Document Obfuscation

January 2022

Contents

1 Problem Introduction

The current pandemic context, which has lasted for more than two years already, has brought a paradigm shift in communication and the presentation of documents in electronic format when you want to deal with official entities reducing physical presence overall. With this transition, the number of official documents in email attachments increased and also we now have to physically present our ID with the new COVID-19 Green Certificate. This is great, officials must know that the certificate that we presented is ours but this method of correlation leaks private information about us but some documents or photos are confidential and are not relevant for the context they are used in. If we upload these documents to the cloud, GDPR reasons may occur even if they are stored securely. A potential solution to this problem would be to obscure these images and documents with a set of randomly generated steps that would alter their content until the owner wants to access them again.

Example: for photos we can apply N layers above it in a random order with visual operations such as changing the contrast, adding effects (blur, pixelate the image), changing various properties, etc. These operations will be retained in order to be able to retrieve the content once the user wants it again. In addition to these operations, various mechanisms for verifying the validity of document integrity can be implemented, such as a signature of the original content and its verification after de-obfuscation.

2 State of the art - Silviu

Data privacy in photos is an important subject that is a major concern in today's world, especially given the compromising data it can contain or the rate at which the information can flow through the internet. So in this paper, we are proposing an approach that can mitigate the danger of sharing things like documents or personal images on the internet. There are multiple research papers on this topic that differ in the real-life problem they are trying to resolve, but ultimately they have the same fundamentals, securely storing sensitive data and being able to recover the original image when needed. For example, this paper [11] proposes a way of solving the previously described problem but it takes a heavy focus on the social media part. It uses several techniques for obfuscating the data, and categorises them into two separate groups:

- Pixel replacement: this technique replaces pixels of an original image with other masks, distortions or patterns. The original pixels are encrypted and embedded together with information about the position and shape of the area.

- Data manipulation: this technique does not replace the original pixels but changes them in a specific way. Typical examples include image encryption and scrambling. Only information about the shape of the protected regions needs to be embedded together.

We focus more on the ability to share specific parts of the document with the intended person. Due to this, we have to first make sure to hide the sensitive information, then to give the ability to restore the initial image when needed. Since the key to restoring the information can be held on the client-side, we have the added security of not being able to recover the original photo without the presence of the user. Three of the most common techniques used to obfuscate data are encryption, tokenization, and data masking.

3 Solution - Robert

We provide a nice web page and two services. The first service is a simple face detection API that use a machine learning in order to detect the faces in a photo in order to resolve the most common use case, hiding the faces of other people in a photo. The second service is the core of our application and handles the obfuscation and de-obfuscation of a document. This will later be split intro the separated micro-services in order to better handle the increase traffic.

The building block of our project is the Layer. In order to obfuscate a document we will execute an obfuscation algorithms specified by the user and save the algorithm ID and the necessary key data for the reverse process. In order to de-obfuscate a document the layers will be processed in reverse order. The layer is concerned with handling one specific algorithm and its required data.

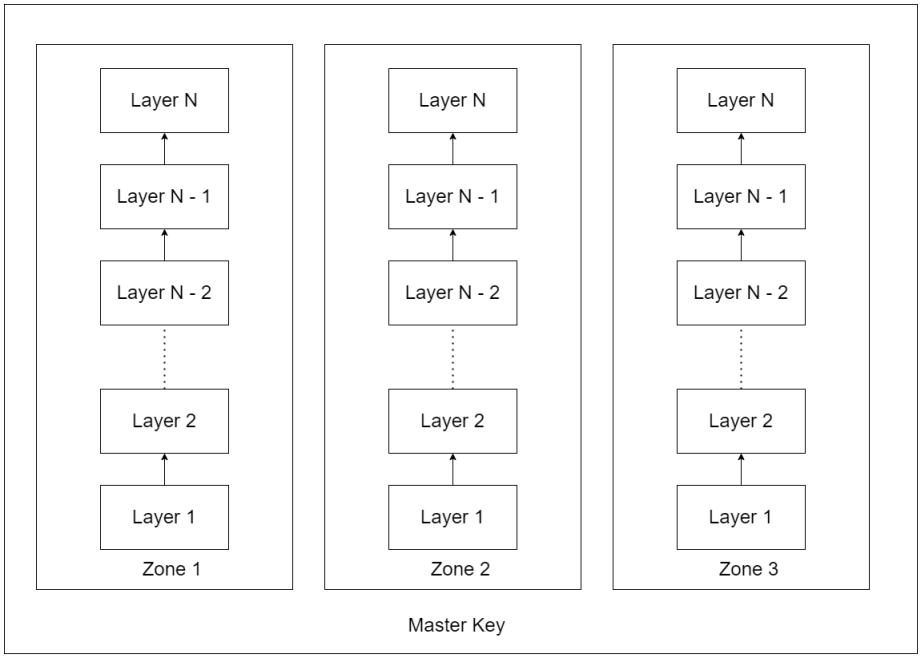

Figure 1: Key Model

Several layers applied on the same area are called a Zone. The Zone holds the redundant information that is used by all the layers and does not need to be stored multiple times as well as the order in which the layers have been applied.

Several zones form a Master Key. The Master key is a container for the

zones and deals with the meta-information about the state of the application. Zones can be removed from the master key in order to keep certain sensitive data hidden. The master keys are not saved on our side and only the client has access to it.

This designed is easy to achieve in code using the "factory" and "chain of responsibility" designed patterns. Each algorithm has an ID that is used to create obfuscation or de-obfuscation chains. In order to create those chains we implemented two specialized factory.

After parsing the input, a user request to obfuscate any number of zones with any combination of layers or a Master Key object we will apply those changes to the document and return it.

In order to better explain the process, we will try to demonstrate it with a visual example. The first step is for the system to automatically identify the faces from the uploaded photo, then the user will have the possibility to select it's own interest areas, this two steps being referred in technical terms as "marking the zones".

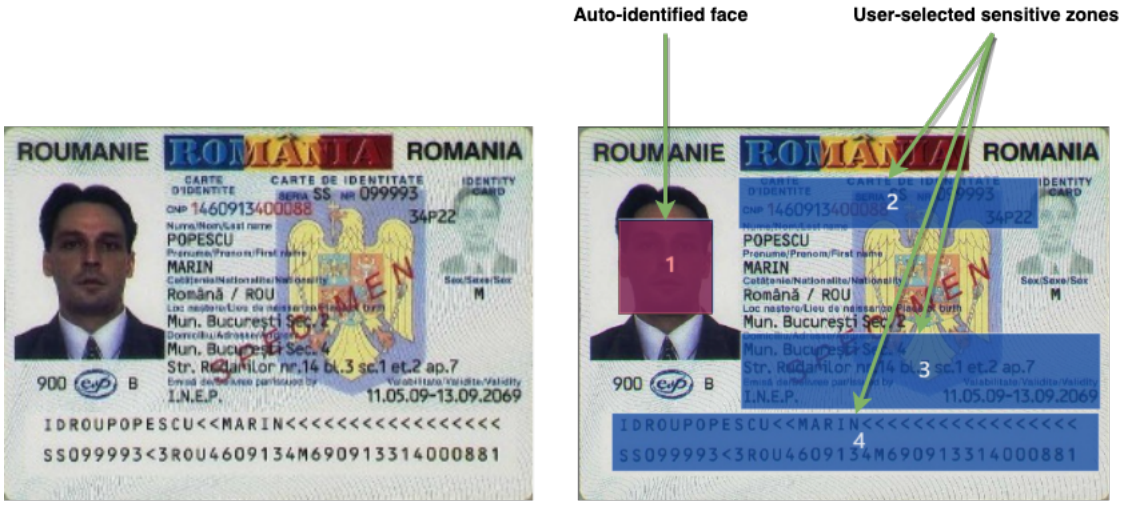

Figure 2: Identification of Zones

After the zones have been selected, the system will allow the user to select the algorithms with which to obfuscate each of them, then generating a unique key for each of these zones.

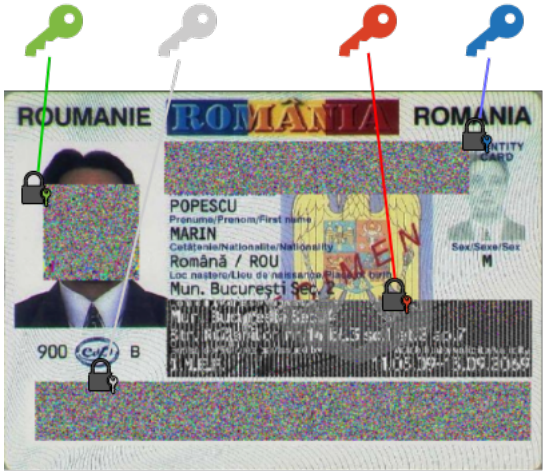

Figure 3: Obfuscated photo with multiple keys

Once the obfuscation process is complete, the user can now share the photo in its new form and also one or more keys. In our case, if the user would like to provide access to the photo and the address of its ID Card, he should use the two relevant keys in this context: the red key and the green key, the result of the de-obfuscation being similar to the one below.

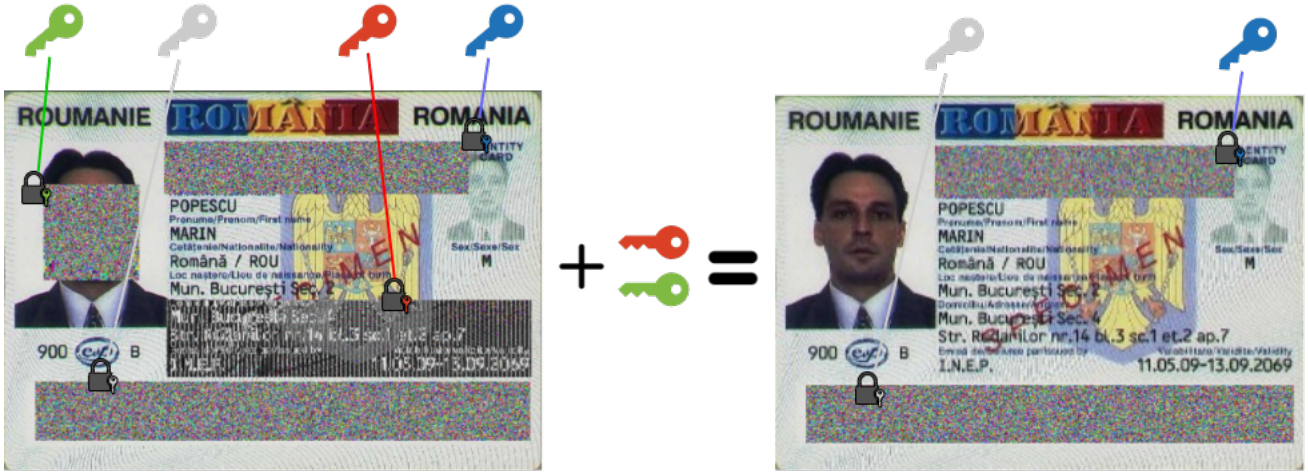

Figure 4: Partial de-obfuscation

4 Results and evaluation - Ana

4.1 AOP and MOP

For continuously evaluating the performance of the application, we used Aspect-Oriented Programming and Monitoring-Oriented Programming.

Aspect-Oriented Programming is used to measure the time that it takes for each obfuscation method to be applied to a certain picture.

Monitoring-Oriented Programming is used inside the obfuscation module for giving warnings inside the application logs about the obfuscation methods that take too long, and in the face detection module to ensure that some operations are done in the correct order (for example, that a picture is saved to the server before trying to detect faces from it).

4.2 Stress testing

In order to see how the application performs in a stress scenario, meaning a big number of users accesses the same page of the application at the same time, we created a script that performs a given number of requests to the server at the same time, and then increases the number of requests sent simultaneously until it reaches a certain threshold. The results obtained for the deobfuscation page are as shown below:

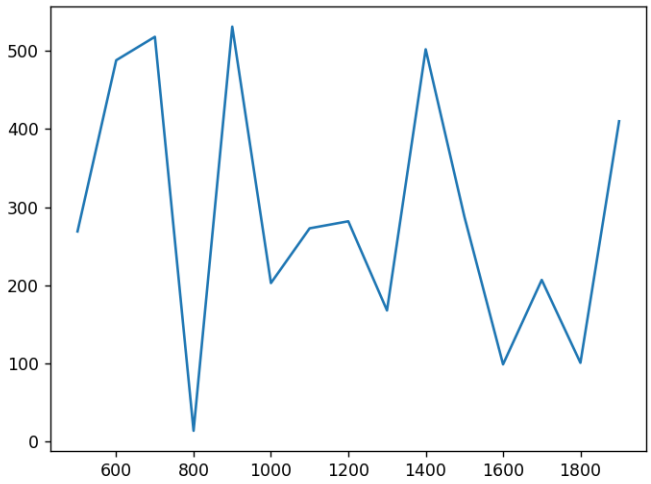

Figure 5: Stress test results

The X axis represents the number of simultaneous requests and the Y axis represents the total time in nanoseconds that it took for the server to respond to all the requests.

5 Comparison with other solutions - Roxana

Surprisingly, even if data privacy is a highly discussed topic nowadays, there are not many tools that can solve the problem of document obfuscation and almost none that take into account reversible document obfuscation.
The existing tools focus on censoring and encrypting the images using simple algorithms. The algorithms cannot be combined and only one region or the whole image can be censored or encrypted.
At the moment, there is no online tool available that uses the theoretical solutions discussed in the State of Art section.

5.1 PineTools - Censor Photo

An online tool that offers the possibility to censor documents is offered by PineTool: PineTools - Censor Photo.
Here a section of an image can be selected and the user can opt for pixelating, blurring the region or just covering the image with a solid color. This is a quick and easy to use tool, perfect for hiding some information in a document.
However, this action is not reversible and the original document cannot be recovered. Moreover, only one region can be selected at one time, therefore if the user wants to censor multiple regions the process is quite tedious.

5.2 Image Online.co - Encryption

Image Online.co offers another tool that can be used to encrypt the whole image using a pixels scrambling algorithm. This process is reversible but just using the description tool provided by them as the encryption key cannot be specified and is not known.
The downsides of this solution is that the whole image must be encrypted and any person that has access to the encrypted photo can decrypt it without other information needed just by using the decryption tool.

6 Future work

Knowing the parts with which our system is great but also the less good parts of it, we intend to add improvements and micro-optimizations in the future to make our application better, faster and with more features. A part where we definitely have work to do is creating client apps for mobile phones, Android and iOS specifically, in order to diversify our user base and have multiple ways to access our app. Especially when our system relies on photos and scans of documents, this feature will also extend our feature list by allowing the user to directly capture his document with a smartphone camera instead of saving it to a file and uploading it to our web interface.

One extra thing we aim to do is also add the ability to create user accounts so that the user can save the keys in a sort of "Secure Vault" in order for them

to be recoverable later on. This would also open up new horizons for features such as permission management, keys that expire after a certain time or the addition of user-groups that automatically take over the keys without having to share them on other communication channels for better security.

7 Conclusions

In addition to the fact that our system offers a multitude of ways to obfuscate images, it can automatically identify faces, has support for multi-zone obfuscation, offers the ability to export keys separately and offers a complete flow to enables the goal that we set from the beginning, namely to give a person the possibility to hide a certain sensitive part of a photo that is not relevant for a certain purpose.

Referring to the already existing solutions, seeing the ways in which a system for image obfuscation of this type is currently working and taking into account the existing standards, we consider that the software built by us is very fast, easy to use and accurate.

Given that the chosen field is one of great importance, where there are many problems and the progress made so far is low, the proposed solution is a breath of fresh air in the industry, with huge potential to solve, at least partially, the problem of excessive shared information with 3rd parties.

References

[1] Khosro Bahrami, Alex C. Kot, Leida Li, and Haoliang Li. Blurred image splicing localization by exposing blur type inconsistency. *IEEE Transactions on Information Forensics and Security*, 10(5):999–1009, 2015.

[2] Finn Brunton and Helen Nissenbaum. *Understanding Obfuscation*, pages 44–44. 2015.

[3] Todd A. Ell. Hypercomplex color affine filters. In *2007 IEEE International Conference on Image Processing*, volume 5, pages V – 249–V – 252, 2007.

[4] Liyue Fan. A demonstration of image obfuscation with provable privacy. In *2019 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pages 608–608, 2019.

[5] Hui Yu Huang and Wei Chang Tsai. Blurred image restoration using fast blur-kernel estimation. In *2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 435–438, 2014.

[6] Anil K. Jain, Brendan Klare, and Unsang Park. Face recognition: Some challenges in forensics. In *2011 IEEE International Conference on Automatic Face Gesture Recognition (FG)*, pages 726–733, 2011.

[7] A. Kovalchuk, D. Peleshko, M. Navytka, and T. Sviridova. Using of affine transformations for the encryption and decryption of two images. In *2011 11th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*, pages 348–349, 2011.

[8] Yuki Sanjo and Takashi Toriu. A method for affine invariant image smoothing. In *2013 Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 242–245, 2013.

[9] Shejin Thavalengal, Ruxandra Vranceanu, Razvan G. Condorovici, and Peter Corcoran. Iris pattern obfuscation in digital images. In *IEEE International Joint Conference on Biometrics*, pages 1–8, 2014.

[10] Ren Wu, Shengen Yan, Yi Shan, Qingqing Dang, and Gang Sun. Deep image: Scaling up image recognition. *arXiv preprint arXiv:1501.02876*, 7(8), 2015.

[11] Lin Yuan, Pavel Korshunov, and Touradj Ebrahimi. Privacy-preserving photo sharing based on a secure jpeg. In *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 185–190. IEEE, 2015.

[12] Ghazi Mohammed Zafaruddin and H. S. Fadewar. Face recognition: A holistic approach review. In *2014 International Conference on Contemporary Computing and Informatics (IC3I)*, pages 175–178, 2014.