k, l) \notin Z_{data} | C_{k,l} \neq C^{(1)}_{k,l}$, then for $\forall (k, l) \notin Z_{data}$, set $C_{k,l} = C^{(1)}_{k,l}$ and go back to stage 1.

Stage 4. Calculate $P^{(2)}_{m,n}$ from $C_{k,l}$ by performing the inverse JPEG DCT transform [5], [6].

Stage 5. If $\exists(m,n)|P^{(2)}_{m,n} \notin \{u^{(0)}, u^{(0)}+1, \ldots, v^{(0)}\}$, then for $\forall(m,n)|P^{(2)}_{m,n} < u^{(0)}$ calculate $s^{(i)} = \max\limits_{(m,n)|P^{(2)}_{m,n}<u^{(0)}} \left(\alpha \left| P^{(2)}_{m,n} - u^{(0)} \right|\right)$ and for $\forall(m,n)|P^{(2)}_{m,n} > v^{(0)}$ calculate $t^{(i)} = \max\limits_{(m,n)|P^{(2)}_{m,n}>v^{(0)}} \left(\alpha \left| P^{(2)}_{m,n} - v^{(0)} \right|\right)$. Set $u^{(i)} = u^{(i-1)} + s^{(i)}$, $v^{(i)} = v^{(i-1)} - t^{(i)}$. For $\forall(m,n)|P_{m,n} < u^{(i)}$, set $P_{m,n} = u^{(i)}$ and for $\forall(m,n)|P_{m,n} > v^{(i)}$, set $P_{m,n} = v^{(i)}$. Increase i by 1 and go back to stage 1.

Stage 6. Set $i = 0$, $\beta = 1$, choose a valid $j|\exists Q^{(j)}$ and for $\forall(k,l)$, set $C^{(0)}_{k,l} = C_{k,l}$.

Stage 7. Calculate $D^{(i,j)}_{k,l} = \left[C^{(i)}_{k,l} \times Q_{k,l}/Q^{(j)}_{k,l}\right]$ and $E^{(i,j)}_{k,l} = \left[D^{(i)}_{k,l} \times Q^{(j)}_{k,l}/Q_{k,l}\right]$

Stage 8. If $\exists(k,l) \in Z_{data}|C^{(i)}_{k,l} \neq E^{(i,j)}_{k,l}$, then set $C^{(i+1)}_{k,l} = C^{(i)}_{k,l} + (-1)^i[(i+1) \times q_{k,l}]$, where $q_{k,l} = \beta 2^{N_{k,l}}$ and $N_{k,l}$ is the number of data bits in $C_{k,l}$. Increase i by 1 and go back to stage 7.

Stage 9. Repeat stages 6 to 8 for $\forall(k,l) \in Z_{data}$ and $\forall j|\exists Q^{(j)}$.

3.5. Application-specific modules and digital watermarking module. The application-specific modules are responsible for the features of data hiding methods which are related to the concrete needs of the user – e.g. verifying the integrity of the image. They combine the information obtained from the image analysis from the basic module with the user-defined binary data which is embedded into the image (Fig. 5).

Transformations such as compression, encryption or the interweaving of error-correcting codes may be applied on the data. Then, it is distributed across the image blocks. This distribution is different for each application-specific module and depends on the data hiding features provided by the method. In the final step, the distributed data is forwarded to the basic module for the actual encoding.

A general scheme of the digital watermarking method which ensures the integrity of the image is shown in Fig. 6. The image is processed by the method before it leaves the sender. If it is modified while it is being communicated to the receiver, the method can both detect the modified image regions and extract the original embedded watermark, which in this case we will refer to as image signature.

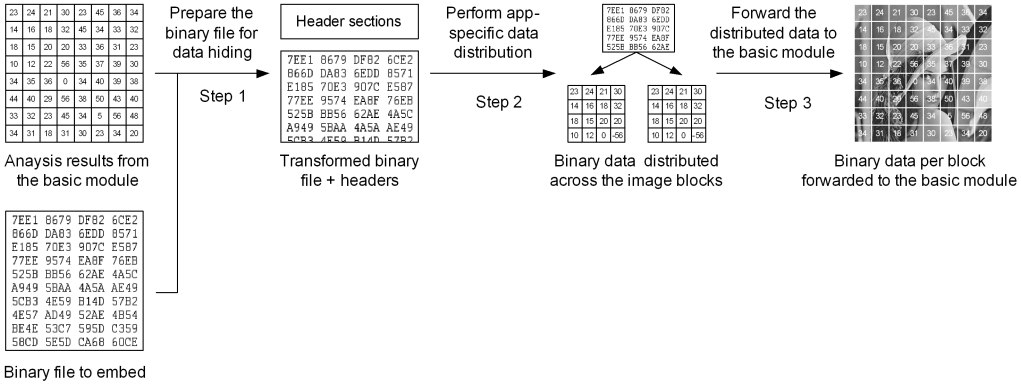

Fig. 5. Application-specific module

This is a blind digital watermarking method in the sense that the original image is not needed at the receiver [4], [13]. For web-based application scenarios, this provides a much needed flexibility but decreases the overall image quality and the maximum length of the embedded signature.

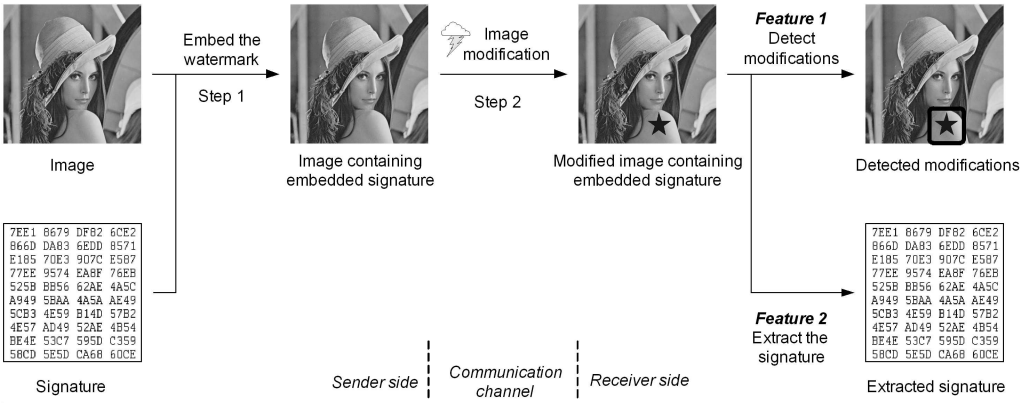

Fig. 6. Digital watermarking method – general scheme

The application-specific module for digital watermarking divides the image into multiple large areas called macroblocks. Each macroblock contains a full copy of the signature. If a macroblock is modified, the embedded signature is changed. The receiver can later detect these changes and identify modified image blocks.

Let us illustrate the signature recovery by means of the image shown in Fig. 7. A copy of the signature is embedded into each of the four macroblocks (A, B, C and D). The signature bits are distributed uniformly across the image

blocks xi ($x \in \{a, b, c, d\}$, $i \in N$), so that each block xi contains B bits.

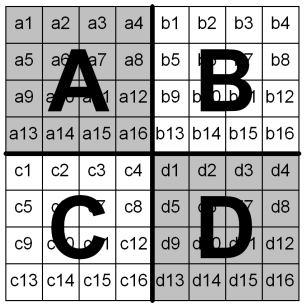

Fig. 7. Image consisting of 64 blocks grouped into 4 macroblocks

At the receiver, the signature copies are extracted from each macroblock and compared to one another. If the image modifications are not too severe, most signature copies will be identical to the original signature. The remaining signatures will contain differences whose locations correspond to the modified blocks of the macroblock containing the modified signature. In our example, each one of the four signature copies is distributed uniformly across the 16 blocks of the corresponding macroblock (Fig. 8). Each block contains $B = 2$ bits of the signature. The signature consists of a total of 32 ($= 1 \times 2$) bits.

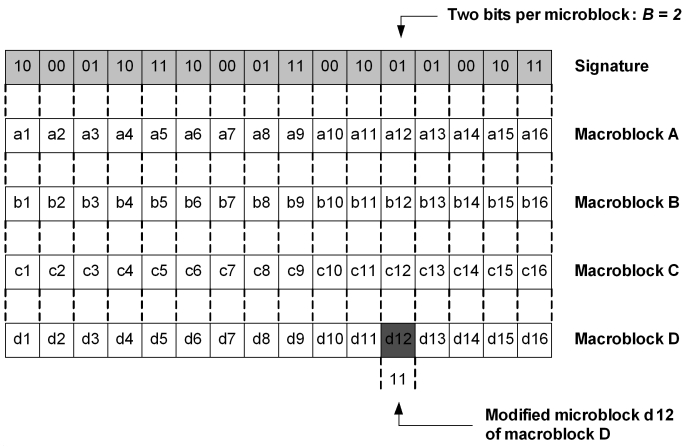

Fig. 8. Detection of modifications during the signature decoding

Let us assume that macroblock D has been modified. In this case, the signature copies extracted from macroblocks A, B and C are identical. The signature copy extracted from macroblock D contains a difference located in the

$d12$ block. Consequently, block $d12$ (see Fig. 7) has been modified after the signature embedding. The original signature can be recovered and corresponds to the signature copies extracted from macroblocks A, B and C.

Because the basic module provides robustness against JPEG transformations, the embedded signatures will not change if any JPEG-related processing takes place. This type of modification will remain undetected by the algorithm described above. It represents a set of acceptable image transformations that do not change the semantic meaning of the image. This is a "semi-fragile" property of the modular digital watermarking which results from the combination of its basic and application-specific modules.

4. Potential web-based applications. Some web-based applications of the modular digital watermarking in the form of a data hiding certification service have been presented in previous papers, e.g. the phishing prevention for bank portals, [10] and the improvement of the legal use of multimedia content in web-based societies [11]. Other use cases encompass the protection of multimedia created and used by artists, photographers, photojournalists and news agencies, the integrity verification of security footage and medical photos, the storage and communication of sensitive private data. They may be of interest to private individuals, corporations or government administrations and institutions.

5. Experimental results and evaluations. We consider three main criteria: the image quality after embedding, the size of the signature and the number and type of features provided by the methods. An optimization with regard to all three criteria simultaneously cannot be achieved.

The modular digital watermarking method has been implemented using the Microsoft .NET framework and native C++, where the execution speed is essential. In the tests, we use 760 image-data pairs. The data set encompasses 10 binary and text data files (archives, executable files, documents, plain text, small images). Their average length is 39 bytes. Each file is embedded into each image of the image set, which is composed of 76 image samples (test images, color photos, scanned images, cartoons, grayscale and black-white images). Their average dimensions are 606×583 pixels. We employ the open source IJG library to conduct tests with different JPEG quality ratios.

A short summary of the results is shown on Table 1. The method achieves accurate data embedding in JPEG files for the tested quality ratios r of 70, 80 and 90. All images keep their embedded data after JPEG transformations with

Table 1. Experimental results

Parameters				Verification results				
Image-data pairs	JPEG quality ratio	Average image size [pixels]	Average data size [bytes]	Robustness against JPEG transformations			Average MSE	Average PSNR [dB]
				Compression [%]	Decompression [%]	Recompression [%]		
760	70	606×583	39	100	100	100	26.3	38.3
760	80	606×583	39	100	100	100	15.0	39.9
760	90	606×583	39	100	100	100	7.0	42.4

any $r_i \geq r$. Error correction is not applied. The average image quality measured as a peak signal-to-noise ratio (PSNR) is between 38.9dB and 42.2dB [21]. It is very similar to the quality achieved by the JPEG compressing algorithm itself and better than many of the data hiding classic monolithic methods. The processing time for the embedding is less than one minute per pair. The PSNR increases with an increase of the quality ratio r, which is in accordance with the higher image quality provided by the JPEG algorithm for higher values of r. In addition, for high values of r, the algorithm needs fewer iterations, fewer coefficient changes and less time.

In comparison with existing monolithic digital watermarking methods,

Fig. 9. Detection of modified image regions

the modular digital watermarking method delivers excellent image quality and embeds a fairly large amount of data.

Image modifications from 1 pixel up to one third of the image can be identified with very high probability. Fig. 9 shows the software prototype implementation. The JPEG photo is an image of a green garden which contains an embedded signature of 25 bytes. It is extracted successfully without any errors. A sketch of a green tree has been drawn in the bottom right area of the image after the signature embedding. There are 63 modified image blocks of size 8×8 pixels. They are identified by the algorithm and indicated by red (dark) rectangular borders. In this case, the information about the modifications is communicated visually directly to the user, but it may also be submitted to an automated system for further processing or storage.

6. Conclusion. The modularity of data hiding methods is an innovative and desirable property with regard to web-based application scenarios. It makes possible the creation of extendable modular data hiding methods suitable for the Internet. Due to the opportunities for code reuse and the easy adaptability to varying user requirements, the new methods can be developed and put into use swiftly and at an affordable price.

The digital watermarking method combines a relatively complex application-specific module with a basic module robust against JPEG transformations. The basic module achieves the error-free retrieval of the embedded data after JPEG transformations. End users may perform common JPEG transformations on the images without affecting the embedded data. The application-specific module handles the signature headers, the optional encryption, compression and error-correction and processes the image blocks and macroblocks. The method makes possible the detection of modified image areas and the reliable recovery of the embedded signatures. It is suitable for scenarios which need a combination of tamper detection with proof of ownership or fingerprinting.

One major advantage is that the method does not confirm the presence of a pre-specified watermark but it extracts a previously unknown signature from the image provided that the image modifications are not too severe. This property makes it more versatile as the only information the decoder needs to extract the signature is the image itself. The user-defined binary signatures may contain arbitrary information appropriate for the particular scenario and do not influence the performance and functionality of the method.

Future research and development will consider the creation of a library of modules. New basic modules are needed to provide robustness against the appli-

cation of other popular lossy image formats such as GIF or JPEG 2000. More sophisticated application-specific modules may provide support for new steganographic or digital watermarking functionality, e.g. the ability to recover modified multimedia areas.

REFERENCES

[1] Chang C., T. Chen, L. Chung. A Steganographic Method Based on JPEG and Quantization Table Modification. *Information Sciences – Informatics and Computer Sciences*, **141** (2002), No. 1–2, 123–138.

[2] Chen B., G. Wornell. Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. IEEE Transactions on Information Theory, IEEE Computer Society, New York, **47** (2001), No 4, 1423–1443.

[3] Cox I., J. Kilian, T. Leighton, T. Shamon. Secure Spread Spectrum Watermarking for Multimedia. *IEEE Transactions on Image Processing*, **6** (1997), No 12, 1673–1687.

[4] Cox I., M. Miller, J. Bloom, J. Fridrich, T. Kalker. Digital Watermarking and Steganography. 2nd ed., Morgan Kaufmann Publishers, 2008.

[5] Fridrich J., M. Goljan. Protection of Digital Images Using Self Embedding. Symposium on Content Security and Data Hiding in Digital Media, New Jersey Institute of Technology, May 14, 1999. (Online) `http://www.ws.binghamton.edu/fridrich/publications.html`, January 2010

[6] Fridrich J., M. Goljan, R. Du. Lossless Data Embedding—New Paradigm in Digital Watermarking. *EURASIP Journal on Applied Signal Processing,* **2002** (2002), No 2, 185–196.

[7] Fridrich J. Image Watermarking for Tamper Detection. In: Proceedings of the IEEE Int. Conference on Image Processing (ICIP), Chicago, Oct. 1998. (Online) `http://www.ws.binghamton.edu/fridrich/publications.html`, 2010

[8] Fridrich J. Steganography in Digital Media: Principles, Algorithms, and Applications. Cambridge University Press, 1st. ed., 2009.

[9] Hamilton E. JPEG File Interchange Format. (Online) `http://www.w3.org/Graphics/JPEG/jfif3.pdf`, January 2010

[10] Ilchev S., V. Ilchev. Modular data hiding for improved web-portal security. In: Proceedings of the 13th International Conference on Computer Systems and Technologies (CompSysTech'12), ISBN 978-1-4503-1193-9, Ruse, Bulgaria, 2012, 187–194. (Best paper award) doi: 10.1145/2383276.2383305.

[11] Ilchev S., Z. Ilcheva. Protection of Intellectual Property in Web Communities by Modular Digital Watermarking. In: Proceedings of the IEEE Signature Conference on Computers, Software and Applications (COMPSAC 2011), 35th IEEE Annual Computer Software and Applications Conference Workshops, Munich, Germany, 2011, e-ISBN 978-0-7695-4459-5, Print ISBN 978-1-4577-0980-7, 374–379. doi: 10.1109/COMPSACW.2011.69, INSPEC 12288790.

[12] Izadinia H., F. Sadeghi, M. Rahmati. A New Steganographic Method Using Quantization Index Modulation. In: Proceedings of the Int. Conference on Computer and Automation Engineering (ICCAE), Bangkok, 2009, 181–185.

[13] Katzenbeisser S., F. Petitcolas. Information Hiding Techniques for Steganography and Digital Watermarking, 1st ed., Artech House, 2000.

[14] Lin C., C. ChangSemi-Fragile. Watermarking for Authenticating JPEG Visual Content. In: Proceedings of the SPIE Int. Conference on Security and Watermarking of Multimedia Contents II, Vol. **3971**, January 2000, San Jose, California, USA. doi:10.1117/12.384968

[15] Li Q., J. Cox. Using Perceptual Models to Improve Fidelity and Provide Resistance to Valumetric Scaling for Quantization Index Modulation Watermarking. *IEEE Transactions on Information Forensics and Security*, **2** (2007), No 2, 127–139.

[16] Lin E., J. Delp. A Review of Data Hiding in Digital Images. In: Proc. of the Multimedia and Security Workshop (ACM Multimedia'99), Orlando, Florida, 1999, 25–29.

[17] Lu C. Multimedia Security: Steganography and Digital Watermarking Techniques for Protection of Intellectual Property. 1st ed., Idea Group Publishing, PA, USA, 2004.

[18] Mao W. Modern Cryptography: Theory and Practice. Prentice Hall PTR, New Jersey, USA, 2003.

[19] Pennebaker W., J. Mitchell. JPEG Still Image Data Compression Standard. 1 st ed., Van Rostrand Reinhold, New York, USA, 1993.

[20] Provos N. OutGuess – Universal Steganography. (Online) http://www.outguess.org, January 2010

[21] Rao K., P. Yip. The Transform and Data Compression Handbook. 1 st. ed., CRC Press, 2001.

[22] Upham D. JSteg. (Online) http://zooid.org/~paul/crypto/jsteg/, January, 2010

[23] Wayner P. Disappearing Cryptography. 2nd ed., Morgan Kaufmann Publishers, 2002.

[24] Westfeld A. F5 – A Steganographic Algorithm. In: Proc. of the 4th Int. Workshop on Information Hiding, Lecture Notes in Computer Sciences, Vol. **2137**, Springer, 2001, 289–302.

[25] Wu M., B. Liu. Watermarking for Image Authentication. In: Proc. of the IEEE Int. Conference on Image Processing, Vol. **2**, Chicago, Illinois, 1998, 437–441.

[26] Yu Y., C. Chang, Y. Hub. Hiding Secret Data in Images via Predictive Coding. *Pattern Recognition*, **38** (2005), No 5, 691–705.

[27] Zhao J., E. Koch. Towards Robust and Hidden Image Copyright Labeling. In: Proceedings of the IEEE Workshop on Nonlinear Signal and Image Processing, Neos Marmaras, Greece, 1995, 452–455.

Svetozar Ilchev
Institute of Computer and Communication Systems
Bulgarian Academy of Sciences
Acad. G. Bonchev str., Bl.2
1113 Sofia, Bulgaria
e-mail: svetozar@ilchev.net

Zlatoliliya Ilcheva
Institute of Computer and Communication Systems
Bulgarian Academy of Sciences
Acad. G. Bonchev str., Bl.2
1113 Sofia, Bulgaria
e-mail: zlat@iccs.isdip.bas.bg

Received June 6, 2014
Final Accepted December 8, 2014