PROTECTING JPEG IMAGES USING VARIABLE THRESHOLD SPLITTING

A Degree Thesis
Submitted to the Faculty of the
Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona
Universitat Politècnica de Catalunya
by
Pau Ballart Godoy

In partial fulfilment
of the requirements for the degree in
AUDIOVISUAL SYSTEMS ENGINEERING

Advisor: Antonio Ortega, Josep Pegueroles

Barcelona, May 2016
Abstract

This thesis describes an improvement of an existing algorithm for privacy preserving in image sharing (P3-Privacy Preserving Photo Sharing). In short, the existing algorithm uses a fixed threshold whereas in this thesis a variable threshold approach is studied.

First, a Region of Interest (ROI) approach is studied, where a given image region is encrypted using a more secure threshold. The second part of the project focuses on analyzing the problem at a DCT block level and deciding for each block which threshold is used.

Some results are presented for both the ROI and the block-based variable threshold cases. For the latter, an optimal curve of points is calculated. This set of points gives the best encryption (fewer edges detected) generating the smallest file size (less overhead).
Resum

Aquesta tesi descriu una millora d'un algorisme existent que tracta de preservar la privacitat quan es comparteixen imatges (P3-Privacy Preserving Photo Sharing). Aquest algorisme utilitza un valor llindar fixe mentre que en aquesta tesi s’estudia una versió on aquest llindar és variable a diferents parts de la imatge.

Primer s’estudia utilitzar diferent llindar en funció d’una regió d’interès (ROI). Donada una ROI, aquesta ROI s’encierra utilitzant un llindar més segur que la resta de la imatge. La segona part del projecte se centra en decidir per cada bloc DCT quin llindar s’utilitza.

Es presenten resultats d’imatges utilitzant el sistema de la ROI i el de llindar variable segons el bloc. Per aquest últim, s’ha calculat la corba òptima de punts. Aquest conjunt de punts donen la millor encriptació (menys contorns detectats) mentre que la mida del fitxer no és massa gran.
Resumen

Esta tesis describe una mejora de un algoritmo existente que trata de preservar la privacidad cuando se comparten imágenes (P3-Privacy Preserving Photo Sharing). Este algoritmo utiliza un valor lindar fijo mientras que en esta tesis se estudia una versión donde este lindar varía en diferentes partes de la imagen.

Primero se estudia utilizar diferente lindar en función de una región de interés (ROI). Dada una ROI, esta ROI se encripta utilizando un lindar más seguro que el resto de la imagen. La segunda parte del proyecto se centra en decidir para cada bloque DCT qué lindar utilizar.

Se presentan resultados de imágenes usando el sistema de la ROI y el de lindar variable según el bloque. Para este último, se ha calculado la curva optima de puntos. Tal conjunto de puntos dan la mejor encriptación (menos contornos detectados) manteniendo un tamaño de fichero razonable.
Acknowledgements

I would like to thank professor Antonio Ortega for the continuous support and guidance during the development of this thesis. Without him, this thesis would not have been possible. In the early stages he helped me to find a good idea to focus on and once I got the focus he helped me with the path to follow to achieve what I’ve got today.

I would also like to thank student Guillem Solé for all the times discussing the algorithms and the results.

I need also to thank my advisor Josep Pegueroles for his advices.

Finally, I would like to express thanks to my family for their support and patience.
### Revision history and approval record

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>25/02/2016</td>
<td>Document creation</td>
</tr>
<tr>
<td>1</td>
<td>14/03/2016</td>
<td>Document revision</td>
</tr>
<tr>
<td>2</td>
<td>12/04/2016</td>
<td>Document revision</td>
</tr>
<tr>
<td>3</td>
<td>28/04/2016</td>
<td>Document revision</td>
</tr>
<tr>
<td>4</td>
<td>06/05/2016</td>
<td>Document revision</td>
</tr>
<tr>
<td>5</td>
<td>10/05/2016</td>
<td>Document revision</td>
</tr>
</tbody>
</table>

### DOCUMENT DISTRIBUTION LIST

<table>
<thead>
<tr>
<th>Name</th>
<th>e-mail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pau Ballart</td>
<td><a href="mailto:pballart@gmail.com">pballart@gmail.com</a></td>
</tr>
<tr>
<td>Antonio Ortega</td>
<td><a href="mailto:antonio.ortega@gmail.com">antonio.ortega@gmail.com</a></td>
</tr>
<tr>
<td>Josep Pegueroles</td>
<td><a href="mailto:josep.pegueroles@upc.edu">josep.pegueroles@upc.edu</a></td>
</tr>
</tbody>
</table>

Written by: Pau Ballart
Reviewed and approved by: Antonio Ortega

<table>
<thead>
<tr>
<th>Date</th>
<th>Name</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/05/2016</td>
<td>Pau Ballart</td>
<td>Project Author</td>
</tr>
<tr>
<td>11/05/2016</td>
<td>Antonio Ortega</td>
<td>Project Supervisor</td>
</tr>
</tbody>
</table>
# Table of contents

Abstract ................................................................................................................................. 1  
Resum .................................................................................................................................. 2  
Resumen ............................................................................................................................... 3  
Acknowledgements .............................................................................................................. 4  
Revision history and approval record .................................................................................. 5  
Table of contents .................................................................................................................. 6  
List of Figures ...................................................................................................................... 8  
List of Tables ....................................................................................................................... 10  
1. Introduction .................................................................................................................... 11  
   1.1. Goal .......................................................................................................................... 11  
   1.2. Requirements and Specifications .......................................................................... 11  
   1.3. Work plan ............................................................................................................... 11  
      1.3.1. Work Packages .............................................................................................. 12  
      1.3.2. Time plan ........................................................................................................ 15  
   1.4. Work plan modifications ....................................................................................... 15  
2. Review of State of the art ............................................................................................... 16  
   2.1. JPEG Images .......................................................................................................... 16  
      2.1.1. The JPEG compression process .................................................................... 16  
   2.2. P3 Algorithm ........................................................................................................... 18  
      2.2.1. Introduction .................................................................................................... 18  
      2.2.2. Sender side ..................................................................................................... 19  
      2.2.3. Recipient side ............................................................................................... 20  
      2.2.4. Evaluation ...................................................................................................... 20  
   2.3. Related work ............................................................................................................ 21  
3. Methodology / project development: ............................................................................. 22  
   3.1. Variable threshold depending on the Region of Interest (ROI) ............................ 22  
      3.1.1. ROI definition .............................................................................................. 23  
      3.1.2. System design ............................................................................................... 23  
   3.2. Variable threshold depending on the DCT block .................................................... 24  
      3.2.1. System design .............................................................................................. 24  
      3.2.2. Optimization ................................................................................................. 26  
4. Results .............................................................................................................................. 30
4.1. Variable threshold depending on the Region of Interest (ROI) .................. 30
4.2. Variable threshold adapted to the DCT block contents.......................... 32
5. Budget ........................................................................................................... 34
6. Conclusions and future development: .......................................................... 35
7. Bibliography: ................................................................................................. 36
8. Appendices: ...................................................................................................... 37
  8.1. Auditorium ................................................................................................ 37
  8.2. Beach ......................................................................................................... 39
  8.3. Concert ....................................................................................................... 41
9. Glossary ........................................................................................................... 43
List of Figures

Figure 1: Gantt Diagram ............................................................................................................. 15
Figure 2: DCT formula ..................................................................................................................... 16
Figure 3: Quantization formula ....................................................................................................... 17
Figure 4: Zigzag matrix .................................................................................................................... 17
Figure 5: P3 algorithm scheme ....................................................................................................... 18
Figure 6: Privacy-Preserving Image Encoding Algorithm ................................................................. 19
Figure 7: Lena image ....................................................................................................................... 20
Figure 8: P3 public part of Lena image ............................................................................................ 20
Figure 9: P3 public image with a face .............................................................................................. 22
Figure 10: ROI using face detection ............................................................................................... 23
Figure 11: User-defined ROI ......................................................................................................... 23
Figure 12: Image representing the THS blocks ............................................................................... 25
Figure 13: Optimization results ....................................................................................................... 27
Figure 14: Graphic representation of the optimal points. ............................................................... 28
Figure 15: Overheads generated in different images ...................................................................... 30
Figure 16: Edges detected in different images ............................................................................... 31
Figure 17: Original P3 image (left) and image using ROI with lower threshold (right) .............. 31
Figure 18: Edges detected in the original image (top), the original P3 public image (left) and the public image using ROI (right) ................................................................. 32
Figure 19: Original image ............................................................................................................... 33
Figure 20: Edges of the original image ............................................................................................ 33
Figure 21: Edges of the original P3 algorithm public image ............................................................ 33
Figure 22: Edges of the block-based variable threshold image ....................................................... 33
Figure 23: Auditorium image .......................................................................................................... 37
Figure 24: Auditorium original image edges .................................................................................... 37
Figure 25: Auditorium P3 image edges ............................................................................................ 38
Figure 26: Auditorium variable P3 image edges ............................................................................ 38
Figure 27: Beach image ................................................................................................................... 39
Figure 28: Beach original image edges ............................................................................................ 39
Figure 29: Beach P3 image edges ................................................................................................... 40
Figure 30: Beach variable P3 image edges ..................................................................................... 40
Figure 31: Concert image ............................................................................................................... 41
Figure 32: Concert original image edges ........................................................................................ 41
Figure 33: Concert P3 image edges

Figure 34: Concert variable P3 image edges
List of Tables

Table 1: Table representation of the optimal points ......................................................... 29
Table 2: Project's software budget ....................................................................................... 34
Table 3: Project's engineers budget ..................................................................................... 34
1. **Introduction**

This thesis has been developed at the University of Southern California thanks to the UPC-USC exchange program. It proposes an improvement to an existing thesis called “P3-Privacy Preserving Photo Sharing” [1], developed by Moo-Ryong Ra, a USC PhD student in 2013.

P3 describes a method for encrypting JPEG images such that only people with access to a private key can view the content (explained in more detail in Section 2.2). The improvement this thesis proposes consists of using a variable threshold depending on the photo content instead of one pre-defined threshold for the whole image.

1.1. **Goal**

P3 uses the same threshold for the whole image, but images have different parts with different information. Given that fact, the idea of using different thresholds depending on the part of the image was proposed. The main objective was to improve security of the public part by using different thresholds in different parts of the image.

1.2. **Requirements and Specifications**

Project requirements:

- Maintain at any time a JPEG standard compliant image.
- Be transparent for third party systems.
- Produce images with faces undetectable for both algorithms and humans.
- Produce images with almost no detected edges without requiring an excessive bit rate overhead.

Project specifications:

- The image file format must be JPEG
- The computation time of the whole progress must be less than 3 seconds in any smartphone that can run the algorithm.
- The overhead in size of the resulting image compared to the original image must be less than 20%

1.3. **Work plan**

The initial work plan has not been modified during the development, except for the last WP that will be explained is Section 1.5 of this document.
### 1.3.1. Work Packages

#### Project: Improving P3 using variable thresholds

<table>
<thead>
<tr>
<th>Major constituent: SW</th>
<th>WP ref: WP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sheet 1 of 7</td>
<td></td>
</tr>
</tbody>
</table>

**Short description:**
Understand the existing “P3” algorithm and workflow for the encryption of JPEG images.

<table>
<thead>
<tr>
<th>Planned start date: 01/11/2015</th>
<th>Planned end date: 15/11/2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start event: 01/11/2015</td>
<td>End event: 15/11/2015</td>
</tr>
</tbody>
</table>

**Internal task T1:**
Read the paper.

**Internal task T2:**
Understand the source code.

#### Project: Improving P3 using variable thresholds

<table>
<thead>
<tr>
<th>Major constituent: SW</th>
<th>WP ref: WP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sheet 2 of 7</td>
<td></td>
</tr>
</tbody>
</table>

**Short description:**
Implement an algorithm for detecting Regions of Interest, starting with a face detector.

<table>
<thead>
<tr>
<th>Planned start date: 15/11/2015</th>
<th>Planned end date: 10/12/2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start event: 15/11/2015</td>
<td>End event: 05/12/2015</td>
</tr>
</tbody>
</table>

**Internal task T1:**
Choose a face detector

**Deliverables:**
Proof of concept

**Dates:**
10/12/2015

#### Project: Improving P3 using variable thresholds

<table>
<thead>
<tr>
<th>Major constituent: SW</th>
<th>WP ref: WP3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sheet 3 of 7</td>
<td></td>
</tr>
</tbody>
</table>

**Short description:**
Embed detector ROI information in JPEG metadata.

<table>
<thead>
<tr>
<th>Planned start date: 10/12/2015</th>
<th>Planned end date: 10/01/2016</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project: Improving P3 using variable thresholds</td>
<td>WP ref: WP4</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>Major constituent: SW</td>
<td>Sheet 4 of 7</td>
</tr>
<tr>
<td>Short description:</td>
<td></td>
</tr>
<tr>
<td>Vary encryption threshold in function of the ROI data.</td>
<td>Planned start date: 10/01/2016</td>
</tr>
<tr>
<td></td>
<td>Planned end date: 10/02/2016</td>
</tr>
<tr>
<td></td>
<td>Start event: 28/12/2015</td>
</tr>
<tr>
<td></td>
<td>End event: 20/01/2016</td>
</tr>
<tr>
<td>Internal task T1:</td>
<td>Deliverables:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project: Improving P3 using variable thresholds</th>
<th>WP ref: WP5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Major constituent: SW</td>
<td>Sheet 5 of 7</td>
</tr>
<tr>
<td>Short description:</td>
<td></td>
</tr>
<tr>
<td>Decrypt correctly the photo.</td>
<td>Planned start date: 10/02/2016</td>
</tr>
<tr>
<td></td>
<td>Planned end date: 25/02/2016</td>
</tr>
<tr>
<td></td>
<td>Start event: 21/01/2016</td>
</tr>
<tr>
<td></td>
<td>End event: 13/02/2016</td>
</tr>
<tr>
<td>Internal task T1:</td>
<td>Deliverables:</td>
</tr>
<tr>
<td>Project: Improving P3 using variable thresholds</td>
<td>WP ref: WP6</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Major constituent: SW</td>
<td>Sheet 6 of 7</td>
</tr>
<tr>
<td>Short description:</td>
<td></td>
</tr>
<tr>
<td>Verify the algorithm with metrics and results.</td>
<td></td>
</tr>
<tr>
<td>Planned start date:</td>
<td>14/02/2016</td>
</tr>
<tr>
<td>Planned end date:</td>
<td>07/03/2016</td>
</tr>
<tr>
<td>Start event:</td>
<td></td>
</tr>
<tr>
<td>End event:</td>
<td></td>
</tr>
<tr>
<td>Internal task T1:</td>
<td></td>
</tr>
<tr>
<td>Find the appropriate metric.</td>
<td></td>
</tr>
<tr>
<td>Internal task T2:</td>
<td></td>
</tr>
<tr>
<td>Extract some results with various images datasets.</td>
<td></td>
</tr>
<tr>
<td>Deliverables:</td>
<td></td>
</tr>
<tr>
<td>Results graphics</td>
<td></td>
</tr>
<tr>
<td>Dates:</td>
<td>07/03/2016</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project: Improving P3 using variable thresholds</th>
<th>WP ref: WP7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Major constituent: SW</td>
<td>Sheet 7 of 7</td>
</tr>
<tr>
<td>Short description:</td>
<td></td>
</tr>
<tr>
<td>Apply the procedure in the production app and website.</td>
<td></td>
</tr>
<tr>
<td>Planned start date:</td>
<td>08/03/2016</td>
</tr>
<tr>
<td>Planned end date:</td>
<td>04/04/2016</td>
</tr>
<tr>
<td>Start event:</td>
<td></td>
</tr>
<tr>
<td>End event:</td>
<td></td>
</tr>
<tr>
<td>Internal task T1:</td>
<td></td>
</tr>
<tr>
<td>Develop iOS app</td>
<td></td>
</tr>
<tr>
<td>Internal task T2:</td>
<td></td>
</tr>
<tr>
<td>Develop website</td>
<td></td>
</tr>
<tr>
<td>Deliverables:</td>
<td></td>
</tr>
<tr>
<td>Source code</td>
<td></td>
</tr>
<tr>
<td>Dates:</td>
<td>04/04/2016</td>
</tr>
</tbody>
</table>
1.3.2. Time plan

The tasks that needed to be done are described in the following Gantt diagram.

![Gantt Diagram](image)

**Figure 1: Gantt Diagram**

1.4. Work plan modifications

There have been some minor timing deviations from the original plan. There has also been a major deviation: after completing the “Verify results” task a new approach of the same system has been studied instead of “Applying on production”. This new approach consists on deciding at a block level instead of deciding depending on the ROI and was proposed at the end of March.
2. Review of State of the Art

2.1. JPEG Images

JPEG is the most used lossy compression method for digital images. Even though it is a relatively old compression system, it is very fast, cheap and simple enough to implement in small devices such as digital cameras. This is a major reason why it is used in all computers, phones and over the internet.

In a lossy compression system (such as JPEG) some image data is lost during the compression process. Knowing how the visual human system works, this data can be chosen so that the human eye cannot tell the difference between the original image and the compressed one [7].

2.1.1. The JPEG compression process

The first thing to do is to convert the image from RGB to YCbCr. The Y component represents the intensity of a pixel and the Cb and Cr components represent the chrominance. This conversion allows greater compression (thanks to downsampling, explained next) without a significant degradation of perceptual image quality.

Once we have the colour space conversion, the next step is to take advantage of the fact that humans do not perceive changes in chrominance very effectively and downsample the Cb and Cr channels. The usual reduction in JPEG is by a factor of 2 in both the horizontal and vertical directions.

The next step is to divide the whole image into small blocks of 8x8 pixels. If the data for a channel (luminance or chrominance) cannot be represented with an integer number of blocks, then the encoder must fill the remaining area of the incomplete blocks with some form of dummy data.

After the block splitting, the Discrete Cosine Transform is applied to each block, converting the whole image to a frequency domain representation.

\[
G_{u,v} = \frac{1}{4} \alpha(u) \alpha(v) \sum_{x=0}^{7} \sum_{y=0}^{7} g_{x,y} \cos \left( \frac{2x + 1}{16} \pi u \right) \cos \left( \frac{2y + 1}{16} \pi v \right)
\]

\[
\alpha(u) = \begin{cases} 
\frac{1}{\sqrt{2}}, & \text{if } u = 0 \\
1, & \text{otherwise}
\end{cases}
\]

*Figure 2: DCT formula*

Where:

- \(u\) is the horizontal spatial frequency, for the integers \(0 \leq u < 8\)
- \(v\) is the vertical spatial frequency, for the integers \(0 \leq v < 8\)
- \(\alpha(u)\) is a normalizing scale factor to make the transformation orthonormal
• \( g_{x,y} \) is the pixel value at coordinates \((x, y)\)
• \( G_{u,v} \) is the DCT coefficient at coordinates \((u, v)\)

The next step is quantization, which leads to loss of image quality, but also allows us to get the most compression. There are quantization tables that have been empirically created by the JPEG committee experts. These tables define a fixed relation across different coefficients. Each DCT block is divided by its quantization table value and rounded to the nearest integer. There is also an optional parameter, called scale factor, that lets the user change the image quality. The JPEG standard defines it with a value of 1. In that case, the corresponding formula for the quantization step is the following.

\[
B_{j,k} = \text{round} \left( \frac{G_{j,k}}{Q_{j,k}} \right) \text{ for } j = 0, 1, 2, \ldots, 7; k = 0, 1, 2, \ldots, 7
\]

*Figure 3: Quantization formula*

Where each \( G \) represents an unquantized DCT coefficient; \( Q \) is the quantization matrix; and \( B \) is a quantized DCT coefficient.

The last step is to encode the quantized DCT blocks. In order to gather the coefficients a run-length coding is proposed. A specific scan is necessary to pass from a 2D to a 1D representation. A zigzag scan allows implementing the run-length coding. When there are only zeroes left, an “End Of Block” (EOB) is sent, so all the likely zeroes of the high frequencies do not need to be sent explicitly. Moreover, the previous quantized DC coefficient is used to predict the current quantized DC coefficient so the difference between the two is encoded rather than the actual value.

*Figure 4: Zigzag matrix*
2.2. **P3 Algorithm**

2.2.1. **Introduction**

Photo sharing Providers (PSPs) are growing and are becoming an essential part of our lives. They are useful because they perform server-side transformations on images (such as resizing or cropping) in order to improve the latency of image downloads (a mobile device, for example, will receive a much smaller image than a device with a large high resolution screen).

However, they have some problems. One of them is that they generate privacy concerns. There have been some photo leaks in the recent past years. Also, once a user uploads an image he/she looses the control of that image and PSPs apply all kind of computer vision to the images to classify them and extract information. Using traditional encryption techniques does not work for us because the server-side transformations applied by the PSPs would not be possible.

The Privacy Preserving Photo Sharing algorithm (P3) is a recently proposed system [2] that ensures photo privacy without sacrificing the latency, storage and bandwidth benefits provided by Photo sharing Service Providers (PSPs).

Given a normal JPEG image, the P3 algorithm splits it into 2 parts using a threshold based-splitting technique. On the one hand side there is the public part, which contains very little visual information but most of the volume in bytes. This public part image is the one shared with the (untrusted) PSPs. On the other hand, there is a secret part, which has most of the visual information and a small number of bytes. This secret part is then encrypted and not shared with the PSPs but shared using other means, such as an independent web server. Then the PSP can perform some linear processing and a recipient with access to both parts can reconstruct the original image.

![Figure 5: P3 algorithm scheme](image)

In some cases, the secret part can be embedded in the JPEG Application Specific Markers so the recipient has all the information in one place and does not need to take each part of the image from a different server.
2.2.2. Sender side

During the JPEG encoding, after the DCT and quantization step, we have the image in the frequency domain divided into $8 \times 8$ blocks [Figure 4]. By definition, in a DCT block we have in the first position the so-called DC coefficient, representing the frequency 0. This means that the DC coefficient contains information about the average intensity (or chrominance) of the block. The average of each block can be very informative (e.g., a smaller image with all the DC values can represent a good approximation to the original), therefore the P3 algorithm sets this coefficient of the block to the secret part. For the rest of the coefficients, it uses a user-defined threshold. The coefficients that are above the threshold are split: the value of the threshold goes to the public part and the difference goes to the secret part. The coefficients that are below the threshold go directly to the public part [Figure 6]. Notice that a higher threshold value means less information included in the secret part, which means less security. A lower threshold will split more coefficients and let less information remain in the public part so less information will be available to the unwanted viewers.

![Figure 6: Privacy-Preserving Image Encoding Algorithm](image)

As mentioned, using this P3 algorithm instead of using full encryption when sharing images allows us to provide information (public part) to the PSPs to apply the server side transformations. For instance, if we upload a picture to Facebook they send different resolutions depending on the device: when viewing the image on a laptop a larger resolution will be given, but if viewing the image on a mobile device a smaller resolution will be downloaded instead. This is good because it matches the image resolution to the display size, resulting in optimized download times and reduced network latency and
bandwidth usage. Encrypting the whole image might be safer than P3 but it means we lose many of the advantages of using PSPs, because the server can no longer process JPEG images (e.g., to reduce their resolution).

2.2.3. Recipient side

A user that has access to both the public part and the secret part can easily reconstruct the original image. Given a block coefficient the recipient only needs to add both parts together taking into account the sign of the coefficient value, because coefficients can be positive or negative. Having the public part coefficient value, the secret part coefficient value, the sign information and the value of threshold used it is easy to recover the original image using the following formula.

\[ y(i) = x_p + x_s + (S_s - S_s^2) \cdot T \]

where:

- \( x_p \) is the public part coefficient
- \( x_s \) is the secret part coefficient
- \( S_s \) is the sign of the secret part coefficient (1 or -1)
- \( T \) is the threshold

2.2.4. Evaluation

The P3 algorithm is fast in terms of computation speed and by removing the private information it breaks computer vision algorithms such as edge detection, SIFT feature detection and face detection. The only bad thing is that it introduces an overhead to the resulting image. This means that the size of the sum of public and secret part is bigger than the size of the original image.

![Figure 7: Lena image](image7.png)

![Figure 8: P3 public part of Lena image](image8.png)
2.3. Related work

Since the publication of P3 [1] numerous papers appeared proposing similar algorithms. It is worth mentioning at least one of them, “PuPPileS: Transformation-Supported Personalized Privacy Preserving Partial Image Sharing” [3]. This paper implements a system where images can be shared in PSPs similarly to P3 [1] but where only a region is encrypted. The authors claim that it could be useful to hide faces, SSN numbers, etc. but keep the background of the image completely visible.

PuPPileS uses the image perturbation technique to “encrypt” the sensitive areas in the original images. It can also support popular image transformations (such as cropping, rotation, etc.) in the server side, like P3.

It points a weak point of P3, and solves this problem by encrypting only a region of the image. A more comprehensive and better privacy protection is proposed in this work.
3. **Methodology / project development:**

This project was developed using the P3 code itself, which was built on top of the Linux Libjpeg library. This library is an open source project written in C that implements JPEG encoding/decoding. It’s one of the main open source JPEG libraries and is maintained by the Independent JPEG group. We used the Mac OS X software Xcode for the development and created a Mac app that uses the modified Libjpeg library to process the images and get the results.

We will use edge detection as a metric to determine how well privacy is protected. We will use the Canny Edge Detector implementation of the OpenCV library in C. We will compare the number of detected edges in our results with the number of detected edges in the original image. We will say that detecting less than 20% of the original image edges provides enough security because there is very little information left considering the discontinuities of the detected edges.

3.1. **Variable threshold depending on the Region of Interest (ROI)**

In this research, we tested P3 with different images and different thresholds and analysed the results. It was noticed that in certain types of images and with certain threshold values the public image provided only limited privacy protection. In particular, using a value of threshold 30 (which is a good trade-off between privacy and overhead) it was noticed that if the image contained a face, the edges of that face could be seen [Figure 6]. As stated in the P3 paper [2], we confirmed that the computer vision algorithms like face detection fail on this image but it is clear that a face can be seen in this type of images. Moreover, we can say that in some cases the person on the image can be recognized by a human, in particular if it is someone famous [Figure 9].

![Figure 9: P3 public image with a face](image-url)
Everybody would agree that a face is a particular case of sensitive information. In an image with a face, probably the face is the most private part, such that if seen by an unwanted viewer it could cause privacy leaks.

Based on the previous statement, we decided to encrypt faces with a lower threshold after applying face detection. The basic idea is that there is no need to encrypt the whole image with a low threshold because this would cause a lot of unnecessary overhead. Only the Region of Interest (ROI) of the face needs to be encrypted with the low threshold.

### 3.1.1. ROI definition

A more general problem is to consider not only faces but regions selected by the user (ROIs) for which protection should be improved. In this project, we developed an algorithm that works with two scenarios. The first one uses the Haar Cascade Face Detector from the OpenCV library to detect a face in the image. If there is a face, the face detector gives the coordinates of the ROI. The second scenario uses user-defined coordinates for the ROI. This is useful because it lets users decide which part of the image needs better security. As an example, see Figures 10-11 showing face detection and user-defined ROIs for the same image. In this case, the user-defined ROI [Figure 11] corresponds to the screen of the laptop, because it may contain confidential information that needs better security.

![Figure 10: ROI using face detection](image1)

![Figure 11: User-defined ROI](image2)

This ROI is represented by two \((x, y)\) points, corresponding to the top-left corner and the bottom right-corner.

### 3.1.2. System design

Once we have the ROI information in pixels we transform it to its corresponding DCT block value. During the P3 process, in the threshold-based splitting part we now consider the DCT blocks inside the ROI and use a lower threshold, which we call “secure threshold”
(THS). The P3 algorithm remains the same but the threshold used is different in the blocks inside the region. In the recovery part, previously the receiver only needed to know the threshold used and this could be defined a priori so there was no place for error. Now the receiver needs to know both the normal threshold and the secure threshold used inside the region. Moreover, it needs to know the ROI where the THS is applied so it can recover the original image properly.

JPEG has different markers that are used to define the start of an image, start of a frame, define quantization tables, etc. One of them is the Application-specific marker (APPn). This can be used to embed data into an image. For instance, the APP1 marker is used for the Exif metadata of a JPEG image. We used one of these markers to embed an array of 4 numbers defining the ROI. That way, the receiver can look at that marker and know which region of the image is using the normal threshold (TH) or the secure threshold (THS) in order to get a perfect recovery.

3.2. Variable threshold depending on the DCT block

The second part of this project focuses on facing the same problem but in a different level. We noticed that the ROI approach worked well but it was not optimal. When the ROI has been defined by the user, there is no room for further optimization. However, in a given region, not all the pixels present relevant information.

It would be useful to remember that the only information remaining in a P3 public part image tends to correspond to the edges. Given that fact, we noticed that not all the pixels or DCT blocks in a region that should be encrypted using THS necessarily represent an edge, thus they could be encrypted with the normal TH. If this is done then the result would be the same in terms of security but will require less overhead.

Based on this observation we decided to develop a system that would be capable of deciding for each single DCT block if it should use TH or THS. This would create an image where instead of having a region encrypted with TH and another with THS, there would be some blocks encrypted using TH and some using THS.

This part is a completely independent idea of the ROI. We just noticed that it is more important to protect certain areas because of their structure and decided to optimize the assignment of using two thresholds.

3.2.1. System design

We can treat each block independently and we need a way to decide if a block should be encrypted using TH or THS. To do that we propose to study the coefficients within the block. Each block has 64 coefficients representing the frequencies. We define a parameter \( k \) that represents the number of coefficients with an absolute value over the THS. We count the number of coefficients with an absolute value over the THS and compare it with \( k \). If it's greater than \( k \) then we consider that block is important so that it
should be encrypted using THS. If it is lower than \( k \) then we consider that the block should use normal TH.

What we have at this point is a P3 image that has some DCT blocks encrypted using the normal threshold (TH) and some block that have been encrypted using the secure threshold (THS). Again, the problem we need to solve now is letting the receiver know how to do the reconstruction. This means the receiver needs to know which blocks have been encrypted using each threshold. To do that, we generate a JPEG image and embed it in the JPEG APPn markers just like before. In this case, we need to embed more information in the marker and using an array with all the block numbers that use THS is not optimal. If the image has very few blocks using the THS, the array approach is better but as images will have several blocks using THS, generating a JPEG image representing that information is better. There is the option of using the JBIG format as the image is binary but this is left for future work.

When doing the threshold-based splitting, the methodology explained above determines if a block will use TH or THS. Then we create a small image that has 1 pixel per DCT block of the original image. Each pixel has 1 bit of information so it will be black if that DCT block is being encrypted using the normal TH or it will be white if that DCT block is being encrypted using the secure THS. Once we have got this image generated [Figure 12] we embed it in a P3 image APPn marker. The overhead this represents is negligible because it is 1/64 bits/pixel.

![Figure 12: Image representing the THS blocks](image)

So far we have a system that decides at a block level which threshold to use and sends that information to the receiver in order to get a complete reconstruction. This system has as input parameters the THS value selected and the \( k \) value, which also can be defined. The TH value has been fixed at 30 as explained before because it gives enough security to not recognize most images (e.g., landscapes) and introduces very little overhead. In the output of the encoder we have a public part image (grey image) that will present a number of edges detected and an overhead with respect the original image.
3.2.2. Optimization

Our system has these 4 parameters described above that can be combined in multiple forms and can produce multiple outputs. We want to find the best combination possible in terms of achieving a good trade-off between having the minimum number of edges detected and generating the minimum overhead. In order to obtain the optimal values, we plot all the possible cases for an experimental dataset and find the optimal curve of the graphic. We use a dataset of 100 random HD images taken with an iPhone 6 camera. We calculated the results for THS values from 1 to 25 and from $k$ values from 1 to 25 also. The result can be seen in Figure 13.
Figure 13: Optimization results
Figure 11 represents all the possible results. The X axis represents the overhead and while on the Y axis we have the detected edges. Both values are shown as a percentage with respect to the normal P3 algorithm. Each line represents a value of THS and each dot in a line is a $k$ value. The optimal case would be the origin $(0, 0)$ but as we can see this is obviously not possible. In the top-left corner we have the normal P3 case, where there are 100% of the edges detected in respect P3 and 0% overhead, so a normal P3 image. On the other extreme we have the most secure image: 0% of edges detected but a lot of overhead. The points that define the curve closer to the origin are the optimal points and are represented in the Figure 12.

![Figure 14: Graphic representation of the optimal points.](image)

<table>
<thead>
<tr>
<th>THS</th>
<th>$k$</th>
<th>EDG (%)</th>
<th>OH (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>3</td>
<td>64.99</td>
<td>2.11</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>39.89</td>
<td>2.89</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>28.21</td>
<td>3.49</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>22.89</td>
<td>3.72</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>17.9</td>
<td>4.26</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>15.85</td>
<td>4.39</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>9.93</td>
<td>5.13</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>9.69</td>
<td>5.23</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>5.03</td>
<td>6.14</td>
</tr>
</tbody>
</table>
Looking at the numbers above we can see that we have a variety of optimal combinations to choose depending on the needs of our system. On one hand, if we want more security we should choose a pair of THS and $k$ from the bottom part of Table 1. On the other hand, if we need very little overhead we should choose a pair of THS and $k$ from the top part of Table 1.

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>1</td>
<td>4.95</td>
<td>6.18</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>2.19</td>
<td>7.25</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>2.28</td>
<td>7.33</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1.86</td>
<td>7.62</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>0.2</td>
<td>8.61</td>
</tr>
</tbody>
</table>

Table 1: Table representation of the optimal points
4. **Results**

4.1. **Variable threshold depending on the Region of Interest (ROI)**

Our algorithm was tested with the MUCT Face Database using the first 500 images [8]. The case of multiple faces in a single image has not been considered and is left for future work. Faces are used as an example, instead of the user-defined ROI, because using the face detector was a fast and easy way of identifying an ROI for each image.

Calculating the average overhead for all the images provided the results shown in Figure 13. The blue line represents the overhead when using normal P3 algorithm with TH=30. The green line represents the overhead when using the THS=5 in the whole image, without taking into account any ROI. The yellow line represents our proposed algorithm (called Variable-P3 or V-P3), where the region of the face is using THS=5 and the rest of the image is using TH=30. As we can see the results of our proposed algorithm introduce a little bit of overhead with respect to normal P3 but it is much less than when using THS for the whole image.

![Figure 13: Overheads generated in different images](image)

After running the edge detector to the same dataset as before, we obtained the results shown in Figure 14. The orange line represents the average of the edges detected in the original images. In blue, there is the number of edges detected in the normal P3 image (TH=30). In green we have the line of the edges detected using the THS=5 in the whole image, without any ROI defined, which is always 0. Finally, in yellow there are the results of using an ROI for the face. As we can see, using our proposed algorithm leads to fewer edges being detected than in the normal P3 image.
After analysing these results, we can see that an improvement of P3 is achieved. We can generate images with an obfuscated region where no edges are detected with only a small amount of overhead [Figure 17].

Looking at the edges detected in the above example image we can see that the edges detected in the original image are pretty clear. It is worth mentioning that the edges in the original P3 are detected but they are affected by some kind of noise and thus the edges are not in the exact same position in both images. Due to the P3 algorithm, the edge detector detects false positives edges that are not present in the original image. When applying the lower threshold (THS) in the face region then the edge detector completely fails in that region and the overhead in this particular case is only increased by a 3% [Figure 16].
These results are good but as the region is a pre-defined parameter there is no place for algorithm optimization. Depending on the original image characteristics and the size of the region the system will output an image with more or less overhead and more or less edges detected.

4.2. **Variable threshold adapted to the DCT block contents**

As explained in Section 3.2.2 there are different combinations of THS and $k$ that one can choose for the system. For the scope of this project we chose a mid-range value of the Table 1 in order to get some results. We chose a value of THS=12 and a value of $k = 2$ which for the studied dataset gives in average an output image with $\sim$10% edges detected with respect of the original P3 algorithm public image and an increase of $\sim$5% in size (overhead).

Figures 17-20 provide an example based on high resolution photos taken with an iPhone 6 [Figure 17] to which the edge detector was applied [Figure 18]. Then we applied the original P3 algorithm and applied the edge detector to the generated grey image [Figure 19]. This grey image generated by P3 had an overhead of 8.63% in size respect the original and when we compared the edges detected we found that only 32% of the edges where detected in the P3 image. Then we applied our proposed algorithm using the
values of THS and $k$ stated before and I applied the edge detector to the output image. This image had an overhead of 10.66% with respect to the original image (a small increase) but when comparing the detected edges this new image had only 10.9% of edges matching those of the original image.

This reduction in the number of detected edges makes a great difference because it makes some key elements of the image disappear (e.g., the speakers hanging from the roof in Figs. 20-23). In the Appendix there are more examples of the results with bigger images so the edges can be better appreciated.
5. **Budget**

In this section there is an analysis of the overall costs of the project. As this is a research project based on software development, there were no physical components needed. Open source libraries and software have been used as much as possible, leaving only one commercial license that needed to be bought: Matlab. We needed the basic Matlab software and a few of extra toolboxes.

<table>
<thead>
<tr>
<th>Software</th>
<th>Units</th>
<th>Price/unit</th>
<th>Total price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matlab</td>
<td>1</td>
<td>105 €</td>
<td>105 €</td>
</tr>
<tr>
<td>Toolbox</td>
<td>15</td>
<td>29 €</td>
<td>435 €</td>
</tr>
</tbody>
</table>

*Table 2: Project's software budget*

Apart from the software, we need to take into account the hours spend by the junior engineer, whose salary is 8€/hour.

<table>
<thead>
<tr>
<th></th>
<th>Price per hour</th>
<th>Months spent in the project</th>
<th>Hours worked per day</th>
<th>Total hours worked</th>
<th>Total cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Junior engineer</td>
<td>8 €</td>
<td>7</td>
<td>6</td>
<td>840</td>
<td>6720 €</td>
</tr>
</tbody>
</table>

*Table 3: Project's engineers budget*

The total cost for this project would be 7260€.

Analysing the cost, we can say that this project required a very reduced budget because of its strong software component and thanks to the open source community.
6. **Conclusions and future development:**

This project is based on another project called P3. The goal was to improve P3 public image’s security by using different thresholds in different parts of the image.

The first part of the project consisted in using a more secure threshold in a specific region, either defined by the user or by a face detector. We found that doing this improved the security of that specific region without compromising overhead as much as the original P3 algorithm.

The second part of the project consisted in looking at the same problem from a different perspective. Instead of looking at the whole image, we focused on each individual DCT block in order to decide the threshold. This development led to experimental results and to the formulation of an optimization problem, which was solved leading to the optimal values for the input parameters.

This thesis left some ideas for future work. One of them is using multiple Regions of Interest. Right now only one user-defined region can be used but there could be scenarios where a user wants to protect different separate regions. Another future work is to use JBIG format for the image that represents which blocks use each threshold. As that image is binary, JBIG format would introduce less overhead.
7. Bibliography:


8. Appendices:

8.1. Auditorium

Figure 23: Auditorium image

Figure 24: Auditorium original image edges
Figure 25: Auditorium P3 image edges

Figure 26: Auditorium variable P3 image edges
8.2. **Beach**

*Figure 27: Beach image*

*Figure 28: Beach original image edges*
Figure 29: Beach P3 image edges

Figure 30: Beach variable P3 image edges
8.3. Concert

Figure 31: Concert image

Figure 32: Concert original image edges
Figure 33: Concert P3 image edges

Figure 34: Concert variable P3 image edges
9. **Glossary**

P3: Privacy Preserving Photo sharing algorithm

ROI: Region Of Interest

DCT: Discrete Cosine Transform

JPEG: Joint Photographic Experts Group

EOB: End Of Block

PSP: Photo Sharing Provider

SIFT: Scale-invariant feature transform

TH: Threshold

THS: Secure Threshold

EDG: Detected Edges (%)

OH: Overhead (%)