A BENCHMARKING PROTOCOL FOR WATERMARKING METHODS

V.Solachidis, A.Tefas, N.Nikolaidis, S.Tsekeridou, A.Nikolaidis and I.Pitas

Department of Informatics, University of Thessaloniki,
Thessaloniki 54006, Greece, e-mail: pitas@zeus.csd.auth.gr

ABSTRACT

In this paper a benchmarking system for watermarking algorithms is described. The proposed benchmarking system can be used to evaluate the performance of watermarking methods used for copyright protection, authentication, fingerprinting, etc. Although the system described in the paper is used for image watermarking, the general framework can be used, by introducing a different set of attacks, for benchmarking of video and audio data.

1. INTRODUCTION

Watermarking research spinned off rapidly in the last years. Many methods have been presented in the literature and several watermarking software packages have been developed [1], thus creating the need for evaluation of the relative performance of these methods against various types of attacks.

A widely used, well known image watermarking benchmark is Stirmark [2, 3]. In Stirmark a set of attacks is applied on a set of watermarked images. The watermarking method under test tries then to detect the watermark and decode the encrypted message if any. If the watermark is detected or the message is fully decoded a value 1 is assigned to the method under test. Otherwise if the watermark is not detected or the message is not fully decoded the score 0 is assigned to the method. The percentage of the correctly detected watermarks is the performance score that is used in order to compare the relative performance of watermarking methods.

Although Stirmark succeeds in revealing certain weaknesses of image watermarking methods, mainly against geometric distortions, it has many drawbacks that do not allow for a solid and thorough performance evaluation of a watermarking method. Its main disadvantage is that it does not take into account the method's false alarm probability (probability to detect a watermark in a non watermarked image). Thus, two methods that have the same false rejection probability, but different false alarm probabilities will score the same marks. Stirmark does not evaluate separately the detector and decoder performance. Moreover, it considers a falsely decoded message as a falsely detected watermark which is not correct. Furthermore, Stirmark uses a single key (watermark) for evaluating the performance of the watermarking method against a specific attack. However, since detection results are key-dependent, a large number of keys should be used in order to obtain an accurate characterization of the detector's and decoder's performance. Another drawback of Stirmark is that the embedding and detection time are not evaluated. Finally, in Stirmark the performance scores obtained for the various attacks are combined with equal weights i.e., Stirmark presumes that all attacks and images have the same probability of occurrence in a real world scenario. In many cases this is not correct since for a certain scenario, some attacks (e.g., compression) might be more frequent than others (e.g., column and line removal).

Apart from the detector/decoder performance two other characteristics of a watermarking system are also of great importance; the payload and the breakdown limit for a certain family of attacks. Payload is the maximum number of bits that can be encrypted in a certain amount of data (e.g., 1Mb) and reliably decoded (decoded with a BER lower than a certain threshold). The breakdown limit of a method against an attack family (e.g., JPEG compression) is the most severe attack that the watermarking method can withstand i.e., the watermark/message can be reliably detected/decoded when the watermarked image has been distorted using this attack (e.g., JPEG compression of quality 10).

The above discussion makes clear that a new benchmarking protocol that will overcome Stirmark drawbacks needs to be devised. Such a benchmark is proposed in this paper. Its main aim is to present a complete set of tests that need to be performed in order to obtain a reliable performance characterization of a watermarking algorithm. A methodology for combining the partial results in order to give a more compact representation of the overall method's performance is also proposed. A complete set of performance indices/plots characterizing the robustness, payload, execution time and the breakdown limits of the watermarking method under test are generated.

2. BENCHMARKING PROTOCOL PARAMETERS

The input of the proposed benchmarking system is the watermark embedding and the detection-decoding software and its outputs are performance indices and plots that illustrate the performance of the watermarking system under test against various attacks. The benchmarking system's parameters are

- a set of images $\mathbf{I} = \{I_i/i = 1...N_I\}$
- a set of keys $\mathbf{W} = \{K_i/i = 1...N_W\}$
- a set of messages $\mathbf{M} = \{M_i/i = 1...N_M\}$
- a set of attacks $\mathbf{A} = \{A_i/i = 1...N_A\}$
- a set of weights $\mathbf{C} = \{C_i/i = 1...N_C\}$
- a set of quality specifications $\mathbf{Q} = \{Q_i/i = 1...N_Q\}$

Image data set: The images that should be used in the benchmarking system should vary in size and frequency content, since these two factors affect the watermarking system performance. Moreover, the types of the images in the image data set (indoor/outdoor scenes, graphic images etc.) should be consistent with those that can be met in a real world application scenario.

This work has been supported by the European Projects LTR-ESPRIT 31103 INSPECT and IST-1999-10987 CERTIMARK

Keys and messages data set: The number of keys used in the benchmark is a very important issue. The reason is that in several watermarking methods the algorithm's performance depends on the watermark key. The number of messages is not as important as the number of keys, since the decoder performance is mostly affected by the accurate watermark detection and not by the embedded message.

Set of image quality specifications: In order to rate the watermark visibility and the perceptual quality of the watermarked image an objective quality measure should be used. The *peak signal to noise ratio* (PSNR) of the watermarked image, has been adopted in the benchmark. The reason for choosing PSNR despite its well-known disadvantages (poor correlation with the perceptual image quality) is that no other globally applicable and acceptable quality measure has been proposed so far. A set of PSNR levels can be used for watermark embedding (e.g., 30dB for heavy, 35dB for medium and 40dB for weak embedding). Experiments should be conducted for all quality levels in this set.

Set of attacks: The set of attacks should include all attacks that the average user or an intelligent pirate can use in order to make the watermark undetectable. It should also include all distortions caused by normal image usage, transmission, storage etc. In this paper the term attack is not used to denote a general attack category e.g. cropping, but a specific attack with certain parameters, e.g. cropping by 25%. The aim of the attacks is to decrease the watermark detectability.

Set of weights: The set of weights is used in order to perform a weighted combination of the performance indices/curves that resulted from a certain quality-image-attack combination thus obtaining the overall system performance. Weights should reflect the probability of occurrence of a quality specification, image or attack in a certain scenario.

3. BENCHMARKING PROTOCOL MODULES

The benchmarking system comprises the watermark embedding module, the attack module, the watermark detection-message decoding module and the performance evaluation module.

Embedding module: The watermark embedding module utilizes the embedding software and takes as inputs the image, quality, key and message sets. The embedding function embeds a watermark W_j and a message M_k to an image I_i. The watermarked image should satisfy the quality specification Q_l, e.g., it should have a prespecified PSNR. The above procedure is repeated for all elements of the sets **I**, **W**, **M** and **Q** and a set of watermarked images is generated. The number of elements (cardinality) of $\mathbf{I}^w$ equals $N_W \times N_I \times N_Q \times N_M$. The time needed for watermark and message embedding in each image is evaluated.

Attack module: The attack module is used to distort the watermarked images that have been generated in the watermark embedding stage. All watermarked images of $\mathbf{I}^w$ are distorted using all the attacks of the set $\mathbf{A}$ and the set $\mathbf{I}^a$ of the attacked images is generated. $\mathbf{I}^a$ is comprised of $N_W \times N_I \times N_Q \times N_M \times N_A$ elements.

Detection/decoding module: The watermark detection-decoding module utilizes the detection-decoding software under test and takes as input the set of attacked images $\mathbf{I}^a$ and the set of watermark keys **W**. It performs watermark detection and message decoding to all the images of $\mathbf{I}^a$ for the correct watermark as well as for an erroneous watermark. First, the detection algorithm detects the watermark W_i that has been indeed embedded in the image I_i^a in the embedding procedure and the message $\widehat{M}$ is decoded. Afterwards, the same procedure is repeated for an erroneous watermark W_j, $(i \neq j)$. Thus, for each attacked image two pairs of detector and decoder outputs are extracted (for the correct and for the erroneous watermark respectively). Let $\mathbf{D}^c$ and $\mathbf{B}^c$ be the sets of the detector and decoder outputs, respectively, for detection with the correct key and $\mathbf{D}^e$ and $\mathbf{B}^e$ the corresponding sets for detection using the erroneous key. During the watermark detection-decoding procedure the execution times are also measured and stored.

The detector output is a number that determines whether a given watermark exists in the image under test. Detector output can be either a binary value (1:watermark detected, 0:watermark not detected) resulting by comparing the test statistic of the corresponding hypothesis test against the decision threshold or the test statistic itself i.e., a real (not binary) number. The decoder output is an estimate of the embedded message. It is evaluated only if the detector results in a positive detection.

Performance evaluation module: The performance evaluation module is used in order to extract the performance scores/plots of the watermarking algorithm under test. It takes as inputs the sets of the detector and decoder outputs $\mathbf{D}^c$, $\mathbf{D}^e$ and $\mathbf{B}^c$, $\mathbf{B}^e$, respectively, the set containing the execution times of each operation and the set of weights $\mathbf{C}$. The outputs of the performance evaluation module are performance scores/plots denoting the relative performance of the algorithm under test or its suitability for a certain application scenario.

4. PERFORMANCE MEASURES AND EVALUATION

After the embedding-attack-detection procedure, the detector and decoder (for correct and erroneous watermarks) outputs and the execution times for embedding, detection and decoding are collected. Using these primary results the method performance against the attacks and the execution time of the method should be evaluated.

4.1. Robustness against attacks

In order to evaluate the robustness of a watermarking method against attacks, the false alarm and the false rejection probabilities of the watermarking method after the attack should be estimated. False alarm probability (P_{fa}) is the probability to detect a watermark in an image that is not watermarked or is watermarked by another key. False rejection probability (P_{fr}) is the probability of not detecting a watermark in an image that is indeed watermarked. In our case P_{fa} has been evaluated by performing detection with an erroneous key because this is equivalent to the worst case scenario.

Non binary detector output: If the detector output is not binary then its empirical distribution can be evaluated. The empirical distribution can be approximated by a theoretical one $f(x)$. Thus, from the sets $\mathbf{D}^c$ and $\mathbf{D}^e$ two distributions, $f_c(x)$ and $f_e(x)$ result. Let T_1 and T_2 be the minimum and the maximum mean value of all these distributions.

Then for each threshold T between T_1 and T_2, P_{fa} and P_{fr} can be calculated.

$$P_{fa}(T) = \int_T^{\infty} f_e(x)dx \ , \qquad P_{fr}(T) = \int_{-\infty}^{T} f_c(x)dx$$

Using P_{fa}, P_{fr} we can evaluate the *Receiver Operating Characteristic* (ROC) i.e. the plot of the probability of false alarm P_{fa} vs

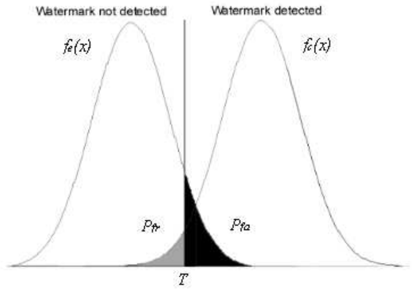

Figure 1: False alarm and false rejection probabilities

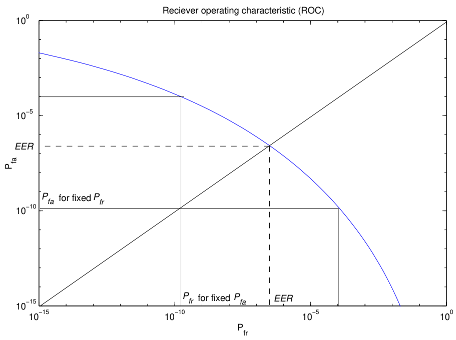

Figure 2: Detector performance measures

probability of false rejection P_{fr}. This can be done by evaluating for each threshold value the area of $f_c(x)$ left of threshold (=P_{fr}) and the area of $f_e(x)$ right of threshold (=P_{fa}) as it is illustrated in Figure 1. ROC curves are extracted for each quality-image-attack combination.

The ROC curves can be also extracted directly from the empirical distributions.

$$P_{fa}(T) = \frac{|\mathbf{D^e_T}|}{|\mathbf{D^e}|}, \text{where } \mathbf{D^e_T} = \{x_i > T/x_i \in \mathbf{D^e}\}$$

$$P_{fr}(T) = \frac{|\mathbf{D^c_T}|}{|\mathbf{D^c}|}, \text{where } \mathbf{D^c_T} = \{x_i < T/x_i \in \mathbf{D^c}\}$$

where $|\mathbf{A}|$ denotes the number of the elements (cardinality) of the set $\mathbf{A}$. However, in order to have accuracy of order 10^{-N} at least 10^N keys have to be used.

The ROC curve has been chosen as a robustness measure because it is the most complete way to describe an algorithm performance with respect to robustness. Having evaluated the ROC one can also evaluate the following performance measures:

- P_{fa} for fixed P_{fr}.
- P_{fr} for fixed P_{fa}.
- Equal error rate (EER): Point on the ROC where P_{fa}=P_{fr}.
- Area A_r below ROC (global performance measure). if A_r =0 the system is perfect.

Any of these measures can be used for evaluating the relative performance of two algorithms or checking the appropriateness of an algorithm for a certain application scenario.

Binary detector output: If the detector output is binary, then ROC curves can not be extracted. The only information at hand is the number N_{fa} of the erroneously detected watermarks and the number N_{fr} of the missed watermarks. Using these numbers a single P_{fa}, P_{fr} can be evaluated. $P_{fa} = \frac{N_{fa}}{|W|}$ and $P_{fr} = \frac{N_{fr}}{|\mathbf{W}|}$, where $|\mathbf{W}|$ is the number of elements in $\mathbf{W}$. As a performance measure the weighted sum $P_{er} = p_1 P_{fa} + p_2 P_{fr}$ can be used. The constants p_1, p_2 are appropriately selected to represent the relative importance of P_{fa}, P_{fr} in a certain application scenario.

Decoder output: If the watermarking method supports message encryption, a message M_i is embedded in every image in addition to the watermark. The message length varies. After the decoding procedure the message M_i that has been embedded is compared with the decoded message $\widehat{M}_i$. This procedure is performed for all the images of $\mathbf{I}_a$ regardless of the watermark that has been used for detection (correct or erroneous).

Afterwards, in case of a not binary detector output we evaluate (for each threshold T):

- The mean number of erroneously decoded bits (BER), for all watermarks (either correct or erroneous) that have resulted in a detector output greater than T i.e., for all detected watermarks.
- The number E of messages for which all message bits have been correctly decoded, for all watermarks (either correct or erroneous) that have resulted in a detector output greater than T i.e., for all detected watermarks.

As a result of the above procedure the plots of BER and E vs threshold T (or equivalently vs the corresponding P_{fa}) are generated for each quality-image-attack combination. The BER for fixed P_{fa} or E for fixed P_{fa} can be used as performance measures.

In case of binary detector output we have a prespecified threshold, i.e. a prespecified operating state in the ROC. As a result, a single BER and E are evaluated for each quality-image-attack combination and they are used as performance measures.

4.2. Execution time

The execution time for watermark embedding and detection is measured for each quality-image-attack-watermark combination. The average embedding and detection/decoding time is evaluated for each quality-image-attack combination by averaging the execution time for all different watermarks and messages.

4.3. Combination of the results

Following the above procedures we obtain performance measures for each quality-image-attack combination concerning the detector output (binary or not), the decoder output and the execution time. However since this would result in an enormous amount of information, results should be combined to obtain a more compact representation. This can be done in a progressive way that leads to a continuously increasing degree of information compaction. Results can be combined by weighting the performance measure of each quality-image-attack combination with weights that correspond to the probability of occurrence of this specific combination in a certain application scenario. An example of choosing the weights in the set $\mathbf{C}$ could be the following; The weights for averaging the results with respect to the various images can be set equal to $\frac{1}{N_I}$, if the assumption that all images used in the benchmarking have equal probability of occurrence in a certain application scenario is adopted. Omitting certain images from the benchmark is equivalent to setting the corresponding weights equal to zero. On the other hand, the weights for combining the results for all attacks within an attack family can be chosen according to the probability of occurrence of this attack, e.g. cropping 25% weight 0.40 (probability of occurrence of such an attack equal to 0.4, cropping 35% weight 0.35, cropping 45% weight 0.25). As an example, the case of combining the ROCs is described below in more detail.

It is easy to verify that combining two or more ROCs is equivalent to obtaining their weighted average value at each threshold. Thus:

- Level 1:ROCs for all images and a certain Attack-Quality pair of parameters can be combined together to obtain a total measure for this pair (e.g. a single ROC for all watermarked images that exhibit a quality of 35dB SNR and have been distorted by cropping 25%)
- Level 2:ROCs for all images and a certain Attack Family-Quality pair can be combined together to obtain a total measure for this pair (e.g. a single ROC for all watermarked images that exhibit a quality of 35dB SNR and have been distorted by cropping (using all cropping parameters e.g. $25\% - 35\% - 45\%$ etc).
- Level 3:ROCs for all images and all Attacks for a certain Quality can be combined together to obtain a total measure for this Quality (e.g. a single ROC for all watermarked images that exhibit a quality of 35dB SNR and have been distorted by all available attacks).

Thus in the last stage we will come up with a single ROC per algorithm per quality specification. The same methodology can be used in order to combine the P_{fa}, P_{fr} in case of a binary detector output. In the case of the message decoder using the weights for each quality-image-attack combination we can obtain the total BER for the watermarking system and the percentage of the correctly decoded messages. Execution time is combined similarly giving the average embedding and detection/decoding time for the watermarking system.

4.4. Evaluation of the breakdown limits of the algorithm

For a chosen performance measure and certain quality, image and attack family the breakdown limit of the algorithm for this attack family can be evaluated. The attack severity is increased in appropriately selected steps (e.g. decrease the JPEG quality factor in steps of 10) until the detector output does not satisfy the chosen performance criterion. The last (strongest) attack for which the algorithm performance is above the selected threshold is the algorithm limit for the selected attack family.

As a result of the above procedure the breakdown limits for each quality, attack family, image combination are evaluated. The breakdown limits for all images and a certain quality-attack family pair can be combined together to obtain a total breakdown limit for this pair. The combination procedure described previously can be also used in this case.

4.5. Payload evaluation

Payload is the largest number of message bits that can be embedded in a fixed amount of data (e.g.1 Mb) and decoded with a certain BER. It is obvious that payload depends on the image characteristics and the Quality $\mathbf{Q}$. In order to estimate the payload of a watermarking method, messages of increasing length are embedded in the host images, until the BER that results during the decoding phase becomes bigger than the prespecified threshold T_{BER}. In case of a not binary detector, BER is evaluated for each value of the sliding threshold T (test statistic), i.e., for each P_{fa} value.

As a result of the above procedure the payload of the method for each threshold i.e., for each operating state, is obtained for each quality, image combination. Since a threshold T corresponds to a certain P_{fa}, P_{fr} pair, plots of payload vs P_{fa} or payload vs P_{fr} can be generated. A possible performance measure for comparing two methods is to compare their payloads for fixed P_{fa}. In case of a binary detector output a single payload is evaluated for the prespecified operating state of the watermarking system. The partial performances for each quality-image pair can be combined, as described above, to give an overall payload for the watermarking algorithm.

4.6. Comparing two methods

In order to compare two methods the performance measures that will be used should be chosen among the ones presented above (P_{fa} vs P_{fr}, or BER, etc). This comparison can be done either for a certain attack family in order to illustrate the superiority of one method against this attack or for the entire set of attacks. For example, if P_{fr} for fixed P_{fa} is chosen to be the performance measure, then the relative performance of two watermarking methods can be evaluated by comparing their overall P_{fr} for fixed P_{fa}.

4.7. Checking a method against requirements of a specific application

It is also possible to check if a watermarking method fulfils the specifications of a certain application. An application scenario corresponds to a specific set of weights $\mathbf{C}$ and a set of performance thresholds. The performance measures obtained by the benchmarking system for the algorithm under test (using weights $\mathbf{C}$) are compared against the scenario thresholds.

5. CONCLUSIONS

A novel benchmarking system for watermarking algorithms that overcomes the drawbacks of Stirmark has been proposed. The benchmarking system can be used to evaluate the performance of watermarking methods used for copyright protection, authentication, fingerprinting, etc. The software can be downloaded from: **http://poseidon.csd.auth.gr/LAB_RESEARCH/watermarking/Benchmarking/index.html**.

6. REFERENCES

[1] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1079–1107, July 1999.

[2] F.A.P. Petitcolas, R.J. Anderson, and M.G. Kuhn, "Attacks on copyright marking systems," in *Second International Workshop on Information Hiding*, Portland, USA, April 15-17 1998, pp. 219–239.

[3] F.A.P. Petitcolas and R.J. Anderson, "Evaluation of copyright marking systems," in *International Conference on Multimedia Systems*, Florence, Italy, June 7-11 1999.