INFORMATICA, 2005, Vol. 16, No. 1, 75–92

Tamper-Proof Image Watermarking, Based on Existing Public Key Infrastructure

Geruta KAZAKEVIČIŪTĖ
Dept. of Graphical Systems, Vilnius Gediminas Technical University
Saulėtekio 11, LT-10223 Vilnius, Lithuania
e-mail: geruta@delfi.lt

Eugenijus JANUŠKEVIČIUS
CAAD Laboratory, Dept. of Architecture, Vilnius Gediminas Technical University
Trakų 1/26, LT-01132 Vilnius, Lithuania
e-mail: ejs@ar.vtu.lt

René ROSENBAUM, Heidrun SCHUMANN
Institute of Computer Graphics, Dept. of CS, University of Rostock
Albert Einstein 21, D-18059 Rostock, Germany
e-mail: rrosen@informatik.uni-rostock.de, schumann@informatik.uni-rostock.de

Received: August 2003

Abstract. The paper describes methods for generating image watermark for asymmetric key based watermark embedding and detection scheme in wavelet domain. The proposed method combines DWT-based watermarking and the ability to verify watermark using reliable methods of asymmetric coding. The watermark scheme is developed to be directly incorporated into JPEG2000 image coding standard, while it can be used in existing image coding standards. In presented watermarking method the application of public key infrastructure is emphasized . We use PGP (Pretty Good Privacy) public key management scheme as the element for verification of sender's identity and also place of storing and retrieval of watermark detection keys. The most public key watermarking schemes use new, independent and cryptologically unverified keys. The reuse existing key verification and exchange infrastructure provides us cryptologically strong and verified keys.

Key words: public watermarking, public key, image hash, wavelets.

1. Introduction

Image watermarking has two main purposes: to protect the author, assuring no other person can claim to be the author of the work; and to protect the customer, assuring he has the same image he was interested in. A customer in today's world is not only a collectioner or movie distribution company; it can be a physician, examining the photo of patient's brain for surgery or a geologist, searching for a new materials in the photo images from the satellites.

There are even more situations, when the need of accurate and reliable visual information is vital. Authenticity of the information must be assured not only physically securing image storage and transmission, but also by protecting the digital media from tampering or altering. There are methods to assure the trustiness of constant digital data, like text, computer programs or databases, which is hard or impossible to change without loosing its meaning. Digital media is not only a stream of digital data. It also has audio-visual properties, which are less sensitive to data alteration than pure digital data, and, besides this, they are invisible when altered. Resizing a picture changes the data stream significantly, while visual properties of the image remain almost the same. On the other hand, replacing part of the image with another one, changes visual properties, but without knowing the image was modified, it is impossible to tell.

To resolve the issue of data authentity, various cryptographical schemes are used. The most secure among them nowadays seem to be the schemes using assymetric cryptography, like X.509 or PGP. However, these schemes can not be directly used for digital media, because they are designed to protect the digital information and does not allow even the smallest change. If we want to use existing assymetric cryptography to prove image authentity, either watermarking or cryptography must be adopted to each other.

Another important aspect is compression of digital data. Wavelet compression, allowing compression rates $1:10 \ldots 1:50$ without visible loss of image quality is the best method of compression known nowadays. Wavelet compression, both lossy and lossless, is the standard type of compression in JPEG2000 image compression and storage standard.

The aim of our investigations is to create a robust method of public watermarking, which could be used for any digital image and could be directly incorporated in JPEG2000 image compression standard.

The paper is organized as follows:

- Section 2 describes the works of other authors, concerning problems of asymmetric watermarking and image authentity;
- Section 3 describes the requirements for a fragile watermarking in JPEG2000 image coding standard;
- Section 4 describes the developed watermarking scheme;
- Section 5 presents and discuss the experimental results of testing the scheme;
- Section 6 summarize the results and propose directions for future research.

2. Related Works

The first idea to obtain a kind of asymmetric watermark known as ”partly known key” was introduced by Hartung and Girod (Hartung and Girod, 1997). It is a modification of spread spectrum watermarking: the person receiving the watermarked signal gets only a part of the watermark key. The recipient can check the presence only of this part of the watermark. The weak point is he can also remove the corresponding part of the water-

mark. Although the other parts of the watermark remains unaltered, a public watermark detector can not detect the whole watermark any more.

Van Schyndel, Tirkel and Svalbe introduced the first autocorrelation scheme, based on Legendre sequence insertion in the signal (Schyndel *et al.*, 1999). The Legendre sequence is invariant in a Fourier transform, therefore it correlates with its conjugate Fourier transform. The private key is the embedded Legendre symbol and the public key is the length of the sequence.

A variation on the autocorrelation scheme was introduced by Eggers in (Eggers *et al.*, 2000). Here the watermark is an eigenvector of a transformation matrix. The eigenvector correlates with its transformation by the matrix. The private key is an eigenvector of the transformation matrix and the public key is the transformation matrix itself.

The main problem of these schemes is that methods used for generating public and private keys, as we know, were not ever proved to be either cryptologically secure or computationally expensive to reverse. Therefore it may exist a way to reveal the missing watermark key from the key pair.

In order to obtain public key, cryptographists use one-way functions with trapdoors (the secret key). These functions are computationally expensive or impossible to invert without knowing the secret values of transformation and they are easy to invert knowing these values. Furon and Duhamel tried to reuse this principle (Furon and Duhamel, 2000). Instead of a one-way trapdoor function, they use a signal processing one-way function, the power density spectrum (PDS). The result of this function does not allow a perfect reconstruction of the signal.

Craver (Craver, 1999) uses a zero knowledge protocol to prove the presence of a watermark in the signal. The secure cryptographic primitive used is a zero-knowledge proof based on the isomorphism of graphs. This scheme does not really solve the industrial problem of the asymmetric watermark because it requires a lot of data to prove the existence of the watermark and it needs a lot of computations and exchanges between two parties – the prover and verifier.

A zero knowledge interactive proof can be transformed into a signature scheme (Fiat and Shamir, 1987). We lose the zero-knowledge property, when such transformation is applied, but it is possible to verify the signature off-line now. On the other hand, if somebody removes the attached signature, watermark can't be verified, even if there is a watermark embedded.

Our proposal is to integrate the cryptographic data inside the watermark. By using cryptography as a subset of watermarking, we can tightly link the payload of the watermark to the signal to be watermarked. This can solve the problem of a new watermarking but the problem of the removal of watermark is still open. We must mention the problem of interaction of all components of the system, as the integration of new components in a protocol must be crafted carefully, or it is likely to lead to new attacks on the system.

3. Requirements for a Fragile Watermarking Scheme, Considering JPEG2000 Image Compression Standard

This section covers requirements for a fragile watermarking scheme. These requirements are of several origins: the requirements of fragile watermarking, image constrains and constrains of other systems that our watermarking scheme uses.

To create really secure watermark for the image, it must fulfill some basic requirements (Wofgang and Delp, 1999; R. Venkatesan *et al.*, 2000):

1. It must depend on the image we are watermarking;
2. It must be unique for every image;
3. It must depend on time and date the image is watermarked.

To fulfill these requirements, watermarking scheme should use some image's property to construct the information, uniquelly defining the image. Usually this information is called "digest" or "hash". Cryptography uses "classical" verified hashes, like SHA1, MD5 or RIPEMD160. These functions were constructed to prevent even smallest alterations of digital data. Digital images can be intentionally or unintentionally altered and these alterations may be not visible without having the source image (ex. de-noising, blurring, sharpening, cropping); these transformations do not change visual perception of the image, but they change digital data significantly. Hash functions for digital images must capture visible properties of images, but not their digital data (del Bimbo *et al.*, 1998; Jacobs *et al.*, 1995). Also, hash function should be somehow "lossy", allowing small alterations to image data, without affecting the resulting hash significantly. On the other hand, if image was modified significantly, image hash must indicate it.

To ensure additional level of randomness for watermark, time and date of watermarking should be used to generate the watermark. Time stamp is the standard element of assymetric cryptography.

3.1. *Requirements for Tamper-Proof Watermarking*

The tamper-proof watermarks can be divided in two types:

- "fragile" watermarks allow to detect any modification of the content, they acts like digital signatures;
- "semi-fragile" or "content-based fragile" watermarks allow to detect and localize modifications of the content changes, while permitting content – preserving processing.

This last category is the most meaningful in terms of functionality and really brings additional features compared to digital signatures. Semi-fragile watermarks are based on features extracted from the image, such edges or objects shapes. These image characteristics are either directly embedded as watermark messages or serve as a seed to generate a watermark pattern.

The following features are desirable to construct an effective tamper-proof watermarking scheme (Min Wu and Bede Liu, 1998):

1. Determination whether an image has been altered or not;

2. Integration of authentication data with host image rather than as a separate data file;
3. Invisibility of the embedded authentication data under normal viewing conditions;
4. Localization of the alteration witch was made on the image;
5. Storage of the watermarked image in lossy compression format, or more generally, to distinguish moderate distortion that does not change the high-level content vs. content tampering.

Semi-fragile watermarks must be robust against one group of attacks and fragile against other attacks (Voyatzis and Pitas, 2000):

1. Fragile in the case of image modifications. Image modification that affects the original image content integrity is cause of watermark distortions and its verification failure. Generally, local image modifications affect image content (e.g., object insertion and extraction). Therefore, the watermark should be very sensitive to such modifications. In that case the detection algorithm should localize the tampered regions and give a negative authenticity answer. High-performance protection demands that watermark can reveal any slight image modifications.
2. Robust in the case of transmission noise and compression. Robustness may be required in some special cases that include modifications that don't harm the original image content, e.g., high-quality compression and necessary insignificant modifications to incorporate the product in a multimedia environment.

In many cases these requirements do not state clear a boundary between what is allowed and what is not. On the other hand, some non - destructive image processing operations, like high compression in DCT domain, alters image and may trigger false alarms.

3.2. *Image-Based Constrains*

Images may be subdivided into two categories: monochrome and multi-spectral images. Monochrome images contain only one band of information, usually brightness, whose values (bit depth) typically are defined by 8 bits, although images with bit depth of 16 or 24 bits are common in some fields. Multi-spectral images usually have more than one band of information, where each band corresponds to a different component, like color, hue, lightness, saturation, brightness. Currently we are dealing only with monochrome images, having in mind the ability to proceed multi-spectral images band-by-band, as a set of monochromes images.

The main requirement any watermarking scheme must satisfy is the requirement of invisibility. No invisible watermark can change the image to such extents that they become visible to human eye. The extents of this requirement is different for watermarking schemes. Usually it is more strict if an original image is required for watermark detection.

Other requirements are set by image storage standards. As we are working with Jpeg2000 image compression standard, we will analyse it more deeply.

Typical image processing pipeline in Jpeg2000 coder is very simple:

1. Color space conversions. Image is converted either to YUV or RGB color space.
2. Tiling. Image is subdivided into tiles. Every tile is processed individually further on.

3. Discrete wavelet transformation, using either reversible 5/3 or irreversible 9/7 transformation kernels.
4. Quantization using dead-zone quantizier.
5. EBCOT coding.
6. Code-stream organization.

For now we must mention that image tile is the main component in Jpeg2000 image compression standard. Every image transmission is based on transmission of required tiles. The watermark, to be detectable in any stage of image transmission, should not cross tile's boundary. There are no special requirements for image tile, except its length and width should fit into four-byte value.

The first step where some of the image's information is lost is DWT transformation using irreversible 9/7 kernel, but due to the next Jpeg2000 processing stage, we can pay no attention to it.

To make compression more effective, quantization is used in the next stage. At this point DWT coefficients are converted into quantization indexes. Quantization step is pre-calculated for every subband of DWT transformation in such way, that the result will not create visible distortions in the restored image. To make the result more effective, DWT coefficients, belonging to range $[-\Delta Q \dots \Delta Q]$ (where ΔQ is quantization step), are set to zero, thus making first quantization zone twice wider the others. De-quantization is performed according to formula $\omega_{i,j} = (Q_{i,j} + 1/2) * \Delta Q$, where $\omega_{i,j}$ is DWT coefficient in the position (i, j) and $Q_{i,j}$ is quantization index in the position (i, j) (Taubman and Marcellin, 2002).

The EBCOT coding stage encodes all bits, starting from the most significant one, into codestream. It is important to notice, that up to now the Jpeg2000 coding was processed in byte level, but now the EBCOT coding switches to bit-planes. During this stage we have no idea which coefficient is beeing processed. The coding is rather complex process (Taubman and Marcellin, 2002) and for us only a few moments are important:

1. Image tile may be subdivided into smaller regions – tile parts. Codestream truncation may appear only at a tile part's boundary.
2. This is a three-pass process, during which we get arithmetically endoded codestream.
3. EBCOT uses statefull arithmetic coding for optimizing perfomance.
4. There is a way to get raw codestream from lower bit-planes by switching encoder to "lazy mode" coding.

Resulting distortion is precalculated for every possible truncation point and is saved for future reference.

The last – codestream organization – stage constructs quality layers according to the pre-calculated codestream truncation points and is responsible for writing them to file.

As we see, there is no way to alter codestream after the EBCOT stage because of arithmetic coding as altering the code will result massive data loss when decoding. We still have some possibilities to insert the watermark (Su and Kuo, 2003; Noda *et al.*, 2003):

- Insert it after wavelet transformation. In this case we must assure our modifications will go to the next quantization index. To accomplish this, we can use the next approach:
- Modify quantization indexes. In this case we must remember that, depending on ΔQ, these indexes accumulate much more energy than wavelet coefficients.
- Insert watermark into bit planes.

As JPEG2000 image coding standard allows lossy image compression, we must take into account the fact that the lowest coefficients in LH, HL and HH subbands may be lost during quantization procedure. This is very serious problem, because LH, HL and HH subbands contain mostly low coefficients. Due to this fact only a small part of total

Table 1
Watermark capacity of DWT subbands

Image name		"Lena"		"Baboon"		"Watch"	
Image size,px		512×512		512×512		1024×768	
Matrix in 5th level, pixels		16×16		16×16		24×32	
Matrix in 5th level, bytes		256		256		768	
Subband	Cut-off, % of max value	Capacity, bytes	Capacity, %	Capacity, bytes	Capacity, %	Capacity, bytes	Capacity, %
HL (H)	10	92	35.93	156	60.93	307	39.97
29.27							
	20	47	18.35	99	38.67	183	23.82
18.26							
	30	26	10.15	62	24.21	102	13.28
11.01							
LH (V)	10	168	65.62	146	57.03	242	31.51
16.52							
	20	99	38.67	70	27.34	113	14.71
10.14							
	30	73	28.51	41	16.01	68	8.85
6.09							
HH (D)	10	109	42.57	173	67.57	244	31.77
13.44							
	20	70	27.34	112	43.75	98	12.76
8.99							
	30	39	15.23	68	26.56	48	6.25
7.25							

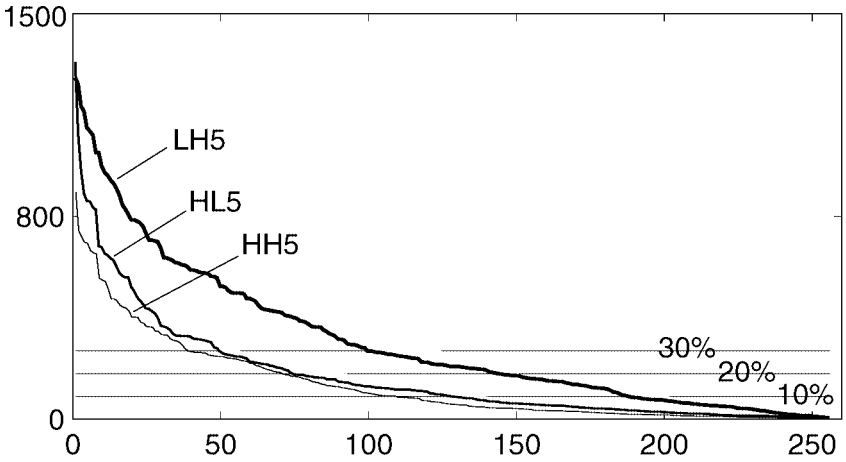

Fig. 1. Typical image capacity for watermarking in different subbands (image "Lena") Vertical axis shows coefficient's value.

number of coefficients of HL, LH and HH matrices could be used for watermark embedding. The number of these coefficients is greatly image-dependent, as these matrices contain high-frequency information – mainly edges of the image. Our preliminary investigations show, that in the case of uncompressed image we can use coefficients up to 10% of maximal value in subbands HL, LH and HH (see Table 1 and Fig. 1). This gives up to 40% of total number of coefficients for hiding watermark. When the image is going to be compressed, we must raise the minimal value of coefficients in the subbands to 30% of maximal value or even higher, what gives 6–20% of available coefficients, depending on image type and subband. This gives the size of watermark – $52 \ldots 104$ bytes for a matrix of 16x16 pixels.

3.3. *Public Key Infrastructure*

Before analyzing our watermarking scheme in detail, we should review the basics of asymmetric coding, our proposal is based on.

Asymmetric coding is based on non-linear, asymmetric, parameter-driven functions, applied to message. Parameters for these functions are called keys. One key from the pair can be distributed and is called "public key", the other is kept secretly and is called "private key". Message, encoded by private key, may be decoded only by corresponding public key and vice versa. The asymmetric properties of transformations do not allow altering or faking trusted digital information, without knowing the pair of transformation keys.

The main idea of asymmetric coding is simplicity of calculations when the information needed for transformation is known, and complexity of brute-force attacks on the system, when information, needed for transformation is unknown. The time of attack on a system can be evaluated as $2^n T(n)$, where $T(n)$ is time for transformation of a mes-

sage of length n. In the case where $n > 1000$ bits, the task becomes computationally unresolvable nowadays.

Asymmetric, or public key cryptography is the most widely used cryptographic standard in digital communication. It combines possibilities of distribution of trusted content, and verification of authenticity of the distributed contents as well as the distributor.

There are two widely used types of asymmetric cryptography: X.509 and PGP. The keys from these systems are not interchangeable, but they are based on the similar principles, so there is no big difference which key management system to use in production. For test environment we have chosen PGP key management system, as there is no certificate authority (CA), providing free, long-term X.509 digital certificates. There exists a possibility for someone to create his own, stand-alone CA, using OpenSSL package from OpenSSL project, but a problem of exchange, validation and verification of keys still arise.

PGP – Pretty Good Privacy is message encryption, signing and signature verification protocol, developed by Philip R. Zimmerman. The binary PGP keys typically is $512 \ldots 2048$ bits long, depending on the intended usage of the key pair. Shorter keys are insecure, as they can be discovered using ”brute force” attacks in reasonable amount of time. Longer keys are more secure, but it takes a lot of computations to generate such keys. PGP keys are generated and managed by the user. This allows to create as many keys as one needs without depending on capabilities of any service provider and assures that the user is the only person, owning private keys from the key pairs. This raises some issues on the identity of the key issuer, which are discussed later in this paper.

Each user possesses an unlimited number of cryptographic key pairs, a private (secret) key and a corresponding public key. Only the owner of the private key is able to encrypt the message, while everyone, possessing corresponding public key is able to decrypt it; anyone, possessing key issuer's public key can encrypt a message to the issuer and only the issuer, having valid private key will be able to decrypt it. This assures the security of private communication between two parties, using open communication channels. Public keys are notarized or certified and can be distributed widely.

The process of ratifying the sender is handled by signing public keys or trusting (accepting) CA's public X.509 type key and automatically trusting all keys, issued by this CA. While issuing X.509 key pair the receiver must prove her/his identity to the CA, providing valid and accurate personal identity data. Usually, the CA is also running a key server for searching, publishing and revoking keys it has issued.

The model of trusting PGP keys is based not on trusting CA, but trusting other person's decisions. The receiver can sign someone's public PGP key K'_{Public} by private key K_{Priv}, thus certifying the public key K'_{Public} presented is really owned by the user, presenting the key. As this is the most important decision in the key management, usually users are verifying each other's identities for not to make mistake. A collection of signed (trusted) keys is called ”keyring”. If receiving person trusts the user, key she/he signed, she/he can trust other users, whose keys were signed by the user the key she/he just signed. To have some control over what is trusted, the ”Trust Level” value is introduced, which allows the signer of the key to rate other users decisions. For example, if we know

a user does not care much while signing keys, we can assign her/him the lowest level of trust and only his key will be trusted, but no other key she/he has signed. On the other hand, if we know a user cares a lot while signing keys, we can assign the highest level of trust to her/him and trust any key the party has signed or any trust relations that are chained via this party. The network of trust relations is called "Web of Trust". Due to security reasons, usual chain of trust relations is limited to 3 to 5 levels of trust. So even without personally signing sender's public key, the receiver can still trust the message received, if she/he can chain trust relations to the sender via "Web of Trust". To have access to someone's PGP public key and keyring, public key servers are used. The main task of these servers is to provide reliable and secure source of public keys and keyrings. They also provide capability of "publishing" public keys and keyrings, thus assuring high level of cooperation between different individuals.

In our scheme PGP public key infrastructure is the element for verification of sender's identity and also place of storing and retrieval of watermark detection keys.

4. Implementation of Watermarking Scheme

4.1. *Generation of Watermark*

We use $(w\mathrm{x}h)$ size blocks of LL_5 subband of DWT transform and calculate the mean value of each block to obtain the image digest block. This algorithm is known as the block-based hash function (BBHF) technique (Wofgang and Delp, 1999). Also, a separate digest may be obtained for each row and each column of an image. This is known as row-column hash function technique (Wofgang and Delp, 1999). Finally, we sign or encrypt hash H of image with private key $K_{Private}$ of image owner to get watermark W. To test our watermarking scheme we have chosen a PGP key management system and created a key ID 0xAC2483FB, fingerprint "876E DE8F 420E F3EE AF5A 5535 7567 CFC7 AC24 83FB". This key can be found on public PGP key servers, like *http://wwwkeys.eu.pgp.net*.

4.2. *Embedding and Detection of Watermark*

The watermark is embedded in the discrete wavelet transform (DWT) domain of the image. As DWT is a part of JPEG2000 image coding pipeline, we can inject the watermark at the moment image or image tile is passing DWT, so time factor of DWT decomposition is not very important for us. JPEG2000 image coding standard implements DWT using (5,3) filter for lossless compression and the bi-orthogonal (9,7) filter for lossy compression (Taubman and Marcellin, 2002).

The problem is to find appropriate place for watermark embedding. The edge, because of peculiarities of human's visual system, masks surrounding areas, so these areas are most suitable for watermark embedding (Langelaar *et al.*, 2000). Wavelet transformation separates high-frequency signal, like edges of an image from medium and low frequency signal, like constant shades, into separate subbands. Also, wavelet decomposition is based on subband coding, so every next level of DW transformation is twice

smaller in size than previous one. This means, that pixels being side-by-side in 5th level of DWT decomposition, will be separated by 31 $(2^n - 1)$ pixels in the original image. This will be quite far away from the edge. These two problems does not allow inserting watermark near the edges: as pointed out earlier, coefficients near edges in LH, HL and HH subbands are almost zeros and may be quantized towards zero when compressing the image, what is not desirable for us. There is still possible to embed watermark into LL subband, but significant changes in this subband will be treated as edges and will migrate to appropriate subbands (either LH, HL or HH). So the only place to embed watermark are the most significant coefficients of every of LH, HL and HH subbands of selected level of decomposition. Inserting actually three watermarks at a time gives us possibility to evaluate the influence of DWT, inverse DWT and image processing operations to the watermark. This also allows to select the best watermark survived, referring the results of cross-correlation between these watermarks. We embed the watermark using multiplicative embedding formula, proposed by Cox (Cox *et al.*, 1997).

$$I_i^w = I_i * (1 + \alpha * W_i), \tag{1}$$

where I_i^w is i-th watermarked byte of appropriate subband, I_i is i-th byte of appropriate subband of original image and W_i denotes i-th byte of watermark sequence.

While applying inverse DWT, we get the watermarked image. The correlation C between watermarked and original image depends on the value of α. We set this parameter individually to every image, to get the resulting value of PSNR (Peak Signal-to-Noise Ratio) $40 \ldots 45$dB, what gives value of C between $0.9800 \ldots 0.9950$. It's not recommended to lower PSNR below 35dB, as artifacts in the watermarked image become noticeable even without the original image.

Discrete wavelet transformation has property of multi-resolution. As DWT is applied to watermarked image to test the presence of watermark, changes in subbands we made only at selected level of DWT decomposition while watermarking, due to redistribution of image energy during second DWT, are present in all levels, starting from the first. This means that even without any attacks on image, perfect reconstruction of plain watermark is hardly possible in DWT domain. To resolve this issue, some assumptions were used:

1. Watermark was quantized to predetermined steps. After signing or encrypting image's hash, watermark's values are in printable ASCII character range – ASCII values $32 \ldots 127$. Due to this, we can look at watermark as unsigned integer (UINT) values.
2. Error correction bits are introduced. We use BCH coder with 7-bit codeword and 4-bit message length. This coding allows to correct 1 bit of error in the message. Using longer, up to 15 bits, codeword would allow to correct up to 3 errors in the message. The price is longer watermark. After signing the hash, length of the watermark is 99 bytes. Error correction makes it 2–4 times longer. Due to this reason, we must embed watermark not into the LH_5, HL_5 and HH_5 bands, but to make inverse DWT a level or two higher and embed watermark into LH_4, HL_4 and HH_4 or even LH_3, HL_3 and HH_3 bands of DW transformation.

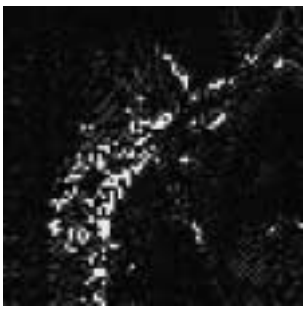

Fig. 2. Watermark in HH_3 subband of 'Lena-bw' image.

For now we assume that the original image is available for watermark recovery, thus intentionally, for research purposes, making public watermarking scheme private.

Inverted and modified embedding formula is used to extract watermarks $W_{LH,HL,HH}$ from LH, HL, HH coefficients of an image.

$$W_i = \left| floor\Big(\frac{I_i^w - I_i}{\alpha * I_i} + 0.5\Big)\right|. \tag{2}$$

The next step is verification of watermarks $W_{LH,HL,HH}$ against hash, obtained from the watermarked image. This stage must give answers to the questions:

- either image is watermarked?
- when it was watermarked?
- who is the watermarker?
- was the image changed after it was watermarked?

The decision of whether watermark is present or no, is made using verification of hash H' of the watermarked image against extracted watermarks and correlations between every key, extracted from the watermarks of HL, LH and HH subbands. Positive answer from the verification procedure assures the image was not changed after it was watermarked. Even with negative verification result, high correlation rates between extracted keys and keyring indicates the presence of watermark. Fig. 2 shows watermark distribution in HH_3 subband. Watermark is inserted in the highest wavelet coefficients and is concentrated near the edges. Restored watermarks from HL_3, LH_3 and HH_3 subbands a little differ from original one even after BCH coder. Watermark length after BCH coding is 198 bytes – twice longer than without using BCH. Restored watermark is shown below graphical presentation of the original watermark and differences between original and extracted one (see Fig. 3), but the correlation rates between these data are high.

5. Testing the Watermarking Scheme

We tested basic functionality of our watermarking scheme using different wavelet filters, image types and hash block sizes.

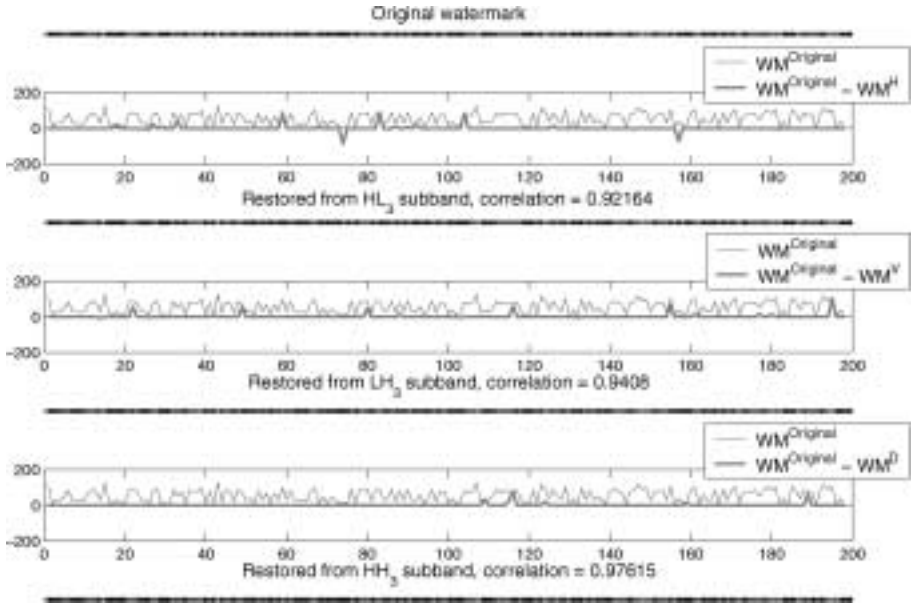

Fig. 3. Results of testing 'Lena-bw' image. Restored watermarks and differences against original watermark. BCH coding applied. Vertical axis shows the value of watermark's index.

5.1. *Testing Strategy*

To estimate influence of size of hash block, its size was variated from 3×3 to 10×10 pixels.

To test the influence of wavelet type, we selected three types of wavelets: Haar, Daubechies and (9,7), used in JPEG2000 image compression standard. Daubechies wavelets were selected of different supporting base size, to find out the influence of wavelet length to the watermarking scheme.

To test the influence of image type, the completely different images were chosen: natural image type – "Lena", high -noise image "baboon" and synthetic image "watch". We took images from publicly available USC-IPI Image Database, located at Signal and Image Processing Institute at the University of Southern California <http://sipi.usc.edu/services//Database.html> to make sure results of our tests could be compared to results of other watermarking systems on the same images.

Testing against intentional and unintentional attacks on watermark was made using StirMark package. We divide tests into two categories: the ones our watermarking scheme must pass: low-pass filtering, sharpening, gamma correction, adding noise, compression using DCT (JPEG) transform; and the ones it must fail: resizing, cropping, rotation.

5.2. *Results of Testing*

Bigger the hash block size, shorter the information, used for generating the watermark. Too short watermark is easy to remove or tamper and it captures too little of image visual

properties. So, total length of watermark should be kept as high as possible. Hash block size should be kept between 3×3 and 6×6 pixels, what gives length of watermark between 9 and 36 bytes in LL_5 subband of 18×18 pixels. Must mention that after hash block is signed with private key, the length of watermark in all cases is 99 bytes.

Testing reveals, that hashes of LL subband of original and watermarked images are almost identical. DWT decomposition, performed on watermarked image produces only small distortions on LL approximation matrix, and they can be compensated while embedding the watermark, for example making preliminary calculations on watermark's effect to LL subband.

Differences $\Delta_{LH,HL,HH}$ between approximation matrices LH, HL and HH of original and watermarked images are much greater and are of two origins: the watermark itself and transposition of energy between subbands because of multiresolutional properties of DWT. Mean values of Δ_{LH} and Δ_{HH} are almost at zero value, while mean of Δ_{HL} is entirely shifted down by 2%. This shift does not allow suitable reconstruction of watermark in HL subband.

The results of testing image processing filters are more scattered. Mean value of correlation in LH, HL and HH subbands varies in wider ranges and depends on the size of watermark and, on the same time, on the size of hash block. Better correlations between embedded and restored watermarks are achieved when using smaller hash blocks and longer watermarks. Critical size of hash block for most filters is 6 pixels: bigger hash block produces worse medium results. The results of the "best correlation" testing are linear for all filters, and, in most cases, dependence of the best correlation and size of hash block are linear too. The best correlation was achieved in HH subband almost in every filter for all images. In some cases correlation in HL subband was as low as 0.320, while correlation in HH subband was 0.990 (filter (9,7), image "Lena-bw"). The testing shows that coefficients of HL subband can be omitted during watermark detection and only subbands LH and HH taken into account.

Watermark testing using StirMark software shows that watermarking scheme is robust against lossy JPEG compression, adding noise up to 40% and median filtering up to 7 pixel box size what greatly changes image's visual perception. This is achieved due to the fact, that wavelet transformation separates low- and high-pass components of the signal, and at the same time modifications, which are not a part of the image signal, like noise, remains on the first levels of wavelet transformation.

Watermark's detection quality fails quickly when image transformations like rotating, scaling or cropping are applied. To test rescaled, rotated or cropped images, image registration, restoring original image size should be applied first. As it is pointed out in (Wang and Wiederhold, 1998), watermarking schemes based on wavelet transform are not sustainable to geometrical attacks, like mirroring, rotation changing of image size. This is because wavelet transformation is made not in 2-D, but in 1-D space, taking image as a linear signal. Resizing or cropping the image tends to changing image's linear data, which are the base for wavelet decomposition. While the information of the image is concentrated in planar data. This issue could be resolved applying DWT to image as a plane, not as linear stream. For this purpose we need 2-D wavelets.

Additionally, StirMark testing revealed the weakness of watermarks with small hash blocks. According to these results, hash block size for watermark should vary from 5 to 7 pixels, to get reliable results for visual transformations of image data.

The best test results and the greatest artifacts introduced were received when using "Haar" wavelet filter.

6. Conclusion

Due to separation of information, needed for embedding and detection process, asymmetric watermarking is the best method today for public watermarking.

Due to errors, introduced by non-reversibility of DWT transformations, coefficients in HL, LH and HH matrices will not be the same as after watermark embedding stage. Limitations on watermark, like quantization, must be set to allow watermarking in wavelet domain.

Errors, introduced during watermark embedding and detection stages should be corrected using longer, supporting up to 3 error corrections in 4-bit message, error correction coding.

Wavelet-based watermarking scheme is robust against noise, lossless and DCT compression, filtering attacks due to the filtering properties of wavelet transformation.

Testing of watermarked images shows that only coefficients of subbands LH and HH retain watermark, while coefficients of subband HL do not preserve the watermark and therefore can not be taken into account, while testing images for presence of watermark.

References

Cox, I.J., J. Killian, F. Thomson Leighton and T. Shamoon (1997). Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, **6**, 1673–1687.

del Bimbo, A., M. Mugnaini, P. Pala, F. Turco (1998). Visual querying by color perceptive regions. *Pattern Recognition*, **31**(9), 1241–1253.

Craver, S. (1999). Zero knowledge watermarking detection. In *A. Pfitzmann, ed. Robustness of an Asymmetric Watermarking Technique, Information Hiding '99*, Vol. 1768 of *Lectures Notes in Computer Science (LNCS)*. Springer-Verlag, Dresden, Germany. pp. 101–116.

Eggers, J.J., J.K. Su and B. Girod (2000). Public key watermarking by eigenvectors of linear transforms. In *Proceedings of the European Signal Processing Conference (EUSIPCO 2000)*, Vol. 3.

Fiat, A., and A. Shamir (1987). How to prove yourself: Practical solutions to identification and signature problems. In A.M. Odlyzko (Ed.), *Advances in Cryptology – CRYPTO '86*, Vol. 263 of *Lectures Notes in Computer Science (LNCS)*. Springer-Verlag. pp. 186–196.

Furon, T., and P. Duhamel (2000). An asymmetric public detection watermarking technique. In A. Pfitzmann(Ed.), *Robustness of an Asymmetric Watermarking Technique, Information Hiding '99*, Vol. 1768 of *Lectures Notes in Computer Science (LNCS)*. Springer-Verlag, Dresden, Germany. pp. 88–100.

Hachez, G., and J.-J. Quisquater (2002). Which directions for asymmetric watermarking? In *Proceedings of the XI European Signal Processing Conference (EUSIPCO 2002)*. Toulouse, France.

Hartung, F., and B. Girod (1997). Fast public-key watermarking of compressed video. In *Proceedings of the IEEE International Conference on Image Processing*. pp. 528–531.

Jacobs, Ch.A., A. Finkelstein and D.H. Salesin (1995). Fast multiresolution image querying. In *Proceedings of SIGGRAPH '95*. ACM, New York. pp. 277–286.

Langelaar, G.C., I. Setyawan, R.L. Lagenadijk (2000). Watermarking digital image and video data. *IEEE Signal Processing Magazine*, **9**, 20–46.

Noda, H., J. Spaulding, M.N. Shirazi, M. Niimi and E. Kawaguchi (2003). Bit-plane decomposition steganography combined with Jpeg2000 compression. In F.A.P. Petitcolas (Ed.), IH 2002, *LNCS 2578*. Springer-Verlag, Berlin, Heidelberg. pp. 295–309.

van Schyndel, R.G., A.Z. Tirkel and I.D. Svalbe (1999). Key independent watermark detection. In *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, Vol. 1. pp. 580–585.

Su, P.-Ch., and C.-C. Jay Kuo (2003). Steganography in Jpeg2000 compressed images. In *IEEE Transactions on Consumer Electronics*, Vol. 49(4). pp. 824–832.

Taubman, D.S., M.W. Marcellin (2002). *JPEG2000. Image Compression Fundamentals, Standarts and Practice*. Kluwer Academic Publishers.

Venkatesan, R., S.-M. Koon, M.H. Jakubowski and P. Moulin (2000). Robust image hashing. In *Proceedings of International Conference on Image Processing*. Vancouver, Canada.

Voyatzis, G., and I. Pitas (2000). Image watermarking for copyright protection and authentication. In Al Bovik (Ed.), *Handbook of Image and Video Processing*. Academic Press. pp. 773–745.

Wang, J.Z., G. Wiederhold (1998). WaveMark: digital image watermarking using daubechies' wavelets and error correcting coding. In *Proceedings of SPIE*, Vol. 3528 Multimedia satellite networks: issues and challenges.

Wolfgang, R.B., and E.J. Delp (1999). Fragile watermarking using the VW2D watermark. In *International Conference on Security and Watermarking of Multimedia Content*. pp. 204–213.

Wu, M., and B. Liu (1998). Watermarking for image authentication. In *Proceedings of 1998 IEEE International Conference on Image Processing*, Vol. 2. Chicago, USA. pp. 437–441.

G. Kazakevičiūtė received PhD degree in technological sciences from Vilnius Gediminas Technical University in 2000. Current research focuses on analysis of information hiding and watermarking methods. Areas of interest are image processing, computer graphics, computer aided design technologies.

E. Januškevičius received MArch degree from Vilnius Gediminas Technical University in 2000. Current areas of interests are image processing, computer graphics, computer aided architectural design, computer networks and programming.

R. Rosenbaum is a research assistant currently employed at the Institute of Computer graphics at University of Rostock/Germany. His personal research interest covers the handling and processing of digital images. He is responsible for different education and research programs, and has conducted and accomplished several projects in Steganography and Watermarking. Beside this, he is working in image communication using JPEG2000, and visualization of large data sets considering limitations of mobile environments.

H. Schumann graduated at the University of Rostock/Germany (1977 Master degree, 1981 PhD, 1989 postdoctoral lecture qualification). Since 1992 she is heading the Computer Graphics Research Group at the Department of Computer Science in Rostock. Her research profile covers information visualization and visual data mining, mobile interfaces, and rendering as well as image display. She was heading the cross-institutional research group “MoVi” Visualization of Multimedia Information on Mobile Computer Systems, supported by the German Research Society – DFG.

Jautrus modifikavimui vaizdo žymėjimo metodas, besiremiantis esama viešų raktų infrastruktūra

Geruta KAZAKEVIČIŪTĖ, Eugenijus JANUŠKEVIČIUS,
René ROSENBAUM, Heidrun SCHUMANN

Straipsnyje aprašomas jautraus modifikavimui asimetriniu kodavimu pagrįsto vandenženklio kūrimas ir jo panaudojimas JPEG2000 vaizdo kodavimo standarte. Siūlomas metodas apjungia vietą ir dažnį lokalizuojančias diskrečiosios vilnelių transformacijos (DWT) savybes bei galimybę patikrinti vandenženklį naudojant asimetrinės kriptografijos metodus. Siūloma schema pritaikyta JPEG2000 vaizdo saugojimo standartui ir išnaudoja jo savybes, bet gali būti taikoma bet kokiame kitame šiuo metu egzistuojančiame vaizdo išsaugojimo standarte. Pateiktame vaizdo žymėjimo metode svarbus vaidmuo tenka viešų raktų infrastruktūros panaudojmui. Mes naudojame PGP (Pretty Good Privacy) viešų raktų valdymo schemą siuntėjo tapatybei nustatyti ir vandenženklio atstatymo raktų saugojimui bei paieškai. Daugumoje vaizdo žymėjimo metodų, kurie remiasi viešų raktų panaudojimu, naudojami nauji, nepriklausomi ir kriptologiškai nepatikrinti raktai. Esamos viešų raktų patikrinimo ir apsikeitimo infrastruktūros panaudojimas suteikia mums galimybę turėti kriptologiškai atsparius ir patikrintus raktus.