May 2003

Compression tolerant DCT based image hash

C. Kailasanathan
University of Wollongong

R. Safavi-Naini
University of Wollongong, rei@uow.edu.au

P. Ogunbona
University of Wollongong, philipo@uow.edu.au

Compression tolerant DCT based image hash

Abstract

With the advent of Internet image authentication has become a central part of research in security. Since JPEG has recommended discrete cosine transform as one of the steps in image compression systems, a hash function which utilizes discrete cosine decomposition is desirable. In this paper, we propose a discrete cosine based hash function which distinguishes acceptable level of compression from image processing modifications such as Median filtering, Gaussian noise addition, and FMLR attack. To increase manipulation detection, we optimize the number of AC coefficients needed in smoothing.

Disciplines

Physical Sciences and Mathematics

Publication Details

This article was originally published as: Kailasanathan, C, Safavi-Naini, R & Ogunbona, P, Compression tolerant DCT based image hash, Proceedings 23rd International Conference on Distributed Computing Systems Workshops, 19-22 May 2003, 562-567. Copyright IEEE 2003.

Compression Tolerant DCT Based Image Hash

C.Kailasanathan and R.Safavi Naini,
Centre for Computer Security Research,
Department of Computer Science, University of Wollongong,
Northfield Avenue,NSW 2522, Australia.
e-mail: ck12@uow.edu.au and rei@uow.edu.au
P.Ogunbona
Motorola Australian Research Centre,
Level 3,12 Lord Street, Botany, NSW 2019,
Australia.
e-mail: pogunbon@arc.corp.mot.com

January 25, 2003

Abstract

With the advent of Internet image authentication has become a central part of research in security. Since JPEG has recommended discrete cosine transform as one of the steps in image compression systems, a hash function which utilizes discrete cosine decomposition is desirable. In this paper, we propose a discrete cosine based hash function which distinguishes acceptable level of compression from image processing modifications such as Median filtering, Gaussian noise addition, and FMLR attack. To increase manipulation detection, we optimize the number of AC coefficients needed in smoothing.

1 Introduction

Image authentication plays an important role in security and communication. Images are being transferred over the Internet and are readily available for access from any part of the world and without introducing authentication mechanism, it is almost impossible to distinguish if an image is original or being manipulated.

Using cryptographic methods to authenticate image data will result in an unworkable systems or unacceptable systems because data authentication are sensitive to single bit change in the original data while image authentication systems need to be mainly content sensitive. This is because images undergo a range of processing including lossy compression that result in changes in bits that are deemed acceptable. Such changes must be tolerable by the authentication system while it is essential for the system to remain sensitive to malicious manipulations.

The three main approaches to image authentication have been *watermarking* in which a fragile watermark is embedded in the image, *image hashing*, also called *feature extraction*, in which a digest of the image capturing its main features is generated, and *message authentication codes* (MAC), also known as keyed hashing which uses a key in hash generation process. The image hash can be encrypted or digitally signed to generate an *authentication tag* to be appended to the image. In all approaches there is also a verification system that takes an image with the authentication information including the secret key, and produces a *true/false* value, depending on the authenticity, of the image. In recent years there has been numerous proposals in each category. However assessment of security in all cases have been ad-hoc and mainly through experiments.

Efficiency of a hash or MAC based authentica-

tion system depends on the size of the key, and the amount of computation required for generation and verification of the authentication tag. In this paper we are mainly concerned with hashing method. We propose hashing methods that use discrete cosine transform of an image.

Two transform based authentication schemes known to date can be found in [1] and [2]. The one proposed by Bhattacharjee and Kutter [1] relies on visually salient image features which are extracted using scale interaction based on Mexican-Hat wavelet. In particular, their scheme looks at if the maximum of $P_{ij} = |W_i(x) - \gamma W_j(x)|$ in a circular neighbourhood with a radius of 5 around a pixel x is above a certain threshold T between scales i and j, and if the variance around x on the original image is above a certain threshold t. If the above two conditions are satisfied, the pixel x becomes a potential candidate for a feature point. The scheme proposed by Chun-Shin Lu and Hong-Yuan Mark Liao [2] relies on the fact that the interscale relationship is difficult to be destroyed by content preserving manipulations and is hard to be preserved by content changing manipulations. In particular, their scheme encodes the parent child relationship $||W_{s+1,o} - Ws, o(2x+i, 2y+j)|| < \rho$ as a sequence of bits, which becomes the hash of an image.

2 Image Hashing

An image hashing algorithm H takes an arbitrary image I and produces a bit string x_I. Unlike cryptographic hash functions, image hashing algorithms must be *bit insensitive* and produce the same value for *similar images* that are defined through a set A of *admissible transformations*. These are transformations that do not modify the main features of the image. We say an image I' *is similar to* I if $I' = a(I), a \in A$. We also need to define the set A' of *inadmissible transformations*, that consists of transformations that are applied to the whole image, such as compression at a low quality factor with the aim of damaging important details of the image, and *localized modification* of the content.

The hashing algorithm must produce 'close' values for images that are obtained through acceptable transformations, and 'distant' values for images that are obtained through forbidden ones. To quantify the notions 'close' and 'distant' we require a distance function.

Some possible measures of distance between two hash values, v and v', are the correlation coefficient $d_{CC}(v, v')$

$$d_{CC}(v, v') = \frac{\sum_{i=0}^{n}(v_i - \overline{v}_i)(v'_i - \overline{v'}_i)}{\sqrt{\sum_{i=0}^{n}(v_i - \overline{v}_i)^2} \cdot \sqrt{\sum_{i=0}^{n}(v'_i - \overline{v'}_i)^2}}$$

and Euclidean distance $d_{ED}(v, v')$.

$$d_{ED}(v, v') = \sqrt{\sum_{i=0}^{n}(v_i - v'_i)^2}$$

If two hash values are close, that is $d_{CC}(H(I), H(I')) \in (1-\epsilon, 1+\epsilon)$ or $d_{ED}(H(I), H(I')) < \epsilon$, we say the two images are similar, or *collide*, and attach a confidence level to this statement. The threshold and the confidence level needs to be determined using the set A and A'.

Image hash algorithms that are used for image authentication must have the property that *it must be difficult to construct two images I and I' that collide and $I \neq a(I')$, $a \in A$.* This property is also called *collision intractability* and ensures that a manipulated image has a different hash value and so the authentication tag is only valid for similar images.

A hash algorithm may produce a close value for an image that is obtained through a non-admissible transformation, that is a collision, or produce a distant value for an image obtained through admissible transformation. These two will result in a *false positive* and *false negative* result, when the hashing is used for image authentication. For a good hash algorithm, the probability of false positive and false negative, taken over all possible inputs must remain small.

The basic idea of our proposal is to find the critical co-ordinates that capture the main features of an image and remain essentially unchanged through acceptable transformations. Here, *essentially unchanged* means the hash value is considered *close*. For A, the set of admissible transformations, we only consider JPEG compression

with different quality levels. All other transformations, including filtering, are considered inadmissible manipulations that must be detectable by the system. However the proposed systems are flexible and can be used with other definitions of A. The larger transformation set means wider band for acceptance and hence higher rate of false acceptance.

The critical co-ordinates are found by taking the difference between the original image pixels and the corresponding low frequency (low pass) image pixels, obtained by applying a discrete cosine transform (DCT), and keeping the co-ordinates of those differences which are above a certain threshold. By doing this we essentially capture the high frequency edge co-ordinates of an image. To be more precise in selecting the final critical set, we take a range of images in an admissible transformation and use the intersection of all those critical sets as the final critical set in the hash value. To make a fair comparison, we fixed the size of the critical set by choosing the other parameters appropriately. In order to optimize detection performance, varying number of AC coefficients are incorporated in smoothing an image. The following section gives a brief description of discrete cosine transform.

2.1 Discrete Cosine Transform

Current standards for compression of still images [3] and moving images use DCT which represents an image as a superposition of cosine functions with different discrete frequencies. The transformed signal is a function of two spatial dimensions, and its components are called DCT coefficients or spatial frequencies. DCT coefficients measure the contribution of the cosine functions at different discrete frequencies. DCT has been recommended by JPEG because of its excellent energy compaction property.

3 Our Image Hash Algorithm

In this section we give the description of our image hash algorithm.

3.1 Basic Description of the Scheme

This method of hashing is based on finding the significant co-ordinates, which are considered at a range of acceptable compression levels. We call this set of points the critical set S_c for compression and keep this set as one of the parameters of the hash value. By keeping this critical set we not only avoid insignificant co-ordinates, but also reduce the length of hash. This critical set will be well estimated if many compression levels are considered.

3.2 Hash Generation Algorithm

1. Given an image I, apply a discrete cosine transform to produce the low pass image I_{LL} without sub-sampling. The low pass image using a discrete cosine transform could be obtained by applying the forward discrete cosine transform, removing the high frequency coefficients by making them 0, and then applying the inverse discrete cosine transform.

2. Let ρ be an appropriately chosen threshold.

3. Compare the corresponding pixel values of I and I_{LL} to determine the set of all pixel co-ordinates whose differences are above ρ. Let us denote this set of pixel co-ordinates as S_0

4. Assuming compression level of 25 to be used as the threshold to distinguish compression from other manipulations, JPEG compress image I at quality levels 75, 50, and 25 to produce the compressed images I_{75}, I_{50}, and I_{25}.

5. Apply steps (1)-(3) on compressed images I_{75}, I_{50}, and I_{25} to produce the set of pixel co-ordinates S_{75}, S_{50}, and S_{25}, using the same ρ.

6. Let us define a critical set $S_c = S_o \cap S_{75} \cap S_{50} \cap S_{25}$. Higher the number of compression levels are considered, the better the critical set S_c is approximated.

7. Since we haven't used many compression levels in determining the critical set S_c, a confi-

dence interval must be set on the number of matching co-ordinates for authentication.

8. Hash of an image will consist of wavelet basis used, S_c, ρ, and t-a confidence measure.

3.3 Verification Algorithm

1. Upon receiving an image I^r and its hash, apply the discrete cosine transform to produce the low pass image I^r_{LL}. This is same as the step 1 above.

2. Compare the corresponding pixel values of I^r and I^r_{LL} to determine the set of all pixel co-ordinates whose differences are above the received ρ. Let us denote this set of pixel co-ordinates as S_r.

3. If the number of matching co-ordinates in S_c and S_r belongs to the interval $[|S_c|-t, |S_c|+t]$, the received image is considered to be authentic.

4 Experimental Results

4.1 Objective and Experiment

As noted before the basic approach in deriving the critical set was to find a smooth version of the image. In DCT based approach the smooth version could be obtained by considering a subset, for example DC together with AC_1 and AC_2, and use the inverse transform to construct the smooth version. The number of coefficients that were used to construct the smooth version was investigated in this experiment. AC coefficients were diagonally incorporated (ie. DC-AC5, DC-AC9, ...). We kept the threshold ρ at a fixed value of 27, and evaluated the size of the critical set as the AC coefficients were being added.

4.2 Observations

The higher the number of low frequency AC coefficients were incorporated in smoothing an image, the lesser the number of critical points were found in the critical set. This was because when more AC coefficients were considered, the low pass image became less smooth (close to original) and hence the number of pixel differences above a certain threshold decreased. This led to reduced number of points in the critical set. Next we kept the size of the hash roughly equal, and evaluated the performance of manipulation detection with respect to various combinations of AC and DC coefficients (ie. DC-AC5, DC-AC9, ...).

Incorporating more number of AC coefficients in smoothing did improve manipulation detection for modifications such as compression at quality level of 10, and Gaussian noise addition. Performance deteriorated for FMLR with the addition of AC coefficients. Mixed results were found for Median filtering.

As we kept on adding AC coefficients, critical points found at the compression level of 25 decreased due to the closeness of compressed image and the low passed version of the compressed image.

Since the compressed image had already under gone a low pass filtering, slight low pass filtering again wouldn't pick many high frequency points in the critical set. This reduced the number of points in the critical set as more and more AC coefficients were being added even if the threshold had been lowered.

Since we included compression quality level of 25 in our critical set computation to give enough tolerance for compression, the property mentioned above limited us from going for higher number of AC coefficients being added if we wanted to have a reasonable size critical set.

From these experiments we found that for a fixed size hash, there was a trade off between the level of compression and the number of AC coefficients considered in smoothing the image. For a fixed level of compression tolerance, adding more number of AC coefficients decreased the size of the hash. This in turn reduced the manipulation detection.

For a reliable system there must be a compromise between the following:

- Size of the hash or critical set,
- Number of AC coefficients used in smoothing,

- Compression tolerance level.

The experiment we did on Lenna and Peppers images revealed that adding 44% of the AC coefficients diagonally from left to right decreased the size of the critical set to 596 (from 738) and 541 (from 1291) even though the threshold had been reduced to 1. Almost a similar observation was found for Pills image. For Paper and Corrosion images, adding 56% of the AC coefficients diagonally from left to right decreased the size of the critical set from their original sizes. Tables 1-5 in appendix illustrates our experimental results.

This shows that the optimal percentages of AC coefficients that give the highest level of detection while keeping the size of the critical set roughly equal are 33% for Lenna, Peppers and Pills images, and 44% for Paper and Corrosion images.

The above analysis explains the limitations of incorporating more and more AC coefficients in smoothing an image in our critical set based hash function surviving acceptable level of compression. We found that the optimum percentage of AC coefficients needed to maximize the manipulation detection for an equal size hash were about 33% to Lenna, Peppers and Pills images and about 44% to Paper and Corrosion images.

5 Security Analysis

5.1 ρ as a Key

Though keeping ρ as a key is not a bad idea, it has a shortcoming. We believe the shortcoming is as follows.

1. If the algorithm to find the critical set is known or public except for the key ρ, it would not be difficult to guess ρ from an image critical set pair.

5.2 Compression Levels as Keys

Keeping compression levels as keys and then choosing the ρ that gives the correct size critical set will not only avoid attacks, but also protect ρ from being guessed. This is because choosing different compression levels does not always lead to the same critical set. We have noticed this from some of our experiments. This is also not a very good key selection because the overlap between two critical sets derived from two sets of compression levels is almost close to 100%.

5.3 Low-pass Filter as a Key

For a fixed size critical set, two distinct filters produce more than 36% of non overlapping points. That is, by knowing a critical set produced using a filter, it is extremely hard to guess another critical set produced using a different filter. This is also evident from some experiments we have performed. Since there are numerous filters we can choose by selecting different basis functions, keeping the low pass filter (ie. filter and its properties) as a key is an attractive idea for avoiding attacks.

6 Size of Key Space

6.1 ρ as a Key

Since the maximum value of pixel value difference between the original image and the lowpass image is 255, ρ can take values ranging from 0 to 255. Keeping ρ as 0 will include all pixel points as critical points and keeping ρ as 255 will include none of the pixel points as critical points. Therefore, the size of the key space of ρ is upper bounded by the value of 255.

6.2 Compression Levels as Keys

JPEG compression quality level range from 0 to 100. Choosing quality levels below certain value l may not produce good quality images and so lead to great proportion of insignificant critical points. Suppose l has been set to an appropriate quality level (close to 100), there are $100-l$ possible values to choose from. Since a quality level chosen to produce a set of points should not be repeated for producing another set of points, each level chosen must be distinct in the range of l to 100. If p (less than or equal to $100-l+1$) levels are chosen, the size of the key space will be $(100-l+1)(100-l)...(100-l-p+2))$ which is equal to $\frac{(100-l+1)!}{(100-l-p+1)!}$

6.3 Low-pass Filter as a Key

In the case of DCT, there are 64 DCT coefficients available for each 8x8 block. In low pass filtering using DCT, if m high frequency coefficients are made 0 uniformly among all blocks, there are m possible keys. If the number of high frequency coefficients that are made 0 is varied among blocks and kept within the range of 1 to m, assuming n blocks are there, the size of the key space will be less than or equal to m^n.

7 A Comparison

In this section we compare our scheme with the Battacharjee's [1] and Chun-Shien's [2] wavelet based authentication schemes with respect to wavelets chosen, scales and size of filters used, computations involved, thresholds selected for compression tolerance, and the amount of redundant information involved.

Battacharjee's scheme uses Mexican-Hat wavelet which is isotropic in nature. Chun-Shien's method does not mention which wavelet is chosen. In our scheme, DCT with varying number of AC coefficients are chosen for smoothing.

Bathacharjee's scheme uses the parent child relationship between the scales which are apart (not adjacent). Sizes of filters are never taken into consideration. Chun-Shien's scheme uses the parent child relationship between adjacent scales. Since ours is a DCT based scheme, filter sizes or scales can not be changed.

Battacharjee's scheme involves a huge amount of computation. In Chun-shien's scheme, computation performance improves with an increase in scale. Our method also involves a lot of computation because of the application of DCT and IDCT.

To give provision for compression tolerance, Battacharjee's scheme must set the thresholds T and t appropriately. Chun-shien's scheme does not look at the performance with respect to different thresholds. Our scheme should set the threshold ρ for adjusting the size of the critical set and then decide on another threshold for giving compression tolerance based on matching critical coordinates.

Battacharee's scheme does not give any provision for frequency bands within each scale because Mexican-hat wavelet is used. Chun-shien's scheme uses information in the higher sub-bands which contains irrelevant information for distingushing JPEG compression, which is a lowpass filtering. Our scheme has been experimented incorporating varying percentage of low frequency AC coefficients.

8 Conclusion

In this paper, we have proposed a variant to the already existing wavelet based image authentication schemes using discrete cosine transform. The use of critical set has increased performance at the cost of more computations. We have showed that the optimal percentages of AC coefficients that give the highest level of detection while keeping the size of the critical set roughly equal are 33% for Lenna, Peppers and Pills images, and 44% for Paper and Corrosion images.

References

[1] Sushil Bhattacharjee and Martin Kutter, Performance Analysis of Image Compression Using Wavelets, Compression Tolerant Image Authentication, Image Processing 1998, ICIP-98 Proceedings, International conference, 1998, Vol-1, pp 435-439

[2] Chun-Shien Lu and Hong-Yuan Mark Liao. Structural Digital Signature for Image Authentication: An Incidental Distortion Resistant Scheme, Proc. ACM Multimedia and Security Workshop at the 8th ACM Int. Conf. on Multimedia, Los Angeles, California, USA, pp 115-118, Nov 4, 2000

[3] Wallace,G.K., "The JPEG still picture compression standard.", Communication of the ACM, vol. 34, no. 4, April 1991, pp 30-40.