Image Hashing Algorithm: An Analysis and Improvement

Seyed Amirhossein Tabatabaei[1]

[1]*Department of Computer Science, Faculty of Mathematical Sciences, University of Guilan, Rasht, Iran*
amirhossein.tabatabaei@guilan.ac.ir

Abstract

Image authentication code algorithms are schemes wherein the authenticity of an image is considered in a robust war. As the images are mainly subjected to some authorized modifications, such schemes must be able to accept the authorized changes while rejecting the malicious ones. This article analyzes an image authentication algorithm based on error detection code. The image authentication scheme utilizes a type of error detection code in order to encode the mixed most significant bits (MSB) intensity value of each image pixel. The secrecy of system is based on two secret keys. The algorithm provides an acceptable robustness and elaborate design however, it suffers from some security features that leave a gap for improvement. These features are analyzed and will be employed to show the vulnerabilities of the scheme. Also, some solutions are proposed to elaborate the algorithm more. The solutions will increase the strength of the scheme while keeping the robustness.

Keywords: *Image Authentication, Robustness, Image Hashing, Error detection and correction.*

1 Introduction

Appearance of the advanced technologies in multimedia processing techniques like image processing softwares facilitates performing the illegal actions on the digital multimedia object. Violation of image ownership, unauthorized duplication and redistribution and malicious copying and manipulating acts are examples which indicate high demand for image authentication. Image authentication aims to verify the authenticity of images which are subjected to some modification and/or unauthorized manipulation. Image authentication schemes are mainly based on the perceptual image hashing or watermarking techniques. A shared secret key is used in hash generation or watermark embedding process and verification to provide security of the scheme. A perceptual image hashing algorithm differs basically from a cryptographic hash function: the generated hash

corresponding to the images with semantically identical content are equal or very close to each other. This property addresses the robustness of perceptual image hashing. However, the independence of the hash output for perceptually different images must be ensured [1]. The concept of security in a hash-based image authentication is a challenging issue. It refers to the ability of the attacker to find perceptually different images with almost equal hashes after observing sufficient number of image-hash pairs. Also, it ensures that the secret key can not be compromised and no image hash or watermark can be generated without knowledge about the secret key.

In this paper, the analysis of an interesting proposed code-based image authentication scheme [2] is presented. The proposed scheme is based on the initial work given in [3] whose perceptual hashing algorithm uses Hamming code technique to generate and embed parity bits in the pixels. To provide the security two secret keys are involved in the embedding process. The main contribution of this paper is the analysis of the security aspects of the authentication scheme for further enhancement. In fact it is shown that the engaged keys do not strengthen sufficiently the security to be met by an image authentication mechanism. Furthermore a solution will be proposed and discussed in details. The rest of this paper is organized as follows. A short description on the related works is given in Section II. Section III describes the subjected coding-based image authentication scheme. The analysis of the scheme is given in Section IV followed by the proposed solution in Section V and security discussion in Section VI. Section VII concludes the paper.

2 Related Works

There is a large body of research works in the literature in the field of image authentication. They are categorized according to their construction technique and application. Classification based on the robustness of the scheme is mostly used in the taxonomy of image authentication methods. According to this classification they are categorized into two major groups performing hard authentication and soft authentication respectively [4]. The main techniques in the first group where the robustness and the number of acceptable modifications is limited are based on standard cryptography and fragile watermarking. In the second group of image authentication schemes, the level of robustness is higher than the first group and a wider range of authorized modifications is accepted while the malicious manipulations are supposed to be detected. The main technique in this group is based on the semi-fragile watermarking and content-based signature wherein the semantic content of an image is extracted as a feature in order to generate a digital signature [4, 5, 6, 7, 8]. Despite of a fast progress in extracting and analyzing the image data used in the image authentication scheme, there are relatively less attention to the corresponding security analysis as one of the main pillars in this design. As one of the leading works regarding to this matter, Swaminathan, et.al., [9] proposed to use differential entropy to evaluate the security of some existing image hashing schemes in the literature. Although it was shown later that the proposed ap-

proach does not justify due to existing scale variant property [10]. In another approach unicity distance was used to determine the maximum number identical used keys for an image authentication scheme [11]. The fundamental works which consider the generic security of perceptual hashing in information theory viewpoint emerge in [1, 12]. Some works also analyze the security problems of the existing schemes individually [13, 14]. A secure framework for general perceptual image hashing also has been proposed in [15].

3 Image Authentication Method based on Applying Hamming Code on Mixed Bits

The proposed method by Chan consists of three procedures including the embedding, the detecting and recovery procedure [2]. It is based on the initial scheme firstly introduced in 2007 [3] where some improvements have been applied. In this section just the recent scheme [2] is recalled and described. The embedding procedure generates the parity check from each pixel intensity value and embed it into another pixel intensity value. In this procedure a Hamming code scheme (Hamming(7, 4)) is used to generate three parity check bits from four most-significant bits of a pixel. To generate the parity check bits, at first the order of the first four most-significant bits of each pixel indicating the data bits is reversed and then a three-bit parity check value is produced. The three-bit parity check value is rotated and embedded into another pixel specified by a transformation. The rotation operation is performed by the use of a sequence of random numbers extracted from a random number generator based on a secret key k_1. Let the image size be $N \times N$, the aforementioned parity check of the pixel P_i with the original bit order J and new bit order J' goes as $J' = (J + R_i) \bmod 3$. For the sake of simplicity the rotated parity check of each pixel is called authentication data. The authentication data bits are embedded into another pixel using the Torus automorphism and Modulus function as follows [2]. Let $P_i = (x_i; y_i)$ be the first pixel to be processed. The next pixel to be processed is specified by Torus automorphism:

$$\begin{bmatrix} x'_i \\ y'_i \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ k_2 & k_2 + 1 \end{bmatrix} \times \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{1}$$

In the above equation k_2 is the second secret key. The authentication data of pixel P_i is modified using the Modulus function before embedding into the new pixel position $P'_i = (x'_i; y_i)$. In the Modulus function a secret value s of k bits length is embedded in a pixel value y. Let $d = s - (y \bmod 2^k)$, the new value of y denoted by y_0 is computed according to $y_0 = d_0 + y$ where d_0 is defined as follows.

$$d_0 = \begin{cases} d & , \text{ if } -\lfloor (2^k - 1)/2 \rfloor \le d \le \lfloor (2^k - 1)/2 \rfloor \\ d + 2^k & , \text{ if } -2^k + 1 \le d \le -\lfloor (2^k - 1)/2 \rfloor \\ d - 2^k & , \text{ if } \lceil (2^k - 1)/2 \rceil \le d \le 2^k \end{cases} \tag{2}$$

This procedure is continuing by embedding the authentication data of pixel P'_i into another pixel specified by the Torus automorphism and the Modulus function until the

cycle is completed by reaching the first pixel P_i. Then the process is repeated to process the whole pixels set. At the end of this procedure the authentication data of each pixel is embedded into another pixel. The detecting procedure localizes and marks the tamper area for the sake of recovery in the recovery procedure. In this phase the authentication data of each pixel is extracted firstly and compared with the regenerated value. The details of this procedure are given in [2].

4 Analysis of Chan's Scheme

The presented image authentication scheme uses two main secret keys in the embedding phase. This first key is used as a seed to generate a sequence of random numbers for the rotation operation and the second key is used in the Torus automorphism to provide ambiguity in locating the host pixel. According to [2, 3], the purpose of employing two secret keys is to provide the security against cases when one key is recovered by the attacker. In fact, if the keys are disclosed, the attacker can design different attack scenarios like impersonation or substitution attack. In this section, we show that when the Chan's image authentication scheme is applied on the chosen images with specified pattern (chosen plaintext attack model) then the second secret key can be recovered trivially followed by an exhaustive search to recover the first key. The second key can be recovered by inquiring two authenticated images I, I' differing in one pixel $(x_i; y_i)$, by solving the Equation (1) for k_2. However, it might happen that more than one pixel in the row $i+j$ in I' will be modified. In this case the key k_2 cannot be determined uniquely. The main security issue of the discussed image authentication scheme is the lack of enough diffusion which is required for a robust image authentication scheme. In fact, the required diffusion in image authentication schemes is not evaluated as in classical cryptographic primitives. It should not hinder the robustness without compromising the security. This issue is used to recover the value of the second key uniquely. To explain this property, let the image I consisting of the pixels all with the same intensity value and its authentication embedded version J are given. The attacker manipulates a pixel value P_i at a chosen position $(x_i; y_i)$ such that $P'_i = P_i + d_i$. The choice of d_i and the constant pixel intensities will be explained later. To bypass the influence of the first key, it is supposed that the parity check (and therefore the authentication data) of the pixel p_i is $(000)_2$ or $(111)_2$. With this assumption reordering the bit positions does not influence on the recovered value. To satisfy the latter assumption, the four MSB values of P_i must be one of the possible values $(0000)_2$, $(0111)_2$, $(1000)_2$ and $(1111)_2$ based on the Hamming code generator matrix. The strategy of the attack is to manipulate one-pixel intensity value such that its authentication data just modifies the host pixel while embedding. To observe how the attack works, the simplest case is considered. Let $P_i = 0$ and $d_i = 96 = (01100000)_2$. Then the authentication data of P'_i is calculated as $s'_i = (101)_2$ or $s'_i = (011)_2$ or $s'_i = (110)_2$ depending on the corresponding random number R_i. Let the host pixel wherein s_i is embedded be denoted by P_j. According to Equation (2), the updated value of P_j indicated by P'_j would be one of the intensity values $(00000101)_2$,

$(00000011)_2$, $(00000110)_2$ receptively. It is observed trivially that as the four MSBs of P'_j are left unchanged, its authentication data will not be affected as well. This indicates the termination of the diffusion in the embedding process. Now, with the knowledge on the position of modified pixel (j), Equation (1) and aforementioned weakness property the value of the second key can be extracted. This attack in a chosen plaintext model can be easily generalized. It is supposed that the attacker has a black-box oracle access to the image authentication scheme indicating that she can request the watermarked image on any chosen input image. The attack is described in the following steps:

- Step One (offline step): The attacker chooses an image I whose pixels have the same intensity value $P_i = p$. To skip the impact of the first secret key (to ease the attack scenario as mentioned before), p can be selected from the set $[0; 15] \cup [112; 143] \cup [240; 255]$.
- Step Two: The watermark image I' is calculated using the image authentication scheme. It is easy to verify analytically that for some values like $p = 0, 8, 112$, $I = I'$.
- Step Three: The attacker modifies a pixel value P_i as $P'_i = p + d_i$. The choice of d is of great importance and plays a crucial role. d_i is selected such that $d > 0$ and the MSBs of the host pixel will not be influenced however LSBs will be modified.
- Step Four: The attacker calculates the watermarked image I'' using the image authentication scheme.
- Step Five: A binary difference matrix indicating the difference between I' and I'' is calculated as $[Dij] = 1 - d_{(I'ij - I''ij)}$.

The elements corresponding to '1' in the above difference matrix is used to solve the system of linear equations 1 to recover the unknown value k_2. The secret permutation generated by the first secret key using a sequence of random numbers is extracted by an exhaustive search using some chosen MSBs which generate the required parity check values. To launch the attack, the attacker just requires two calls to the image authentication scheme for recovering the second key. Also, secret permutation used before embedding can be extracted with $2N^2$ calls resulting in maximum $2N^2 + 2$ total calls with chosen images. Considering the typical image sizes corresponding to N = 512, 1024, the complexity is of order 219 or 221 which is negligible in cryptanalysis.

This attack can be extended to the earlier version of the Chan's image authentication scheme presented in [3] trivially. In the initial scheme, the LSB replacement is used instead of the Modulus function and the pixels are being processed one by one from left to right according to the Torus automorphism. Also the bit reversion on the MSBs are not applied. The attack conditions are met much easier in this case. In fact the choice of difference value di in the third step is less restricted: all of the values of d which keep the encoded value of $(p + d) >> 4$ unchanged (while changing the LSBs of $p + d$) would be valid for this attack. However the complexity of the attack remains unchanged.

5 Improving and Enhancing the Authentication Model

5.1 Enhancing the Scheme

To enhance the Chan's scheme and increase the security against the aforementioned attack while keeping its original elaborated robustness, the following modifications will be suggested on two steps of the scheme.

1) At first as a preprocessing step, the image undergoes some preprocessing steps including a bilinear interpolation and mapping to a fixed-size square image. An optional low-pass filtering can be applied on the image to provide minor robustness.

2) To start processing the image pixels, four pixels are chosen at random. The main modification applies in calculating the parity check parts of the pixels in the processing step. Similar to the original scheme a Hamming code with parameters $n = 7$ and $k = 4$ is used for encoding purpose. However all four pixels contribute in generating each parity check value in the proposed scheme. The details are as follows: let the four selected pixels are denoted by P_1, P_2, P_3, and P_4 whose first four MSBs are indicated by $p_{i1}p_{i2}p_{i3}p_{i4}$ for $i = 1, 2, 3, 4$.

$$W_1 = p_{11}p_{22}p_{33}p_{44}, W_2 = p_{21}p_{12}p_{43}p_{34}, W_3 = p_{31}p_{42}p_{13}p_{24}, W_4 = p_{41}p_{32}p_{23}p_{14} \quad (3)$$

By this arrangement of the MSBs, each generated authentication bit would be a function of all processed pixels. Also each new nibble W_1, W_2, W_3, W_4 contains all significant values. In the proposed modified scheme four pixels are used in each step for extracting the authentication data. To increase the entropy of the intermediate bits, the nibbles are added via XOR operation with random words (R_1, R_2, R_3, R_4) generated from a key-based pseudo random number generator PRNGk as follows.

$$W'_i = W_i \oplus R_i \quad , i = 1, 2, 3, 4 \quad (4)$$

Some proper Boolean functions can be used instead to introduce some nonlinearity into the scheme.

3) Four 3-bit parity check values as authentication data of four pixels are computed by applying the Hamming code on W'_1, W'_2, W'_3, W'_4 respectively. The authentication bits are firstly concatenated and then permuted using a key-based random permutation before embedding. Finally the next four pixels to be processed are determined by Torus transformation wherein the permuted parity check bits are embedded.

The recovery procedure in the modified image authentication is almost the same as recovery procedure in the original Chan's scheme. In the recovery phase, the original pixel values (MSBs) are recovered by applying reverse permutation and decoding operation followed by XORing with the random nibble words. However in the modified scheme, four pixels are recovered at the same time. Hamming codes are able to detect up to two erroneous bits whereas are able to correct one bit without detection. The possible tampered or manipulated areas of the image are localized further and tried to

be corrected using the correction capability of the error-correcting code. Whenever the occurred errors are beyond the capability of Hamming code (more than one bits) the correction attempt is further continued using adjacent pixels. If the latter trial fails, the erroneous pixels are marked as tampered or manipulated area of the image.

5.2 Extension to 16-bit Grayscale Images

The scheme has been so far set for the cases wherein images are grayscale and the pixel depth is an 8-bit value. However, there are many applications specially in medical images in which the pixel depth is 16 bits. The proposed scheme with the existing coding structure is not applicable in such cases due to the code word length. To extend the scheme, the Hamming error correcting code with parameters $n = 15$, $k = 11$ ($[15, 11, 3]_2$) with higher coding rate can be used. In the latter code, four parity check bits for eleven MSBs of each pixel are generated and embedded into the four LSBs of other pixels values corresponding to the revised algorithm described in Section V. However the embedding procedure is modified slightly and eleven pixels are selected in each step to generate four parity check words for embedding in the eleven target pixels determined by Torus transformation (the value of k is set to 4). When the image size number is not a multiple of 11 which is in the most of the cases, the last pixels to be processed are processed naively one by one which does not significant impact on the security. Similar to the initially proposed scheme, each selected pixel will contribute in generating each parity check word. However to avoid further computational cost and size adjustment problem one can update four pixels in each step like the latter enhanced scenario as follows.

$$
\begin{aligned}
W_1 &= p_{11}p_{22}p_{33}p_{44}p_{15}p_{16}p_{17}p_{18}p_{19}p_{110}p_{111} \\
W_2 &= p_{21}p_{12}p_{43}p_{34}p_{25}p_{26}p_{27}p_{28}p_{29}p_{210}p_{211} \\
W_3 &= p_{31}p_{42}p_{13}p_{24}p_{35}p_{36}p_{37}p_{38}p_{39}p_{310}p_{311} \\
W_4 &= p_{41}p_{32}p_{23}p_{14}p_{45}p_{46}p_{47}p_{48}p_{49}p_{410}p_{411}
\end{aligned}
$$

Further processing is the same as the enhanced scheme in the latter subsection in which the 11-bit words are added via XOR operation with random words of the same size before encoding. The generated parity check bits are permuted and then embedded into the LSBs accordingly. Like the original scheme, erroneous bits can be detected or one can be corrected without detection of other erroneous bits. However the security of the scheme will be strengthened due to longer parity check values. This point is addressed in the following section.

6 Revisiting the Analysis of the Modified Scheme

The main security aspect of the modified scheme is that any significant changes in a pixel (MSBs) will be reflected to up four host pixels which was one pixel at max in the original Chan's scheme. Also expanding the pixels processing to four branches simultaneously

instead of one in the original scheme will increase the diffusion. However, it is worth to mention that the robustness of the scheme is not influenced in the proposed modification due to using the same coding and embedding building blocks. Here it is shown that the proposed modification strengthens the security of the original scheme significantly against the mentioned key recovery attack. Let the image be of size $N \times N$. To recover the first secret key corresponding to Torus automorphism uniquely, the attacker does not have difficulties due to Equation 1 and negligible image size in this sense. However the cost of finding the positions is of order $O(N^6)$. Also the cost of recovering the secret random permutation is approximately 12!. So the whole computational complexity required for recovery of secret components excluding the generated random numbers would be $C = O(N^6) + 16 \times 12!$. The generated random numbers must be disclosed via exhaustive search as well which increases the complexity.

The complexity would be higher for the extended version as well. In fact, the cost of recovering secret random permutation used in the parity check part is 16! which simply affects the second term of the equation. Concerning the commonly used values for N and pixel depth, the attack complexity C has increased significantly however some low thresholds on N and pixel depth can be set to satisfy desired security strength according to the existing computational power. To enhance the security more, a keyed pseudo-random function whose label is dependent to the image can be used instead to embed the generated random output into the MSBs. In this case the labeled function provides the intrinsic security strength of a cryptographic authentication function. Further improvement in the accuracy of the tamper localization is possible as well by using another code structure with higher detection and correction capability. The latter suggestions are considered as future work since there is not enough room left in this paper.

7 Conclusion

An image authentication scheme based on Hamming code and rearrange MSBs of pixel intensity values has been analyzed. Some modifications in preprocessing step and computation of authentication data have been proposed to enhance the security and accuracy of the scheme. The proposed modifications strengthen the scheme against the aforementioned flaw and generalize the scheme to be used on grayscale images with 16 bits pixel intensity values. However there is still some room left for further enhancement to elaborate the design more. Using different types of systematic error-correcting schemes which increases the detection and correction capability and engaging a keyed labeled cryptographic pseudorandom function would be a potential track for the improvement and further secure designs.

References

[1] O. Koval, S. Voloshynovskiy, F. Beekhof, and T. Pun, “Security analysis of robust perceptual hashing,” in Electronic Imaging 2008, pp. 681906–681906–10, International Society for

Optics and Photonics, 2008.

[2] C.-S. Chan, "An image authentication method by applying hamming code on rearranged bits," Pattern Recognition Letters, vol. 32, no. 14, pp. 1679 – 1690, 2011.

[3] C.-S. Chan and C.-C. Chang, "An efficient image authentication method based on hamming code," Pattern Recogn., vol. 40, pp. 681–690, Feb. 2007.

[4] A. Haouzia and R. Noumeir, "Methods for image authentication: a survey," Multimedia Tools and Applications, vol. 39, no. 1, pp. 1–46, 2008.

[5] Ling Du, Anthony T.S. Ho, Runmin Cong, Perceptual hashing for image authentication: A survey, Signal Processing: Image Communication, Volume 81, 2020, pp. 115713.

[6] M. Sajjad, I. U. Haq, J. Lloret, W. Ding and K. Muhammad, "Robust Image Hashing Based Efficient Authentication for Smart Industrial Environment," in IEEE Transactions on Industrial Informatics, vol. 15, no. 12, pp. 6541-6550, Dec. 2019, doi: 10.1109/TII.2019.2921652.

[7] Karsh, R.K. LWT-DCT based image hashing for image authentication via blind geometric correction. Multimed Tools Appl 82, pp. 22083–22101, 2023.

[8] Thabit, R. Review of medical image authentication techniques and their recent trends. Multimed Tools Appl 80, 13439–13473, 2021.

[9] A. Swaminathan, Y. Mao, and M. Wu, "Robust and secure image hashing," Information Forensics and Security, IEEE Transactions on, vol. 1, pp. 215 – 230, 2006/06// 2006

[10] G. Zhu, J. Huang, S. Kwong, and J. Yang, "A study on the randomness measure of image hashing," IEEE Transactions on Information Forensics and Security, vol. 4, pp. 928–932, Dec 2009.

[11] Y. Mao and M. Wu, "Unicity distance of robust image hashing," IEEE Transactions on Information Forensics and Security, vol. 2, pp. 462–467, Sept 2007.

[12] O. Koval, S. Voloshynovskiy, P. Bas, and F. Cayre, "On security threats for robust perceptual hashing," in Media Forensics and Security, vol. 7254, p. 72540H, feb 2009.

[13] T. Uehara and R. Safavi-Naini, "On (In)security of A Robust Image Authentication Method", pp. 1025–1032. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002.

[14] M. Heidari, S. Samavi, S. M. R. Soroushmehr, S. Shirani, N. Karimi, and K. Najarian, "Framework for robust blind image watermarking based on classification of attacks," Multimedia Tools and Applications, Nov 2016.

[15] D. Hu, B. Su, S. Zheng, and Z. Zhang, "Secure architecture and protocols for robust perceptual hashing," in Computational Intelligence and Security (CIS), 2013 9th International Conference on, pp. 550–554, Dec 2013.