Anti-cropping digital image watermarking using Sudoku

Shamsul Kamal Ahmad Khalid*

Department of Information Security,
Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia,
Johor, Malaysia
Email: shamsulk@uthm.edu.my
*Corresponding author

Mustafa Mat Deris

Department of Software Engineering,
Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia,
Johor, Malaysia
Email: mmustafa@uthm.edu.my

Kamaruddin Malik Mohamad

Department of Information Security,
Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia,
Johor, Malaysia
Email: malik@uthm.edu.my

Abstract: Many digital image watermarking schemes have been developed to embed copyright information into an image. However, an attacker may reuse parts of a watermarked image by cropping out unwanted parts. Several techniques have been designed to overcome this attack but due to their limited redundancy approach, some section of the images can still be retrieved without detectable watermark. In this paper, a new watermarking scheme that is robust against severe cropping using Sudoku is proposed. It is based on Sudoku's permutation property that allows evenly distributed copies of watermark pieces in all parts of the cover image. A valid Sudoku solution is used during the embedding as well as during the detection of the watermark. Using classic 9×9 Sudoku, the scheme demonstrated robustness of up to 94% of random cropping.

Keywords: watermarking; Sudoku; cropping; redundant embedding.

Reference to this paper should be made as follows: Khalid, S.K.A., Deris, M.M. and Mohamad, K.M. (XXXX) 'Anti-cropping digital image watermarking using Sudoku', *Int. J. Grid and Utility Computing*, Vol. X, No. Y, pp.xx–xx.

Biographical notes: Shamsul Kamal Ahmad Khalid is a PhD candidate at Universiti Tun Hussein Onn Malaysia. He received his degree in Computer Science from New York University in 1995. His research interests are in steganography, watermarking and program analysis.

Mustafa Mat Deris is a Professor of Computer Science in the Faculty of Computer Science and Information Technology, UTHM, Malaysia. He received PhD from University Putra Malaysia in 2002. He has successfully supervised five PhD students and currently he is supervising eleven PhD students and published more than 160 papers in journals and conference proceedings.

Kamaruddin Malik Mohamad received his degree in Computer Science from Acadia University, Nova Scotia, Canada in 1992. He was awarded the PhD degree from Universiti Tun Hussein Onn Malaysia in 2011. His research interests are in file carving, digital forensic and steganography.

1 Introduction

With the proliferation of digital multimedia content on the internet, content owners and service providers require good mechanism to protect their work. Digital watermarking is a technique used to embed certain information into the media to be protected, such as a company's logo or product number. Such information can later be extracted and used to detect forgery, authentication and unauthorised usage. The embedded information is called watermark; and the media being protected is called host or cover media. Recently, many watermarking schemes have been proposed in the literature for digital images.

A digital image watermarking scheme must at least satisfy the requirements of robustness, imperceptibility and reasonable capacity. A watermarking system is considered robust if the embedded watermark remains detectable or retrievable under various attacks on the watermarked host, such as cropping, filtering, noise addition, geometric distortions and others. Although a visible watermark is possible, an invisible watermark provides another layer of protection to the digital content. Imperceptibility is the measure of the quality of the watermarked image compared to its original host image. To have good imperceptibility, a watermarked image must appear the same as its original host image. Capacity is the size of the embedded information. Increasing capacity usually degrades the imperceptibility property.

Watermark embedding can be implemented either in spatial domain or transform domain (Fung et al., 2011). In spatial domain technique, the watermark embedding is done by directly modifying the pixel values of the host image (Fu et al., 2008; Aggarwal and Singla, 2011). In transform domain technique, the host image is first converted into frequency domain by a transformation method such as the Discrete Cosine Transform (DCT) or Discrete Wavelet Transform (DWT) (Reddy and Varadarajan, 2010; Kundur and Hatzinakos, 2004). Then, watermark is embedded by modifying its coefficients. Modifications are done by changing one or more of the bit-planes of the pixel values or the coefficients in such a way that they do not perceptibly change the host image.

A stronger watermarking technique is to have more than one copies of the watermark at almost all locations in the host image. Image watermarking systems commonly use redundant embedding to handle cropping, filtering and addition of band-limited noise (Cox et al., 2008). Having redundancies like this will facilitate successful detection or retrieval of the watermark being attacked by the adversaries. One way of doing this is by embedding a greyscale watermark (e.g. 8-bit greyscale). If some of the bit-planes are damaged, then the remaining bit-planes can be used to detect or reconstruct the watermark from the watermarked image. Another way is by having copies of a binary watermark evenly distributed in the host image. If a copy of the watermarks is damaged, then the remaining copies can be used to detect or reconstruct from the watermarked image. The watermarking systems proposed by Aggarwal and Singla (2011), Reddy and Varadarajan (2010) and Fang et al. (2004) are not robust against cropping attack. One of the main reasons is that the watermarks are not well distributed in the host image. Cropping done on such watermarked image can get away with no detectable watermark.

In this paper, a new watermarking scheme that is robust against severe cropping using Sudoku is proposed. It is based on Sudoku's permutation property that allows evenly distributed copies of watermark pieces in all parts of the cover image. A Sudoku solution is used during the embedding as well as during the detection of the watermark. Using a classic 9×9 Sudoku, the scheme demonstrated robustness of up to 94% of random cropping.

The rest of this paper is organised as follows. In Section 2, related work will be discussed. The details of our approach are discussed in Section 3. The result and discussion of experiments will be covered in Section 4. Finally, Section 5 is for the conclusion.

2 Related work

2.1 Cropping

Once a hiding place has been decided (i.e. either in spatial or transform domain), a hiding scheme must be designed to be robust enough against various watermarking attacks. We are particularly interested in investigating and designing a scheme that is robust against cropping. Cropping is defined as cutting unwanted parts from a watermarked image. We review here how current approaches fare against cropping attacks. This is summarised in Table 1.

Table 1 shows that most recent approaches could only handle cropping in the maximum range of 50–75% of the watermarked image. Cox et al. (1996) proposed a transform domain watermarking system in which a single watermark spread over the host image. Due to its image content-analysis based approach, the length of watermark is not fixed. Cox approach requires the cover image to retrieve the watermark. Although Fang et al. (2004) offers a blind watermarking technique, it supports only up to 60% cropping with 1 kB of watermark and does not support random cropping. Aggarwal and Singla (2011) try to use redundant watermark, but due to its limited copies and uneven positions, random cropping will produce a cropped image without watermark. It can support 75% cropping with 4 kB of watermark and it needs the original host image to recover the watermark. Although Rawat and Raman (2010) can embed bigger watermark (average is 22 kB watermark per colour component) and support random cropping, it cannot handle very well cropping more than 50% and requires the cover image to extract the watermark. Therefore, generally, most of these schemes cannot support severe random cropping (larger than 75%), have limited watermark redundancies and limited watermark size.

Table 1 Comparison of approaches against cropping attack

Scheme	*Maximum Cropping Ratio Supported (*)*	*Number of Watermark*	*Watermark Size*	*Support Random Cropping*	*Support Blind Retrieval (**)*
Cox et al. (1996)	75%	A single watermark seems to be spread all over the cover	Not fixed. Depending on the content of the cover image	Yes	No
Fang et al. (2004)	50–60%	A single watermark seems to be spread all over the cover	1 kB on one colour component	No	Yes
Aggarwal and Singla (2011)	75%	5 watermarks. Watermarks are fixed to the 4 corners and 1 in the middle of the cover image	4 kB on one colour component	No	No
Rawat and Raman (2010)	50%	A single watermark seems to be spread all over the cover image using all the subbands	66 kB on three colour components (i.e. 22 KB on a single component)	Yes	No

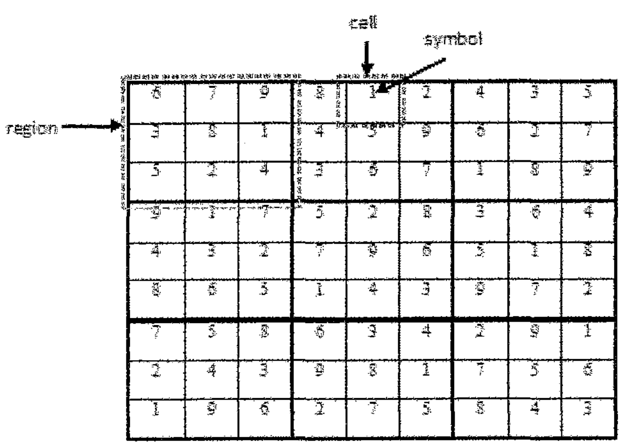
Note (*) as reported in their corresponding paper. NCC (Normalised Cross Correlation) must be greater than 0.8 and BER (Bit Error Rate) must be less than 20%.

(**) Blind retrieval is the extraction of watermark without needing the cover file.

2.2 *Sudoku*

A Sudoku puzzle consists of a partially completed row-column grid of cells partitioned into N regions each of size N cells, to be filled in using a set of N distinct symbols (for example, the digits $\{1,2,\ldots,N\}$). A digit must be assigned to each cell in the grid with only one restriction: a given digit cannot appear twice in a row, in a column or in a block (region) (Jussien, 2007). A classic Sudoku is a puzzle whose objective is using the digits from one to nine to fill a 9 × 9 grid. A solution of this type of Sudoku grid satisfies the following properties. First, a Sudoku grid contains nine 3 × 3 regions, each containing different digits from one to nine. Second, each row and each column of a Sudoku grid also contain different digits from one to nine. Figure 1 shows an example of a Sudoku solution.

Figure 1 Example of Sudoku solution (see online version for colours)



One of the most important properties of Sudoku is that its constraints enforce evenly spread symbols/numbers across the board. In virtually all sections of the board, almost all tiles' numbers can be gathered to form a complete set of symbols/numbers. Another important property of Sudoku is its number of unique solutions. Having a unique solution guarantees correct and unique sequence must be achieved horizontally, vertically and diagonally around a particular tile. Felgenhauer and Jarvis (2006) analyse the classic 9 × 9 Sudoku solutions to show that total number of possible solutions is $\approx 6.671 \times 10^{21}$. The result was derived through logic and brute force computation. Russell and Jarvis (2007) showed that if various possible symmetries (e.g. rotation, reflection and so on) are allowed, then the number of fundamental solutions of 9 × 9 Sudoku grid is 5,472,730,538. The number of valid Sudoku solution grids for the 16 × 16 derivation is unknown.

2.3 *Sudoku approach in security and data hiding*

Sudoku pattern has been employed in relatively few works in security and data hiding applications (Naini et al., 2010). Wu and Ren (2009) proposed an image authentication system using Sudoku and chaotic map. A selected Sudoku solution is used to guide cover pixels' modification in order to imply secret data. In another experiment, using Sudoku pairs, blocks scrambling and bits scrambling are applied to a cover image to completely scatter image contents (Zou et al., 2011). Chou et al. (2010) proposed a data hiding scheme using Sudoku to spread out original image into three shadows images carrying the secret data. Retrieving requires a pairing of at least two shadow images. This is also done in the work of Chang et al. (2010) with lossless recovery of the embedded secret. Yet another extension to the 'shadow-Sudoku' technique is done by Roshan et al. (2009) by extending the work to use pairs from colour images (e g. red and green components) and, use 27 × 27 reference matrix instead of 256 × 256. Naini et al. (2010) proposed a watermarking scheme using Sudoku that is robust against JPEG compression. Bits of the secret message are embedded along an edge using 16 × 16 Sudoku's non-repeating numbers. The authors said the scheme is also robust against cropping but mentioned 'the robustness against cropping attack depends on the cropped region'. It is inferred that the scheme does not support random cropping. No cropping percentage is provided in their experiments.

3 Anti-cropping approach

The proposed watermarking system makes use of the excellent redundancy property of Sudoku to solve cropping problem in watermarking.

3.1 Embedding procedure

Consider a cover image C has $m_c \times n_c$ pixels and a watermark image W has $m_w \times n_w$ pixels, where m is the image height and n is the image width. A Sudoku solution S consists of row-column grid of cells, partitioned into N regions each of size N cells, to be filled in using a set of N distinct symbols.

Sudoku cells: A Sudoku cell, $S_{i,j}$ denotes a cell where i is the position of the cell in a region and j is the position of the region in S. For example, a third cell in the forth region of S will be denoted as $S_{3,4}$. A value, v can be assigned to a cell where v ranges from 1..N, which is constrained by the Sudoku requirements R – each rows, columns and regions must contain all the numbers (1..N) and no repeat (e.g. see Figure 1). Therefore, a value $v_{i,j}$ assigned to a Sudoku cell $S_{i,j}$ can be represented as:

$$S_{i,j} = v_{i,j} \text{ where } v \in 1..N, 1 \le i \le N, 1 \le j \le N \text{ and } \boldsymbol{R} \text{ is true}$$

Region size: The region size RS of a Sudoku S can be calculated as:

$$RS_{row} = \frac{m_c}{m_w} / m_w * \sqrt{N}$$

$$RS_{column} = \frac{n_c}{n_w} / n_w * \sqrt{N}$$

To get a watermark that can fit a region, W_t the original watermark, W_{orig} need to be shrunk to a region size RS, which can be represented by:

$$W_t = \text{resize}\left(W_{orig}, RS_{row}, RS_{column}\right)$$

Watermark pieces: As each region must have N symbols, W_t will be divided into $\sqrt{N} \times \sqrt{N}$ tiles. For example, if $N = 9$, W_t will be divided into 3 × 3 tiles:

$$W_t = \begin{bmatrix} W_{t1} & W_{t2} & W_{t3} \\ W_{t4} & W_{t5} & W_{t6} \\ W_{t7} & W_{t8} & W_{t9} \end{bmatrix}$$

Notice that the watermark tiles/pieces are numbered from left to right, top to bottom. More generally, we write:

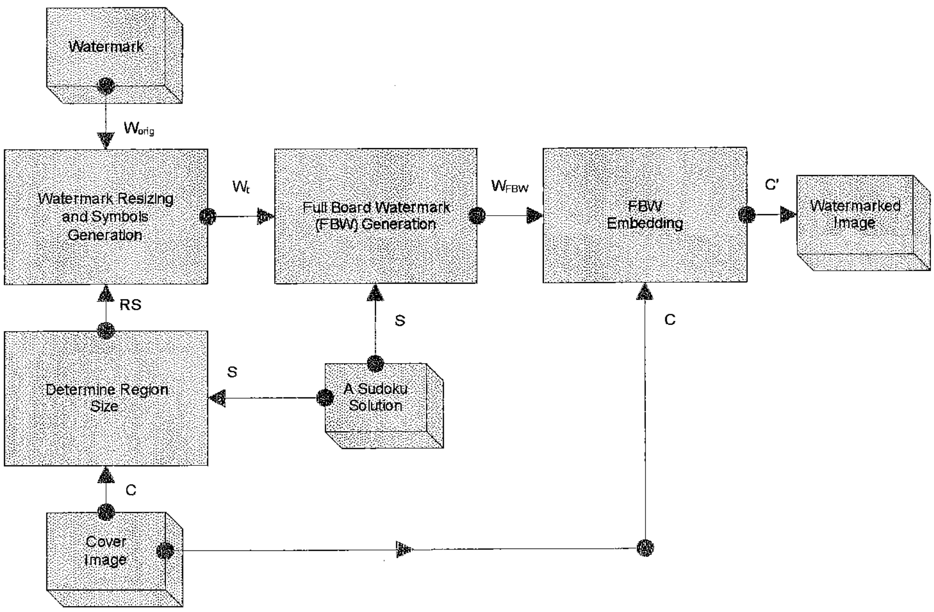
$$W_t = \{W_{t1}, W_{t2}, \ldots, W_{tN}\}$$

Full board watermark: Using W_t tiles and the Sudoku solution S, a full board watermark image, W_{FBW} can be constructed by mapping each cell $S_{i,j}$ to the corresponding W_t tiles. The mapping can be represented by the following formula:

$$W_{FBW} = \sum_{j=1}^{N} \sum_{i=1}^{N} S_{i,j} \rightarrow W_{tk} \text{ where} \begin{Bmatrix} k = 1 \, if \, S_{i,j} = 1 \\ k = 2 \, if \, S_{i,j} = 2 \\ k = 3 \, if \, S_{i,j} = 3 \\ k = 4 \, if \, S_{i,j} = 4 \\ k = 5 \, if \, S_{i,j} = 5 \\ \ldots \\ \ldots \\ k = N \, if \, S_{i,j} = N \end{Bmatrix}$$

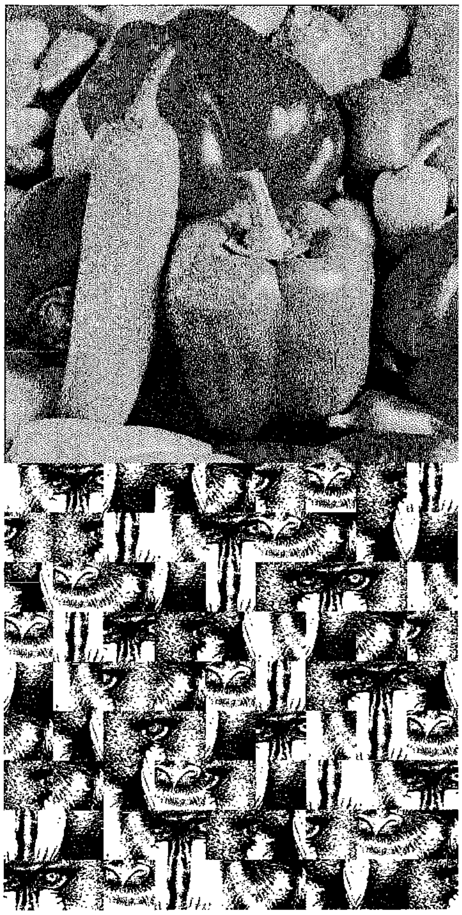
Figure 2 shows the embedding process. It starts with two processes: (a) regions mapping of the cover image to the Sudoku regions (nine regions in total); (b) symbols generation of the watermark image by breaking it into 3 × 3 = 9 distinct symbols or tiles (in this paper, we use 'symbols' and 'tiles' interchangeably). The tiles are numbered from left to right, top to bottom. Based on the Sudoku solution, the watermark symbols will be re-arranged and embedded into each region of the cover image.

Figure 2 The embedding process of anti-cropping (see online version for colours)



The end result will be nine copies of binary watermarks being distributed in 81 tiles which is not overlapping and evenly spread in the cover image (see Figure 3).

Figure 3 A watermarked *pepper* cover image with *baboon* inside it (top). The 81 *baboon* watermark tiles (bottom)



Changing a Sudoku solution will accordingly change the watermark tiles arrangement, but preserving its distribution property. Figure 3 illustrates the watermarked image (top) and the watermark tiles embedded inside the cover image (bottom).

3.2 Detection procedure

Prior to finding the watermark in the cropped image C', the watermark tiles, W_t need to be calculated from W_{orig}. Once the raw embedded watermark is retrieved from the cropped image or a clean watermarked image, symbols searching can be performed. Using symbols from W_t, a search of the tiles begins by recording the sequence of the detected tiles, Q, represented by:

$$Q = \text{sym\_search}(W_t, W_{ret})$$

The sequence information in Q is matched with the one in the Sudoku solution S. From the matches, the detection result D will indicate a successful or a failed detection.

$$D = \begin{Bmatrix} yes\,if\, Q \cap S \neq \varnothing \\ no\,if\, Q \cap S = \varnothing \end{Bmatrix}$$

Figure 4 shows the process of watermark detection. Using the same watermark used in the embedding process, nine symbols will be generated. Then, a raw watermark will be retrieved from the cropped image or from the original watermarked image. It follows with searching each of the watermark symbols in the retrieved raw watermark. The outcome will consist of complete and partial tiles as shown in Figure 5. During the search, the sequence of the complete tile(s) is recorded. Then, the sequence analysis engine will check if the sequence matches with the one in the supplied Sudoku solution – horizontally and vertically around the detected tile(s).

Figure 4 The watermark detection process of anti-cropping (see online version for colours)

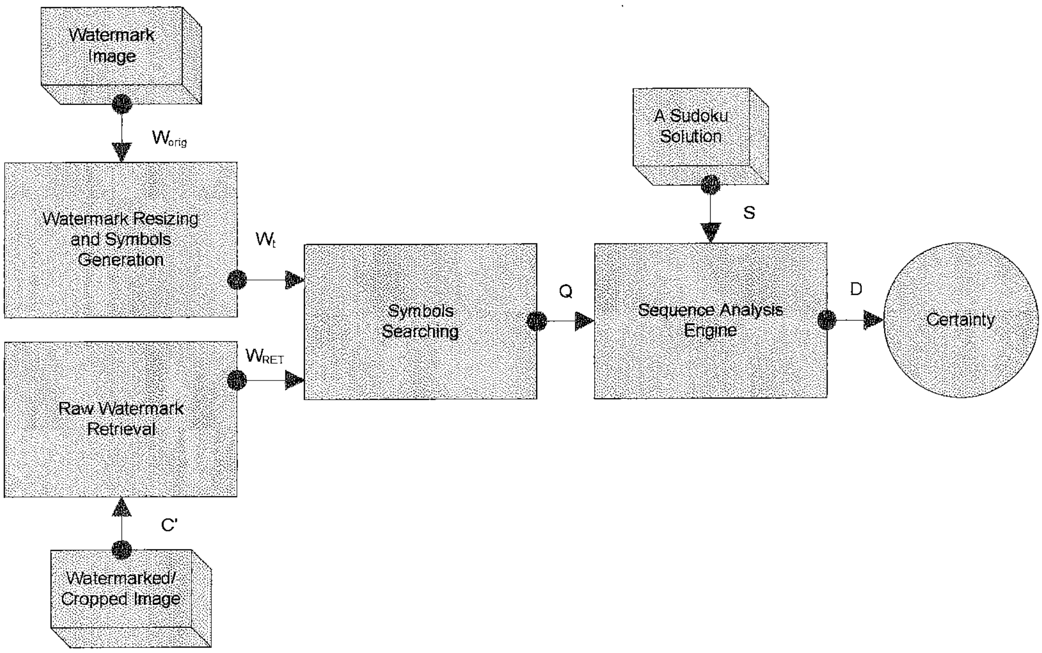


Figure 5 A cropped image and watermark tiles embedded in it

4 Results and discussion

The proposed watermarking system was tested with standard images using Matlab. Figure 6 partially shows the Matlab codes for the embedding process. Notice the Sudoku solution is put into a matrix of cell to assist the construction of the full board watermark image.

Table 2 illustrates 28 of 56 random cropping performed on eight watermarked images and its detection outcomes. The average PSNR is well above normal watermarking schemes (30–40dB). To detect a presence of watermarking a minimum of one full tile is required, together with its surrounding partial tiles (neighbours). The detected tiles and its neighbouring tiles' sequences must match the Sudoku solution on all sides.

Figure 7 shows that for all cases, at least one full tile will be successfully detected with 94% random cropping. This single tile together with its neighbouring sequence and location of tiles (the top, right, bottom and left), can be used to determine if the cover has been watermarked by the Sudoku watermarking scheme. As a note, successful detection expects exact same Sudoku solution is supplied to the watermark detector.

Figure 6 A Matlab code snippet for the embedding process

```
% read in the cover image
file_name='testfiles/boat.tiff';
cover_image=imread(file_name);

% read in the watermark image
file_name='testfiles/elaine.tiff';
watermark=imread(file_name);

% conversions needed to spread the image values on
a 256 gray-scale
watermark=double(watermark);
watermark=round(watermark./256);
watermark=uint8(watermark);

% determine the size of the cover image
Mc=size(cover_image,1);                % height
Nc=size(cover_image,2);                % width

% determine the size of the watermark
Mm=size(watermark,1);                  % height
Nm=size(watermark,2);                  % width

% determine the region size
rregionsize = round(Mm/3)-1;           % row
cregionsize = round(Nm/3)-1;           % column

% resize the watermark to fit the region
smaller = imresize(watermark,[regionsize
cregionsize]);

% trim the watermark to get congruent size
rcellsize =round(regionsize/3)-1;      % row
ccellsize =round(cregionsize/3)-1;     % column
nsmaller = smaller(1:rcellsize*3, 1:ccellsize*3);

% separate the watermark to 9 pieces
awatermark = [];
awatermark = mat2cell(nsmaller,[rcellsize
rcellsize rcellsize],[ccellsize ccellsize
ccellsize]);

% the Sudoku solution
vsodoku =[[7 2 6 3 1 5 4 8 9],[4 9 3 7 2 8 6 5 1],
         [8 1 5 9 4 6 2 3 7];
         [8 5 2 6 7 3 9 4 1], [1 4 7 9 8 5 3 6 2],
         [6 9 3 1 2 4 7 5 8];
         [1 9 4 5 6 7 2 3 8], [8 3 6 2 1 4 5 7 9],
         [5 7 2 3 8 9 4 6 1]];
avsodoku = mat2cell(vsodoku,[1 1 1],[9 9 9]);

% Full Board Watermark Generation
nwatermark = [];  % cell watermark accumulator
mwatermark = [];  % region watermark accumulator
rvwatermark = []; % board watermark accumulator
```

Figure 6 A Matlab code snippet for the embedding process (continued)

```
for i = 1:3
    for j = 1:3
        region = avsodoku{i,j};
        for k = 1:9
                if region(1,k) == 1
                        nwatermark = [nwatermark,
                                awatermark{1,1}];
                end
                if region(1,k) == 2
                        nwatermark = [nwatermark,
                        awatermark{1,2}];
                end
                if region(1,k) == 3
                        nwatermark = [nwatermark,
                        awatermark{1,3}];
                end
                if region(1,k) == 4
                        nwatermark = [nwatermark,
                        awatermark{2,1}];
                end
                if region(1,k) == 5
                        nwatermark = [nwatermark,
                        awatermark{2,2}];
                end
                if region(1,k) == 6
                        nwatermark = [nwatermark,
                        awatermark{2,3}];
                end
                if region(1,k) == 7
                        nwatermark = [nwatermark,
                        awatermark{3,1}];
                end
                if region(1,k) == 8
                        nwatermark = [nwatermark,
                        awatermark{3,2}];
                end
                if region(1,k) == 9
                        nwatermark = [nwatermark,
                        awatermark{3,3}];
                end
        end

        mwatermark = [
                nwatermark(1:rcellsize,1:ccellsize*
                3);
                nwatermark(1:rcellsize,ccellsize*3+
                1:ccellsize*6);
                nwatermark(1:rcellsize,ccellsize*6+
                1:ccellsize*9)];
        rvwatermark = [rvwatermark, mwatermark];
        nwatermark = [];
    end
end

rvvwatermark = [
        rvwatermark(1:rcellsize*3,1:ccellsize*3*3);
        rvwatermark(1:rcellsize*3,ccellsize*3*3+1:c
        cellsize*3*6);
        rvwatermark(1:rcellsize*3,ccellsize*3*6+1:c
        cellsize*3*9)];
arvvwatermark = mat2cell(rvvwatermark,[rregionsize
        rregionsize rregionsize],[cregionsize
        cregionsize cregionsize]);
```

Table 2 Watermarks images and their corresponding PSNR, cropping ratios and detection result

Cover image	*Watermark*	*PSNR (dB)*	*Cropping ratio*	*Number of full tiles detected*	*Follow correct sequence?*	*Have correct neighbouring pattern*
Lena	Pepper	45.28	99.51	0	NA	NA
			98.29	0	NA	NA
			97.05	0	NA	NA
			95.13	1	NA	Yes
			87.64	4	Yes	Yes
			69.66	16	Yes	Yes
			50.84	36	Yes	Yes
Pepper	Baboon	45.28	99.36	0	NA	NA
			96.34	1	NA	Yes
			94.23	1	NA	Yes
			90.36	4	Yes	Yes
			79.64	9	Yes	Yes
			66.35	16	Yes	Yes
			43.46	36	Yes	Yes
Baboon	Lena	45.29	98.97	0	NA	NA
			97.43	1	NA	Yes
			95.95	1	NA	Yes
			87.64	9	Yes	Yes
			79.64	16	Yes	Yes
			58.96	25	Yes	Yes
			41.98	36	Yes	Yes
Boat	Elain	45.29	99.74	0	NA	NA
			98.97	0	NA	NA
			97.74	1	NA	Yes
			93.65	4	Yes	Yes
			83.97	9	Yes	Yes
			63.81	25	Yes	Yes
			43.46	36	Yes	Yes

Figure 7 Graph of cropping versus tiles detected

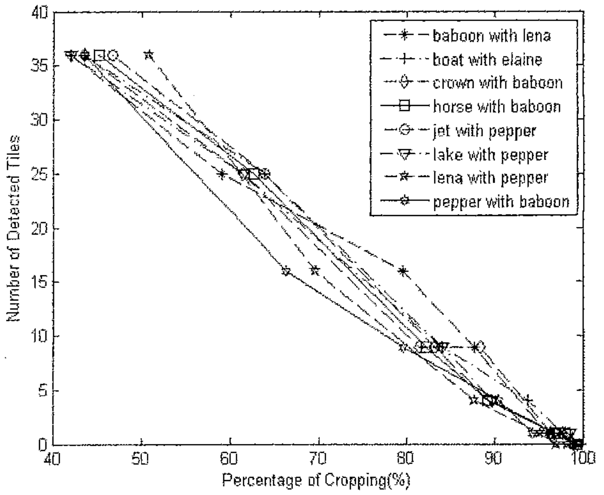


Table 3 shows the comparison of anti-cropping approach with other watermarking schemes. The detection capability of Soduku-based anti-cropping approach under severe cropping far exceeded the rest. Another important feature of anti-cropping is its ability to support random cropping. The Sudoku's unique symbol permutation and its evenly distributed tiles effectively make its watermarked image robust against random cropping. As a blind watermarking scheme, anti-cropping offers greater practicality. It appears that the work of Aggarwal and Singla (2011) and anti-cropping systems have distributed copies of the watermarks. However, some parts of the host image are not covered by Aggarwal and Singla (2011) system. Furthermore, anti-cropping is able to embed 28 kB of watermark at nine non-overlapping places on a grey cover image. With colour images, it can have more redundancies.

A simple alternative to anti-cropping would be to randomly embed watermark tiles in each region of a cover image. Using a good random function generator, the watermark tiles can be uniformly distributed into each region of the cover image. However, such approach cannot guarantee good tiles recovery over cross-regions cropping. For example, in the case of a long rectangular cropping, Sudoku immediately offers non-overlapping tiles that can be used to assist watermark detection/recovery; while random tiles do not – as every row, columns and regions have no tight relationship with each others.

Table 3 Comparison of anti-cropping with other watermarking schemes

	Maximum cropping ratio supported	*Number of watermark*	*Watermark size*	*Random cropping ratio*	*Blind retrieval*
Other schemes	50–75%	1–5 copies	1–22 kB per colour component	Mixed	Mixed
Anti-cropping	94%	9 copies with 81 evenly distributed pieces	28 kB per colour component	Yes	Blind (*)

Note (*) Original cover image is not needed to extract the watermark. The anti-cropping watermark detector can be distributed with a fixed and encrypted or hand coded watermark. As there is no relationship between the watermark and the cover image (i.e. one cannot derive the other), anti-cropping can be considered a blind watermarking scheme. Refer to blind watermarking discussion in Cox et al.'s book (2008).

5 Conclusion

In this paper, a novel watermarking scheme based on Sudoku's permutation property is introduced, which embed watermark pieces randomly and evenly in a chosen cover image in order to be robust against random cropping (called anti-cropping). The proposed watermarking scheme was tested using various standard cover and watermark images and, with different cropping ratios. Using 81 9×9 Sudoku tiles, the experiments shows that anti-cropping is more robust by being able to withstand 94% random cropping attacks. Different sizes of Sudoku tiles will be used in our future work (such as 256 16×16 Sudoku tiles). There are two advantages of using such configuration. Firstly, more tiles can be detected within a cropping ratio. Secondly, more Sudoku tiles provide higher randomness, therefore, more secure. Other experiments will also be conducted on Sudoku's ability at countering other types of attacks such as salt-n-pepper and JPEG compression.

Acknowledgements

This work is supported by Universiti Tun Hussein Onn Malaysia. The grant number is FRGS/1/2012/TK08/UTHM/03/5 (FRGS 1051).

References

Aggarwal, A. and Singla, M. (2011) 'Robust watermarking of color image under noise and cropping attack in spatial domain', *International Journal of Computer Science and Information Technologies*, Vol. 2, No. 5, pp.2036–2041.

Chang, C.C., Lin, P.Y., Wang, Z.H. and Li, M.C. (2010) 'A Sudoku-based secret image sharing scheme with reversibility', *Journal of Communications*, Academy Publisher, Vol. 5, No. 1.

Chou, Y.C., Lin, C.H., Li, P.C. and Li, Y.C. (2010) 'A (2,3) threshold secret sharing scheme using Sudoku', *Proceedings of the IEEE 6th International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp.43–46.

Cox, I.J., Kilian, J., Leighton, T. and Shamoon, T. (1996) *A Secure, Robust Watermark for Multimedia*, Workshop on Information Hiding, University of Cambridge.

Cox, I.J., Miller, M.L., Bloom, J.A., Fridrich, J. and Kalker, T. (2008) *Digital Watermarking and Steganography*, Morgan Kaufmann, Burlington, MA.

Fang, Y.M., Huang, J.W. and Wu, S.Q. (2004) 'CDMA-based watermarking resisting to cropping', *Proceedings of the ISCAS 2004*, pp.25–28.

Felgenhauer, B. and Jarvis, F. (2006) 'Mathematics of Sudoku I', *Mathematical Spectrum*, Vol. 39, No. 1, pp.15–22.

Fu, Y.G. and Shen, R.M. (2008) 'Color image watermarking scheme based on linear discriminant analysis', *Computer Standards and Interfaces*, Vol. 30, pp.115–120.

Fung, C.W.H., Gortan, A. and Godoy, W., Jr. (2011) 'A review study on image digital watermarking', *Proceedings of the 10th International Conference on Networks (ICN2011)*, pp.24–28.

Jussien, N. (2007) *A to Z of Sudoku*, ISTE Ltd, London.

Kundur, D. and Hatzinakos, D. (2004) 'Towards robust logo watermarking using multi-resolution image fusion', *IEEE Transactions on Multimedia*, Vol. 6, pp.185–197.

Naini, P.M., Fakhraie, S.M. and Avanaki, A.N. (2010) 'Sudoku bit arrangement for combined demosaicking and watermarking in digital camera', *Proceedings of the IEEE 2nd International Conference on Advances in Databases, Knowledge, and Data Applications*, pp.41–44.

Rawat, S. and Raman, B. (2010) 'A new robust watermarking scheme for color images', *Proceedings of the IEEE 2nd International Advance Computing Conference*, pp.206–209.

Reddy, V.P. and Varadarajan, S. (2010) 'An effective wavelet-based watermarking scheme using human visual system for protecting copyrights of digital images', *International Journal of Computer and Electrical Engineering*, Vol. 2, No. 1, pp.24–27.

Roshan Shetty, B.R., Rohith, J., Mukund, V. and Rohan, H. (2009) 'Steganography using Sudoku Puzzle', in *Proceedings of the IEEE International Conference on Advances in Recent Technologies in Communication and Computing*, pp.623–626.

Russell, E. and Jarvis, F. (2007) 'Mathematics of Sudoku II', *Mathematical Spectrum*, Vol. 39, No. 2, pp.54–58.

Wu, W.C. and Ren, G.R. (2009) 'A new approach to image authentication using chaotic map and Sudoku puzzle', *Proceedings of the IEEE 5th International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp.628–631.

Zou, Y., Tian, X.L., Xia, S.W. and Song, Y. (2011) 'A novel image scrambling algorithm based on Sudoku puzzle', *Proceedings of the IEEE 4th International Congress on Image and Signal Processing*, pp.737–740.