Detecting Data Leakage from Hard Copy Documents

Jijnasa Nayak, Shweta Singh, Saheb Chhabra, Gaurav Gupta, Monika Gupta, Garima Gupta

Chapter 7

DETECTING DATA LEAKAGE FROM HARD COPY DOCUMENTS

Jijnasa Nayak, Shweta Singh, Saheb Chhabra, Gaurav Gupta, Monika Gupta and Garima Gupta

Abstract Document fraud has evolved to become a significant threat to individuals and organizations. Data leakage from hard copy documents is a common type of fraud. This chapter proposes a methodology for analyzing printed and photocopied versions of confidential documents to identify the source of a leak. The methodology incorporates a novel font pixel manipulation algorithm that embeds data in the pixels of certain characters of confidential documents in a manner that is imperceptible to the human eye. The embedded data is extracted from a leaked printed or photocopied document to identify the specific document that served as the source. The embedded data is robust in that it can withstand errors introduced by printing, scanning and photocopying documents. Experimental results demonstrate the efficiency, robustness and security of the methodology.

Keywords: Document fraud, data leakage detection, font pixel manipulation

1. Introduction

People and organizations depend on documents for their day-to-day activities. The importance of documents has driven criminals to perpetrate a number of digitized document frauds. Document frauds involve manufacturing, counterfeiting, altering, selling and/or misusing documents for criminal purposes [4, 5]. Indeed, document frauds have become a global problem that requires serious attention on the part of digital forensic researchers and investigators.

Data leakage is a common but most serious threat. Incidents involving data breaches are reported almost daily. Large tranches of sensitive documents are lost or stolen; in many cases, they are posted on Internet sites such as Wikileaks. According to a 2016 report by In-

foWatch [8], the United States ranked first with 451 leakage incidents, 54% of the total number of incidents; Russia was second with 110 leaks and the United Kingdom third with 39 leaks. Depending on the leaked documents, the incidents could impact national security, cause business losses, tarnish reputations and result in staggering financial penalties due to non-compliance of regulations or lawsuits.

To address the data leakage problem, researchers have proposed digital watermarking and data hiding techniques for a variety of digital media applications, including ownership protection, copy control, annotation and authentication. Data hiding has attracted the interest of the signal processing research community as a means for detecting and preventing data leakage. It is the art and science of inserting payloads (external information) in host content. Some techniques employ cryptographic algorithms [9]. Others leverage steganography to hide secret messages in host data while concealing the very existence of the secret messages [13].

While numerous algorithms and techniques have been proposed for hiding data in images, audios and videos, very little work has focused on data hiding in documents. The principal reason is that the continuous tone property inherent to images and videos does not hold for digital text documents. Furthermore, data hiding in binary images is challenging due to the lack of redundancy in the image carrier and the arbitrary flipping of pixel values that produces noticeable noise.

This chapter describes a novel methodology for detecting leakage from hard copy documents. The methodology embeds a quick response (QR) code in an original "cover" document in a manner that is imperceptible to the human eye. The embedded data is extracted later to identify if a leaked document is a printed version or photocopy of the original cover document. The methodology, which can withstand errors introduced by printing, scanning and photocopying the original document, does not require access to the original document to identify the source of the leak.

2. Related Work

Zou and Shi [16] have proposed a data hiding technique involving inter-word space modulation that embeds exactly one bit of information in one line of text; the technique has been shown to be robust to printing, photocopying and scanning. He et al. [6] have developed a novel data hiding algorithm that combines block partitioning, discrete cosine transforms and pixel flipping. Block partitioning is performed, the matrix of characteristic values of each block is converted into the discrete cosine transform space and a coefficient matrix is generated; high fre-

quency coefficients are modified based on a threshold. The detection process, which checks if the characteristic values change after applying the inverse discrete cosine transform, is robust to printing and scanning.

Wu and Liu [15] have developed a data hiding method that manipulates flippable pixels based on specific block-based relationships, enabling a significant amount of data to be embedded without creating noticeable artifacts. Shuffling is applied before embedding to equalize uneven embedding capacities from region to region. The hidden data can be extracted without using the original image. Moreover, the data can be extracted after high quality printing and scanning with the help of few registration marks.

Odeh et al. [11] have investigated steganographic algorithms that employ text files as carriers. Secret data is hidden in a text file by manipulating the fonts or inserting special symbols in the text file; the algorithms can be applied to Unicode and ASCII code regardless of the text file format. Villan et al. [14] have applied color quantization, which has a high information embedding rate, to digital and printed text documents. The color or luminance intensity of each character is quantized such that the human visual system is unable to distinguish between the original and quantized characters; however, the embedding can be detected by a specialized reader.

Por et al. [12] have developed a data hiding method that inserts external information into a Microsoft Word document in a manner that addresses the low embedding efficiency of text-based data hiding. The data hiding is reversible in that the embedded information can be extracted to completely reconstruct the original Word document. Dasare and Dhore [3] leverage the Microsoft Compound Document File Format (MCDFF) to hide data in unused areas of a Word document while ensuring that the changes made to the document are not visible. Culnane et al. [1] have enhanced the method of Zou and Shi [16] by applying multi-set modulation techniques to increase the data hiding capacities of binary document images; their approach employs automatic threshold computation, threshold buffering, shifted space distribution and letter space compensation techniques. In other work, Culnane et al. [2] have proposed a watermarking method for formatted text documents that is robust to printing and scanning. This method, which builds on their earlier work [1], treats a cover document as one long line of text and uses all the word spaces. Culnane et al. compute the threshold between letter and word spaces based on frequency distributions, and employ a new approach for threshold buffering.

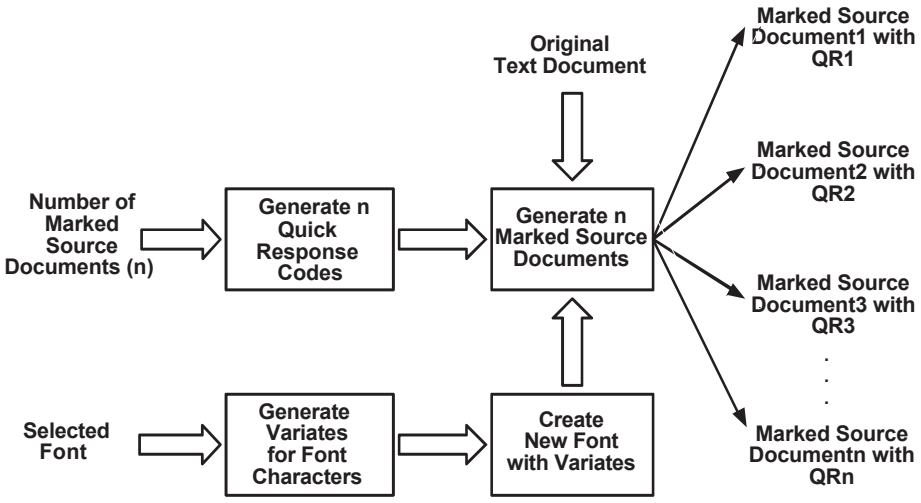

Figure 1. Document generation.

3. Methodology Overview

Identifying the specific document that leaked secret information is a challenging problem. During the past decade, researchers have developed several techniques for identifying the sources of leaks of digital documents. However, the problem is more challenging in the case of physical documents. In a typical use case, confidential documents with the same information are disseminated. Later, it is discovered that someone has leaked the information by taking a photographic image or making a photocopy of one of the disseminated documents. The determination of the source document that leaked the confidential information involves two processes: (i) document generation; and (ii) source identification.

3.1 Document Generation

Figure 1 shows the document generation process. The original text document, the number of marked source documents (copies) required and the selected font are provided as inputs. The user then identifies the specific font characters (letters and/or symbols) that should be modified to create variates whose changes are imperceptible to the naked eye. A new font is then created that contains the variate characters and the characters from the original font.

Let n be the number of marked source document copies that the user wishes to produce for dissemination. Then, n quick response codes are created, each with unique encoded data. Note that the user may

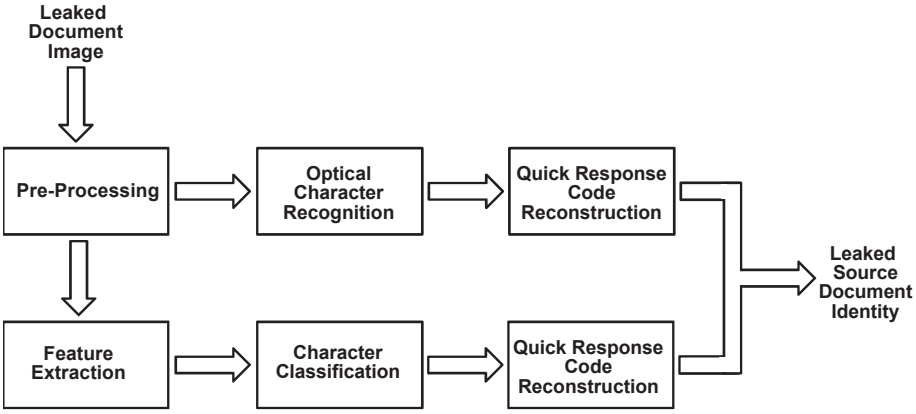

Figure 2. Source identification.

provide input for generating the quick response codes. Following this, the original text document, new font with variates and n quick response codes are used to produce the n marked source document copies, each embedded with a unique quick response code.

3.2 Source Identification

Figure 2 shows the source identification process. The image of the leaked document (camera capture or scanned image) is provided as input. The image is pre-processed, which involves document alignment, noise removal, binary conversion and text segmentation. The segmented character images are input to an optical character recognition (OCR) engine and to a feature extraction module that operate in parallel.

The optical character recognition engine produces the document text and reconstructs the quick response code embedded in the leaked document based on the occurrences and positions of the variate characters.

The feature extraction module extracts the features of each segmented character and passes them to a trained machine-learning-based classifier. The quick response code is then reconstructed based on the classified output characters and their occurrences and positions.

In the final step, the quick response codes generated by the optical character recognition engine and the machine-learning-based classifier are combined to produce a single quick response code; this serves to reduce the error rate. The quick response code is then decoded to extract the encoded information, which identifies the specific marked source document that was responsible for the leak.

Table 1. Printed and photocopied documents used in the experiments.

Printed Documents			
Times New Roman	**Calibri**	**Arial**	**Comic Sans**
50	50	50	50
Photocopied Documents			
Times New Roman	**Calibri**	**Arial**	**Comic Sans**
50	50	50	50

4. Experimental Setup

The proposed methodology identifies the specific source document responsible for a data leakage. This is accomplished by embedding secret information in the form of a unique quick response code in each source (cover) document. Certain font characters in the source documents are altered in a unique and imperceptible manner without affecting the document content. The source documents marked with the embedded quick response codes are then printed and distributed.

A marked document could be leaked in several ways. The original marked document could be released, or the marked original document could be photographed and the camera image could be disseminated, or the original marked document could be photocopied and the photocopy disseminated, or the original marked document could be scanned, emailed and subsequently printed. Additionally, a camera image of the document could be printed, photocopied and then disseminated, or a photocopy of the document could be photographed and the camera image disseminated, and so on.

The proposed solution requires a digitized version of the leaked physical document. It extracts the structure of the quick response code from the digitized version to determine which original marked source document was responsible for the leak.

In order to implement the proposed methodology, multiple commonly-used fonts were chosen in order to create confidential documents. Following this, various characters were selected from each font, a modified version or variate of each character was generated and a unique quick response code was embedded in each confidential document.

In the experiments, four fonts – Times New Roman, Calibri, Arial and Comic Sans – were chosen (Table 1). The A, E, a and g characters of the Times New Roman, Calibri, Arial and Comic Sans fonts were selected

a **g**enuine mess**a**ge

a **,**enuine mess**%**ge

Figure 3. Selected font with variates.

for creating variates and embedding the quick response codes. For each font, 50 documents were created and printed using an HP Color LaserJet Pro MFP M177 printer. This yielded 200 ($= 50 \times 4$) printed documents.

In order to verify the robustness of the proposed methodology, photocopies of the printed documents were considered in addition to the printed documents. Hence, the total number of documents used in the experiments was 400 ($= 200 \times 2$). A Canon CanoScan Mark II scanner was used to produce digitized images of the 400 documents at 400 dpi.

5. Technical Details

This section provides additional technical details about the document generation and source identification processes.

5.1 Document Generation

The two principal steps involved in document generation are: (i) font pixel manipulation; and (ii) quick response code embedding.

Font Pixel Manipulation. The font pixel manipulation technique is used to manipulate certain characters of a font to create variates. The variates are minor modifications of the original characters. The variates are assigned new ASCII values associated with special characters that are rarely used. A new font is created that contains the original characters and the variates.

Figure 3 shows the original text "a genuine message" where the first instance of "g" and the second instance of "a" are shown in boldface for emphasis. The text on the second line replaces these two instances with their variates "," and "%," respectively. Note that the variates are denoted by different symbols ("," and "%") because they are essentially indistinguishable from the original characters ("g" and "a"). While the differences are imperceptible to the human eye in a low-resolution image, they can be detected in a high-resolution image.

Quick Response Code Embedding. The printing, scanning and photocopying processes add noise to a document, which makes it difficult to extract a watermark. Therefore, the proposed methodology incorporates a novel embedding technique in which data (i.e., quick re-

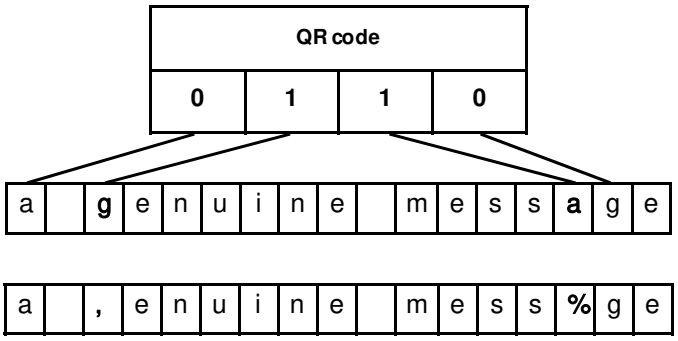

Figure 4. Mapping of a quick response code to text.

sponse code) is hidden in text using modified font characters. In the technique, the original two-dimensional quick response code, which consists of ones and zeros, is first converted to a single-dimensional array. The array is then mapped to the text in the document.

Again, assume that the text is "a genuine message" (Figure 4) and the chosen characters are "a" and "g." Furthermore, assume that the one-dimensional quick response code array that uniquely identifies the document is "0110." Only instances of "a" and "g" in the text may be replaced by their variates. A value of zero in the quick response code array indicates that the corresponding character is not replaced by its variate and a value of one indicates that the corresponding character is replaced by its variate. For example, since the array is "0110," the first instance of a chosen character ("a") is not replaced, the second instance of a chosen character ("g") is replaced with its variate, the third instance of a chosen character ("a") is replaced with its variate and the fourth instance of a chosen character ("g") is not replaced.

Figure 4 shows the mapping of the quick response code array to the text. Because the quick response code array is "0110," only the two characters ("g" and "a") highlighted in boldface are replaced by their variates (once again, denoted by the symbols "," and "%" because the variates are visually indistinguishable from the original characters). Thus, the marked document has miniscule differences from the original document that are imperceptible to humans.

5.2 Source Identification

Source identification involves the extraction of the quick response code from the leaked document image. Next, the quick response code is decoded to identify the original marked source document that was respon-

sible for the leak. This section provides key technical details about the source identification process.

Pre-Processing In order to improve accuracy, the document image is pre-processed before submitting it to an optical character recognition engine. First, the orientation of the document is extracted using the Hough transform. Following this, the image is de-skewed.

Next, the image is converted to the CMYK subtractive color model; the K-channel is selected because it highlights only the black printed text and suppresses the other colors. The final pre-processing step involves the segmentation of each character in the image using vertical and horizontal profiling.

Feature Extraction. An optical character recognition engine and a machine-learning-based classifier are used to identify individual characters (including variates) in the segmented image. Of course, both the systems must be trained to accurately recognize characters using appropriate training sets.

In order to further improve the accuracy of character recognition, a feature extraction technique is applied to the pre-processed image. Because an image can be distorted by scaling, rotation and translation, the invariant moments feature extraction technique [10], which is independent of these operations, is employed. In this technique, the scaling, rotation and translation features corresponding to each character are extracted and fed to the classifier.

In the experiments, the extracted features comprised seven invariant moments that express the shape descriptors of characters. The invariant moments were computed using the following equations:

$$
\begin{aligned}
\phi_1 &= \eta_{20} - \eta_{02} \\
\phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
\phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \mu_{03})^2 \\
\phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21}\mu_{03})^2 \\
\phi_5 &= (\eta_{30}3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 \cdot 3(\eta_{21} + \eta 03)^2] + \\
&\quad (3\eta_{21}\eta_{03})(\eta_{21} + \eta_{03})(3(\eta_{30} + \eta_{12})^2 \cdot (\eta_{21} + \eta_{03})^2) \\
\phi_6 &= (\eta_{20} - \eta_{02})((\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2) - 4\eta_{11}(\eta_{30} + \\
&\quad \eta_{12})(\eta_{21} + \eta_{03}) \\
\phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \\
&\quad (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2)]
\end{aligned}
$$

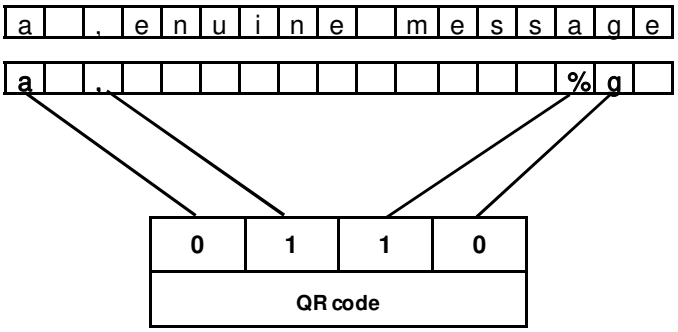

Figure 5. Mapping of text to a quick response code.

where

$$\eta_{pq} = \mu_{pq}/\mu_{oo}^{\gamma} \quad \gamma = (p+q+2)/2, \; p+q = 2,3,...$$
$$\mu_{pq} = \int_{+\infty}^{-\infty} (x-\bar{x})^p (y-\bar{y})^q f(x,y) dx dy$$

Thus, the classifier is able to correctly identify the characters in a segmented document image.

Character Recognition and Code Reconstruction. In the final step, the segmented document image is passed to the trained optical character recognition engine and the machine-learning-based classifier, which output the document text. The output text includes all the variates of the chosen characters. When reconstructing the quick response code from the output text, only the chosen characters are retained because they may have been replaced with their variates based on the quick response code; the remaining characters in the text are ignored.

Figure 5 shows the mapping of the text "a ,enuine mess%ge" in the example document above, where (as before) the "," and "%" denote the variates of the characters "g" and "a," respectively. Since only the second and third instances of the chosen characters were replaced by their variates, the quick response code is determined to be "0110."

6. Experimental Results

Experiments were performed on 400 documents, 200 printed documents and 200 photocopied documents. The images were input to a trained optical character recognition engine and two trained machine-learning-based classifiers in order to reconstruct the quick response codes.

Table 2. Accuracy of quick response code reconstruction.

	OCR-Based System		SVM-Based System		KNN-Based System	
			Times New Roman			
Characters	**Printed**	**Photocopied**	**Printed**	**Photocopied**	**Printed**	**Photocopied**
A,a	82.1	78.3	87.4	83.3	75.6	72.8
A,a,g	80.7	79.2	86.7	82.9	73.3	71.5
A,a,E,g	81.5	80.0	86.1	84.7	72.8	72.3
			Calibri			
Characters	**Printed**	**Photocopied**	**Printed**	**Photocopied**	**Printed**	**Photocopied**
A,a	81.2	81.0	86.1	82.7	73.2	72.1
A,a,g	80.8	79.1	85.3	81.1	74.9	73.7
A,a,E,g	80.6	78.2	84.2	83.1	73.3	71.6
			Arial			
Characters	**Printed**	**Photocopied**	**Printed**	**Photocopied**	**Printed**	**Photocopied**
A,a	83.4	83.3	85.9	84.3	73.5	72.2
A,a,g	82.3	81.7	86.2	85.6	76.6	74.3
A,a,E,g	82.8	81.8	83.4	84.9	74.1	73.8
			Comic Sans			
Characters	**Printed**	**Photocopied**	**Printed**	**Photocopied**	**Printed**	**Photocopied**
A,a	84.1	83.3	86.5	83.7	71.9	70.2
A,a,g	85.1	81.7	84.7	84.6	73.4	71.4
A,a,E,g	84.6	77.8	84.9	83.8	74.5	73.7

6.1 Optical Character Recognition

A trained Tesseract (version 3.02) optical character recognition engine was used to identify the characters and variates in the segmented document images. The Tesseract engine incorporates algorithms that learn features and classify characters. Before using the engine in the experiments, it was trained using 50 images of each character in each of the four fonts (including the variates).

Table 2 presents the quick response code reconstruction results (as percentages) for the printed and photocopied documents. The results clearly show that the trained optical character recognition engine performed better with the printed documents than with the photocopied documents; the errors were likely introduced while training the engine. All the reconstructed quick response codes were decoded successfully us-

ing the ZXing library. Additional experiments revealed that the quick response codes could be extracted and decoded successfully when the optical character recognition accuracy was greater than 90%.

6.2 Machine-Learning-Based Classification

Two machine-learning classifiers, a support vector machine (SVM) and a k-nearest neighbor (KNN) classifier, were trained using 184 samples of each character in the four fonts used in the experiments. For each font, 728, 754 and 13,780 samples were used, corresponding to alterations to two, three and four characters, respectively.

Table 2 shows the quick response code reconstruction accuracy rates (as percentages) for the machine-learning-based classifiers with printed and photocopied documents. The results demonstrate that the support vector machine performed better than the k-nearest neighbor classifier. Moreover, both the classifiers performed better with printed documents than with photocopied documents. All the quick response code were decoded successfully using the ZXing library.

7. Conclusions

Data leakage from hard copy documents is a common type of document fraud. However, existing solutions for identifying the specific confidential document that was the source of a leak are not robust to printing, photocopying and scanning. The methodology described in this chapter incorporates a novel font pixel manipulation algorithm that embeds unique data in the pixels of certain characters of confidential documents in a manner that is imperceptible to humans. The embedded data is extracted from a leaked printed or photocopied version of an original confidential document to identify the specific document that was the source of the leak. Experimental results demonstrate that the methodology is robust in that it can withstand errors introduced by printing, scanning and photocopying documents.

Future research will focus on improving the identification accuracy using machine learning. Research will also extend the methodology to embed high-dimension color quick response codes in source documents.

References

[1] C. Culnane, H Treharne and A. Ho, A new multi-set modulation technique for increasing hiding capacity of binary watermarks for print and scan processes, *Proceedings of the International Workshop on Digital Watermarking*, pp. 96–110, 2006.

[2] C. Culnane, H. Treharne and A. Ho, Improving multi-set formatted binary text watermarking using continuous line embedding, *Proceedings of the Second International Conference on Innovative Computing, Information and Control*, 2007.

[3] A. Dasare and M. Dhore, Secure approach for hiding data in MS Word documents using MCDFF, *Proceedings of the International Conference on Computing, Communication, Control and Automation*, pp. 296–300, 2015.

[4] G. Gupta, C. Mazumdar, M. Rao and R. Bhosale, Paradigm shift in document related frauds: Characteristics identification for development of a non-destructive automated system for printed documents, *Digital Investigation*, vol. 3(1), pp. 43–55, 2006.

[5] G. Gupta, S. Saha, S. Chakraborty and C. Mazumdar, Document frauds: Identification and linking fake documents to scanners and printers, *Proceedings of the International Conference on Computing: Theory and Applications*, pp. 497–501, 2007.

[6] B. He, Y. Wu, K. Kang and W. Guo, A robust binary text digital watermarking algorithm for the print-scan process, *Proceedings of the WRI World Congress on Computer Science and Information Engineering*, pp. 290–294, 2009.

[7] Z. Huang and J. Leng, Analysis of Hu's moment invariants on image scaling and rotation, *Proceedings of the Second International Conference on Computer Engineering and Technology*, vol. 7, pp. 476–480, 2010.

[8] InfoWatch, Global Data Leakage Report 2016, Moscow, Russia (`info watch.com/report2016`), 2016.

[9] A. Menezes, P. van Oorschot and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, Florida, 2001.

[10] J. Noh and K. Rhee, Palmprint identification algorithm using Hu invariant moments and Otsu binarization, *Proceedings of the Fourth Annual ACIS International Conference on Computer and Information Science*, pp. 94–99, 2005.

[11] A. Odeh, K. Elleithy, M. Faezipour and E. Abdelfattah, Highly efficient novel text steganography algorithms, *Proceedings of the Long Island Systems, Applications and Technology Conference*, 2015.

[12] L. Por, K. Wong and K. Chee, UniSpaCh: A text-based data hiding method using Unicode space characters, *Journal of Systems and Software*, vol. 85(5), pp. 1075–1082, 2012.

[13] V. Potdar and E. Chang, Visibly invisible: Ciphertext as a steganographic carrier, *Proceedings of the Fourth International Network Conference*, pp. 385–391, 2004.

[14] R. Villan, S. Voloshynovskiy, O. Koval, J. Vila, E. Topak, F. Deguillaume, Y. Rytsar and T. Pun, Text data-hiding for digital and printed documents: Theoretical and practical considerations, *Proceedings of SPIE–IS&T Electronic Imaging*, vol. 6072, pp. 607212-1–607212-11, 2006.

[15] M. Wu and B. Liu, Data hiding in binary images for authentication and annotation, *IEEE Transactions on Multimedia*, vol. 6(4), pp. 528–538, 2004.

[16] D. Zou and Y. Shi, Formatted text document data hiding robust to printing, copying and scanning, *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. 5, pp. 4971–4974, 2005.