# Multimedia Data Security - Competition Rules

## What is the competition about?

The Multimedia Data Security competition is essentially a group activity which allows to apply already learned concepts in a fun way. The objectives are two:

- Applying an **embedding** strategy that is both **robust and unperceivable**
- **Attacking** the watermarked information of other groups while **preserving image quality**

NOTE: In the following document, each part is discussed thoroughly. If you still have doubts, the teaching assistants are more than happy to answer your questions, so make sure everything is as clear as possible!

## Scheduling

Deadlines:

| | |
|---|---|
| **Friday, 08th October at midnight** | **GROUP FORMATION** |
| **Sunday, 24th October at midnight** | **RUNNING CODE SUBMISSION** |

Events:

| | |
|---|---|
| **Wednesday, 27th October from 8.30 to 12** | **COMPETITION** |
| **10st November from 8.30 to 11.30** | **GROUPS PRESENTATIONS** |

Specific labs will be devoted to code development.

## Group Formation

All participating students need to be divided in groups of **3/4 people**. Each group must select

- a **spokesperson**, that will take care of the communication with the teaching assistant
- a **nickname** that only your group will know. Group nicknames must not contain spaces/symbols/capital letters.

Once everything is ready, the first job of the spokesperson is to send an e-mail to the teaching assistants with the name of the group and the name, surname and e-mail (preferably academic address) of each member of the team. This must be done by the following **deadline:** **8th October at midnight** .

After the deadline for the registration, you'll receive an e-mail providing:

- A randomly generated **watermark** which will be strictly associated to your group. All information about the watermark is given in the *Additional Information* section at the end of this document.
- A randomly generated **password** that you will need to access the website of the competition (so keep it safe and secret).

## Running Code Submission

Each group should work on their strategy. You can work also before the watermark is given to you: just create a dummy watermark (you can see the structure in the *Additional Information* section) to test your code with.

By **24th October at midnight** you will be asked to deliver a working code. This is just to avoid you having problems the day of the competition, and have the backup of a working solution. You are asked to deliver also the ROC-curve code so as we can check how you set the threshold. You can still change and fine tune your solution after this deadline.

For more details see the section “How to prepare your code”.

## Competition

The competition will take place on **27th October**, please **be ready to start at 8.30**.

The website will be your main tool to upload and download the materials. The competition is divided in three phases: defense phase, attack phase and a verification phase. Each phase is strictly timed, so be ready!

**Important note**: For the competition you are allowed to used as many computers as you like (parallelization of work is advised). However, due to the limited networking resources, you are asked to **have only one device connected on the website at a time** to perform all the necessary uploads/downloads. Also beware that if all of you try to access the website the last minutes of the competition, you might experience problems in uploading content.

### Defense phase → 8.30 - 9.30

During the defense phase, you will have to

1. Download the competition images (three 512 × 512 grayscale images)
2. Download the WPSNR code (if needed, as it will be the same code provided during laboratories)
3. Use your embedding strategy to insert your watermark in each of the three images
4. Upload the embedded images in the website. You can do this multiple times, each time you do it a score will be assigned to you based on the quality of the embedding (see Scoring Information for more details).
5. Upload the code of your detection strategy (other groups will use it during the attack phase)

During this phase you can tune some parameters of your embedding/detection strategy, but make sure that you correct everything before uploading the materials on the competition website. **Remember to check if your detection code works properly!**

### Attack phase → 9.30 - 12

The second phase of the competition consists in targeting the other groups and attack their images following these steps:

1. download other groups images from the list that will appear on the website

2. perform some image processing attack(s) on the images watermarked by other groups to remove the watermark while keeping image quality as high as possible.
   **Important note:** the list of attacks that you are free to use, both singularly or combining them, is reported in the *Additional Information* section. **Note:** no external software (e.g.: Photoshop, Paint, GIMP, ... ) is allowed.

3. upload the attacked images on the website, indicating which group was attacked. If you perform multiple successful attacks on a group's images, just upload the case in which you have the highest WPSNR!

## Verification phase

After the competition you are required to send a log of your attacks (you can use excel or a csv). The required structure is the following:

| **Image** | **Group** | **WPSNR** | **Attack(s) with parameters** |
|---|---|---|---|
| lena | attackedGroup1 | 37 | JPEG QF=90, Median 5×5 |
| ... | ... | ... | ... |

We will use this log file to check your results. Please, be aware that cheating will be penalized. We will also check any other problem that might arise during the competition and update scores accordingly.

## Presentation

Each group will have to present its approach on 10$^{rd}$ **November** as part of the exam: each group member has to be present, take part to the presentation and answer to possible questions. Once we will be aware about the number of groups participating we will provide more details about the timing of the presentation. The presentation should include the following elements:

- defense strategy
- attack strategy
- results

## How to prepare your code

It is very important that your code is as correct as possible and strictly follows the rules here presented: penalties will be given to those not following the rules. [See the *Additional Information* section for more details]
So make sure everything of this section is clear to you!

Both the embedding and the detection codes must be sent to the teaching assistant for a final check and evaluation before the following **24$^{th}$ October at midnight**. Only minor changes to the code will be allowed after that date. If during the final check the assistant find errors or uncertain steps, an e-mail will be sent to the corresponding group, asking for clarifications and corrections, to avoid mistakes and cheating during the day of the competition.

## Embedding

Make sure to use `opencv` library to read, write and generally handle images. All the images will be in BMP format. During the competition, you will be provided with three grayscale images of fixed dimensions 512x512. As explained later, points are assigned on basis of the **WPSNR value** resulting from a comparison between the original image and its corresponding watermarked version: the higher the WPSNR, the more the points. [See the *Additional Information* section for more details]

## Threshold computation

Once you gathered what should be the embedded watermark, the next step is to understand if the extracted data corresponds to the original watermark or not. For this competition, **a similarity evaluation** must be performed. For this purpose, the detection strategy is prepared in two phases: **threshold computation** and **comparison**. [See detection section for the comparison]

The threshold computation must be performed **once** and not in the actual detection code that you will deliver on the competition day. Since it is unrealistic to have image-specific thresholds **only one threshold** value must be selected and used **for all the images**. The code to compute the threshold **should not be included in the detection algorithm**.
Once you completed your watermark extraction strategy, the similarity threshold **have to** be estimate by modifying the code of Lab3, topic ROC-curves example 2, as follows:

1. Embed your watermark $W_{original}$ to a set of images
2. In a loop, attack one by one these images (with random attacks or the strategy you prefer)
3. Extract the watermark with your planned technique $W_{extracted}$
4. Compute `sim`$(W_{original}, W_{extracted})$ and append it in the *scores* array and the value 1 in the *labels* array. These values will correspond to the true positive hypothesis.
5. Generate a random watermark $W_{random}$ and compute `sim`$(W_{random}, W_{extracted})$ to append it in the *scores* array and the value 0 in the *labels* array. These values will correspond to the true negative hypothesis.
6. with *scores* and *labels*, generate the ROC and choose the best threshold $\tau$ corresponding to a False Positive Rate $FPR \in [0, 0.1]$.

Take note of the $\tau$ value: it will be used to assess if the data extracted with your strategy from a filtered/attacked image still corresponds to the original watermark. **This code has to be send within the 24$^{th}$ of October at midnight**

## Detection

The detection of a watermark is strictly related to your chosen embedding strategy. For the purposes of our competition, the watermark detection follows a **non-blind** strategy.

The detection function must be **a single file** named `detection_groupname.py`, no external functions are allowed (except those of the WPSNR).
The function will make use of the **WPSNR** code seen during the laboratory sessions.

You need to make sure that **input and output values are correct** and follow this structure:

```
def detection(input1, input2, input3):
    '''
    YOUR CODE
    '''
    return output1, output2
```

- `input1` corresponds to the **string** of the name of the **original** image
- `input2` corresponds to the **string** of the name of the **watermarked** image
- `input3` corresponds to the **string** of the name of the **attacked** image
- `output1`: if the attacked image contains the watermark it is equal to **1** , otherwise it is equal to **0**
- `output2` corresponds to the **WPSNR** value between the **watermarked** and the **attacked** image.

To assess whether an attack is successful or not, you you must compare the watermarked extracted from the watermarked image (i.e.: $W_{extracted}$ ) and one extracted from the attacked image (i.e.: $W_{attacked}$). Therefore, the detection code should not rely on the watermark file nor have the watermark hard-coded. If the similarity beween the two values (i.e. `sim`($W_{extracted}$ $W_{attacked}$), **is equal or above** the previously calculated threshold $T$, then the watermark is assumed to be present and the attack is considered failed. An **attack is considered successful** if

- the similarity is below the threshold $T$ (i.e., `output1` $= 0$)
- the WPSNR $\geq 35$ [dB] (i.e., `output2` $\geq 35$).

Summing up:

- Read all the images inside the function using the three input!
- Don't read the original watermark: extract it!
- Attacks are considered successful only if the watermark is destroyed and WPSNR $\geq 35$ [dB]
- Attacks are considered failed if the watermark is present or, if destroyed, with WPSNR $< 35$ [dB]

**Important note:** Make sure that if you provide the original image also as attacked input to your detection function, the watermark is not present.

**Final remark:** your code must complete the detection within 5 seconds and must not open any pop-up windows or print anything on screen.

## Attack

During the attacking phase, you are asked to work using image processing techniques.
You are only allowed to use a limited list of attacks, that you can tune and combine to destroy other groups watermarks. **Permitted attacks** for the competition are:

- AWGN
- Blurring
- Sharpening
- JPEG Compression
- Resizing
- Median filtering

## Additional Information

### Watermark

Each group will be assigned a specific watermark containing the assigned mark (a 1024 numpy array of zeros and ones), that will be used on the day of the competition.

### Naming convention

Assuming that `groupA` is your group name, the embedded images must be named

`imageName_groupA.bmp`

Assuming that `groupB` is the group that you want to attack, images watermarked by that group downloaded from the website will be named

`groupB_imageName.bmp`

Assuming that `groupB` is the group that you want to attack and you are `groupA`, images attacked by your group must be named

`groupA_groupB_imageName.bmp`

### Scoring Information

Scores will be assigned to each group's performance during the competition according to the following tables:

- **EMBEDDING QUALITY**: aim at a higher quality of the watermarked image for more points.

| WPSNR | POINTS |
|---|---|
| **35 ≤ WPSNR < 50** | 1 |
| **50 ≤ WPSNR < 54** | 2 |
| **54 ≤ WPSNR < 58** | 3 |
| **58 ≤ WPSNR < 62** | 4 |
| **62 ≤ WPSNR < 66** | 5 |
| **WPSNR ≥ 66** | 6 |

- **ROBUSTNESS**: average WPSNR of the group's watermarked images successfully attacked by other groups

| WPSNR | POINTS |
|---|---|
| **35 ≤ WPSNR < 38** | 6 |
| **38 ≤ WPSNR < 41** | 5 |
| **41 ≤ WPSNR < 44** | 4 |
| **44 ≤ WPSNR < 47** | 3 |
| **47 ≤ WPSNR < 50** | 2 |
| **50 ≤ WPSNR < 53** | 1 |
| **WPSNR ≥ 53** | 0 |

- **ACTIVITY:** percentage of groups attacked

| % OF GROUPS ATTACKED | POINTS |
|---|---|
| $> 30\%$ | 2 |
| $> 60\%$ | 4 |
| $> 90\%$ | 6 |

- **QUALITY:** number of images that you attacked with a WPSNR higher that the average WPSNR of the other attacks to that group's images (e.g. if a group has robustness 46 and you attack an image of that group achieving a WPSNR=47, you'll score a point).

| ATTACKED IMAGES WITH WPSNR >ROBUSTNESS | POINTS |
|---|---|
| **1-5** | 1 |
| **6-10** | 2 |
| **11-15** | 3 |
| **> 15** | 4 |

- **BONUS:** you will be awarded 2 extra points
  - if you successfully attacked a group that no one else attacked
  - if you were not attacked by anyone (assuming that your code is working)

  This bonuses will be removed during the verification phase if they are caused by errors in the code or images.

## Penalties

A 2-point penalty in the competition results will be applied for each transgression of the discussed rules, including the following.

- You miss a deadline.
- You have to change one of the files uploaded to the website after the conclusion of the defense phase.
- The detection function successfully finds a watermark in the original image, in several unrelated images (e.g., images watermarked by other groups) or completely destroyed images (WPSNR $\leq$ 25db)
- Your detection code opens pop-up windows or prints on screen.
- Your detection code takes more than 5 seconds to run.