Real-time screen watermarking using overlaying layer

Maciej Piec
Vienna PhD School of Informatics
Vienna University of Technology
& Secure Business Austria
Vienna, Austria
e-mail: mpiec@sba-research.org

Andreas Rauber
Vienna University of Technology
& Secure Business Austria
Vienna, Austria
e-mail: rauber@ifs.tuwien.ac.at

***Abstract*—Protection of intellectual property is a well researched area of computer science. There are many methods that allow securing information that should not be available to unauthorized parties. Many of these techniques provide protection on a relatively low level such as file encryption or filtering of network traffic. However, many of the mechanisms fail as soon as someone attempts to save the content of the screen that is displaying the sensitive data. Such a malicious insider can cause serious harm to an organization's reputation and finances. It is important in case of such incidents to be able to track the source of the leak.**

In this paper a method for watermarking the screen image, that can be used in forensic investigations, is presented. The imperceptible digital watermark is placed on the overlay layer over the whole computer screen. Our method leverages the Human Visual System (HVS) properties and allows to generate dynamically adaptable watermarks that respond to the currently displayed content by using the FAST feature detection algorithm. We use QR Codes that provide spatial non-uniformity and error correction capabilities. The evaluation shows that our scheme works well in the selected use-case scenarios. We show that the watermark is robust against modifications typical for office processing, such as cropping and quality preserving JPEG compression.

***Keywords*-real-time digital watermarking, overlaying layer, screenshot protection, malicious insider, digital forensic.**

I. Introduction

The protection of intellectual property or sensitive information is a well known challenge in the domain of computer security. There are many ways to protect digital objects from being copied or misused on different stages of processing in a computer system. Files can be encrypted and password protected, but after providing a correct password they will be presented in the plain form to the user. Certain file types can be excluded from the list of potential e-mail attachments to prevent sending out confidential data. However, although these mechanism are effective during object processing, they do not prevent or help when tracing screen capture. Even using Digital Rigts Managements mechanisms can be insufficient to protect the content displayed on the screen. The challenge of the problem is summarized by Raymond Chen: "The graphics system doesn't know, 'Oh, wait, this pixel is special. Don't let anybody read this pixel.' " [5].

A malicious insider may want to steal a piece of intellectual property. To do so, he might try to bypass the first line of defense such as encryption and e-mail filtering by taking screenshots of the protected data and thus passing content-type filters or password protection. As a result, he can cause a information leak that might have severe consequences. If the data was not secured against such an attack by including visible or invisible information unambiguously determining the leak source, it might be impossible to detect the point of failure in the organization.

Tracking the origin of screenshots was widely discussed in the context of the actions of one of the most influential companies in computer games industry - Blizzard. In 2012, a scandal emerged in the online community gathered around a popular MMORPG game, World of Warcraft[1] the . It was related to the content of the screenshots taken while playing the game. Through investigation of the screenshot contents by the player community [34] it turned out that the screenshots contained a hidden watermark. The secret information consisted of data enabling the identification of the screenshot source, i.e. account ID/name, realm date/time the screenshot was taken, IP of the server and information about the realm. This example shows that screen protection against grabbing or capturing its content is getting attention in a business environment.

Real-time screen watermarking provides a method to identify the source of the leak based on the screenshot taken regardless of the content being displayed at the time. In this paper we present a novel watermarking methodology that utilizes the imperfections of the human eye to place an imperceptible watermark over the protected content using the layer overlaying the whole screen. The method presented does not require any dedicated hardware and can be applied to any computer system. High adaptation dynamics of the algorithm allows reacting to the changes on the user's screen with minimal, hardly noticeable delay. Imperceptibility and robustness of the watermark are investigated for varying algorithm configurations. Additionally using QR codes as a watermark increases both the imperceptibility and robustness

[1] www.warcraft.com/

978-1-4799-4223-7/14 $31.00 © 2014 IEEE
DOI 10.1109/ARES.2014.83

by spatial dispersion and built-in error correction. Placement of the watermarks is determined based on the feature points of the image obtained using the FAST Feature Detector [29]. The evaluation was conducted via three scenarios which represent typical office activities, i.e. programming (editing a source code), working with spreadsheets and reading documentation. The results show that the method proposed in this paper can be effectively used to protect the intellectual property and sensitive information in a typical office-like environment, where most of the data is a combination of text, schemes and charts. The contributions of this work are: definition of the requirements for the watermarking on the overlaying layer, system design and implementation using the properties of the HVS, features of the QR codes and the feature points of the displayed image, obtained by using the FAST algorithm.

The remainder of the paper is organized as follows: section II provides the overview of the watermarking classification and presents the state of the art in the fields closely related to our work. The definition of real-time screen watermarking is given is section III, together with a comparison to existing approaches and with the elaboration of the approach-specific needs. Section IV gives the detailed description of our method for applying and detecting the watermark on the overlaying layer. Evaluation of the watermarking scheme is based on the use-case scenarios is presented in section V. Section VI concludes the paper and shows the directions for further work in the field.

II. Background

Digital watermarking is one of the methods that helps to protect intellectual property for various media types. This field was intensively researched for decades and there are many algorithms and methods that allow to combine the watermark with the protected digital content. This section gives an overview of the watermarking techniques classification, methods and their applications. The subsection "Related work" deals with research achievements directly related to our solution.

A. Classification of Watermarking Techniques

According to [32], the most general classification of the digital watermarks is the distinction between visible and invisible watermarks. Invisible watermarks can be classified into fragile, semi-fragile and robust regarding attacks. Fragile watermarks are mostly used to prove data authenticity, because any change in the content results in a destruction of the watermark. Semi-fragile watermarks can survive certain types of attacks, while robust watermarks are designed in a way that any attempt to remove them significantly degenerates content quality. Additionally, a robust watermark itself should resist a typical signal processing operation on the protected media. Another distinction can be made when discussing the need to have original content in order to extract the watermark. In non-blind detection schemes, the original content is needed to extract hidden content. Blind techniques allow to extract the watermark directly from the content. Another classification takes into consideration a domain in which the watermark is applied. The choice of the domain depends on the media type and application. Most frequently used domains are the ones that use mathematical transforms of the signals: Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT). Apart from the transform domain also the spatial domain and the time-frequency domains can be used. Some of these techniques make use of the file specific meta-data [14] or format-related features [9], [28], [3]. Most of the work in digital watermarking focuses on multimedia watermarking.

B. Image and Video Watermarking Techniques

Although historically the most attention has been paid to securing still images, along with the rising bandwidth of the Internet connection and ease of sharing a great amount of data, video watermarking techniques are addressed by many researchers. Given a multitude of the watermark insertion possibilities, a great number of methods have been studied, allowing efficient watermarking of both media types. The overview of existing image watermarking algorithms can be found in recent surveys [27], [4], [18]. The survey by Le [19] focuses on summarizing the watermarking tools, watermark attacks, and benchmarking tools for image watermarking. An overview of robust video watermarking techniques can be found in [25]. Methods of securing video data using watermarks are also a part of the works of Arnold [1] and Hartung [13].

C. Related Work

This subsection presents the state of the art in the fields directly related to the method presented in this publication.

1) Human Visual System and Just Noticeable Distortion: To achieve imperceptibility of the watermark, the watermarking techniques often take advantage of the studies of the Human Visual System (HVS). The HVS model simplifies and describes how the human eye processes and interprets various characteristics of light such as sensitivity to color- and luminescence changes [6]. Studies of the HVS resulted in defining the notion of Just Noticeable Distortion (JND) referring to the maximal distortion not perceived by the Human Visual System [24], [7]. This concept has origins at Weber's Law stating that the just noticeable difference between two values is approximately proportional to the values themselves. Usage of the perceptual models in the context of digital watermarking and data hiding is further described in chapter 8 of [8]. Methods for perceptually adaptive watermarking can also use a correlation-based method to provide a good level of robustness together with a high level of imperceptibility. The example of securing print-scan channel using a watermark based on luminance changes can be also found in [2]. The work of Tao [33] shows the importance of considering the HVS when embedding the watermark and introduces the adaptive watermarking scheme in the DCT domain.

2) Screenshot Identification: The paper by Lee [21] deals with the problem of screenshot identification by analyzing artifacts from an interlaced video. A very similar technique is presented in [20]. However, these techniques assume the identification based on video files and they would not work with motionless screen images. Additionally, they do not give any possibility to identify the originating display, but only solve the classification problem of distinguishing images from a screenshot.

3) Watermarking on Overlay Layer: Within the great amount of various algorithms developed for both images and videos, little attention has been paid to imperceptible watermarking techniques that make use of the overlaying layer. The technique leveraging the overlay layer to protect video content is presented in [22]. The paper discusses a relation between JND and opacity of the overlaying layer containing a watermark generated by the set-top box and presented over the image displayed on the multimedia device. Contrary to the aforementioned solution, this paper proposes a method by which the overlaying layer is generated by the same device as the original content. Additionally, in our solution the watermark position is adapted to the currently displayed content and also the watermark itself contains the information uniquely identifying time and device.

4) Real-time Watermarking: In most cases, the watermarking algorithms focus on processing data files stored on a disc. However, some approaches focus on a real-time content processing. In [17] the authors present a method for real-time watermarking of HD MPEG-2 encoded video that uses low level instructions of the processor. A hardware-based approach for image watermarking is presented in [26]. The paper also describes an extension of the method which allows real-time video-watermarking using a specialized processor. Additionally, the aforementioned technique for the video watermarking on the overlay layer can also be classified as a real-time watermarking scheme. Compared to most of the solutions, our system is designed to work regardless of the format of the data that is being displayed.

5) 2-D Codes in Digital Watermarking: The recent popularity of 2-D codes is also reflected in the area of digital watermarking. A number of methods shows the way to employ watermarks to 2-D code images [35] while others use the code as a watermark that is combined with the original content [15], [31], [23], [11], [36], [16]. The popularity of the latter is mostly caused by the fact that it allows the information to be restored even from a partially damaged code and is generally insensitive to noise [16], [11].

III. Real-time screen watermarking

This section describes the perceptual basis and a high-level design of the system that uses the overlaying layer for watermark placement. The following section (section IV) contains a detailed description of our system design. The aim of our work is to secure the intellectual property that is being displayed on a user's screen from being redistributed by taking a screenshot. Each grabbed screen image should contain the information that allows to identify the originating machine and the person who has taken the screenshot. We show the general solution that uses the overlaying layer to place the watermark on top of the windows stack. In case of an attempt to save the content of the screen using screenshot, the overlaying layer is also stored and therefore allows the identification of the source of the image. Such a design makes the watermarking scheme independent from the type of data format and software that are used on the protected system. Figure 1 shows the general approach that we propose.

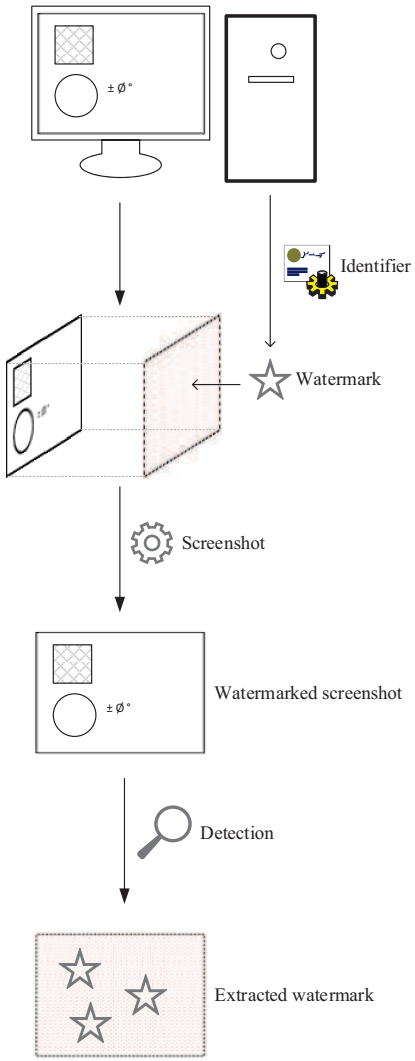

Figure 1. Main idea of the watermarking on overlaying layer.

A. Perceptual Basis

As mentioned in the section II, due to inequalities in sensitivity, the human eye cannot perceive minor luminance changes. This fact can be used to construct a watermark that

hides the information by slightly changing the luminance of the screen. This can be achieved by introducing an overlay layer consisting of white and black pixels. By manipulating opacity of the pixels on the layer, we can effectively change the local luminance of the displayed pixels in a way imperceptible to the user. However, what cannot be noticed by humans can be processed by the computer. The changes in luminance are amplified in a way that the hidden watermark becomes visible.

B. Watermarking System Requirements

Thr real-time screen watermarking scheme shares many features with other watermarking techniques, but also has some more specific requirements and characteristics. These characteristics, same as those of traditional techniques, are concerning the requirements for the watermark itself:

- Imperceptibility: The watermarked image should not be distinguishable from the non-watermarked one.
- Robustness: The watermark should be resistant to typical signal processing and geometrical distortions, such as cropping, scaling and compression

Additionally the method proposed by us has a set of specific requirements that must be satisfied in order to successfully mark the host media. The requirements for the watermarking scheme are:

- Adaption: The algorithm must be adaptable to be able to fit to a changing content on the screen.
- Blind-detection: The watermarking scheme must be a blind scheme, because there is no access to the frames displayed previously. The watermark must be recoverable without using the original, not watermarked image.
- Low computational complexity: The algorithm has to be able to work in real-time in order to quickly respond to the changing content of the protected screen.

C. Comparison with Existing Methods

There are some important differences between the real-time screen watermarking using overlaying layer and traditional video- and image watermarking methods. The major limitation is that there is no possibility of pre-processing the content in order to embed the watermark (as it is possible when watermarking media files). The screen image is not formatted, so there is no possibility to use the format-specific meta-data as a space for the watermark insertion, as it is used in some methods, for example in [14], [3]. Contrary to video watermarking, the dimension of time-sequence domain is also not applicable.

D. Watermark Content

In order to achieve the goal of being able to investigate the source of a screenshot, the watermark should contain information unique to the machine on which it has been taken. The insertion of currently logged user ID together with a time-stamp could greatly increase the chances of finding the source of a leak in case of a security incident. We elaborate more on the information encoded in a watermark in our system design in the section IV-C.

E. Watermark Localization

Generally, the watermark should be put over a content that should be protected in a way that even the processed screenshot contains the information sufficient to extract the watermark. Depending on the domain, various configurations are possible. The section IV-B shows a possible method that works well with content consisting of text, charts and schemes together with some pictures.

IV. System design

This chapter presents the structure of the system fulfilling the requirements defined in the section III. The system makes use of the HVS characteristics to manipulate the color and opacity of the overlaying layer, which results in the changes of the brightness of the currently displayed image. There is a hardly noticeable delay between the moment when the screen changes and the moment when the watermarking algorithm adapts to the new screen image. Thanks to the keeping algorithm simple, we achieved low computational complexity and close-to-real-time functioning. It is worth noting that the delay is proportional to the size of the screen which is watermarked. The block diagram presented in Figure 2 shows the method overview.

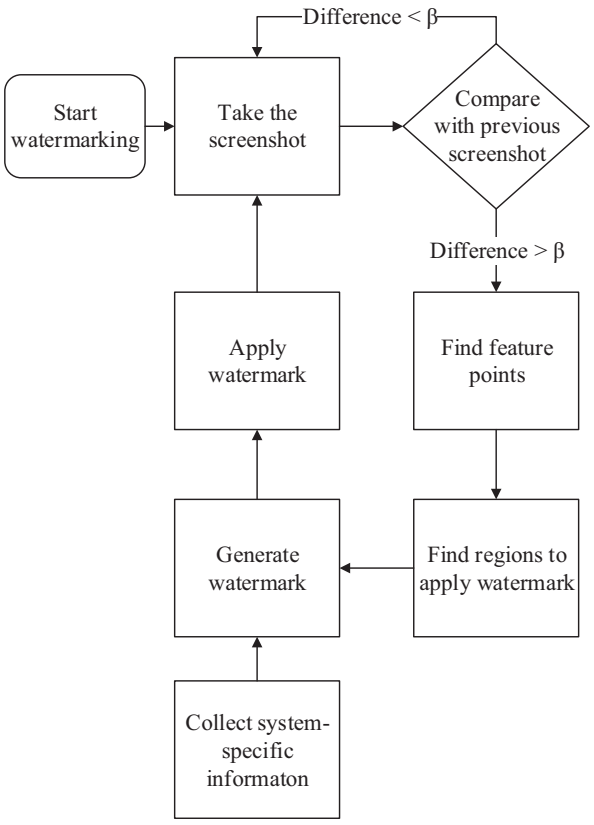

Figure 2. System overview.

A. JND and Overlay Layer Opacity

One of the requirements for this scheme is that the watermark should be imperceptible. The watermark consists of white and black pixels with varying opacity levels denoted α. Therefore we need to know the opacity difference threshold, not perceived by the human eye. Basically, for achromatic pictures, there are two most important factors that have influence on the perception of brightness change. Namely, the luminance difference and the spatial non-uniformity of the background luminance [7]. However, the algorithm used by Chou to calculate the JND has a too high computational complexity and results in unacceptable delays in watermark generation. The authors of [22] simplified the formula proposed in [7] and translated the luminance values into the opacity levels for the white and black pixels. According to their experiments, the imperceptibility threshold for the white pixels of the watermark equals $\alpha_w = 2$ and for the black pixels $\alpha_b = 3$. These values were taken by us as the basis for the watermark generation. Although these values are generally correct, we noticed that it is possible to use $\alpha_w = 3$ and $\alpha_b = 4$ respectively in the areas that are less texture-uniformed in order to increase the watermark strength and to enable easier extraction. As a criterion for the uniformity we take the number of feature points in the image region to be watermarked in relation to the average number of feature points per region. More details of the division of the screen bitmap into regions and the method to find feature points are presented in the following subsection.

B. Watermark Placement

To be able to effectively protect the data, we should keep in mind that the watermark must be imperceptible. We designed the algorithm for placing the watermark on the overlaying layer. The algorithm is based on the feature points detection. The feature point is a term derived from the computer vision terminology and denotes the well defined points of the image, rich in the local information context and resistant to the illumination perturbations [30]. Among many existing detectors we decided to choose the Fast Feature Detector (FAST) introduced by Rosten and Drummond in [29], mainly because of its speed and low computational complexity.

1) Screen Taint: It is crucial to mention that the image of the screen used for the watermarking algorithm should not contain the overlaying layer used for watermarking. The presence of the watermark in the processed image would result in the incorrect feature point detection and, consequently, in an improper watermark placement.

2) Watermarked regions: Distributing the watermark equally over the whole screen would result in a higher visibility of the watermark, because positioning the watermark in the regions having small textural masking capabilities could result in exposing the watermarking. To determine the best watermark placement in terms of imperceptibility and covering the content, the feature points of the current screen image are calculated using the chosen algorithm. Next, the screen image is divided into rectangular regions of size $w \times h$ pixels. The lower bound of the region size is determined by the watermark size (see section IV-C) and the upper bound is limited by the size of the screen ($W \times H$). The influence of the region size on the placement of the watermarks is presented in Figure 3. The smaller the region is, the better the watermark covers protected content. On the other hand, as it is presented in section V, with decreasing size of the region the robustness of the watermark also decreases.

For each region the number of feature points is calculated. If the number of feature points in a given region is higher than the threshold γ, the region will be used as a watermark host. Otherwise, the region remains empty. It should be noted that threshold γ can be calculated dynamically, depending on the number of feature points found. The example results of the placing algorithm are presented in Figure 4, in which the screen is divided into 12 regions, and 4 of them are chosen to be watermarked.

C. Watermark Design

To be able to effectively track the potential source of a leak, the watermark must include some unique tracking information. In our case we used the operating system (OS) identifier (1 character: W for Windows, M for MacOS, U for Unix/Linux), the computer serial number (7 characters), user ID (10 characters) and the timestamp (12 characters). The format of the identifier is presented in Figure 5. For the application, which requires strong privacy protection, the identifying sequence can be additionally encrypted, and cyphertext can be used instead.

We decided that the watermark will be placed in form of the QR Code [12]. The main motivations are that its distributed structure is harder to be noticed by the user when applied on the watermark layer, and the QR code has implemented error correction capabilities, enabling to correctly decode the message even when up to 30% of the data is lost. The standard specifies 40 versions of the QR code from the smallest containing of 21×21 up to 177×177 modules. Regarding the identifier used and the maximal data correction capabilities, the QR code in version 3 has sufficient capacity for the needs of the scheme. The code in that version consists of 29×29 modules allowing to encode up to 35 alphanumerical characters with the highest error correction rate. It should be noticed that raising the length of the message might result in the need to use a QR code version that is capable of storing more data than in our solution.

If the underlying screen image is black, further reduction of the luminance would be undetectable. Therefore, for dark background, the watermark is inverted and white pixels are drawn on the overlay layer instead of black ones.

D. Watermark Detection

To extract the watermark from the screenshot, basic image-processing operations must be applied to the image. Since the watermark is changing the local luminance of the pixels, the detection process focuses on amplifying the difference of luminance between pixels. Depending on the captured image, different techniques can be applied to extract the watermark.

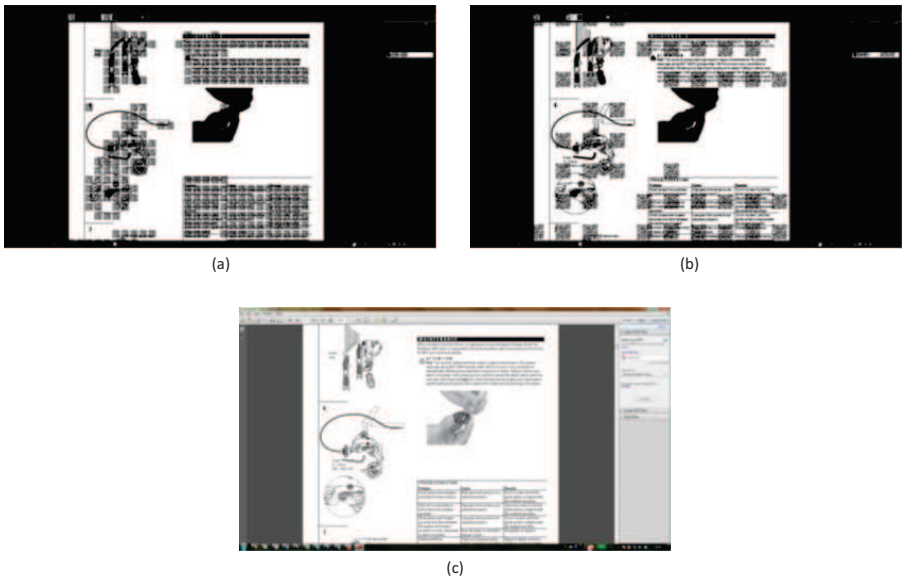

Figure 3. Watermark placement depending on the region size. Smaller region size (a), bigger region size (b), original image (c).

10	7	9
3	25	10
22	19	15
5	3	2

Figure 4. The choice of the regions to place the watermark with $\gamma > avg$, where avg is the arithmetic average of the number of feature points in regions. The numbers in the rectangles denote the number of feature points detected in the region. For the presented example $avg = 10,83$. Green marked regions are used as watermark hosts.

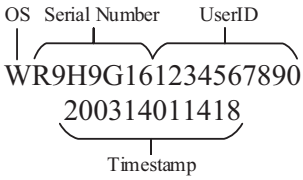

Figure 5. Structure of the identifier used in the system.

In the simplest cases, when the watermark was located over the relatively uniform area, a modification of histogram curves, that results in changes to contrast and brightness, brings satisfying watermark readability. In more complex situations, when the watermark was located over the non-uniform elements, such as colorful pictures, the method allowing to find the watermark is histogram-based image segmentation. In extreme situations it might be necessary to locate the well-preserved parts of the watermark in different regions of the image and to recreate the QR code by overlapping these parts. The example showing the watermarked region and extracted watermark is presented in Figure 6.

E. Optimization

This subsection presents some modification of the core concept, increasing the algorithm's dynamics and robustness.

1) Reusing Watermark Locations: This optimization enables to reduce the algorithm's iteration execution time by using the watermark locations calculated in a previous iteration. It is motivated by the observation that in the typical office usage scenario the displayed screen changes much less frequently than the algorithm iteration time. Reducing the time by reusing the previously calculated locations increases the dynamic adaptation features of the method, since in the moment of the significant change of the pixels' configuration on the screen it is able to faster adapt the watermark layer to the new configuration. The decision wether recalculating or reusing the location points is based on the comparison of the previously taken image screen with the current one.

Numerous methods exist to compare two images, as presented in [10]. In our design, that is oriented according to the computational efficiency, we decided to use the proportion of the pixels changed to the total number of pixels as the deciding factor. This metric can be effectively computed and provides sufficient accuracy. If the ratio between the screen's pixels changed to the total number of pixels is higher than a defined

(a) (b)

Figure 6. Part of the screenshot before (a) and after (b) watermark extraction.

threshold β, the watermark locations are being recalculated.

2) QR Code Rotation: To increase the robustness of the watermark, in some of the regions a rotated watermark can be used instead of the originally generated one. Rotation increases the chance to recreate the watermark from existing parts even if the watermarked image was substantially modified. Since QR codes are geometrically well-defined, it is relatively easy to find matching points and to recreate the code that is correctly interpreted by the reading device.

V. Evaluation

This section contains the results of the evaluation of the presented method for watermarking screens in a real time. The configuration of the system was tuned to effectively watermark the screen of the computer used in an office environment, where most of the textual data is displayed together with schemes, small pictures and charts.

A. Use Case Scenarios

For the evaluation of our method we considered three scenarios in which a user interacts with different types of data and interfaces, namely: programming environment, spreadsheet work and documentation reading. During the interaction, the user can use the screenshot feature (or a snipping tool) at any moment to save the image of the screen. After applying possible distortions, the resulting images are analyzed in order to retrieve the watermark. Next we attempt and read it using the QR code scanner of a smartphone.

a) Programming Environment: In this scenario, user is presented with a source code of a Java program using Eclipse Integrated Development Environment[2]. His interaction with the environment consists of editing the .java file by adding to, deleting and editing the code. The user can also run the program, enter the debug mode and use all options offered by the environment.

[2]https://www.eclipse.org

b) Spreadsheet Work: In this scenario the user is presented with a spreadsheet containing the imported data set together with corresponding charts. The user can modify the data, scroll through the rows, change the chart type, add and delete charts, etc. This use case scenario uses Microsoft Office Excel.

c) Documentation Reading: The user is presented with a manual in the pdf format, consisting of logos, tables, schemes, icons and text. The user may read the manual using Adobe Reader software and use all of the features provided.

B. Environment Configuration

The evaluation was conducted by using the prototype application providing real-time screen watermarking using overlaying layer. The software was written in Java and run on a typical, modern PC-class computer. The overlaying layer is technically a window that is always displayed on the top of other windows. The overlaying layer does not capture any input signals from peripherals and therefore provides both visual and behavioral transparency. To effectively process images, the software makes use of the java bindings of the OpenCV[3] image processing library. The display connected to computer was a Samsung SyncMaster P2770. The screen resolution was set to 1920×1080 pixels. For that screen size the average algorithm iteration time is about 1s. The smartphone used to read the QR codes was HTC Desire with ”Neo Reader” QR codes detector. The watermarks were detected by manipulating the screenshot image with Gimp[4] software.

1) Algorithm Parameters: Parameters W and H are screen width and height respectively. The method was evaluated for four different watermark regions sizes:

- small (S), where $w = 40, h = 40$
- medium (M), where $w = 120, h = 135$
- large (L), where $w = 320, h = 360$
- extra large (XL), where $w = 480, h = 540$

[3]http://opencv.org/
[4]http://gimp.org/

Parameters β and γ were obtained empirically and set to $\beta = 0.4 * avg$ where avg is the average number of the feature points in the regions and $\gamma = 0.001$.

C. Experiment Design

The experiments aimed to check the watermark imperceptibility, the efficiency of marking the screen andthe robustness against the most possible distractions (attacks). For each of the screenshots taken the following operations were applied:

- JPEG compression (with quality 100%, 95%, 90%, 85%, 80%, 75%, 60%, 50%)
- resizing (70% and 130%)
- cropping (400×400 px at point (200,200))

. The experiment was performed for each of the aforementioned use case scenarios.

D. Experiment Results

Tables I, II, III show the results of the experiment. Symbols for watermark imperceptibility:

- × denotes high visibility, meaning that it was possible to perceive the watermark on the screen
- ⋆ denotes minor artifacts visible, meaning that small parts of the watermark were noticeable
- ⋆⋆ denotes imperceptible watermar.

Symbols for attacks :

- × denotes unreadable watermark
- ⋆ denotes possibility to extract the watermark after multistage processing or recreating the watermark from the parts preserved in a different parts of the screenshot image
- ⋆⋆ denotes possibility to read the watermark directly from the processed image using the application installed on the smartphone, after basic image processing

E. Results Discussion

The results show that the system is able to fulfill the requirements for the real-time screen watermarking defined in the section III. The following subsection contains the discussion of the results regarding imperceptibility and robustness.

1) Imperceptibility: For medium and small settings of the watermarked region size the watermark was imperceptible in all of the use case scenarios. For bigger regions' sizes the watermark was just noticeable. It shows the trade-off between the level of visibility of the watermark and the accuracy of the placement, meaning covering only the regions of the screen containing content to be protected.

2) Robustness: For the unprocessed screenshots it was possible to extract the watermark in all tested cases. When it comes to the robustness of the watermark, bigger regions' sizes have better robustness against most of the tested attacks. The exception is cropping, which is better survived by the smaller watermarks. For medium region sizes JPEG compression preserving a good quality of the image enables the watermark to be extracted. Watermarks located in the small region were fragile even to the JPEG compression preserving relatively good quality of the picture. The increase of the robustness against compression grows with the size of the watermark. The biggest watermark can survive JPEG compression up to a quality of 50%.

Table I
RESULTS OF THE EXPERIMENTS FOR USE CASE SCENARIO - PROGRAMMING ENVIRONMENT.

Programming environment				
Impercepctibility	⋆⋆	⋆⋆	⋆	⋆
Attack/Size	**S**	**M**	**L**	**XL**
None	⋆⋆	⋆⋆	⋆⋆	⋆⋆
JPEG100	⋆	⋆⋆	⋆⋆	⋆
JPEG95	×	⋆	⋆	⋆
JPEG90	×	⋆	⋆	⋆
JPEG85	×	⋆	⋆	⋆
JPEG80	×	×	⋆	⋆
JPEG75	×	×	⋆	⋆
JPEG60	×	×	×	⋆
JPEG50	×	×	×	×
Crop	⋆⋆	⋆⋆	⋆	×
Scale Up	×	×	⋆⋆	⋆⋆
Scale Down	×	×	⋆⋆	⋆⋆

VI. CONCLUSIONS AND FUTURE WORK

In this work, a new method to protect the intellectual property and sensitive information displayed on the screen was presented. It is achieved by watermarking the screen image, placing the watermark on the overlay layer. This solution enables to watermark the underlying screen image close to real time, regardless of the format of the displayed data. The watermark hidden in the screenshot can be used to track the source of the information leak, that were caused by grabbing the content of the screen and saving it as an image. The content of the watermark enables to identify the machine, the user and the time, which are all important in a forensic investigation. The requirements for this kind of system were discussed and the features of such a system were described.

The presented solution makes use of the Human Vision System's features and the imperfections of the human eye. The designed system does not require any additional hardware equipment and is realized using only software. The watermark is located based on the feature points of the image calculated using the FAST Feature Detector. As a watermark we are using a QR code due to features such as spatial dispersion and error correction, allowing to increase the watermark's imperceptibility as well as robustness. The number of parameters allows to adjust the algorithm to the particular needs and to increase

Table II
RESULTS OF THE EXPERIMENTS FOR USE CASE SCENARIO - SPREADSHEET WORK.

Spreadsheet work				
Impercepctibility	⋆⋆	⋆⋆	⋆	⋆
Attack/Size	**S**	**M**	**L**	**XL**
None	⋆⋆	⋆⋆	⋆⋆	⋆
JPEG100	⋆⋆	⋆⋆	⋆⋆	⋆
JPEG95	×	⋆	⋆⋆	⋆
JPEG90	×	⋆⋆	⋆	⋆
JPEG85	×	×	⋆	⋆
JPEG80	×	×	⋆	⋆
JPEG75	×	×	⋆	⋆
JPEG60	×	×	×	⋆
JPEG50	×	×	×	×
Crop	⋆⋆	⋆⋆	⋆	×
Scale Up	×	×	⋆⋆	⋆⋆
Scale Down	×	×	⋆⋆	⋆⋆

Table III
RESULTS OF THE EXPERIMENTS FOR USE CASE SCENARIO - DOCUMENTATION READING.

Documentation reading				
Impercepcibility	⋆⋆	⋆⋆	⋆	⋆
Attack/Size	**S**	**M**	**L**	**XL**
None	⋆⋆	⋆⋆	⋆⋆	⋆
JPEG100	⋆⋆	⋆⋆	⋆⋆	⋆
JPEG95	×	⋆	⋆⋆	⋆
JPEG90	×	⋆⋆	⋆	⋆
JPEG85	×	⋆	⋆	⋆
JPEG80	×	×	⋆	⋆
JPEG75	×	×	⋆	⋆
JPEG60	×	×	×	⋆
JPEG50	×	×	×	×
Crop	⋆⋆	⋆⋆	⋆	×
Scale Up	×	×	⋆⋆	⋆⋆
Scale Down	×	×	⋆⋆	⋆⋆

the robustness of the watermark against various attacks. Three use case scenarios - programming, work with spreadsheets and reading a documentation - were tested to show the applicability of this method to typical office work with documents of mixed structure consisting mostly of text, charts and small pictures.

The results of the evaluation show that a combination of multiple watermark sizes in the same time should significantly increase the robustness of the watermark. Another improvement that in our opinion could be a valuable extension to our work is the automation of the watermark detection process. For more strict robustness requirements the watermarking techniques using the overlay layer to embed the watermark in the transformed domain should be further explored. Our work shows that the overlay layer is an up-and-coming solution to watermark content regardless of what is being displayed. Based on our experience, investigating techniques that modify the screen image before its being displayed is also a promising research direction.

REFERENCES

[1] Michael Konrad Arnold, Martin Schmucker, and Stephen D Wolthusen. *Techniques and applications of digital watermarking and content protection*. Artech House, 2003.

[2] Paulo Vinicius Koerich Borges and Joceli Mayer. Document watermarking via character luminance modulation. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 2, pages II–II. IEEE, 2006.

[3] M Talstra Celik, J Lemma, and S A Katzenbeisser. Camcorder capture robust low-complexity watermarking of mpeg-2 bit-streams. *Image*, 2007.

[4] Abbas Cheddad, Joan Condell, Kevin Curran, and Paul Mc Kevitt. Digital image steganography: Survey and analysis of current methods. *Signal Processing*, 90(3):727–752, 2010.

[5] Raymond Chen. Windows confidential: What you see is what you see, 2011. http://technet.microsoft.com/en-us/magazine/hh241037.aspx.

[6] Chun-Hsien Chou and Chi-Wei Chen. A perceptually optimized 3-d subband codec for video communication over wireless channels. *Circuits and Systems for Video Technology, IEEE Transactions on*, 6(2):143–156, 1996.

[7] Chun-Hsien Chou and Yun-Chin Li. A perceptually tuned subband image coder based on the measure of just-noticeable-distortion profile. *Circuits and Systems for Video Technology, IEEE Transactions on*, 5(6):467–476, 1995.

[8] Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker. *Digital watermarking and steganography*. Morgan Kaufmann, 2007.

[9] Wu Ge and Li Guiying. Collusion-resistant blind video watermarking based on h264. In *Computer Application and System Modeling (ICCASM), 2010 International Conference on*, volume 11, pages V11–9. IEEE, 2010.

[10] A Ardeshir Goshtasby. Similarity and dissimilarity measures. In *Image Registration*, pages 7–66. Springer, 2012.

[11] Su-Young Han, Eui-Hyun Jung, and Seong-Yun Cho. A robust digital watermarking adopting 2d barcode. In *Computer Analysis of Images and Patterns*, pages 717–723. Springer, 2005.

[12] M. Hara, T. Nagaya, T. Nojiri, Y. Uchiyama, and M. Watabe. Optically readable two-dimensional code and method and apparatus using the same, 1998. US Patent 5,726,435.

[13] Frank Hartung and Martin Kutter. Multimedia watermarking techniques. *Proceedings of the IEEE*, 87(7):1079–1107, 1999.

[14] Hsiang-Cheh Huang and Wai-Chi Fang. Metadata-based image watermarking for copyright protection. *Simulation Modelling Practice and Theory*, 18(4):436–445, 2010.

[15] Xun Jin and JongWeon Kim. Imperceptibility improvement of image watermarking using variance selection. In *Computer Applications for*

Web, Human Computer Interaction, Signal and Image Processing, and Pattern Recognition, pages 31–38. Springer, 2012.

[16] Eui-Hyun Jung and Seong-Yun Cho. A robust digital watermarking system adopting 2d barcode against digital piracy on p2p network. *IJCSNS International Journal of computer science and netwoork secuity*, 6(10):263, 2006.

[17] In-Koo Kang, Dong-Hyuck Im, Heung-Kyu Lee, and Young-Ho Suh. Implementation of real-time watermarking scheme for high-quality video. In *Proceedings of the 8th workshop on Multimedia and security*, pages 124–129. ACM, 2006.

[18] Manpreet Kaur, Sonika Jindal, and Sunny Behal. A study of digital image watermarking. *Journal of Research in Engineering and Applied Sciences*, 2(2):126–136, 2012.

[19] Thi Hoang Ngan Le, Kim Hung Nguyen, and Hoai Bac Le. Literature survey on image watermarking tools, watermark attacks and benchmarking tools. In *Advances in Multimedia (MMEDIA), 2010 Second International Conferences on*, pages 67–73. IEEE, 2010.

[20] Ji Won Lee, Min-Jeong Lee, Hae-Yeoun Lee, and Heung-Kyu Lee. Screenshot identification by analysis of directional inequality of interlaced video. *EURASIP J. Image and Video Processing*, 2012:7, 2012.

[21] Ji-Won Lee, Min-Jeong Lee, Tae-Woo Oh, Seung-Jin Ryu, and Heung-Kyu Lee. Screenshot identification using combing artifact from interlaced video. In *Proceedings of the 12th ACM workshop on Multimedia and security*, pages 49–54. ACM, 2010.

[22] Ji-Won Lee, Tae-Woo Oh, Min-Jeong Lee, Heung-Kyu Lee, and Hae-Yeoun Lee. Video watermarking on overlay layer. In *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2011 Seventh International Conference on*, pages 85–88. IEEE, 2011.

[23] Jihah Nah, Jizhe Cui, and Jongweon Kim. A multiple audio watermarking algorithm using 2d code and hadamard transform. *matrix*, 100:6, 2012.

[24] Yaqing Niu, Matthew Kyan, Sridhar Krishnan, and Qin Zhang. A combined just noticeable distortion model–guided image watermarking. *Signal, Image and Video Processing*, 5(4):517–526, 2011.

[25] Rini T Paul. Review of robust video watermarking techniques. *IJCA Special Issue on Computational Science*, 3:90–95, 2011.

[26] Guillaume Petitjean, J-L Dugelay, Sophie Gabriele, Christian Rey, and Jean Nicolai. Towards real-time video watermarking for system-on-chip. In *Multimedia and Expo, 2002. ICME'02. Proceedings. 2002 IEEE International Conference on*, volume 1, pages 597–600. IEEE, 2002.

[27] Vidyasagar M Potdar, Song Han, and Elizabeth Chang. A survey of digital image watermarking techniques. In *Industrial Informatics, 2005. INDIN'05. 2005 3rd IEEE International Conference on*, pages 709–716. IEEE, 2005.

[28] Radu O Preda and Dragos N Vizireanu. Blind watermarking capacity analysis of mpeg2 coded video. In *Telecommunications in Modern Satellite, Cable and Broadcasting Services, 2007. TELSIKS 2007. 8th International Conference on*, pages 465–468. IEEE, 2007.

[29] Edward Rosten and Tom Drummond. Fusing points and lines for high performance tracking. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1508–1515. IEEE, 2005.

[30] Cordelia Schmid, Roger Mohr, and Christian Bauckhage. Evaluation of interest point detectors. *International Journal of computer vision*, 37(2):151–172, 2000.

[31] V Seenivasagam and R Velumani. A qr code based zero-watermarking scheme for authentication of medical images in teleradiology cloud. *Computational and mathematical methods in medicine*, 2013, 2013.

[32] Srdjan Stankovi, Irena Orovi, and Ervin Sejdi. Digital watermarking. In *Multimedia Signals and Systems*, pages 255–284. Springer US, 2012.

[33] Bo Tao and Bradley Dickinson. Adaptive watermarking in the dct domain. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, volume 4, pages 2985–2985. IEEE Computer Society, 1997.

[34] user: Sendatsu. Looking inside your screenshots, 2012. Available at http://www.ownedcore.com/forums/world-of-warcraft/world-of-warcraft-general/375573-looking-inside-your-screenshots.html. Accessed 03.2014.

[35] Rong-sheng Xie, Ke-shou Wu, Gao-pan Xu, and Miao Ouyang. Research on anti-counterfeiting quick response 2d barcode techniques based on digital watermark. *Journal of Shanghai Jiaotong University (Science)*, 18(4):443–447, 2013.

[36] Shanjun Zhang and Kazuyoshi Yoshino. Embedding qr code watermark in divided wavelet domain. In *TENCON 2004. 2004 IEEE Region 10 Conference*, pages 287–290. IEEE, 2004.