Combining digital Watermarks and collusion secure Fingerprints for customer copy monitoring

Jana Dittmann

GMD - German National Research Center for Information Technology, Darmstadt, Germany
jana.dittmann@gmd.de

ABSTRACT

Digital watermarking is the enabling technology to prove ownership on copyrighted material, detect originators of illegally made copies, monitor the usage of the copyrighted multimedia data and analyze the spread spectrum of the data over networks and servers. Embedding of unique customer identification as a watermark into data is called fingerprinting to identify illegal copies of documents. Basically, watermarks embedded into multimedia data for enforcing copyrights must uniquely identify the data and must be difficult to remove, even after various media transformation processes. Digital fingerprinting raises the additional problem that we produce different copies for each customer. Attackers can compare several fingerprinted copies to find and destroy the embedded identification string by altering the data in those places where a difference was detected.

In our paper we present a technology for combining a collusion-secure fingerprinting scheme based on finite geometries and a watermarking mechanism with special marking points for digital images. The only marking positions the pirates can not detect are those positions which contain the same letter in all the compared documents, called intersection of different fingerprints. The proposed technology for a maximal number d of pirates, puts enough information in the intersection of up to d fingerprints to uniquely identify all the pirates.

Keywords: Collusion secure fingerprinting, watermarking, copyright protection, customer copy identification

1. MOTIVATION

Digital watermarking is the enabling technology to prove ownership on copyrighted material, detect the originator of illegally made copies, monitor the usage of the copyrighted multimedia data and analyze the spread spectrum of the data over networks and servers. Embedding of unique customer identification as a watermark into data is called fingerprinting to identify illegal copies of documents. Basically, watermarks, labels or codes embedded into multimedia data for enforcing a copyright must uniquely identify the data as property of the copyright holder, and must be difficult to be removed, even after various media transformation processes. Thus the goal of a label is to always remain present in the data. Digital fingerprinting which embeds customer information into the data to enable detection of license infringement raises the additional problem that we produce different copies for each customer. Attackers can compare several fingerprinted copies to find and destroy the embedded identification string by altering the data in those places where a difference was detected.

In our paper we present a technology for combining a collusion-secure fingerprinting scheme based on finite geometries and a watermarking mechanism with special marking points for digital images. The only marking positions the pirates can not detect are those positions which contain the same letter in all the compared images, called intersection of different fingerprints. The proposed technology for a maximal number d of pirates, puts enough information in the intersection of up to d fingerprints to uniquely identify all the pirates. The next chapter explains the fingerprinting algorithm. Based on the fingerprinting algorithm we present then the watermarking algorithm and the test results.

2. COLLUSION SECURE FINGERPRINTING ALGORITHM

A digital fingerprinting scheme consists of a number of marking positions in the document, a watermarking algorithm to embed letters from a certain alphabet at the marking positions, a fingerprinting algorithm which selects the letters to be embedded for each marking position depending on the number i of the copy and a pirate tracing algorithm which, on input of a modified document, outputs at least one number i of a copy that was used in constructing the modified document.

Different copies of a document containing digital fingerprints differ at most at these marking positions. A powerful attack to remove a fingerprint therefore consists of comparing two or more fingerprinted documents and to alter these documents randomly in those places where a difference was detected. If three or more documents are compared, a majority decision can be applied to improve this kind of attack: For the area where the documents differ, choose the value that is present in most of the documents.
The only marking positions the pirates can not detect are those positions which contain the same letter in all the compared documents. We call the set of these marking positions the intersection of the different fingerprints.

In this section we propose a fingerprinting algorithm that, for a maximal number \( d \) of pirates, puts enough information in the intersection of up to \( d \) fingerprints to uniquely identify all the pirates. A fingerprinting scheme with this property is called \( d \)-detecting. Another important parameter is the number \( n \) of copies that can be generated with such a scheme. We use techniques from finite projective geometry \([1, 6]\) to construct \( d \)-detecting fingerprinting schemes with \( q+1 \) possible copies. These schemes need \( n = q^d + q^{d-1} + \ldots + q + 1 \) marking positions in the document.

We use a binary alphabet in our scheme: Therefore marking positions with a "1" embedded will be called "marked", those with a "0" will be called "unmarked". Unmarked positions are not altered compared to the original document.

The problem of collusion-secure fingerprinting has originally been described and solved by D. Boneh and J. Shaw \([2]\). Our approach, is different from \([2]\), since we put the information to trace the pirates into the intersection of up to \( d \) fingerprints. In the best case (e.g. automated attacks like computing the average of fingerprinted images) this allows us to detect all pirates, in the worst case (removal of individually selected marks) we can detect the pirates with a negligibly small one-sided error probability, i.e. we will never accuse innocent customers.

2.1. Two simple examples

The smallest possible example of a fingerprinting scheme (and the smallest projective space) is shown in Picture 1. The projective space \( \text{PG}(2,2) \) of dimension 2 (i.e. it is a plane) and order 2 (i.e. there are \( 2+1=3 \) points on each line) has 7 points and 7 lines (the circle through the points 2, 4 and 6 counting as a line).

![Figure 1: A 2-detecting fingerprinting scheme with 3 possible copies in the finite projective space PG(2,2).](image)

To implement this scheme, we need 7 marking positions in the document, each associated with one point of \( \text{PG}(2,2) \). This "association" must be secret and highly nonlinear to destroy all purely geometric information in the document. (In the following, if no confusion is possible, we will not distinguish between the terms "point" and "marking position"). Each fingerprint consists of 3 marked and 4 unmarked points. E.g. to embed fingerprint 2 in a document, the marking positions corresponding to the points 1, 2 and 3 will be marked, the rest remains unmarked.

This scheme is 2-detecting because any two of the lines \( \{1,2,3\}, \{3,4,5\} \) and \( \{1,5,6\} \) intersect in a unique point. A possible attack could be the following: Customer 1 buys a copy of the document with fingerprint 1, and customer 2 gets a copy with fingerprint 2. They compare their documents to generate a pirate copy. The two documents differ at the marking positions 2, 3, 5 and 6. In the worst case, they can unmark all those points. However, they can not detect marking position 1 if a "good" watermarking scheme has been used.

If they sell the pirated copy, it will eventually fall into the hands of the copyright owner. The copyright owner will then start the pirate tracing algorithm and detect point 1 and from this point the two pirate customers 1 and 2.

Remark: Following a different strategy, it is possible for the pirates to generate with probability \(\frac{1}{q}\) a document where detection of pirates is not possible. This can be done by guessing, in each document, the marking point that belongs to point 5 or 3, resp. By leaving these marks unchanged, the pirates make it impossible to decide whether customer 1 and 2, or 2 and 3, or 1 and 3, have generated this document. This problem has already been described in \([2]\). However, when \( d \) and \( q \) increase, this probability becomes negligible.

For details of the mathematical model and the generalization see \([8]\). The main disadvantage of the Schwenk/Uneberberg scheme is the limited number of copies that can be produced. This situation can be improved by combining two (or more) randomly chosen hyperplanes into a fingerprint. If we use each hyperplane only once, then this scheme is still \( d \)-detecting, but allows for \(\frac{1}{2}(q^d + q^{d-1} + \ldots + q + 1) \) copies. This scheme poses some problems concerning the traitor tracing algorithm and will be studied in the next phase of our project.
3. DIGITAL WATERMARKING FOR COLLUSION SECURE FINGERPRINTING FOR IMAGES

Usually the known watermarking techniques spread the watermarking information all over the image data [3,4,5]. The embedding of the fingerprinting information with watermarking techniques requires an optimized watermarking scheme in the context of digital fingerprints since we have different requirements for the watermarking scheme here. Our watermarking scheme needs special marking points to integrate the fingerprinting information and to build the intersection of remaining fingerprinting elements after an attack in the intersection region.

In the first section we present the watermarking algorithm in general for a maximal number d of pirates and for number q of copies that can be generated with the proposed fingerprinting scheme. In the following sections we describe the detailed embedding and retrieval algorithm. The watermarking algorithm is designed to use the original image in the retrieval process to get better results and avoid failures of customer detection.

3.1. Watermarking Algorithm

Digital Watermarking is used to embed customer information generated by the fingerprinting algorithm to trace illegal image copies. Current digital watermarking techniques usually would embed the generated fingerprinting information FP randomly all over the image with the disadvantage, that the intersection of the proposed fingerprints can not be used to find attackers after comparing attacks of different customer copies. To use the excellent properties of the fingerprint to conclude to the customers which attacked the watermark we build a watermarking scheme with a fixed number of marking points in each copy of the image. The fingerprinting algorithm selects the letters, the FP vector over the binary alphabet \([0,1]\). The watermarking algorithm embeds this binary FP vector at the chosen marking positions.

The fingerprinting algorithm generates binary FP vectors of length \(q^d+q^{d-1}+\ldots+q+1\) for each image. With this fingerprinting scheme a d-detecting binary FP vector can be generated to build q customer copies of the image. Each customer gets his specific binary FP vector which elements will be embedded at the same marking positions of the image. With these construction the only marking positions the pirates can detect are the differences in the binary FP vectors and they do not detect those positions which contain the same letter in all the compared images, the intersection of the fingerprints FP. Based on the remaining information it is possible to conclude to the customers.

To give the watermarking scheme more robustness we embed each binary FP vector \(r_1\) times, \(r_1\) redundant, into the image. With these construction we need \(r_1 \cdot (q^d+q^{d-1}+\ldots+q+1)\) marking positions. The marking positions will be generated randomly with a image specific secret user key of the copyright holder and image size as parameter. The embedding of the binary FP vector into the image at the defined marking positions is performed in the frequency domain to be more robust against compression. The image is divided into blocks, transformed into the frequency domain using a discrete cosine transformation and quantized. The blocks correspond to the marking positions. If the block was selected randomly as marking position the DCT coefficients are modified with a watermarking sequence depending on the FP vector element. The retrieval uses the original and check image to evaluate the embedded watermarking sequence at the marking positions and retrieve the binary FP vector. The fingerprinting algorithm gets the vector as input parameter and produces the customer list.

3.2. Fingerprint Embedding

The detailed embedding of the fingerprinting vector for each customer copy is performed in three steps.

In the first step the fingerprinting vector FP for the customer is generated. The number of customer which can be delivered with a d-detecting fingerprint depends on the maximal number of available marking positions of the image. In step 2 the position sequences, the marking positions, are generated from the user key as a seed with a secure random number generator. In the order of the generated position sequence every marking position block is now discrete cosine transformed and the fingerprinting vector is embedded in the following way:

1. Parameter:
   - Image I with height h and width w
   - Binary FP vector \(v\) with length \(n = q^d+q^{d-1}+\ldots+q+1\) for example \(n=13, d=2, q=3\) (3 customer): \(v_1=(0001001011000), v_2=(0010001100100), v_3=(0100100100100)\),

2. Redundancy for vector embedding: e.g. \(r_1 = 3\)

3. Redundancy for watermarking intensity of the single marking points \(r_2\): e.g. \(r_2 = 10\)

4. Embedding sequence for each vector element at the marking positions \(R\): \([-k,k]\), \(-4 \leq k \leq 4\), parameter value \(k\) influences robustness and visibility

5. Secret user key: \(k\)

6. Watermarking strength: WMStrength

7. Calculation of the 8x8 blocks of the image: \(m\)

8. Calculation of the redundancy factor \(r_1\), maximal \(n \cdot r_1 = m\)

9. Pseudo random generation of \(r_1 \cdot n\) block marking positions \(x_{ij}\) with the user key UK

6/3
5. Before we embed the binary bits into the DCT coefficients of the marking blocks directly, we increase the signal difference by a generation of the n random sequences R_i [-k,k], k=4, for embedding the single n fingerprinting information bits. Each FP vector element will be then embedded r_1 (e.g. 3) times and modifies r_2 (e.g. 10) DCT coefficients, by using the random sequence R_{ij} for each FP vector bit with r_1 * r_2 (30) redundancy:

<table>
<thead>
<tr>
<th>FP vector bits</th>
<th>Random sequences for the marking Positions n * r_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>x_i = (x_0, x_1, x_2, x_3, ..., x_n)</td>
<td>R_{i,1}, R_{i,2}, R_{i,3}, ..., R_{i,r_1}</td>
</tr>
</tbody>
</table>

Table 2: Random marking sequences R_{ij} for the marking positions, each R_{ij} has length r_2

For example, r_1 = 3, r_2 = 10
R_0 = (R_{0,1}, R_{0,2}, R_{0,3}) = (3,0,3,-1,1,0,3,-4,1|2,2,0,1|3,1,1,-4,1,0,0,3,1)
... R_{n-1,123} = (R_{12,1}, R_{12,2}, R_{12,3}) = (4,0,-4,-1,1,1,0,2,-4,1,2,2,0,1|1,-4,1,-2,1,0,3,4,3,1)

6. Embedding of the FP vector v = (x_0, x_1, x_2, x_3, ..., x_n) with redundancy r_1, e.g. v_{ij} = (x_{ij}, x_{ij}, 0,0,0,0,0,0,0) using the random sequence R_i. For each vector bit we use R_i = (R_{ij}, R_{ij}, R_{ij}) to embed v_i = (x_{ij}, x_{ij}, 0,0,0,0,0,0) with redundancy r_2 by adding the R_{ij} to the DCT coefficients when x_i is 1 otherwise we embed nothing.

6. Retransform the luminance values, inverse quantization, inverse DCT and replacing original luminance of the block with the modified one.

Figure 2 Embedding into the marking blocks DCT coefficients, e.g. x_{ij}

The image has now modified luminance blocks if the FP vector bit is one else the original image block remains. The attacker can now work together, find differences, but the intersections of the fingerprinting information remains present for evaluation during retrieval.
3.3. Fingerprint Retrieval

The retrieval of the fingerprinting information is performed with the check image and the original image to find the embedded watermarking sequence in the marking positions.

We have three main steps to retrieve the customer fingerprints back, the position generation, the retrieval of the watermarking information, which is the fingerprinting vector, and the evaluation of the extracted fingerprint to find the customer IDs. The input parameter of the position generation and retrieval are the user key, the original image and check image.

First we calculate the differences between the original and the check image. All following steps use this difference image for retrieval. It is decoded into RGB values and the position generation of the marking points is performed. Then the marking blocks luminance values are DCT transformed and quantized. The random sequences \( R_i \) are generated and will be searched in the next steps over all marking position points. If we find a conformity we interpret the marking position with a 1 otherwise 0. The detailed retrieval steps are as follows:

1. Parameter:
   - \( d \) and \( q \) for the fingerprinting system
   - Redundancy factor \( r_1 \), e.g. \( r_1 = 3 \)
   - Redundancy for watermarking intensity of the single marking points \( r_2 \); e.g. \( r_2 = 10 \)
   - Search sequence for each vector element at the marking positions \( R: [-k,k] \), \(-4 \leq k \leq 4\), parameter \( k \) influences robustness and visibility, \([-4,4]\)
   - Secret user key: UK
   - Tolerance level
   - Geometric transformation estimation: user input of scaling, rotation and/or cropping

2. Evaluation of geometric distortions (comparison of original and watermarked image, best matching decision) and calculation of difference image between original and check image

3. Pseudo random generation of \( n \) block positions with the user key UK

4. Generation of the \( n \) random sequences \( R_i: [-k,k] \), \( k=4 \):

   - In our example:
     \( R_0 = (R_{01}, R_{02}, R_{03}) = (3,0,3,-1,1,0,3,-4,1,4,2,2...0,1,3,1,1,-4,1,0,0,0,3,1) \)
     ... 
     \( R_{n=12} = (B_{12,1}, B_{12,2}, B_{12,3}) = (4,0,-4,-1,1,0,2,-4,1,2,3,2...0,1,-4,1,1,-2,1,0,3,4,3,1) \)

5. Comparing the generated \( R_i \) with appropriate DCT coefficients at the marking positions. If there is a conformity we retrieve a 1 else a 0

4. TESTRESULTS

We have tested our implementation with different images and performed first a robustness test of the watermarking algorithm and second the recognition test of attackers, the evaluation of the fingerprinting information.

Ten examples demonstrate the capabilities and the shortcomings of the algorithm. We tested compression, scaling and StirMark [7] operations. StirMark combines various attacks. It simulates distortions caused by a printing and rescanning process. Furthermore it introduces some minor geometric distortions like stretching, shearing, rotations and shifting. It is reported that StirMark is very effective against most even commercial watermarking techniques. However, the distortions introduced by StirMark are unrecognisable.

The tests show excellent visual quality factors due to the use of our smooth block edge detection. About 30% of the images could withstand very high compression of 90%, and 90% of the images could withstand 50% compression with the appropriate watermarking parameters.

The StirMark attack was very successful if we used the original image and the stirmarked watermarked image. No fingerprint was retrieved correctly. We could improve the retrieval by performing also the StirMark attack with the same parameter on the original image. In this extended retrieval we used the stirmarked original image and the stirmarked watermarked image. 70% of the images could be checked successful and the fingerprinting information was retrieved correctly. The remaining 30% error rate are caused by the scratching and shifting operations of StirMark where our watermarking positions could not be found correctly.

The next table contains the error rates after comparing attack of different copies to remove the fingerprinting information.

Out of a great variety of attacks, we have chosen the most significant one:

- Comparison of copies from 2 customer and selecting the average of the difference values
- Comparison of copies from 2 customer and replacing differences with surrounding colours
- Comparison of \( n \) copies and selecting the most frequently used position differences for all \( n \) image copies
- Comparison of \( n \) copies and selecting the less frequently used position differences for all \( n \) image copies

The attack where the difference blocks are replaced by the average value of the images produces partly additional occurrences of 1 instead of 0 in the watermarking retrieval process. The current implementation of the fingerprinting evaluation tool supports only the strategy of maximal removal of the fingerprinting information. Therefore the creation of additional ones are not supported yet and the tool can not detect the concerned customers correctly.
The attack of replacing surrounding image part with the difference blocks can be detected correctly with the fingerprinting tool. The original image is nearly reconstructed and produces 0 instead of 1 simulating maximal removal of the fingerprinting information.

<table>
<thead>
<tr>
<th>Two customer attack</th>
<th>Two and more customers (maximal d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average calculation</td>
<td>Replacing by surroundings (no differences)</td>
</tr>
<tr>
<td>Correct customer recognition</td>
<td>70%</td>
</tr>
</tbody>
</table>

Table 3 Fingerprint attacks test results

5. OPEN PROBLEMS

The watermarking algorithm can handle geometric distortions if the user enter the performed distortions (scaling, rotation, cropping) in the retrieval process. The watermarking algorithm evaluates the distortion parameter for example scale factors, cropping region or rotation degrees. If we have combined all transformations we have still problems with robustness. The retrieval after the StirMark attack can be increased by using the detected difference blocks in the difference image between the stirmarked original and stirmarked watermarked image instead of using only the pseudo random sequence as correct marking positions. If we have an additional orientation for our watermarking blocks we could find the shifted watermarked blocks easier and improve the error rates of 70% easily.

Our prototype implementation of the fingerprinting algorithm can only identify the customers if the strategy of maximal removal of the detected marking point differences has been used. All other kind of attacks, like partly modifications of differences are not supported today in our implementation but will be addressed in further work.

6. CONCLUSION

In our paper we described a fingerprinting mechanism and an appropriate watermarking scheme to embed customer information into images to trace illegal copies. Current digital fingerprinting has the disadvantage, that customers could work together as attackers to destroy the watermarking information by comparing their image data and remove the differences. With our proposed technique attackers could still work together, but our mechanisms provide the possibility to conclude to the customers which attacked the watermark by comparing differences.

Our robustness tests of the watermarking scheme based on DCT coefficients are mainly based on compression, format conversions and geometrical transformations. Format conversions are handled with very low error rates. Especially StirMark attacks could be handled more efficiently by performing the StirMark on the original image too to find the correct marking points back. The fingerprinting tests, comparing different image copies, are successful and the remaining intersection gives the possibility to trace the attackers. Altogether our tests show that the watermarking technology with special marking points are satisfying to de-motivate the illegal use of the copied data. Future work will address further security problems if the customer use blind mixing methods instead of removing recognized differences.

ACKNOWLEDGMENTS

The research work described in this paper was performed jointly with Eva Saar and Jörg Schwenk from Telekorn and we would like to thank them for many stimulating discussions and support.

REFERENCES