DOI: 10.15439/2014F250
ACSIS, Vol. 2

Image Hashing Secured With Chaotic Sequences

Relu-Laurentiu Tataru
Faculty of Electronics, Telecommunications and
Information Technology,
POLITEHNICA University of Bucharest, 1-3, Iuliu
Maniu Bvd., Bucharest 6, Romania
Email: tataru.relu@yahoo.com

Abstract—**This paper presents an image hashing algorithm using robust features from jointed frequency domains. Extracted features are enciphered using a secure chaotic system. The proposed hashing scheme is robust to JPEG compression with low quality factors. This scheme also withstands several image processing attacks such us filtering, noise addition and some geometric transforms. All attacks were conducted using Checkmark benchmark. A detailed analysis was conducted on a set of 3000 color and gray images from three different image databases. The security of the method is assured by the robustness of the chaotic PRNG and the secrecy of the cryptographic key.**

I. Introduction

THE INCREASING popularity of mobile devices (smartphones, tablets and other gadgets) with high quality cameras and the development of high resolution digital cameras raised the number of digital images published over the Internet. An important contribution to this evolution was due to fast development of social networks and services dedicated to media sharing. All these services increased the number of images released in the online environment. This evolution raised the need of addressing some important issues such as: ownership protection, content authentication, fast image indexing and retrieval.

Digital watermarking was proposed as a first solution for these issues. Watermarking techniques are used to embed binary signatures (watermarks) modifying directly the content of the image. Watermarked images are identified detecting the presence of the watermark inside the image. However, many watermarking methods do not solve the problem of content identification. Most of the embedded watermarks are independent of the image content. An alternative of these problems was provided by the concept of image hashing.

Perceptual image hashing was used in the last years to solve ownership disputes, authentication and image retrieval problems. An image hashing scheme usually provides a sequence of values defining the visual characteristics of the image. The result is an image fingerprint usually protected with cryptographic techniques. Compared with conventional hash functions from cryptography, perceptual hash functions designed for images tolerate those modifications which do not affect the content of the image (i.e. compression, filtering, noise addition etc.). Thus, images with different representations, but with the same visual content, provide the same or very close hash values.

In the current work we propose a new algorithm which includes a perceptual hashing scheme for digital images secured with chaotic sequences. The interest was to find a suitable image representation space providing robust features to create an image hash for serving authentication of digital images, copyright protection and easy image database management. Our idea was to combine efficiently existent spaces in frequency domain for feature extraction.

This paper is organized as follows: Section II describes the design principles and properties of image hashing regarding some state of the art principles, Section III presents the construction of the image hashing scheme, Section IV illustrates the simulation results, Section V points some practical applications of the proposed algorithm and Section VI concludes the work.

II. Image Hashing – design principles and properties

A. Design Principles

It is widely accepted that the basic components of perceptual image hashing are image pre-processing, feature extraction, feature post-processing and randomization.

By image pre-processing a new scaled version of the digital image is obtained. This is usually achieved reducing the representation space of the original image without losing the significant features of the content. The result is usually a scaled version of the input image, facilitating operations with low computational cost.

Feature extraction is the next phase in the construction of the image hash. This is an important stage because the feature space influences directly the robustness of the hash function. Depending on the application type of the image hashing scheme, a certain domain for feature extraction could be imposed. Robust algorithms are usually relied on frequency transforms for feature extraction. A scheme resilient to JPEG compression may use Discrete Cosine Transform (*DCT*) for the feature extraction stage. In [3], the authors proposed a scheme based on the statistical modeling

978-83-60810-58-3/$25.00 © 2014, IEEE

of DCT coefficients as a Gaussian distribution. The authors assert that invariance of DCT coefficients achieves robustness against attacks such as JPEG compression, filtering, scaling, brightness adjustment, histogram equalization and even small angle rotations. In [6], Fridrich and Gojan illustrate the advantage of considering low frequencies in DCT domain for feature extraction. Their reasoning is based on the properties of low frequency DCT coefficients which preserve the significant information of the image. Any modification in these frequencies is noticeable on the host image. Other transforms are also used in achieving perceptual image hashing. Guo and Dimitros propose the extraction of a robust feature set by using Discrete Wavelet Transform (DWT) followed by Radon transform [9]. The hash value is generated using a probabilistic quantization. This hash value is resilient to image compression, filtering, scaling and rotations. The authors assert good results even for image tampering. When high robustness against geometric attacks is required, the feature set may be extracted using transforms such as Discrete Fourier Transform or Mellin Fourier Transform. Swaminathan et al. [7] propose in their work an image hashing algorithm based on Mellin Fourier Transform. They claim to obtain good results against rotation operations up to 10^o and 20% cropping. This class of methods usually performs well against this type of attacks. However, they may be less robust against other common attacks such as noise addition.

The feature extraction process could be also realized in other transform domains such as Singular Value Decomposition (SVD) [5] or Fast Johnson-Lindenstrauss Transform (FJLT) [10].

The feature set extracted from a transform domain is generally built to assure the goals of the image hashing scheme.

Post-processing stage is usually a compression of the previously extracted features. A feature reduction technique is usually applied in this purpose in order to obtain a final binary feature set. This step is commonly realized using one of the following techniques: random projection of the feature set in another space, direct compression of the feature set, feature set quantization, clustering or by computing a cryptographic hash of the feature set.

Randomization is the last step in achieving the final perceptual hash value of the image. This step is mandatory and assures the unpredictability of the hash value obtained for each digital image, using a secret key.

B. Properties of Image Hashing

The final hash algorithm should provide the following features: one-way i.e. hard to recover the input from the hash value, collision resistant i.e. perceptually different images provide totally different hash values and key-dependence i.e. the hash value is highly dependent on the secret key.

The use of the hash value in verifying an image with a pair is resumed to the direct comparison of the two binary hashes. Few or zero differences between the hash values validate the authenticity of one image with respect to the other one.

The goal of this paper is to propose a robust hash function which respects both design principles and general features of image hashing algorithms. The proposed algorithm is potentially capable of solving copyright disputes, authenticating similar images and retrieving the image content from large image databases.

III. Proposed Image Hashing Scheme

As most of the image hashing algorithms, our scheme computes a global set of features from a digital image. A feature set is used to compute a perceptual hash value. The feature set is enciphered using a chaotic system. The novelty of the proposed algorithm is given by the feature set construction and the use of a proven secure chaotic system for the feature set encryption. A description of the proposed image hashing scheme is illustrated in the following subsections.

A. Image Pre-Processing

A color digital image is converted to grayscale and resized to a default size mxm. The resizing procedure allows fast operations on the grayscale image. Comparing to the original input image, the content of the new image is not changed.

B. Feature Extraction

For the feature extraction step, the grayscale image is converted in frequency domain. This is the most significant stage in computing a robust feature set. A feature set built in frequency domain provides good robustness to certain classes of attacks.

Discrete Wavelet Transform (DWT) and Discrete Cosine Transform (DCT) are jointly used for extracting the global feature set. The first level DWT decomposition assures the separation of the information from the grayscale image in frequency sub-bands LL_1 (low-low), HL_1 (high-low), LH_1 (low-high) and HH_1 (high-high). The LL_1 sub-band carries most of the information from the grayscale image. For this reason, we consider the LL_1 sub-band in the feature generation process. A n –level decomposition is performed, considering LL_{n-1} sub-band ($n \geq 2$) at each iteration. At the n -level decomposition, the LL_n sub-band is obtained. This sub-band provides a matrix preserving most of the correlations from the original grayscale image. The DCT transform is applied for this sub-band on blocks of size kxk . The Wavelet distribution from the LL_n sub-band is changed at the block level, and the new distribution follows the properties of DCT transform. Most significant frequencies are positioned in the top-left corner of the block, and the less significant frequencies are grouped in the bottom-right, according to the DCT distribution. The first term of each DCT block, i.e. the (0,0) frequency, integrates the most important part of information from the block. This frequency, also called DC term, is extracted from each DCT block.

C. Feature Post-Processing

A feature vector containing the DC's of all DCT blocks computed from the LL_n sub-band is obtained at the previous step. At the current step, we apply a binarization technique for the feature vector. This is achieved by comparing each component of the feature set with the global mean of the feature set. Binary 0 is used to represent DC values under the mean and binary 1 is used to represent DC values above the mean. Thus, we obtain a binary fingerprint of the digital image.

D. Feature Randomization

This step is mandatory in order to assure the confidence of the binary feature set. The security of the feature set is obtained by direct enciphering with a recently proposed chaotic system. The chaotic generator proposed by Vlad et al. [8] is used as a stream cipher. This generator is based on tent-map and the running-key principle. The tent-map has the following formula:

$$x_{n+1} = f(x_n) = \begin{cases} \frac{x_n}{a}, & 0 \le x_n \le a \\ \frac{1-x_n}{1-a}, & a < x_n \le 1 \end{cases} \tag{1}$$

where $a \in (0,1) \setminus \{0.5\}$ is the control parameter of tent-map, x_n is the n^{th} value of the chaotic sequence generated using the tent-map and x_0 is the initial value from $(0,1)$ range. Binary sequences Z_i are generated using the X_i real value sequences generated with the tent-map, and binarization threshold c.

$$z_{i,j} = \begin{cases} 0, & 0 \le x_{i,j} \le c \\ 1, & c < x_{i,j} \le 1 \end{cases}, \tag{2}$$

where $z_{i,j}$ is the j^{th} element of the chaotic binary sequence Z_i and $x_{i,j}$ is the j^{th} element of the chaotic non-binary sequence X_i.

A running-key procedure is applied for typical Z_i binary sequences in order to obtain binary i.i.d. sequences compatible with the fair coin model.

The enciphering key used for the proposed image hashing algorithm is a binary sequence based on five additions of typical sequences Z_i, as shown in equation 3.

$$Y = \sum_{i=0}^{4} Z_i \bmod 2 \tag{3}$$

According to [8], the binarization threshold was considered equal to the control parameter $(c = a)$. The security of the method was theoretically and experimentally proven for a control parameter a in the range $(0.39, 0.61) \setminus \{0.5\}$ for 5 modulo 2 additions.

The secret key K of the system is given by the initial values of each non-binary sequence X_i and the control parameter:

$$K = \left(x_{00} \,||\, x_{01} \,||\, x_{02} \,||\, x_{03} \,||\, x_{04} \,||\, a \right) \tag{4}$$

Note: As already suggested in [8], the additions number could be increased, extending the range of the control parameter a. This result leads to a larger selection of the secret key. The construction principle of the chaotic system used to generate the pseudo-random key to encipher the feature set is presented in Fig. 1.

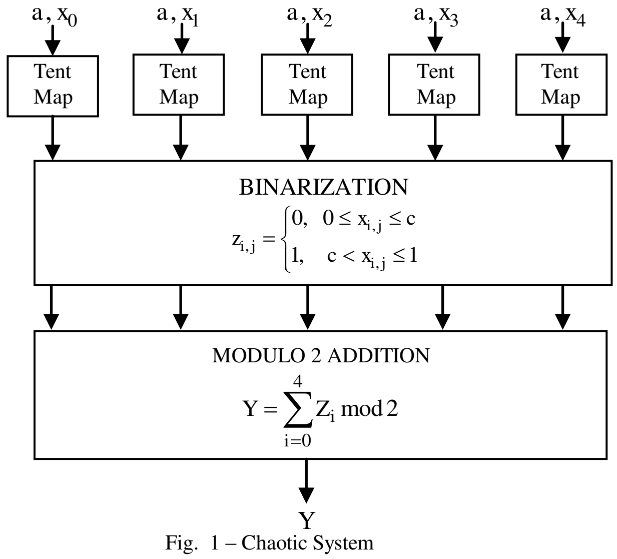

a, x_0
a, x_1
a, x_2
a, x_3
a, x_4
Tent Map
Tent Map
Tent Map
Tent Map
Tent Map
BINARIZATION
$z_{i,j} = \begin{cases} 0, & 0 \le x_{i,j} \le c \\ 1, & c < x_{i,j} \le 1 \end{cases}$
MODULO 2 ADDITION
$Y = \sum_{i=0}^{4} Z_i \bmod 2$
Y

Fig. 1 – Chaotic System

The step-by-step construction of the perceptual image hashing scheme according to the description is next presented:

1. The input color image is transformed to grayscale.
2. The grayscale image is resized to a standard mxm dimension using the bicubic interpolation.
3. A n–level DWT Haar decomposition is applied on the grayscale image.
4. LL_n sub-band is divided in non-overlapping kxk blocks and DCT transform is applied on each block.
5. DC coefficients are extracted from all DCT blocks and the vector V containing the features of the image is built. The

length of the feature vector V is given by the formula:

$$l = \left(\frac{m}{k \cdot 2^n} \right)^2 \quad (5)$$

6. The mean value m_{dc} of the feature vector is computed.

7. The feature vector V is binarized and a new binary feature vector W is obtained according to the formula:

$$w_i = \begin{cases} 0, & v_i < m_{dc} \\ 1, & v_i \geq m_{dc} \end{cases} \quad (6)$$

where $V = (v_i)_{i=1...l}$ and $W = (w_i)_{i=1...l}$

8. A pseudo-random sequence $Y = (y_i)_{i=1...l}$ is generated using the chaotic system presented in Fig. 1, with the secret key K.

9. The binary feature vector W is enciphered using the pseudo-random sequence Y and the final hash value $H = (h_i)_{i=1...l}$ is obtained, where: $h_i = w_i \otimes y_i$, $i = 1...l$

At the end of all steps, a hash value with $l-$ bits length is obtained. The proposed system is illustrated in Fig. 2.

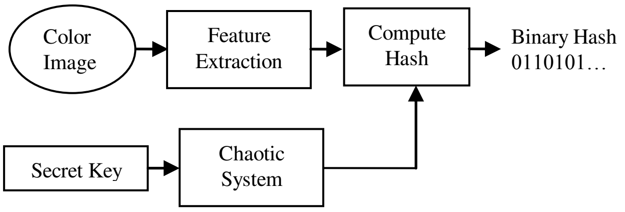


Fig. 2 – Proposed Hashing System

IV. Experimental Results

A. Simulations description

Several parameters were tested during the simulations. This parameters were the dimension of the resized image (M), the $n-$ level of the DWT transform and the block size kxk of the DCT transform. In this paper we illustrate the performances of the proposed algorithm for the following parameters:

a) $M = 256,\ n = 1,\ k = 8,\ l = 256$

b) $M = 256,\ n = 2,\ k = 4,\ l = 256$

The investigation of the proposed scheme was performed for hash values with constant binary lengths $l = 256$. All feature sets were encrypted using the 256–bit length chaotic sequences generated according to [8]. Each bit error from the binary hash contributes to the final error with the value $\frac{1}{l}$ (i. e. err ≈ 0.0039).

For our purposes we used three different databases with resized images between 512x384 and 1024x1024. The investigation of the proposed algorithm was conducted using the following databases: Uncompressed Color Image Database (UCID) (color images), Break Our Steganographic System (BOSS) database (color images) and Break Our Watermarking System (BOWS) database (gray images). 1000 uncompressed images with different formats (tif, bmp and pgm) were randomly chosen from each database to create our testing set containing 3000 images. Images from BOSS database were converted from CR2 (Cannon Raw file format) in bmp format.

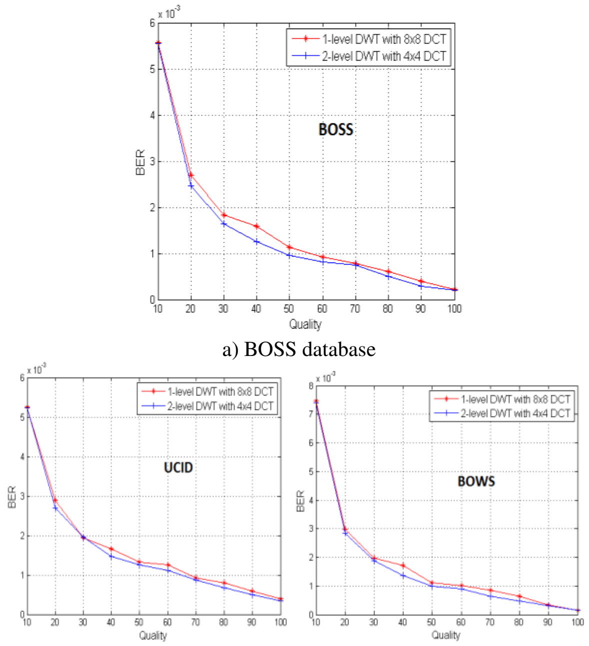
To define the similarity between the reference hash and the target hash, we use the Bit Error Rate (BER) as a measure of number of differences. The BER value for two hashes is given by the ratio between the number of erroneous bits and the total number of bits. A perfect similarity equates with a 0 BER value and two completely different images should provide a BER value close to 0.5 (not similar).

B. Robustness against JPEG Compression

Our tests for the proposed hash function aimed primarily the resilience of the method to the JPEG compression with different quality factors $(Q = 10,\ 20...100)$. A number of 1000 uncompressed digital images from each database were compressed with different quality factors, from 10 to 100. All hash values computed from uncompressed images were compared with hash values calculated for the corresponding JPEG image compressed with quality factor Q. The robustness achieved under JPEG compression for all three databases are independently illustrated in Fig. 3. Our results prove the robustness of the proposed method for both color and gray images at compression factors down to $Q = 10$. A DWT 2-level decomposition of the image jointed with the DCT 4x4 decomposition proved slightly better results in terms of robustness against JPEG compression for all three image sets.

C. Robustness against other image processing attacks

The proposed hashing algorithm exploits the advantage of transform domain and also provides some robustness against common image processing attacks such as filtering, noise addition and some geometric transforms. The BER value calculated between the hashes was also used as metric to measure the similarity between the original and attacked images. Several attacks were applied on the Lena image stored in JPEG format compressed with $Q = 90$ and with size 512x512. All attacks were conducted using the Checkmark framework [11] with the specific parameters and the results are presented in Table 1. The proposed hashing scheme performed well under filtering, noise addition and some geometric attacks. However, the method proved to be vulnerable against geometric manipulations such as rotations greater than 2^o and certain cropping attacks.



a) BOSS database

b) UCID database c) BOWS database

Fig. 3 - Robustness against JPEG compression

TABLE I.
ROBUSTNESS AGAINST CHECKMARK ATTACKS FOR IMAGE LENA

Attacks [11]	**Bit-Error-Rate (BER)**
Gaussian Noise(1)	0
Hard Thresholding(1)	0.0039
Soft Thresholding(1)	0.0039
Wiener Filtering	0.0039
Median Filtering(1)	0.0156
Sharpening	0.0117
Shearing(1)	0.1133
Stirmak(1)	0.0156
JPEG compression 10	0
Wavelet Compression 10	0
Denoising with Remodulation(1)	0.0039
Sample Down(1)	0.0117
Template Remove	0.0039
NullLineRemove(1)	0.0078
Rotation 2°	0.0273
Rotation 10°	0.4961
Scale 40%	0.0117
Crop 40%	0.1796

(1) This test is available in Checkmark Benchmark in several variants. Worst result is presented.

D. Robustness against malicious attacks

An attacker may perform two types of malicious attacks. The former implies the counterfeiting of both digital image and hash value. A second type of manipulation is by direct modification of the image content, while retaining the hash value of the image. The first class of attacks is unfeasible for the proposed scheme due to the secrecy of the enciphering key of the chaotic system. The resilience of the proposed algorithm against this class of attacks is given by the strength of the chaotic system and the secrecy of the key. For the latter class of attacks, the image may be maliciously distorted using the following techniques: object addition, object removal and object altering. Our block based hashing scheme is less sensitive to local modifications. Changing small parts of the image is reflected by the DC coefficients obtained for the DCT transform applied for the LL sub-band. However, these changes are not fully reflected in the binary feature vector. This is because the averaging procedure used to collect the feature set is not always sensitive to this type of modification. The use of a threshold very close to 0 may assure a partial robustness against this class of attacks.

A feasible solution for images containing text elements (letters, numbers, visible watermarks etc.) is using character identification techniques. All extracted text elements may be hashed using a robust cryptographic hash function. This hash value is concatenated to the perceptual hash value and a final hash is built. The use of SHA-256 as cryptographic hash function assures a 512 bit length of the final hash value.

E. Collision Resistance

A perceptual hashing scheme should provide different hashes for dissimilar images. The proposed algorithm complies with this requirement and provides different hash values for different images. In order to illustrate the collision resistance property of the proposed image hashing scheme an example is illustrated in Fig. 4 for images AudiA4_1.jpg and AudiA4_2.jpg (source: www.autovit.ro) The BER value calculated for the images presented in figure 0.5078 indicates the total difference between the hashes of the two distinct images.

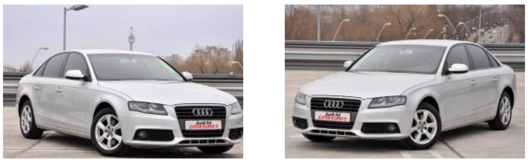


Fig. 4 – a) AudiA4_1.jpg b) AudiA4_2.jpg
BER = 0.5078

However, the BER value of dissimilar images is not always close to 0.5. In Fig. 5 we illustrate the discriminative capability of the proposed algorithm, by computing the probability density function of BER values for dissimilar images. This result was obtained for 1000 image pairs, randomly extracted from the test databases. The BER values calculated between perceptual hashes of distinct images have a Gaussian distribution, with the mean 0.4763, which is close to the theoretical value 0.5.

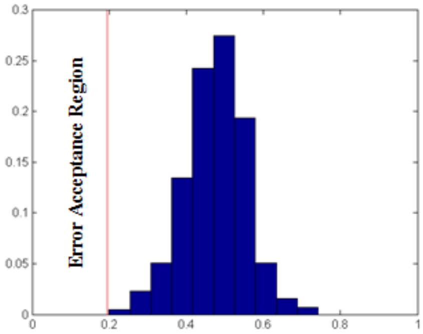


Fig. 5 – Probability density function for 1000 dissimilar image pairs

The minimum value (i. e. 0.1992) is far enough from most of the values obtained for Lena image and its attacked versions using Checkmark Benchmark. A BER threshold value fixed at 0.05 assures very good performances for our perceptual hashing method.

III. Practical Applications of the proposed hashing scheme

The goal of the proposed image hashing system was to cover the following three topics: image authentication, image retrieval and copyright protection. In all three cases the reference image is required. For each target image the hash value is computed using the same secret key as for the reference image. Two hashes are computed at the bit level in order to determine the similarity level.

The verification system is built according to Fig. 6.

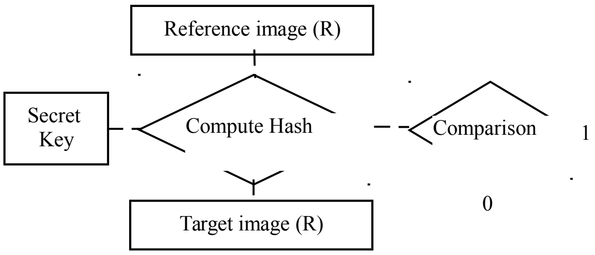


Fig. 6 Verification System

After comparing the two hashes at bit level, a *BER* value is computed. Depending on the sensitivity of the application integrating the image hashing scheme, different threshold value of the BER should be chosen. *BER* values above the threshold outputs a binary 0 (non-authentic image) and *BER* values under the threshold outputs 1 (authentic image). A *BER* value should be close to 0 when very high sensitivity is required (e.g. authentication of tampered images) and the may be increased when the application is not very restrictive (e.g. applications with content identification such as *TinEye*, *Google Images*).

IV. Conclusions and future works

In this paper we investigated the concept of image hashing in frequency domain secured with the aid of chaotic sequences. Our feature points are extracted using jointed *DWT* and *DCT* transforms. The feature set is enciphered using a robust chaotic map. A large set of natural images was tested in order to measure the performances of the proposed method. Experimental results illustrated a good robustness of the proposed method against known attacks such as compression, filtering, noise addition and slight geometric transformations. The proposed scheme may be applicable for image authentication, copyright protection and image retrieval. In part of future research, we will concentrate on an alternative approach which is more robust against geometric transforms and tampering attacks.

Acknowledgment

The author thanks Professor Adriana Vlad for the guidance concerning the use of the chaotic system from this work and for all technical advices and discussions about chaos based cryptography.

References

[1] V. Monga and B. Evans, “Robust perceptual image hashing using feature points,*” in Proc. IEEE Int. Conf. Image Processing, Singapore*, pp. 677-680, 2004, doi: 10.1109/ICIP.2004.1418845.

[2] L. Weng and B. Preneel, “A secure perceptual hash algorithm for image content authentication,” in *Proc. Of IEEE International Conference on Signal Processing and Communications*, pp. 1063-1066, 2007.

[3] F.-X. Yu, Y.-Q. Lei, Y.-G. Wang and Z.-M. Lu, “Robust image hashing based on invariance of DCT coefficients,” JIH-MSP 2010, vol.1, pp.286-291.

[4] M. K. Mihcak and R. Venkatesan, “New iterative geometric methods for robust perceptual image hashing,” in *Proc. ACM Workshop Security and Privacy in Digital Rights Management*, Philadelphia, 2005, doi: 10.1007/3-540-47870-1_2.

[5] S. S. Kozat, K. Mihcak, and R. Venkatesan, “Robust perceptual image hashing via matrix invariances," *Proc. IEEE Conf. on Image Processing*, pp. 3443-3446, 2004, doi: 10.1109/ICIP.2004.1421855.

[6] J. Fridrich and M. Goljan, “Robust hash functions for digital watermarking,” in *Proc. IEEE Int. Conf. Information Technology: Coding Computing*, pp. 178-183, 2000.

[7] A. Swaminathan, Y. Mao and M. Wu, “Image hashing resilient to geometric and filtering operations,” in *Proc. IEEE Workshop on Multimedia Signal Processing*, Siena, Italy, Sep. 2004, doi: 10.1109/MMSP.2004.1436566.

[8] A. Vlad, A. Luca, O. Hodea and R. Tataru, “Generating chaotic secure sequences using tent map and a running-key approach,” in *Proc. of The Romanian Academy*, Series A, vol. 14, pp. 292-302, 2013.

[9] X. X. Guo and H. Dimitrios, “Content based image via wavelet and radon transform,” in *Proc. Of the 8th Pacific Rim Conference on Multimedia*, Hongkong, China, vol. 4810, pp. 755-764, 2007, doi 10.1109/ICIEA.2007.4318736.

[10] X. Lv and Z. J. Wang, "Fast Johnson-Lindenstrauss Transform for Robust and Secure Image Hashing," *Proc. of the IEEE 10th Workshop on Multimedia Signal Processing (MMSP)*, pp: 725-729, 2008, doi:10.1109/MMSP.2008.4665170.

[11] S. Pereira, S. Voloshynovskiy, M. Madueno, S. Marchand-Maillet and T. Pun, “Second generation benchmarking and application oriented evaluation,” in *Information Hiding Workshop*, Pitsburgh, PA, USA, 2001.

[12] G. Schaefer and M. Stich, “Ucid – An uncompressed colour image database,” in *Proc. SPIE, Storage and Retreval Methods and Applications for Multimedia*, pp. 472-480, San Jose, U.K, 2004.

[13] P. Bas, T. Filler and T. Pevny, “Break our steganographic system – the ins and outs of organizing BOSS,” in *Proc. of Information Hiding Conference*, Prague, 2011, doi: 10.1007/978-3-642-24178-9_5.

[14] T. Furon and P. Bas, “Broken arrows,” *EURASIP Journal on Information Security*, 2008, doi:10.1155/2008/597040.