A Robust Image Hashing with Enhanced Randomness by using Random Walk on Zigzag Blocking

Xi Huang, Xiaoguang Liu, Gang Wang, and Ming Su*
Nankai-Baidu Joint Lab, Department of Computer Science, Nankai University, China

***Abstract*—Security issues are important concerns of image hashing. A type of image hashing needs to randomly divide a given image into several parts, usually rectangles. However, the security problem arises due to the rectangular shapes. So, we propose a new method which enhances the randomness by dividing the image into zigzag blocks with random walk. Security analyses are provided, and experiments show our method has good distinguishing ability compared with existing methods.**

1. Introduction

With the increasing application of multimedia data, efficient management of digital content and verifying content integrity have become important concerns. Since multimedia data can allow moderate levels of modifications, the cryptographical hash usually used to authenticate text messages bit-by-bit is not applicable. An image hashing algorithm takes an image as input and outputs a hash value. When there are slight modifications on the image, the hash output should be approximately the same. Since image hashing has promising applications in image authentication and image retrieval, researchers are interested in improving the robustness of image hashing [1], [2], [3], [4]. Meanwhile, security has become another important concern of image hashing [5], [6] because of the risk that an attacker may forge a visually different image having almost the same hash value.

Generally, an image hashing includes robust feature extraction, feature compression, and feature concatenation into final hash output, where a secret key is usually used in feature extraction or feature compression process. We particularly focus on the feature extraction, where the key is used in the random blocking. To express the image hashing concisely, we use the function $V = ImageHash(Image, Key)$, where V represents the output sequence and Key the key used in the blocking. In feature extraction stage, random process can be involved to enhance security. Suppose the attacker can observe the $(Image, V)$ pairs and know the image hashing scheme, but he does not know the Key. Venkatesan et al. [7] proposed an image hashing scheme in which the original image was divided into several rectangles controlled by a key. Intuitively this method is secure since we may use the Key to control the random dividing process. However, since a rectangle can be uniquely determined by its top-left coordinate and bottom-right coordinate, there was an iterative search attack [8, p. 465]. The divided regions controlled by the Key can be approximated by using a number of pairs of $(Image, V)$ observed by the attacker. The *unicity distance* proposed by Shannon is a security measure for encryption system, which is the least number of $(Image, V)$ pairs required to ensure crack success of image hashing[8]. Different from sensitive to the input changes of cryptographic hash function [9], image hashing output should be almost the same when the input images are perceptually similar. On the one hand, it is believed that key disclosure problem exists for robust image hashing schemes due to the Key reuse. On the other hand, block or stream cipher is believed to be secure supporting key reuse because of elaborate components with convincing security. Though in a dilemma, we are still possible to provide a secure robust image hashing scheme by introducing carefully designed randomness.

Matias and Shamir proposed a video scrambling scheme on Space-Filling Curves (SFC) in [10], where an SFC is a Hamiltonian path that traverses every pixel one time in a frame. Since searching space for such SFCs is exponential, security is guaranteed, and see further discussions in [11]. However, this method aimed at video scrambling and was based on pixels, which is sensitive to slight modifications on images and not applicable in the robust image hashing. On the one hand, rectangular blocks are considered for division, and a local feature such as expectation or variance of a block is adopted because the scheme [7] was fairly robust to image manipulations. On the other hand, a security problem arises since the rectangular blocks related to the key can be estimated in [8]. In order to enhance the security of image hashing, we propose a new method by using *random walk* to divide an image into several blocks. Random walk has been widely used in randomized algorithms [12], and see some theoretical analysis as well as experiments in [13]. In our method, two keys are used to divide image into non-rectangular, zigzag blocks instead of rectangles, which makes it infeasible to estimate the shape and location of them. The unicity distance of our image hashing should be very large and the iterative search attack does not work. Meanwhile we do not lose the robustness of image hashing, and the performance of the ability to distinguish still remains.

The rest is organized as follows. In section 2, we give a detailed description on the random walk method. Then analyses on security are provided in section 3, and experimental

2324-9013/16 $31.00 © 2016 IEEE
DOI 10.1109/TrustCom.2016.39

results are in section 4. Finally we conclude in section 5.

2. Random Walk Method

Different from cryptographical hash, image hash keeps almost the same with pixel value changes in small areas of an image, which is not a successful attack as shown in Fig.1. But once we know the division of image hash, we can forge a visually different image by replacing other content in blocks with the same statistical feature as shown in Fig.2, which is a successful attack. The distortion can be estimated by the peak signal to noise ratio (PSNR), where

$$PSNR = 10 \cdot \log_{10} \frac{MAX_I^2}{MSE},$$

and MSE is the mean squared error between two $m \times n$ gray-scale images I and K,

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |I(i,j) - K(i,j)|^2,$$

MAX_I is the maximum pixel value of the image. The PSNR between two images is 28.5db in Fig. 1, a regular acceptable value for similar images. In contrast, that value is 9.2db in Fig. 2, which is small enough to verify the success of the attack. Now we briefly describe the iterative search attack [8, p. 465] as follows.

Suppose the attacker observes the pairs $(I_1, V_1), (I_2, V_2), \ldots, (I_n, V_n)$, and let $\tilde{V} = [V_1, V_2, \ldots, V_n]$. If an estimated key (rectangle parameters) $\hat{p}$ is used to produce $\hat{V} = [\bar{V}_1, \bar{V}_2, \ldots, \bar{V}_n]$, where $\bar{V}_i = ImageHash(I_i, \hat{p})$, $i = 1, 2, \ldots, n$. Denote the *normalized correlation* $\eta(\tilde{V}, \hat{V})$ by

$$\eta(\tilde{V}, \hat{V}) = \frac{\tilde{V}^T \hat{V}}{\|\tilde{V}\| \cdot \|\hat{V}\|}.$$

Then we can use the normalized correlation between $\tilde{V}$ and $\hat{V}$ to indicate the accuracy of the estimation $\hat{p}$, and find $\hat{p}$ which is the maximum of the normalized correlation $\eta(\tilde{V}, \hat{V})$.

At the initialization stage, they partitioned the entire image area using three methods and obtained 96 rectangular blocks, and a rough region with the parameter set achieving the maximum normalized correlation among 96 blocks was estimated. Then in the search refinement stage, they updated the rectangle parameter iteratively. In each iteration they updated the existing parameter set to obtain a larger normalized correlation with 12 sets of parameter increments for one rectangle block. Finally a more exact region was obtained once the termination conditions were satisfied.

Due to the security weakness in image hashing, we propose a new method to divide images including three steps as follows. First the image is divided into small rectangles called *grids*. Second we use random walk based algorithm to combine these grids into several zigzag blocks. Both these two steps need Key_1 or Key_2 to produce pseudo-random numbers, where the pseudorandom generator $Rand(\cdot)$ is a linear congruential generator of 64 bits uniformly distributed

(a) (b)

Figure 1. Visually similar images having almost the same hash output: (b) is derived from (a) by luminance changes in small areas, PSNR=28.5db

(a) (b)

Figure 2. Valid attack: visually different images with the same hash output, PSNR=9.2db

in $[0, 1]$ in our experiments, and we can choose other stronger non-linear pseudorandom generator instead. Finally we'll adjust these parts into a specific number of blocks.

Step 1: ***Dividing into grids*** In this step, we regard the whole image as a square of size $L * L$. Then we divide the square into $n * n$ grids with horizontal and vertical parts. For the vertical direction, we cut the image from left to right $n - 1$ times. Given a rectangle with size $W * L$ each time, we select a width w from a set $\{w_1, w_2, ..., w_t\}$, where t is the number of choices, and cut the rectangle into two rectangles with size $w * L$ and $(W - w) * L$, then keep on cutting the latter rectangle until we have n divisions. A pseudo-random number controlled by Key_1 determines the width w each time. For horizontal direction we do similarly. The divisions are shown in Fig. 3. Note that if all grids are of the same size, equivalently they are squares, then the attacker can enumerate the size of grids resulting in security risks.

Step 2: ***Random walk*** For convenience, we mark every grid with a Cartesian coordinate (x, y), and $grid(x, y)$ denotes the grid lies on row x and column y. Next we'll combine these grids into several connected blocks. A block is said to be *connected* if and only if any two grids in the block are connected. At the beginning, no grid belongs to any block. Then we use a random walk method to connect grids into a block assigned some color c. More specifically, we'll traverse these grids from left to right, from up to down. If $grid(x, y)$ is not colored, we'll regard it as the *center grid* and execute the random walk algorithm. For every

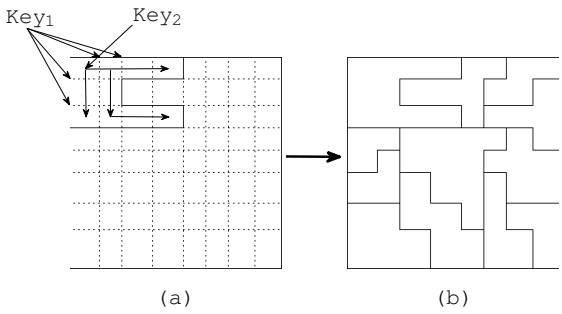

Figure 3. Random blocking by two keys: (a) the image was first divided into $8 * 8$ grids; (b) the grids was combined into several blocks

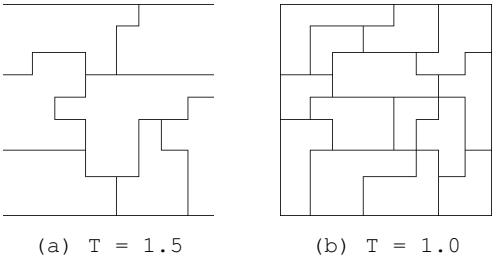

Figure 4. Image division with different T

$gird(x', y')$, we define a distance measure $D(x', y', x, y)$ from the center grid $grid(x, y)$:

$$D(x', y', x, y) = |x - x'| \cdot Rand(Key_2) + |y - y'| \cdot Rand(Key_2).$$

The Traverse algorithm is to combine the grids into zigzag blocks. Every time a new block is formed by the *Random_Walk* algorithm, and the zigzag blocks are shown in Fig. 3 (b). The pseudo-code is as follows.

Algorithm 1 $Traverse(grid)$

Input: The $grid(1..n, 1..n)$
Output: Randomly divided zigzag blocks

```
c ← 0
for x = 1 to n do
  for y = 1 to n do
    if grid(x, y) hasn't been colored then
      c ← c + 1
      Random_Walk(x, y, x, y, c)
    end if
  end for
end for
```

Note that T is a threshold closely related to the number of zigzag blocks as shown in Fig. 4, and $D(i, j, x, y)$ decides the probability whether the grid will be colored in the loop of the random walk algorithm. Larger $D(i, j, x, y)$ implies larger probability of adding $grid(i, j)$ into a block centered at $grid(x, y)$. We expect that a center grid is surrounded by connected blocks, so that extracted statistical features from each block well represent local information, and $D(i, j, x, y)$ just helps us on this.

Algorithm 2 $Random\_Walk(x, y, x', y', c)$

Input: The *center* $grid(x, y)$, the present $grid(x', y')$ and the color c
Output: A connected block colored with c

```
color grid(x', y') with c
for all uncolored grid(i, j) adjacent to grid(x', y') do
  if D(i, j, x, y) ≤ T then
    Random_Walk(x, y, i, j, c)
  end if
end for
```

Step 3: ***Adjusting*** Now we have divided the image into several parts. To ensure no rectangle exists, we split every rectangle by random walk again. Actually the chances are very slim to obtain a rectangle in the divided blocks. Then we can extract statistical feature, e. g., expectation of luminance for every block to produce a final hash output. For the hash output with fixed size, we first choose an appropriate threshold T to generate a little more blocks than expected, and then we iteratively choose the block with the smallest size and combine it with a connected block until we get the desired number of blocks. The shapes of the blocks are hard to estimate because of the randomness.

3. Security Analysis

For security reason, we increase the randomness by involving two keys Key_1 and Key_2. In this section, we'll show how security is enhanced and how this method protects image hashing from the iterative search attack.

3.1. Contribution of Random Grids Controlled by Key_1

With the knowledge of division of blocks, replacing a block with the same statistical property would be an effective attack. If grids are of the same size, i. e., they are squares, the attacker may simply enumerate the size of the square. This motivates us to randomize the size of every grid.

For the first i, $1 \le i \le n-1$, steps, choose the $width(i)$ randomly from a set $\{w_1, w_2, ..., w_t\}$ whose elements are different, where each w_i is close to $\lfloor \frac{L}{n} \rfloor$. For the final step, we just let $width(n) = L - \sum_{i=1}^{n-1} width(i)$. Consequently, there are t^{n-1} ways to divide the image horizontally since we choose a width from that set each time with equal possibility, and cutting vertically is similar. So, the total number of ways for division is t^{2n-2}. It's infeasible for the attacker to find how we divide image by using exhaustive search for the appropriate t and n.

We also require that the greatest common divisor of the sizes of grids equals 1. Otherwise, the attacker could divide the image into grids whose size is the greatest common divisor of the numbers in the set, and then launch an attack by changing units with size of that common divisor.

3.2. Contribution of Random Walk by Key_2

With a number of pairs $(Image, V)$ similar as chosen-plaintext attack in cryptography [9], the original image hashing can be cracked because one rectangle can be uniquely determined by its top-left and bottom-right points. To avoid such kind of attacks, we divide image into zigzag blocks by using another Key_2 to randomize the ways of combining the grids.

3.3. Enhanced Security Explanation

A. Swaminathan et al. [14] proposed the differential entrophy as the security measure for some image hashing schemes, where they assumed that the attacker only knew the original image and the hashing algorithm, but did not know the Key used and the actual hash values. But for our scenario we assume that the attacker can observe the $(Image, V)$ pairs and know the image hashing scheme. Next we will give a description why our proposed random zigzag blocking has enhanced security, in terms of order of magnitude of searching space for the number of possible shapes of one block. Some basic notations on computational complexity can be found in [15].

Consider a block $\mathcal{B}$ in our division scheme and let C be its convex rectangle cover. Then the number of internal points(pixels) of C is right its area, $AREA(C)$. The search strategy proposed in [8, p. 465] is to check all possible rectangles nearby C and find the best location of one rectangle according to the normalized correlation. Since one rectangle can be uniquely determined by top-left point and bottom-right point, the number of possible nearby rectangles can be estimated as $\Theta(AREA^2(C))$ in terms of order of magnitude. The unicity distance of approximate 40 was given for determining the suitable rectangle [8].

However, for our case the $\mathcal{B}$ is of zigzag shapes, the search space should be $\Theta(2^{AREA(C)})$, excluding $\Theta(AREA^2(C))$ number of rectangles.

Because $AREA(C) \geq AREA(\mathcal{B})$ (identity holds only for rectangle case), and the minimum area of all blocks can be larger than a specific value under control, e. g., $3*10^2$ in our experiment, searching for the zigzag shape of a given block can not be expected even for estimating approximate shape of the zigzag block, equivalently considering downsampled version of the image or smaller $AREA(\mathcal{B})$.

Overall, the searching space for determining the location of a given block $\mathcal{B}$ is $\Omega(AREA^2(\mathcal{B}))$ for rectangle case, and $\Omega(2^{AREA(\mathcal{B})})$ for zigzag block case. Provided the adversary has m pairs of $(Image, V)$, then the success probability of the iterative search attack is approximately $\Theta(\frac{m \cdot AREA^2(\mathcal{B})}{40 \cdot 2^{AREA(\mathcal{B})}})$.

Therefore, the existing iterative search attack is infeasible due to the essential difference of the order of magnitude for searching space, and the unicity distance of our proposed scheme should be very huge, which guarantees the security.

4. Experimental Results

In our method, we just use $Key = Key_1 \| Key_2$ to divide images into several blocks. When repeatedly executing the random walk algorithm, the number of initial blocks are stable. In our experiments, with $n = 20$ and $T = 1.5$, the number of initial blocks is in $[55, 70]$. If we want to divide the image into a fixed number of blocks, we may select appropriate parameters and adjust some of the initial blocks.

By using the random walk method, we enhance the security of the image hashing. Meanwhile we do not sacrifice the robustness. We used an image dataset including 30 images from the USC-SIPI dataset and 1467 images from VOC2012 [16] dataset. The size of every image was fixed to $256*256$. Then we created some different versions for each image by several content-preserving modifications. In practice, we transformed each image into an 8-bit grey-scale image, which is divided into $20 * 20$ grids. Then we used the random walk obtaining 48 blocks. Expectations of luminance were extracted from those blocks. After quantization and compression, every block was represented by 3 bits and the final hash output was of 144 bits. The Normalized hamming distance of some content-preserving modifications for two blocking methods is shown in Fig. 5, which reveals our proposed image hashing scheme by the random walk is comparable with previous method on distinguishing ability for those perceptually similar images. For perceptually different images, the mean value of normalized hamming distance is very close to 0.5. Moreover, the variance of normalized hamming distance is 0.0030 for our method, and 0.0038 for previous method. Based on the experimental results, the normalized Hamming distance is effective to determine the similarity of images under several modifications. We suggest 0.15 an upper bound to distinguish similar images. Besides, the PSNR varies from 10db to more than 30db between the same images under these modifications. Compared with the normalized Hamming distance, it's difficult to use PSNR to distinguish similar images, since the PSNR between perceptually different images is usually under 15db.

5. Conclusion

We have proposed an image hashing scheme including random walk on zigzag blocking. We divide a given image into zigzag blocks during the feature extraction stage, in which Key_1 is used to generate a random grid and another Key_2 is used for combination of small grids by using random walk. From discussions on security, the division of zigzag blocks related to $Key = Key_1 \| Key_2$ can hardly be analyzed from the existing methods due to exponential searching space, so the iterative search attack in [8] is infeasible and the security is enhanced. Additionally, the proposed scheme is robust to slightly incorrect segmentation since quantization is used in each block, and experiments of some content-preserving modifications demonstrate the robustness, implying promising applications in image retrieval or authentication without additionally frequent key updates on image hashing. We will introduce an appropriate measure of perceptual similarity, and consider how to choose the optimal parameter T combining security and robustness in future.

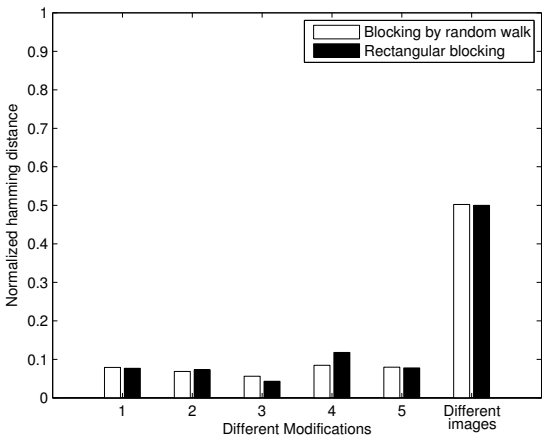

Figure 5. Normalized hamming distance under different modifications: 1-adding 0.05 salt and pepper noise; 2-deleting 5 lines randomly; 3-4 $*$ 4 median filtering; 4-cropping up to 10% of image area; 5-affine1 attack from stirmark software. The last one indicates the normalized hamming distance between perceptually different images.

Acknowledgment

The authors also show their sincere thanks to Prof. Jufeng Yang, and Prof. Mingming Cheng for their valuable comments and discussions on earlier versions of this work. This work is partially supported by NSF of China (Grant No. 61003070, 61373018, and 11301288), Program for New Century Excellent Talents in University (grant number: NCET130301) and the Fundamental Research Funds for the Central Universities(Grant No. 65141021). The last corresponding author is also supported by China Scholarship Council.

References

[1] V. Monga and M. K. Mihak, "Robust and secure image hashing via non-negative matrix factorizations," *Information Forensics and Security, IEEE Transactions on*, vol. 2, no. 3, pp. 376–390, Sep. 2007.

[2] I. H. Laradji, L. Ghouti, and E. H. Khiari, "Perceptual hashing of color images using hypercomplex representations," in *Image Processing, IEEE International Conference on*, Sep. 2013, pp. 4402–4406.

[3] S. S. Kozat, R. Venkatesan, and M. K. Mihcak, "Robust perceptual image hashing via matrix invariants," in *Image Processing, IEEE International Conference on*, Oct. 2004, vol. 5, pp. 3443–3446 Vol. 5.

[4] W. Li and N. Yu, "Rotation robust detection of copy-move forgery," in *Image Processing, IEEE International Conference on*, Sep. 2010, pp. 2113–2116.

[5] C. Hsu, C. Lu, and S. Pei, "Secure image hashing via minimum distortion estimation," in *Image Processing, IEEE International Conference on*, Nov. 2009, pp. 1281–1284.

[6] M. Johnson and K. Ramchandran, "Dither-based secure image hashing using distributed coding," in *Image Processing, IEEE International Conference on*. IEEE, 2003, vol. 2, pp. II–751.

[7] R. Venkatesan, S. M. Koon, M. H. Jakubowski, and P. Moulin, "Robust image hashing," in *Image Processing, IEEE International Conference on*, 2000, vol. 3, pp. 664–666 vol.3.

[8] Y. Mao and M. Wu, "Unicity distance of robust image hashing," *Information Forensics and Security, IEEE Transactions on*, vol. 2, no. 3, pp. 462–467, Sep. 2007.

[9] B. Schneier, *Applied Cryptography*, John Wiley & Sons, 1996.

[10] Y. Matias and A. Shamir, "A video scrambling technique based on space filling curves," in *Advances in Cryptology - CRYPTO '87, A Conference on the Theory and Applications of Cryptographic Techniques, Santa Barbara, California, USA, August 16-20, 1987, Proceedings*, 1987, pp. 398–417.

[11] A. Massoudi, F. Lefèbvre, and M. Joye, "Cryptanalysis of a video scrambling based on space filling curves," in *Proceedings of the 2007 IEEE International Conference on Multimedia and Expo, ICME 2007, July 2-5, 2007, Beijing, China*, 2007, pp. 1683–1686.

[12] S. Arora and B. Barak, *Computational complexity: a modern approach*, Cambridge University Press, 2009.

[13] C. M. Grinstead and J. L. Snell, *Introduction to Probability*, American Mathematical Society, 2012.

[14] Ashwin Swaminathan, Yinian Mao, and Min Wu, "Robust and secure image hashing," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 215–230, 2006.

[15] J. Kleinberg and E. Tardos, *Algorithm Design*, Prentice Hall, 2006.

[16] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge 2012 (voc2012) results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index. html.