Perceptual Hashing for Authentication using Physically Unclonable Functions

João Barbosa Tareco

Thesis to obtain the Master of Science Degree in

Engenharia Electrotécnica e de Computadores

Supervisors: Professor Doutor Paulo Luís Serras Lobato Correia
Professor Doutor Paulo Sérgio de Brito André

Examination Committee

Chairperson: Professor Doutor José Eduardo Charters Ribeiro da Cunha Sanguino
Supervisor: Professor Doutor Paulo Luís Serras Lobato Correia
Member of the Committee: Professor Doutor Luís Eduardo de Pinho Ducla Soares

November 2021

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

Ao Professor Doutor Paulo Luís Serras Lobato Correia quero expressar o meu sincero reconhecimento pela disponibilidade e encorajamento manifestados na orientação deste trabalho.

Ao Professor Doutor Paulo Sérgio de Brito André agradeço, de igual forma, todo o incentivo, apoio e orientação prestados.

Ao Mestre Tiago Filipe Santos Silvério, que se demonstrou sempre disponível para ajudar, desde o apoio na compreensão do seu trabalho, no qual esta tese se baseia, até à obtenção de dados. Sem ele esta tese não existiria.

Finalmente, ao Instituto de Telecomunicações agradeço o acolhimento do tema conducente à presente dissertação.

Abstract

The development of authentication methods is an increasingly important topic, which is also the focus of a wide discussion in the public sphere. Optical PUFs are a type of cryptographic device that leverages the inherent randomness of certain objects (tokens) for authentication purposes, by probing them with a coherent light, generating unique speckle patterns. However, a problem arises due to intra-variability, where speckle patterns obtained from the same token can vary depending on the acquisition environment and system alignment. This work shows how a Physically Unclonable Function (PUF) system utilizing tracing paper tokens can be employed for authentication purposes, despite the intra-variability between acquired speckle pattern images. Two datasets were acquired with the purpose of simulating intra-variability (turning the system ON and OFF and changing the camera positioning). Non-reflective black tape around the region of interest was also used to automate the cropping process. In the pre-processing stage, warping was performed, as well as pixel intensity normalization, followed by Gabor Filtering. Two feature extraction methods were tested for hash generation using: (i) Discrete Cosine Transform and (ii) Principal Component Analysis. Two classification approaches were tested: (i) a Hamming Distance based classification and (ii) machine learning classifiers. For a data independent method, the Discrete Cosine Transform (DCT) combined with Hamming Distance (HD)-based classification achieved the best results. For a data dependent method, the Principal Component Analysis (PCA) with machine learning classifiers performed the best overall. Gabor filtering provided an authentication performance boost, but the kernels used may need to be calibrated between datasets.

Keywords

Discrete Cosine Transform; Principal Component Analysis; Support Vector Machines; k-Nearest Neighbour; Random Forest; Machine Learning; Authentication; Gabor Filtering; Intra-Variability;

Resumo

O desenvolvimento de métodos de autenticação é um tema cada vez mais importante e também o foco de ampla discussão no domínio público. As funções fisicamente não clonáveis (PUF) ópticas são um tipo de dispositivo criptográfico que dá uso à aleatoriedade inerente de certos objectos (tokens) que quando irradiados com luz coerente, produzem padrões de speckle únicos, podendo ser utilizados para fins de autenticação. Contudo, padrões de speckle obtidos pelo mesmo token podem variar (intra-variabilidade) dependendo das condições de aquisição, impedindo a sua autenticação. Este trabalho demonstra como uma PUF que utiliza tokens de papel vegetal pode ser utilizada para fins de autenticação, apesar da intra-variabilidade entre padrões de speckle adquiridos. Dois conjuntos de dados foram adquiridos com a finalidade de simular intra-variabilidade (ligar e desligar o sistema e alterar o posicionamento da câmara). Para recorte automático da região de interesse das imagens foi utilizada fita preta não refletiva. Na fase de pré-processamento foi também efetuado warping, bem como normalização da intensidade dos pixels, seguida de filtragem de Gabor. Foram testados dois métodos de extração de características para gerar chaves: (i) Transformada Discreta de Cosseno (DCT) e (ii) Análise de Componentes Principais (PCA). Utilizaram-se duas abordagens de classificação: (i) baseada em distância de Hamming e (ii) classificadores de aprendizagem automática. Para uma abordagem de dados independentes, a DCT combinada com uma classificação baseada em distância de Hamming permitiu obter os melhores resultados. Para uma abordagem de dados dependentes, revelou melhor desempenho a PCA combinada com classificadores de aprendizagem automática.

Palavras Chave

Transformada Discreta de Fourier; Análise de Componentes Principais; Aprendizagem Automática; Autenticação; Funções Fisicamente Não Clonáveis; Intra-Variabilidade; Transformada de Gabor;

Contents

1 Introduction 1

1.1 Motivation . . . 2

1.2 Core Concepts . . . 2

1.2.1 What are Optical PUFs? . . . 2

1.2.2 What is Perceptual Hashing? . . . 3

1.3 Problem Formulation . . . 4

1.4 Objectives . . . 4

1.5 Contributions . . . 5

1.6 Dissertation Outline . . . 5

2 Review of Authentication Techniques using Optical Physical Unclonable Functions 7

2.1 Overview of Optical PUFs . . . 8

2.2 Overview of Perceptual Hashing Methods . . . 11

2.3 Gabor Transform for Speckle Pattern Analysis . . . 17

2.4 DCT Based Hashing Solutions . . . 19

2.5 PCA Based Hashing Solutions . . . 20

2.6 Classification Algorithms . . . 22

3 The PUF System Used 25

3.1 Introduction . . . 26

3.2 The PUF System . . . 26

3.3 Image Acquisition . . . 27

4 Proposed Approach for Authenticating a PUF 31

4.1 Introduction . . . 32

4.1.1 Desired Features in an Authentication Algorithm for PUFs . . . 32

4.1.2 Proposed Approach . . . 34

4.2 Pre-Processing . . . 36

4.2.1 Warping and Cropping Speckle Pattern Images . . . 37

4.2.2 Gabor Filtering . . . 41

4.3 Feature Extraction and Hash Generation 42
4.3.1 DCT for Feature Extraction 42
4.3.2 PCA for Feature Extraction 44
4.4 Classification 44
4.5 Pipelines of the Proposed Approaches 45
4.5.1 DCT Based Approaches 45
4.5.2 PCA Based Approaches 47
4.6 Discussion of the Proposed Approaches 50
4.6.1 Advantages and Drawbacks: Gabor filtering 50
4.6.2 Advantages and Drawbacks: DCT versus PCA Based Approaches for the PUF Scenario 51
5 Experimental Results **53**
5.1 Introduction 54
5.2 Experimental Setup 54
5.2.1 Evaluation Metrics 54
5.2.2 Parameter Tuning 55
5.2.2.A Speckle Pattern Image Resizing 55
5.2.2.B Gabor Kernel Parameters 57
5.3 Performance Results with Hamming Distance Based Classification 59
5.3.1 Testing with Dataset A 59
5.3.1.A DCT based algorithm 59
5.3.1.B PCA based algorithm 62
5.3.2 Testing with Dataset B 64
5.3.2.A DCT based algorithm 64
5.3.2.B PCA based algorithm 64
5.4 Performance Results with Machine Learning Classifiers 67
5.4.1 Testing with Dataset A 67
5.4.1.A DCT based algorithm 67
5.4.1.B PCA based algorithm 68
5.4.2 Testing with Dataset B 69
5.4.2.A DCT based algorithm 69
5.4.2.B PCA based algorithm 71
5.5 Discussion of the Results 71

6 Conclusions and Future Work 73

6.1 Conclusions 74

6.2 Future Work 75

List of Figures

1.1 Example of using PUFs for document authentication. 4

2.1 Schematic of a standard optical PUF based on an optical diffuser. Taken from: [1] 9

2.2 Operations performed when creating a digital identity for each PUF and its storage in a registry. Taken from: [2] 10

2.3 Schematic of perceptual hashing methods. 12

2.4 Illustration of the Scale-invariant Feature Transform (SIFT) algorithm operation. Taken from: [3] 13

2.5 Typical matching result by using Oriented FAST and rotated BRIEF (ORB) where the green lines are valid matched points and red circles indicate unmatched ones. Taken from [4] 14

2.6 An example of a decision threshold adapted to a specific dataset. Taken from [5] 23

2.7 An example of two hyperplanes maximally separating two classes. Taken from [6] 24

3.1 Illustration of the physical PUF system used to generate and acquire speckle pattern images. 26

3.2 Exemplification of the variance introduced by changing the positioning/orientation of the camera; (a) Camera; (b) Target; (c) PUF Token; (e) Beam Expander; 28

3.3 Sample image belonging to the first dataset A. 29

3.4 Sample image belonging to the second dataset, where black tape is utilized to aid in the cropping of the speckle patterns. 30

4.1 Diagram and description of each module in a perceptual hashing based authentication algorithm. 34

4.2 (a) Registration phase of token A. A speckle pattern image of token A is acquired and an hash is generated and stored in a database; (b) Authentication phase of token A. A speckle pattern image of token A is acquired, an hash is generated and compared to all other hashes in the database. Because it has been previously registered, the authentication is correct; (c) Authentication phase for an unregistered token. A speckle pattern image of token B is acquired, an hash is generated and compared to all other hashes in the database. Because this token has not yet been registered, the authentication process fails; 35

4.3 Pre-processing diagram all implementations. 37

4.4 Hough transform combined with Canny edge detection. 38

4.5 Bounding rectangle (in white) computed that encompasses the entire region of interest of the speckle pattern image. 39

4.6 Matched key points by ORB between images with different camera positions. 40

4.7 Diagram of the warping algorithm implemented, utilizing ORB and DCT. 41

4.8 Comparison of Gabor kernels with a diagonal orientation and an horizontal orientation. A larger variation can be seen between images filtered with an horizontal kernel: a) 45º Gabor Kernel with standard camera position; b) 45º Gabor Kernel with different camera position; c) 180º Gabor Kernel with standard camera position; d) 180ª Gabor Kernel with different camera position; 42

4.9 Example of a DCT matrix obtained from a 64x64 resized speckle pattern. Most of the energy tends to be compacted in the first coefficients of the matrix, i.e. the upper left corner. 43

4.10 Registration Phase - Approach 1: Pipeline for the registration of new tokens with a hamming distance based classification. 46

4.11 Authentication Phase - Approach 1: Pipeline for the authentication of tokens against registered tokens in a database with a hamming distance based classification. 46

4.12 Registration Phase - Approach 2: Pipeline for the registration of new tokens utilizing the DCT and a Classifier. 47

4.13 Authentication Phase - Approach 2: Pipeline for the authentication of tokens against other registered tokens in a database utilizing the DCT and a Classifier. 47

4.14 Registration Phase - Approach 3: Registration phase for the PCA based implementation using a hash hamming distance threshold as classification. Multiple token images are used to create a matrix on which PCA is applied and a final token image is used to generate an actual hash value for the PUF token. 48

4.15 Authentication Phase - Approach 3: Authentication phase for the PCA based implementation using a hash hamming distance threshold as classification. Data needs to be pulled from databases in two separate instances during the authentication process. 49

4.16 Registration Phase - Approach 4: Registration phase for the PCA based implementation using a classifier. Multiple token images are used to create a matrix on which PCA is applied and the generated hashes are used to train a classifier. 49

4.17 Authentication Phase - Approach 4: Authentication phase for the PCA based implementation using a classifier. 50

5.1 Impact of resizing the speckle pattern images to different sizes in terms of computational time taken and hamming distance difference between the same tokens and different tokens. 57

5.2 Impact of Gabor filter frequency in authentication performance, comparing the mean hamming distance difference. It is possible to see the benefit of considering only the imaginary part of the Gabor kernels. 58

5.3 Map of the normalized hamming distances, obtained with the DCT based algorithm on dataset A. Higher hamming distances are associated with a progressively redder colour. Lower hamming distances are associated with a green colour. 60

5.4 (a) Histogram obtained with the DCT based algorithm and Gabor filtering on dataset A; (b) Histogram obtained with the DCT based algorithm without Gabor filtering on dataset A; 61

5.5 (a) Histogram obtained with the PCA based algorithm and Gabor filtering on dataset A; (b) Histogram obtained with the PCA based algorithm without Gabor filtering on dataset A; 63

5.6 (a) Histogram obtained with the DCT based algorithm and Gabor filtering on dataset B; (b) Histogram obtained with the DCT based algorithm without Gabor filtering on dataset B; 65

5.7 (a) Histogram obtained with the PCA based algorithm and Gabor filtering on dataset B; (b) Histogram obtained with the PCA based algorithm without Gabor filtering on dataset B; 66

5.8 (a) Dataset B - Image (taken with standard camera position/orientation) after automatic cropping. Dimensions of the circled feature are approximately 150x245 pixels; (b) Dataset B - Image (taken with different camera position/orientation) after automatic cropping and warping. There is a slight variation in alignment and size of features. Dimensions of the circled feature are approximately 150x240 pixels; (c) Dataset A - Image (taken with standard camera position/orientation) after manual cropping. Dimensions of the circled feature are approximately 185x220 pixels; (d) Dataset A - Image (taken with different camera position/orientation) after manual cropping and automatic warping. There is a larger variation in alignment and size of features. Dimensions of the circled feature are approximately 155x200 pixels; 70

List of Tables

2.1 Overview and categorization of some widely used machine learning classifiers. 24

3.1 Overview of the obtained datasets. 29

4.1 Authentication approaches implemented and tested. 36

4.2 Overview and categorization of the machine learning classifiers tested in the implemented algorithms. 45

5.1 Confusion matrix for a binary classification problem. 54

5.2 Main characteristics of the Gabor kernel used in the pre-processing stage of the implemented algorithms. 59

5.3 Metrics obtained from the hamming distance histograms of the DCT based approach on dataset A. 60

5.4 Metrics obtained from the hamming distance histograms of the PCA based approach on dataset A. 62

5.5 Metrics obtained from the hamming distance histograms of the DCT based approach on dataset B. 64

5.6 Metrics obtained from the hamming distance histograms of the PCA based approach on dataset B. 64

5.7 Authentication results with dataset A and the DCT based approach. 68

5.8 Authentication results with dataset A and the PCA based approach. 69

5.9 Authentication results with dataset B and the DCT based approach. 70

5.10 Authentication results with dataset B and the PCA based approach. 71

Acronyms

Acc	Accuracy Rate
BRIEF	Binary Robust Independent Elementary Features
DoG	Difference of Gaussians
DCT	Discrete Cosine Transform
Ecc	Error Rate
FAST	Features from Accelerated Segment Test
GLCM	Gray Level Co-occurrence Matrix
HD	Hamming Distance
LLE	Locally Linear Embedding
NMF	Non-negative Matrix Factorisation
ORB	Oriented FAST and rotated BRIEF
PCA	Principal Component Analysis
POF	Polymer Optical Fiber
PVA	Polyvinyl Acetate
PUF	Physically Unclonable Function
Q-SVD	Quaternion Singular Value Decomposition
RPIVD	Ring partition and invariant vector distance
SIFT	Scale-invariant Feature Transform
SVD	Singular Value Decomposition
SVM	Support Vector Machines
SURF	Speeded up Robust Features

1

Introduction

Contents

1.1 Motivation . . . 2

1.2 Core Concepts . . . 2

1.3 Problem Formulation . . . 4

1.4 Objectives . . . 4

1.5 Contributions . . . 5

1.6 Dissertation Outline . . . 5

1.1 Motivation

The development of authentication methods is an increasingly important topic, which is also the focus of a wide discussion in the public sphere. From maintaining security in communications to preventing counterfeiting of documents, authentication has a necessary role in our society and everyday life. International counterfeiting has negatively impacted a growing number of industries. The expansion of counterfeiting into diverse areas means that damages from counterfeiting are measured not only in loss of profit, but also in loss of jobs and even lives. Physically Unclonable Functions (PUFs) [7] present an attractive solution to these exceedingly important problems. PUFs are physical devices characterized by having uniquely random and practically impossible to reproduce structures, which can therefore be used for authentication purposes.

1.2 Core Concepts

1.2.1 What are Optical PUFs?

(PUFs) are a type of cryptographic device that leverages the inherent randomness of certain objects as an authentication token, which can be used as a fundamental layer in the design of security systems. PUFs are characterized by challenge-response pairs, in which a certain input to the system generates an uniquely related output. This behaviour can be interpreted as a one-way function, which is why these devices are referred to as PUFs. The usage of physically unclonable functions has several advantages [8]:

- The power consumption of these devices is usually lower and the hardware is easier and cheaper to fabricate, when compared to other cryptographic digital systems. In fact, because the secret is obtained from an intrinsic physical property of the device/material used, no power is required for storing the characteristics from which the key will be derived.
- If a third party were to access the secret, that would have to be done while the system supplying the PUF would be active.
- Keys generated from PUFs are much harder to replicate, given that the key itself is associated to a physical property of the system, which is truly random.

PUFs exploit the inherent randomness introduced during the manufacturing of an object to produce a unique ’fingerprint’. They are the physical equivalent of one-way mathematical transformations that, upon external excitation, can generate irreversible responses. PUFs can be categorized in two classes [8]: i) Weak PUFs; and ii) Strong PUFS. They differ in the amount of challenge/response pairs each one supports. Challenge/response pairs refer to the input/output relations each PUF has. In other words, for each input or challenge given to the PUF, a certain output or response occurs. Weak PUFs have few challenge/response pairs (in most cases only one) and are mostly used for secret key storage. Strong PUFs have enough challenge/response pairs that within a limited time frame one can not feasibly determine them all. This type of PUF is mostly used for authentication purposes.

There are many types of PUFs that exploit the randomness and features of different objects. This thesis will focus on Optical PUFs. These consist of an inhomogeneous material that contains inherent imperfections and random structures that, when challenged with a coherent light source, creates an unique speckle image response. This speckle pattern, obtained from the scattering of light through the object, can be used for authentication purposes. Image based PUFs are especially useful for anti-counterfeiting purposes. Not only can an object be used to generate a cryptographic key, but this key can also be used to authenticate the object itself.

Image-based PUFs are implementations of strong PUFs, so they are attractive for authentication purposes. The majority of Optical PUFs consist of three primary components: i) a light source; ii) a token produced with a physical scattering medium (an optical diffuser); and iii) an imaging device.

The light source illuminates the physical scattering medium that sits stationary along the light source path. This creates a speckle pattern that is focused by a lens and projected in a target. This pattern is then captured by the imaging device.

1.2.2 What is Perceptual Hashing?

Images taken from PUF tokens need to be aligned and normalised to be perceptually analysed and matched for authentication purposes. Perceptual image hashing is required for this process, notably as a tool to help overcome possible noise and distortions between token images. Perceptual hashing consists of hash functions that produce similar hash values for similar input images. This can be used for authentication purposes, like the authentication of optical PUF tokens. Perceptual hashing is the tool that allows the translation of the unique ”fingerprint” obtained from a PUF token.

Perceptual hashing, in general, consists of three separate stages [9]: (i) pre-processing; (ii) perceptual feature extraction; and (iii) post-processing. The pre-processing and post-processing stages have the purpose of getting a more robust hash, but are not mandatory.

Pre-processing typically includes filtering operations, with the purpose of removing unwanted variability, such as low-pass filtering, Gaussian blurring and order statistic filtering.

Perceptual feature extraction aims to achieve a robust operation, ensuring the image hashing algorithm’s invariance to tampering and content-preserving attacks, as well as the ability to provide similar hash values for similar input images, despite the distortions that may affect them [10]. Robustness largely depends on the features extracted from the image, making the feature extraction stage crucial in the success of the image hashing process.

Post-processing can consist of operations such as quantization or compactation of features.

1.3 Problem Formulation

A common sheet of paper, for instance used as the support of an important document, can be used as an optical diffuser, or token, since the microscopic structures of the paper diffuse light in random directions. This token can be used to generate an unique speckle pattern when challenged with a coherent light. This would be an example of a PUF system used for authentication purposes, where the sheet of paper acts as the authentication token. As illustrated in Figure 1.1, the document can be authenticated by generating a hash key from the perceptual analysis of the speckle image, obtained by the PUF system, and matching it to the hash key value stored in a database.

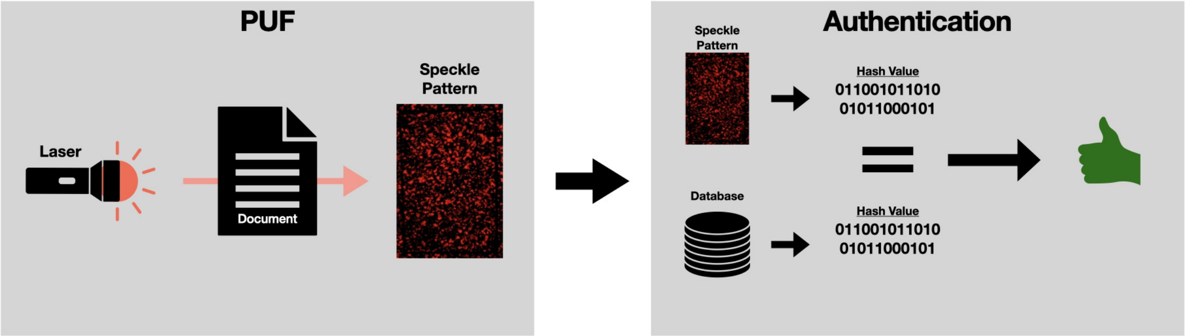


Figure 1.1: Example of using PUFs for document authentication.

However, a problem arises due to the variability in challenge/response pairs. In fact, a same challenge can generate different responses depending on the time instant the image was recorded at. The same light probing the document paper token, previously described, can generate slightly different speckle patterns.

1.4 Objectives

The main objective of this thesis is to deal with the variations between different acquisitions from the same token by developing a perceptual hash algorithm that is robust against intra-variability.

The goal is to test multiple perceptual hashing methods to understand what would holistically work best in a PUF authentication system implementation. In a first step, the speckle pattern images obtained will be processed, namely, cropped, scaled and corrected in illumination variations. This will yield images with standard geometry that can be compared to a reference. After this, perceptual hash algorithms can be applied to the normalised images, to produce responses that are not affected by intra-variability. So, in a given PUF, different acquisitions of the same challenge taken at different temporal instants will yield the same responses. This will allow accurate authentication of PUF tokens.

The main methods for feature extraction utilized are the Discrete Cosine Transform (DCT) and Principal Component Analysis (PCA). Several other techniques are also explored and utilized for the pre-processing of speckle pattern images, like Oriented FAST and rotated BRIEF (ORB) and the Gabor Transform.

The physical PUF system utilized was developed in [11], having been employed to acquire the datasets used in this work.

1.5 Contributions

This dissertation shows how a PUF system utilizing tracing paper tokens can be employed for authentication purposes, despite the intra-variability between acquired speckle pattern images. The main contributions of this present work are the following:

- An objective comparison between the application of the DCT and PCA for feature extraction in speckle pattern images obtained from tracing paper tokens.

- An objective comparison between the usage of a normalized hamming distance based classification or machine learning classifiers for the classification of speckle pattern images obtained from tracing paper tokens.

- Fully implemented perceptual hashing based authentication algorithms, both data dependent and independent, with proven efficacy in authenticating the considered PUF system, utilizing tracing paper tokens.

- An automatic cropping and warping scheme that can be used to possibly use a smartphone as the imaging device of this specific PUF system. This scheme consists on the usage of non-reflective black tape to isolate the region of interest in the target of the PUF system, as well as the usage of image processing tools, namely, thresholding, finding contours based on pixel intensity and ORB.

1.6 Dissertation Outline

This dissertation will is composed of six chapters, whose summarized contents are the following:

- **Introduction**: In this chapter, the motivation for this dissertation is presented. Some core concepts of this thesis are explained namely Optical PUFs and perceptual hashing. The base problem this work aims to solve is also described.

- **Review of Authentication Techniques using Optical Physical Unclonable Functions**: In this chapter, an overview of the concepts introduced in the previous chapter is presented. The state of the art for optical PUF devices and perceptual hashing is presented. A review introducing image and data processing methods, such as the DCT and PCA, as well as classification methods, is also included.

- **The PUF System Used**: In this chapter, the physical PUF system used is described. The characteristics of the acquired datasets are also explained.

- **Proposed Approach for Authenticating a PUF**: In this chapter, the proposed approach for authenticating the PUF tokens, despite intra-variability, is explained. An introduction outlining the desired features in this type of algorithms is also included. A brief discussion of the benefits of each proposed approach concludes the chapter.

- **Experimental Results**: In this chapter, the experimental results obtained from the proposed algorithms are presented and discussed. The parameter tuning done for each algorithm is also described.

- **Conclusion and Future Work**: This chapter presents the conclusions of this work, as well as possible approaches for future work.

2

Review of Authentication Techniques using Optical Physical Unclonable Functions

Contents

2.1 Overview of Optical PUFs 8

2.2 Overview of Perceptual Hashing Methods 11

2.3 Gabor Transform for Speckle Pattern Analysis 17

2.4 DCT Based Hashing Solutions 19

2.5 PCA Based Hashing Solutions 20

2.6 Classification Algorithms 22

2.1 Overview of Optical PUFs

Even thought the concept of a PUF has been known for some time, it has gained traction mostly in the last decade. Several implementations of PUF designs that have been documented and used with success. As summarised in [1], the ”optical PUF’s physical mechanism relies on the random interference pattern (speckle) created when a laser beam propagates through an inhomogeous material”.

Once a speckle pattern is captured by the imaging device, we would like to use it for image authentication purposes. However, this acquisition process is expected to be affected by intra-variabily, as when making different acquisitions in very similar conditions, the resulting pattern is expected to show slight changes. Perceptual hashing can be employed to overcome this intra-variability between speckle patterns of the same challenge, which would otherwise prevent it from being used for authentication purposes. Features that are invariant to distortions and noise need to be extracted from the image, through perceptual hashing. The hash value is then generated from these features, allowing the system to overcome the expected intra-variability.

Optical PUF’s were first proposed by Pappu *et al.* [7]. In their implementation the light source can move in a 3D space pointing to a stationary scattering medium. The challenge is the position of the laser beam and the response is the speckle pattern recorded. The scattering medium considered was composed of a large number of randomly positioned silica spheres embedded in hardened epoxy. Since then, many iterations of this fundamental concept have been proposed.

A cost effective and relevant example of a PUF implementation is the called PaperSpeckle [12], which consists of a portable paper fingerprinting system that can identify and authenticate paper. The main use for this is to prevent document forgery and counterfeiting, which is a very relevant problem around the world. In PaperSpeckle the paper works as the scattering medium, having random and hard to replicate structures that create unique speckle images. PaperSpeckle showed that it is possible to extract repeatable speckle patterns from microscopic regions of paper, with just paper, pen and a microscope. These speckle patterns can then be turned into unique fingerprints associated with the document. This is a robust PUF implementation.

Robustness is a characteristic of PUFs that is defined as the probability to reconstruct the output of a PUF system that has been produced in setup mode [13]. In other words, a robust system is one that can reproduce the same response to a certain challenge multiple times, overcoming any intra-variability that may alter results. This is because the operations of the system must be time-invariant. PaperSpeckle developed a fingerprinting algorithm that makes their system robust across adverse environmental conditions. It consists of two operations, namely computing Gabor transforms combined with Singular Value Decomposition (SVD) of a large matrix, which will be discussed later on.

Another relevant example is the implementation proposed by Mesaritakis *et al.* [1], where an optical waveguide is used. This waveguide is a Polymer Optical Fiber (POF) that has inherent random physical

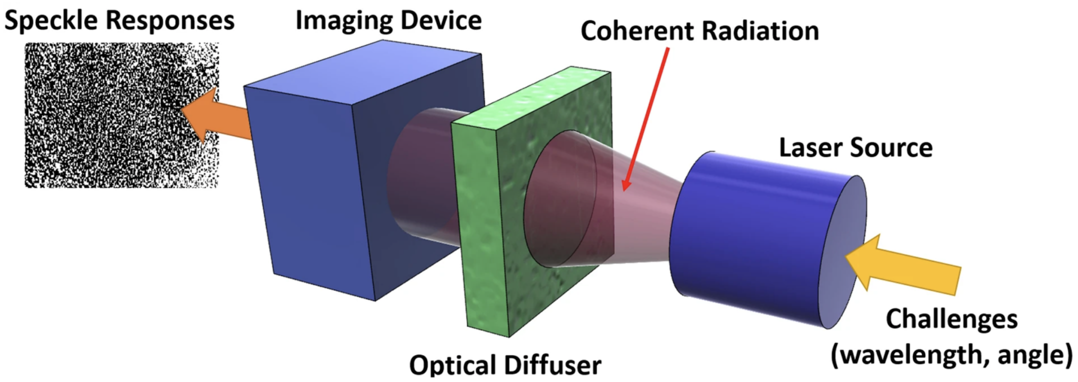


Figure 2.1: Schematic of a standard optical PUF based on an optical diffuser. Taken from: [1]

features at the extremity of the fiber line (it's facet). These defects may come from scratches, noise-driven mechanical friction and irregularities from manufacturing. Given this, the facet can be used as an ordinary optical diffuser. The light source is placed at the input of the line. The angle of the laser and the wavelength of the light are the challenges. A schematic of this system is shown in Figure 2.1. It is noted that the POF may also have irregularities, like scratches and manufacturing defects, along its sides. This may alter the response in ways that are unaccounted for, which is why the length of the optical fiber should be kept to a minimum. The ending facet has the same characteristics as the input one, working as a second optical diffuser. The imaging device is set at this facet of the POF. Both of the facets unique physical characteristics function as an authentication token in this implementation. In order to combat intra-variability and generate time-invariant responses, as well as remove noise originated from unwanted defects inherent to the system, the raw output captured by the imaging device of the PUF is processed through fuzzy extractor techniques as well as hashing procedures, like the Gabor binary method or the random binary method. These image processing methods are discussed later on.

Arppe-Tabbara *et al.* [2] also proposed and validated a versatile authentication system based on optical PUFs. It's main application lies in preventing counterfeiting of products, but can be applied elsewhere. The system is based around tagging products with PUFs that can later be authenticated as official. When the product is made, the manufacturer inserts into it a tamperproof tag, which consists of a unique scattering of particles. The end-user can always validate whether the product is genuine by authenticating it using the tag. Arppe-Tabbara *et al.* show that tags can be created from an array of different carrier materials, such as epoxy and Polyvinyl Acetate (PVA), different inks and regular printing technologies. The PUF's imaging device is a smartphone fitted with a macro lens, which increases the versatility of the system. An image of TiO2 in PVA is incorporated into a QR code, printed in office

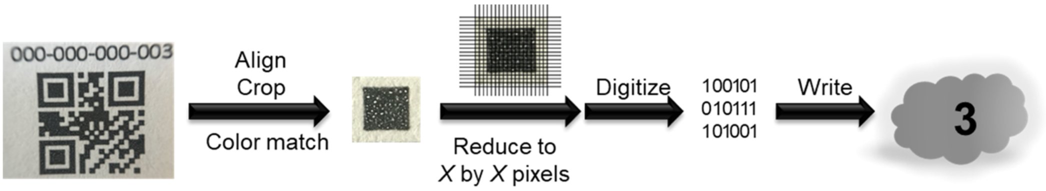


Figure 2.2: Operations performed when creating a digital identity for each PUF and its storage in a registry. Taken from: [2]

quality paper and laminated, for validation purposes. The image, in this implementation, is taken with a smartphone camera that is reasonably recent, and converted into an 8 bit grayscale PNG file. The image is then cropped, rotated, scaled and corrected for aberrations. This results in an image with a standard geometry that can be matched to the reference tag. This morphology transformation is possible due to the application of a non-degenerate affine transformation from the four corners of any quadrilateral to any other quadrilateral. By identifying the four corners of the QR code in the image taken, this transformation can be computed. This is a robust process as corner identification is reliable and works well in good lighting conditions. In the end, the inverse of the affine transformation is applied to the image which makes it match the reference in terms of scale and morphology. Non-linear filters are also applied to the image with the purpose of increasing the contrast and removing some noise. Matching isn't done by comparing both images pixel-by-pixel. Instead, key features are compared between the acquired image and the reference, which increases robustness to noise and slight mismatches in morphology. Low-dimensional feature representation falls short in representing the qualities of the tags, which makes them difficult to counterfeit. Using the method described, with images of size 200 x 200 pixels, ensures that each PUF's acquired image can be recognised as unique. False positives where observed with resolutions lower than 100 x 100 pixels, whereas higher resolutions resulted in larger files and slower matching.

A schematic representation of the operations performed for creating a digital entry for each PUF is shown in Figure 2.2. In a first step the PUF incorporated into the printed QR code is selected, aligned and color-corrected. It is then digitised and stored in a registry. This system is relevant given that the PUF's imaging device is a smartphone with which a person takes the image. This introduces human error, considerable variations in lighting conditions and angle of image capture, which in turn increases intra-variability. However, several morphology transformations and feature matching were used to combat that.

2.2 Overview of Perceptual Hashing Methods

There are several methods of perceptual hashing that have been proposed over the last decades, each with different approaches, and exploring distinct concepts. A recent publication by Du *et al.* [10] describes the results of a survey where they compare and categorise existing perceptual hashing methods. They propose that perceptual hashing can be grouped into five main categories:

- **Invariant feature transform methods** - Methods in this category explore a representation of the input image in a transformed domain. They generally have the advantage of being robust against certain types of distortion and noise attacks. Wavelet and Quaternion based hash functions fall under this category;
- **Local feature methods** - Methods in this category leverage local features that are invariant under content preserving attacks. Feature-point based hash functions fall under this category, including examples such as SIFT, Speeded up Robust Features (SURF) and ORB;
- **Dimension reduction methods** - Methods in this category make use of dimension reduction techniques. SVD based hash functions fall under this category;
- **Statistic feature methods** - Methods in this category take advantage of image statistics for the calculation of the hash value. Ring partition and invariant vector distance (RPIVD) and histogram based hash functions fall under this category;
- **Learning methods** - Methods in this category take advantage of efficient learning algorithms that can be implemented to generate hash values based on parameters learned from the training of data;

Categorising perceptual hashing methods in this way makes sense since it encompasses most of the existing methods. A simplified diagram of this categorisation is proposed in Figure 2.3. A few examples of algorithms belonging to each category are presented below for illustration purposes.

Invariant feature transform methods make use of frequency coefficients in a transform domain. The input image is transformed into the frequency domain so that it's features depend on the image's frequency coefficients. These can be used to extract robust features from the image, making this type of methods more robust to certain attacks and distortions. Fourier transform, Discrete Cosine transform and Wavelet transform are a few of the transformations that can be considered. Different types of transforms have different types of properties that the hashing approaches can exploit. Most of the invariant feature transform based methods are robust against one or a few types of attacks. There is not, as of yet, an universally robust method that can be applied to every scenario [10].

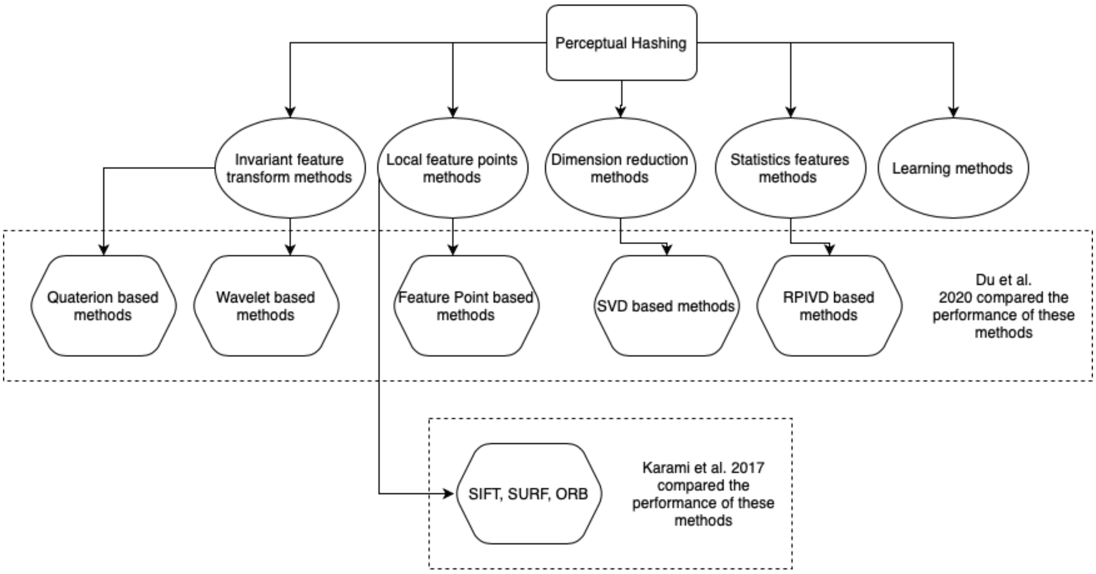

Perceptual Hashing
Invariant feature transform methods
Local feature points methods
Dimension reduction methods
Statistics features methods
Learning methods
Quaterion based methods
Wavelet based methods
Feature Point based methods
SVD based methods
RPIVD based methods
Du et al. 2020 compared the performance of these methods
SIFT, SURF, ORB
Karami et al. 2017 compared the performance of these methods

Figure 2.3: Schematic of perceptual hashing methods.

Lei *et al.* proposed a perceptual hashing algorithm based on the Radon transform [14]. After the transform was performed on the image the moment features in the projection space were calculated. To resist rotation, the discrete Fourier transform was also applied on the moment features. The image hash bits are finally obtained from the quantized magnitude of the significant coefficients from the discrete Fourier transform. This algorithm was mainly focused on robustness. Another image hashing algorithm that makes use of the Radon transform was proposed by Nguyen *et al.* [15]. Due to this transform's inherent invariance, the algorithm was robust against noise addition, JPEG compression, filtering and geometrical distortions. A mechanism to make the hashing process dependent on a cryptographic key was also implemented.

An example of a Quaternion-based image hashing algorithm is proposed by Yan *et al.*, allowing a better identification of localised attacks [16]. It used the Quaternion Fourier-Mellin transform to obtain a geometric hash and the Quaternion Fourier transform to obtain a image feature hash. To improve the robustness against localised tampering, an adaptive algorithm was also proposed to improve the detection accuracy. A Quaternion is an hyper-complex number with four parts (three imaginary and one real part), which can be used to represent color images. The Quaternion Fourier-Mellin is useful for eliminating the influence of geometric distortions. Moreover, the Quaternion Fourier transform is used to calculate coefficients, a selection of which being selected as the image feature hash used to detect tampering.

Local feature methods are centred around the identification of corners, edges, salient regions and so on. These are known as local feature patterns, which should be generally invariant and robust, since image hashes should be the same between equal images even when one is tampered with. These types of methods are widely used for robust matching and object detection.

Feature point detection is important when it comes to robust feature extraction. Scale-invariant Feature Transform (SIFT) is a feature detection algorithm commonly used to extract key points from images and provide a feature description of the image [17]. It operates in monochrome images to mitigate the impact of varying lighting conditions. In order to perform reliable recognition of images, these points are usually in high-contrast zones of the image, for example object edges, which are less affected by different lighting, scaling or the addition of noise in the image. These features shouldn't change their relative distance to one another at different image acquisition moments, as that would mean that the object, which they describe, had changed. SIFT key points are extracted from a series of differently scaled and smoothed versions of the input image, to which the difference of Gaussian's function (DoG) is applied. The maxima and minima of these results are the SIFT key points, which are obtained by applying a high-pass filter on the extrema. Points of low contrast are discarded. To ensure a more stable and robust matching, gradient, orientation and positions of each feature are determined [3]. A summarised illustration of the SIFT algorithm operation is presented in Figure 2.4.

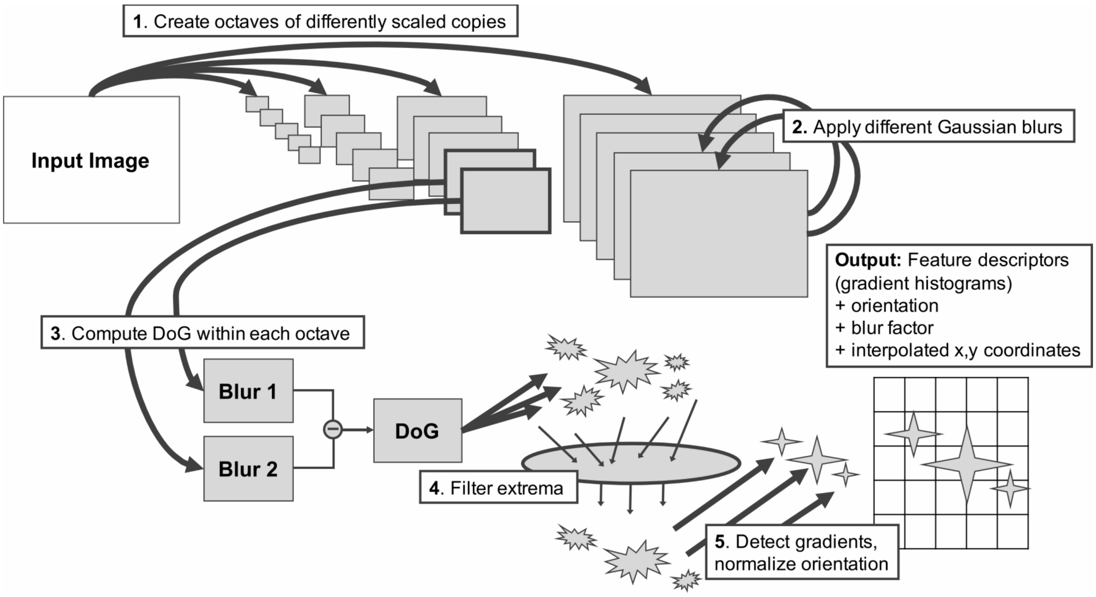

1. Create octaves of differently scaled copies
Input Image
2. Apply different Gaussian blurs
Output: Feature descriptors
(gradient histograms)
+ orientation
+ blur factor
+ interpolated x,y coordinates
3. Compute DoG within each octave
Blur 1
Blur 2
DoG
4. Filter extrema
5. Detect gradients,
normalize orientation

Figure 2.4: Illustration of the SIFT algorithm operation. Taken from: [3]

An example of an image hashing approach in this category, using SIFT, is proposed by Lv *et al.*, consisting of a hashing approach using robust local feature points [18]. They incorporated the Harris criterion with SIFT to detect the most stable and robust feature points, which are less likely to change with image processing attacks. To make this approach more robust, shape context was introduced into the hash generation, creating a representation of the structure of the image, with the geometric distribution of the detected feature points being embedded into the hash. The proposed shape-context-based hashes could also be used and implemented to detect localised content tampering. This was tested with a database with over 107 000 images that were tampered with a wide range of distortions and content-preserving attacks, showing the approach is robust, having an increase in performance under geometric attacks such as rotation and brightness changes and comparable identification outcomes under more classical distortions like blurring, noise addition and compression when compared to previously proposed approaches.

Speeded up Robust Features (SURF) was proposed by Bay *et al.* as an approximation of SIFT with the intention of outperforming it in terms of computational speed [19]. SURF also relies on Gaussian scale space analysis of images. It uses a Hessian matrix-based measure for the detector and a distribution-based descriptor. Features extracted by SURF are invariant to scale and rotation but have limited affine invariance. One main advantage of SURF over SIFT is the lower computational cost.

Oriented FAST and Rotated BRIEF (ORB) was proposed by Rublee *et al.*, using a modified Features from Accelerated Segment Test (FAST) detection as well as direction-normalised Binary Robust Independent Elementary Features (BRIEF) based descriptors, which are rotation invariant and resistant to noise [4]. FAST finds corner keypoints using an augmented pyramid scheme for scale and a Harris corner score to filter out poor quality points. A modified version of BRIEF descriptor was employed to handle rotation attacks. ORB features are therefore invariant to rotation, scale and to limited affine changes. An example of a typical matching result obtained by using ORB is shown in Figure 2.5.

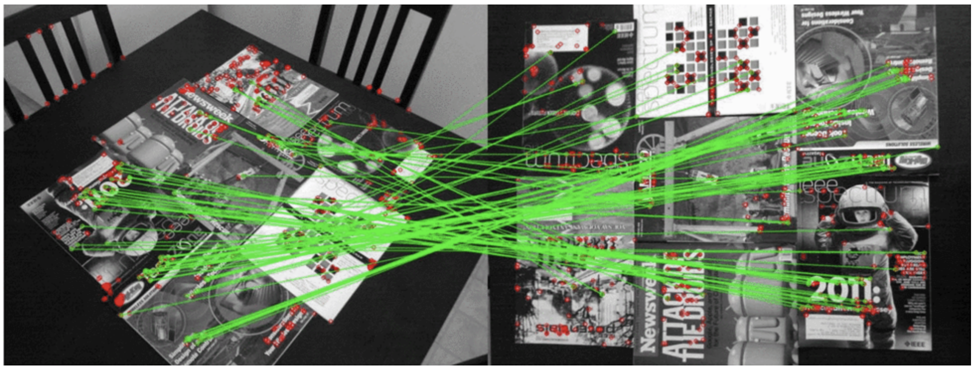


Figure 2.5: Typical matching result by using ORB where the green lines are valid matched points and red circles indicate unmatched ones. Taken from [4]

Local feature methods are generally robust, especially against geometric transforms like rotation attacks. This is an intrinsic advantage of local features based algorithms. However, there are a also few limitations. One is that, given the variety in number of local feature points from image to image, the hash size will vary and depend on the image size and texture. Another limitation, and perhaps more relevant in the scope of this work, is that images with the same feature points may not have the same content. This raises concerns about the perceptual hash algorithm's security.

Dimension reduction methods capture the essential features that are invariant under many image processing attacks. This type of methods are especially robust against geometric attacks, like rotations. There are several dimension reduction techniques like Singular Value Decomposition (SVD), Locally Linear Embedding (LLE), or Non-negative Matrix Factorisation (NMF), among others.

SVD is important because when applied to an image, the selected components capture the essence of the geometric information and the semi-global features of the image. This was observed by Kozat *et al.*, whose experiments showed that these components of the SVD are mostly invariant under content-preserving attacks [20]. SVD is widely applied in many perceptual hashing algorithms.

Ghouti *et al.* proposed a hashing algorithm aimed at coloured images, which is something that is widely ignored in perceptual hashing as most algorithms are designed for gray-level images [21]. They used Quaternion Singular Value Decomposition (Q-SVD), where quaternions represent extensions of the 2D complex domain space to the 3D and 4D spaces. They concluded that Q-SVD provided the best low-rank approximation of color images. This can be used to generate robust hash codes, that consist of Q-SVD singular vectors, in an efficient manner. The proposed algorithm yielded the lowest misclassification rate when compared to other SVD-based hashing algorithms. It also had better security and robustness against standard attacks.

Statistics feature methods take advantage of the assumption that the relative relationship between pixels remains the same after the image is tampered with some distortion attack. In these methods, the hash value is generated from estimated statistical properties of the image, including histogram, mean, variance, or moments of image blocks, amongst others.

Huang *et al.* proposed a hashing method targeting a good balance between robustness and discrimination [22]. The proposed hashing method uses two types of complementary features: (i) global statistical texture features; and ii) local invariant frequency features, using the DCT. Texture is a characteristic inherent to all image surfaces, related for instance with the image gray-level statistics, or its spatial distribution and structure, and the Gray Level Co-occurrence Matrix (GLCM) is used to describe the texture of the image. GLCM calculates how many times pairs of pixels with specific values and spatial relationship appear in the image, resulting in a co-occurrence matrix. From this matrix a set of statistical features can be extracted, such as energy, contrast, correlation and homogeneity. Local feature based methods are generally limited because of their locality aspect. This paper effectively combined these coefficient features via DCT with global image texture, achieving a better balance between robustness and discrimination.

Statistic features methods, using statistical characteristics of the image, are generally robust against distortions like noise and blurring. This means that they are largely invariant to tampering with content preserving attacks to the image. However, this comes at the cost of distinctiveness, a very important concern for the uniqueness of image hashes, which is relevant in the scope of PUF implementation.

Learning methods utilise training with data to provide a better binary representation of the image. It is important to note that, although promising and avidly studied at the moment, there are still few studies that have focused on applying these perceptual hashing methods to image authentication, which is something that is important in the scope of the present work. Nevertheless, these types of algorithms can produce high quality hashing, possibly with higher complexity, thus taking more processing time, than data independent methods.

An analysis of how perceptual hashing is used in the actual implementation of optical PUFs is important. Different implementations work with different types of speckle patterns and need to overcome different problems in order to make the system robust against varied operational settings. Speckle patterns with no apparent or salient features are more challenging to implement, especially when the system is meant to be used for authentication purposes. Good authentication results must be achieved with good discrimination. Studies where the speckle pattern of the optical PUF was obtained from blank paper or an optical fiber are the most relevant and therefore must be reviewed.

A general overview of an optical waveguide based PUF proposed by Mesaritakis *et al.* [1] was discussed earlier. The speckle pattern is created by the fiber's facets which exhibit random defects. In order to correct detrimental experimental noise and generate time invariant binary strings, the raw output of the PUF is first processed through fuzzy extractor techniques. These techniques use fuzzy set theory [23] to achieve more accurate filtering of the image. Fuzzy sets can be used for intensity transformations, like contrast enhancement, which in turn makes features more apparent. When compared to, for example, histogram equalisation, there is a considerable improvement in tonality, having a higher level of detail. Fuzzy sets can also be used for spatial filtering, which can be utilised for boundary extraction between regions in an image. This is useful for processing speckle patterns obtained from an optical waveguide, for example, where the pattern is only a small portion of the image and the rest should not be considered for the generation of the hash value. These fuzzy extractor techniques are combined with hashing approaches, like the random binary method or the Gabor binary method. In short, for the random binary method, random pixels are chosen and sampled from the previously processed image. These are multiplied with a matrix containing entries randomly chosen from a normal distribution and then quantized using the mean intensity of the image. If instead Gabor hashing is used, Gabor coefficients are chosen, instead of pixels from the raw image, and post processing that involves Gabor filters is applied. These metodologies are integrated in a general security framework.

Gabor based hashing offers high robustness and discrimination, so it is attractive for feature extraction. It makes use of the Gabor transform and can effectively distinguish different patterns, so it has been successfully applied in various applications [24], such as texture analysis, which is important in this context, as we will see in the next section.

2.3 Gabor Transform for Speckle Pattern Analysis

Speckle patterns, in general, have very few predominant features. To improve their perceptual analysis, the Gabor transform can be useful. Theoretically, Gabor filtering is closely related to the primary visual cortex [25], in terms of perceiving texture and detecting edges. This makes it interesting for computer vision purposes. It has been used in various applications like image coding, enhancement and compression. In the realm of optical PUFs, the Gabor transform is often used in speckle pattern authentication systems.

A Gabor filter is a linear filter. In the spatial domain, it is derived from the modulation between a Gaussian kernel and a sinusoidal plane wave. Because of this, the parameters of a 2-D Gabor function include wavelength λ, orientation θ, phase offset ϕ, standard deviation of the Gaussian envelope σ as well as aspect ratio $\gamma < 1$, and bandwidth b . The gabor kernel can be defined as:

$$g(x, y; \lambda, \theta, \phi, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi\frac{x'}{\lambda} + \phi\right) \tag{2.1}$$

where

$$x' = x\cos\theta + y\sin\theta,$$

$$y' = -x\sin\theta + y\cos\theta.$$

Due to the inherent orientation of a Gabor kernel, certain features of the unfiltered image that are aligned with that orientation will become more prevalent in the filtered image. This means that different orientations will be useful for different image types or purposes. The same goes for kernels with different λ or kernels applied at specific locations. This allows Gabor filters to select specific features from images. Different combinations of these parameters must be tested to understand which kernel, or kernels, better accentuate and discriminate the intended features of the images. Usually, for these reasons, a Gabor kernel bank is utilized. A Gabor kernel bank consists of multiple Gabor kernels with various distinct λ and θ. In section we will see how different orientations and frequencies of Gabor kernels affect the speckle patterns of the PUF system.

It is important to note that Gabor filtering is not intrinsically a feature extraction method, even though it can be. It is mainly utilized as a pre-processing tool to accent features and improve discrimination and robustness when combined with other methods, like the ones discussed earlier.

In the first optical PUF implementation, proposed by Pappu *et al.* [7], Gabor filtering is utilized for the generation of hash values. In short, speckle patterns are down-sampled by applying Gabor filters, thresholding at zero and scaled down. This method can be represented as a multi-resolution pyramid, in which a level of the pyramid refers to one iteration of filtering and sub-sampling of the image. At each level of the pyramid, four Gabor kernels are utilized independently in four copies of the image. The orientations of these four filters are $\theta = 0, 45, 90$ and 135. However, it is important to note that, in this implementation, only the two diagonal orientations are used. This is mainly due to the fact that the values of the Gabor transform along the diagonals are much less sensitive to small changes in the horizontal or vertical positioning of the token. After Gabor filtering, the filtered images are scaled down by a factor of their measurements divided by the number of the level they are on.

Let's assume, as an example, that at level 4 the four images have been sub-sampled to 30x40 pixels. Because only the two diagonal orientations are used, the two images can be treated as a long string of 2400 bits. A subset of bits can be chosen as the fingerprint of the PUF token. Note that the images are binary due to thresholding at zero. The hamming distances at level 4 proved to be small when comparing hashes from the same token and large when comparing hashes from different tokens.

The work carried out by Pappu *et al.* not only introduced optical PUFs, but established the Gabor transform as a capable tool for the analysis of these types of images. In the implementation of Paper-Speckle [12], which was presented in section **??**, the Gabor transform is also used. This implementation consists of a PUF where the scattering medium is paper, which is highly related to the present work. Here, the same hashing method as [7] is used. It is important to note that in both papers [12] , [7] only the imaginary part of the Gabor wavelet is used to extract bits. By doing this, any illumination defects, contrast variations or poor focus that are present in the speckle image are eliminated. This improves the robustness of the system. After images are sub-sampled, the resulting bit sequence is converted into a binary matrix, on which SVD is performed. The resulting singular values are used as the fingerprint of the speckle pattern.

One potential caveat of conventional Gabor filtering is that the filter's response to the input image is sensitive to its orientation. This can be beneficial in some applications, but not in others. PUF systems that utilize optical waveguides as scattering medium are more prone to rotation distortions, reducing the effectiveness of Gabor filtering. However, the Gabor filter can be designed to be independent to orientation, like the one proposed by Li *et al.* [24]. The inherent orientation of Gabor kernels can also be utilized to add another layer of security to hashing algorithms, like in [26]. By dividing an image in blocks, each one can be filtered independently, with various kernels of distinct orientations. A tent map is utilized to generate different directions that can be replicated by saving the initial tent map values as a helper secret key. Each block is filtered with a different, apparently random, direction, making the final hash more secure.

2.4 DCT Based Hashing Solutions

The Discrete Cosine Transform [27] is widely used in image processing. It has notable applications in image compression, in methods like JPEG and HEIF, and perceptual hashing. Its image processing applications rely on the premise that pixels in an image exhibit a certain level of correlation with their neighbouring pixels.

Similarly to the Discrete Fourier Transform, the DCT expresses a signal in terms of a sum of cosine functions with different frequencies and amplitudes. These frequency coefficients in the transformed domain can translate image features that are robust against distortions and attacks. It uses only the real part of the Fourier Transform, so it is computationally less taxing when compared to the Discrete Fourier Transform and even the Fast Fourier Transform. Different variations of the DCT exist, but the most common one is the type-II DCT.

The DCT is usually defined as

$$X[n] = \sqrt{\frac{2}{N}} \times \sum_{m=0}^{N-1} x[m] \times \cos(\frac{(2m+1) * n\pi}{2N}) \tag{2.2}$$

$$,(n = 0, ..., N-1).$$

Which can also be expressed as

$$X[n] = \sum_{m=0}^{N-1} c[n,m] \times x[m] \tag{2.3}$$

$$,(m, n = 0, ..., N-1)$$

where $c[n,m]$ is the DCT matrix and n and m are the row and column numbers, respectively. The DCT matrix can be computed in advance, so equation 2.4 is useful for when the DCT is implemented in code.

To use the DCT on two-dimensional signals, i.e. images, it needs to be extended to a two-dimensional space. Because the DCT is a separable linear transformation, the 2-D DCT is a direct extension of the one-dimensional case. It is equivalent to a 1-D DCT applied along a single dimension and another 1-D DCT in the other dimension.

In the DCT matrix, the frequency of both vertical and horizontal components of the coefficients increase with the line and column indexes. The first entry in the matrix is referred to as the DC coefficient. Because most of the image information tends to be concentrated in a few low-frequency components of the DCT, the low-frequency coefficients are generally robust against noise and variability. Thus, in both image compression and image hashing, this property is used to represent the most perceptually significant features.

2.5 PCA Based Hashing Solutions

High dimensionality is a reoccurring challenge in several fields, including image matching and authentication. Images typically contain a large number of measurements (in the form of pixels) [6]. This makes the task of finding matches in image datasets very time intensive and computationally taxing. However, as it was shown earlier with the application of the DCT, images can be compressed. A lot of their relevant information can be represented in a much lower-dimensional sub-space.

Principal Component Analyses (PCA) is a dimensionality reduction technique that is based around the SVD. It is of great significance in the realm of probability and statistics, but also very commonly used for data analysis and machine learning applications. It was first introduced over a hundred years ago, so it is a very well established technique with a lot of theory behind it.

To introduce PCA theory, the SVD needs to be discussed. Singular Value Decomposition (SVD) allows for the determination of a low-dimensional approximation of the high-dimensional data, with regard to dominant patterns. The SVD provides a hierarchical representation of the data, analysing dominant correlations, in terms of a new coordinate system.

In the perceptual hashing context, it is the optimal low-rank approximation of a matrix X that consists of several concatenated columns. Each column consists of an image that is reshaped as a vector, where each element corresponds to a pixel. This matrix can be decomposed in the following way:

$$X = U\Sigma V^T = \begin{bmatrix} | & | & & | \\ u_1 & u_2 & ... & u_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & ... & \\ & & & \sigma_m \end{bmatrix} \begin{bmatrix} | & | & & | \\ v_1 & v_2 & ... & v_m \\ | & | & & | \end{bmatrix}^T \tag{2.4}$$

Where U and V are unitary, or orthogonal, matrices and Σ is a diagonal matrix. Each matrix can be interpreted in a physical way with regard to our perceptual hashing application. Let's consider the example of face recognition, where the N columns of X consist of reshaped pictures of faces:

- Each column of the matrix U is denoted as an ”eigenface”. The eigenfaces in U are hierarchically arranged by the order of their ability to describe the variance between the pictures of faces that make up the X matrix. The columns in U are all orthogonal and have unit length, so they provide a basis for the N-dimensional vector space in which the columns of X can be represented.

- The values in the matrix Σ are the singular values, which are all non-negative. Singular values can be viewed as scaling factors. They are also hierarchically arranged, where the first value of the matrix is the largest.

- Each column of the transpose of the V matrix can be interpreted as the composition of the different vectors of U that, when scaled by the singular value, make up a face, or column, in the X matrix.

Images can be compressed, or have relevant features extracted from them, by selecting only a portion of the first, most important vectors and highest valued singular values from matrices U, Σ and V.

The PCA leverages the fact that dominant features in high-dimensional data, like images, can be well described by a smaller set of values, that result from the SVD. PCA adds to the SVD by pre-processing the data by mean subtraction and setting the variance to unity first. In this way, the PCA can be viewed as the SVD on mean-subtracted data. Principal components (PC) make up the resulting coordinate system. These PCs are orthogonal to each other but have maximal correlation with the actual measurements that compose the initial data.

Let's take a look at the computation of the generalized PCA. First, we compute the mean of all m columns of matrix X:

$$\psi = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{2.5}$$

We then subtract the mean from each row, x_i, in matrix X, which results in the mean-subtracted data B:

$$b_i = x_i - \psi \tag{2.6}$$

$$\boldsymbol{B} = \begin{bmatrix} b_1 & b_2 & ... & b_m \end{bmatrix} \tag{2.7}$$

We can now apply SVD. The co-variance matrix of the rows of B results from

$$\boldsymbol{C} = \frac{1}{n-1}\boldsymbol{B} * \boldsymbol{B} \tag{2.8}$$

The principal components can be obtained by computing the eigen-decomposition of C:

$$\boldsymbol{CV} = \boldsymbol{VD} \tag{2.9}$$

where V is the matrix that contains the eigenvectors and D is a diagonal matrix that contains the singular values. The principal components are the eigenvectors corresponding to a certain eigenvalue.

Considering the example of face recognition again, the PCA is applied to a dataset of pictures of faces to extract the most dominant correlations between these images. From this decomposition, we are left with a new coordinate system that is defined from a set of eigenfaces. New images, that we want to recognize as faces in our dataset, can be represented in these coordinates by taking the scalar product with each of the principal components. Images of the same person tend to cluster in the eigenface space, allowing for their classification in this facial recognition system, as we will see in section 2.6. This also means that any generated hash value from images of the same person will be similar, allowing for the successful application of classification algorithms.

2.6 Classification Algorithms

In the context of PUF authentication, hash generation is only a means to an end, which is classification. After an hash is generated from a particular image, it needs to be classified. To accomplish this, classifying algorithms are employed.

A normalized hamming distance (HD) classification approach is widely used in this space, with a few notable mentions being [1], [7], [28]. The normalized hamming distance can be interpreted as the number of different bits between two binary hashes of the same size. Because it is normalized, when its value is zero, it means that both hashes are the same. On the contrary, if the hamming distance is 1, both hashes are 100% different. Utilizing a hamming distance threshold for classification means that, in short, if the hamming distance between two hashes is below a certain threshold, they are considered to be from the same image. A benefit of this classification approach is that it allows for the use of error correction codes, which could be employed to improve the authentication performance of the entire PUF system. It is also data independent, meaning it does not need a training set to be able to classify new occurrences in a testing dataset.

Classification is also the driving force of machine learning. In the topics of authentication systems, such as face recognition applications, machine learning plays an important role, and is served as the fundamental technique in many existing literatures. Machine learning utilizes previously obtained data, which is denominated as the training set, to make accurate predictions in new data, which is denominated as the test set [5]. Machine learning classifiers aim to find a decision threshold, in the same way as in a hamming distance classifier. When data that needs to be classified is multi-dimensional, machine learning classifiers pose as a solution. And example of multi-dimensional data classification is the eigenface example. Each principal component is a dimension of the data. A classification threshold needs to be discerned for each principal component that is analysed. In figure 2.6, an example of a decision threshold for a specific principal component of a dataset is presented. This decision threshold is calculated with the purpose of separating as best as possible the differently labeled data.

In short, machine learning, for classification purposes, has two main categories: Unsupervised learning and supervised learning. Unsupervised learning uses an unlabeled dataset (i.e. the feature vectors from the dataset don't contain a label, like a specific PUF token) to train a certain model and then classifies new data based on it. PCA is an unsupervised learning technique, on which a vector space is built so that new data can be projected upon it. This projected data tends to cluster when it is similar,

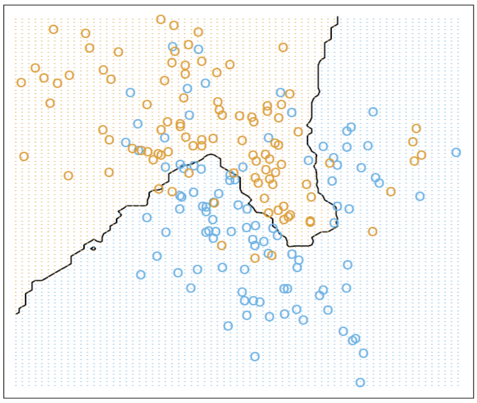


Figure 2.6: An example of a decision threshold adapted to a specific dataset. Taken from [5]

therefore allowing it's classification. The results from unsupervised learning could be further used for supervised learning. Supervised learning utilizes labeled datasets to train a model and make predictions on it.

In [5], categories of supervised learning are considered:

- Linear model: the model is specified as a linear combination of features.
- Parametric model: once the parameters of the model are learnt from the the training set, the training set could be discarded and only the parameters saved.
- Non-Parametric model: the model assumes that similar feature vectors have similar labels. The model finds similar feature instances in the training set, using a suitable measure, and determines the final output label.
- Non-metric model: the model functions on feature vectors with elements of non-comparable data. Comparable data would be, for example, 2 is closer to 1 than to 8. Non-metric models utilize data without any natural similarity metric.

On Table 2.1, an overview of some important machine learning classifiers is presented. All four algorithms employ different classification mechanisms.

Support vector machines (SVM) is one of the most successful classification algorithms developed to date, being widely used and often providing results that are better than competing methods [6]. SVM tries to make a decision boundary so that the separation between two classes the widest it can be. To accomplish this, the distance between the decision threshold and the closest points from each class (support vectors) are calculated. The hyperplane for which the margin is maximized is returned as the optimal hyperplane. A linear classification problem is characterized by being solved with a simple

Table 2.1: Overview and categorization of some widely used machine learning classifiers.

Method	Category
Support vector machine (Support Vector Machines (SVM))	Linear Model
K-nearest neighbors	Non-parametric model
Decision tree	Non-metric model
Random forest	Mixed method

straight line that can classify all data considered. An example of this type of problem is shown in fig. 2.7, where two generated hyperplanes are presented. A non-linear classification problem can not be solved with a straight line. In SVM, in non-linear problems, the data is mapped into a higher dimensional space. Afterwards, an optimal hyperplane is computed in this non-linear space.

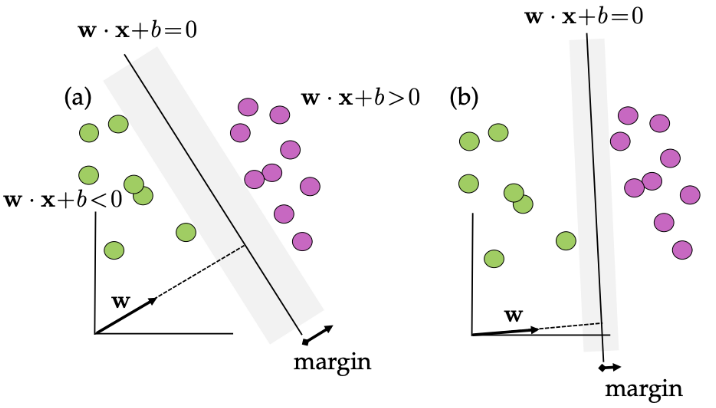

$\mathbf{w}\cdot\mathbf{x}+b=0$
(a)
$\mathbf{w}\cdot\mathbf{x}+b>0$
$\mathbf{w}\cdot\mathbf{x}+b<0$
w
margin
$\mathbf{w}\cdot\mathbf{x}+b=0$
(b)
w
margin

Figure 2.7: An example of two hyperplanes maximally separating two classes. Taken from [6]

K-nearest neighbors is likely the simplest supervised learning algorithm to understand. Given a new unlabeled data point x_a, simply find the k nearest neighbouring points. The label of x_a is determined by the majority of the labels of the k nearest neighbours.

The decision tree is a hierarchical construct that looks for optimal ways to split the data in order to provide a robust classification and regression. These decision trees generally lack robustness to different samples of the data. Thus, random forest works by constructing various decision trees at training time. New data is then classified by the class given as output by the majority of the trees.

3

The PUF System Used

Contents

3.1 Introduction 26

3.2 The PUF System 26

3.3 Image Acquisition 27

3.1 Introduction

This chapter covers, in a first section, the physical implementation of the PUF system that is used to acquire the speckle pattern images utilized. This system was built in [11]. The system allows for the usage of different types of tokens (different objects made of different materials). In this thesis, tokens made of non-woven polyester paper are used.

In the following section, the process and characteristics of each obtained dataset are discussed. In order to study how the impact of intra-variability can be reduced, we need to induce it purposefully. Intra-variability is introduced in the system by switching it OFF and ON between different acquisitions and changing the camera's position in the system.

3.2 The PUF System

The physical experimental system used to obtain speckle pattern images is described in figure 3.1. This system was developed in [11]. Because the focus of this work is on computer vision and hash generation, this subsection will not be as detailed.

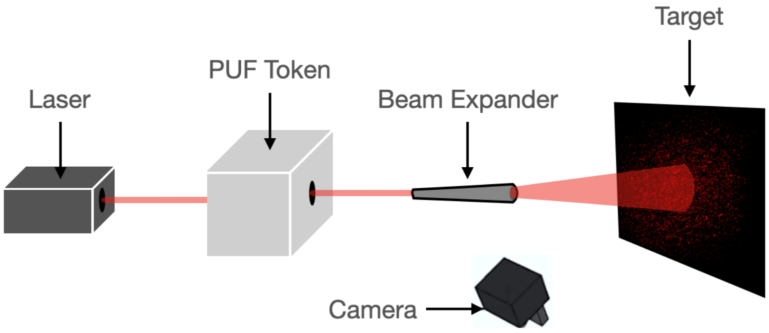


Figure 3.1: Illustration of the physical PUF system used to generate and acquire speckle pattern images.

The light source of this optical PUF system is a coherent He-Ne Laser (ref. HNLS008R by Thorlabs). The beam passes through the PUF, which consists of a non-woven polyester paper fabric that is characterized by being water resistant, capable of withstanding high mechanical strain and with a density of 250 g/cm^2. This type of tracing paper was chosen because it is a translucent object, meaning that the light emitted by the laser will be transmitted and result in a clear speckle pattern. The paper is placed in a sample holder which consists of a compact structure covered with black tape to prevent any external interference from light sources other than the laser. The light that passes through the PUF is directed into a beam expander, that serves the purpose of increasing its diameter. The beam expander used is the GBE20-A - 20X Achromatic Galilean Beam Expander (by Thorlabs). The speckle pattern is then projected onto a sheet of paper with a black rectangle printed on it, acting as a target. The purpose of this black rectangle is to have have a focus area of the speckle pattern which will later be perceptually analysed. If the entire obtained image were to be analysed, some control over certain conditions of the system would be lost (like camera focus a lighting conditions).

To acquire the images, the Camera Module V2 connected to a Raspberry PI is used. However, the authentication algorithms developed should be robust enough to allow the use of other cameras, such as a smartphone camera.

3.3 Image Acquisition

To test the authentication algorithms implemented, a dataset that purposefully introduces intra-variability between acquisitions is necessary. The intra-variability that results from normal usage of the system is the most relevant from an applicability of the algorithm's point of view. The main objective is to make the entire system (hardware + software) the most robust it can be in its normal usage. This means focusing on the elements of the environment that vary the most between usages of the system.

To understand what varies the most between acquisitions taken at different times, it is important to define how the system is operated. Initially, the laser is switched off. After a PUF token is inserted the camera should be positioned in as close of a position as possible as previous acquisitions of the same token. The paper token, which is placed under the sample holder, should also be positioned with maximal aligning to prevent fluctuations in the speckle pattern.

The three main sources of intra-variability in the system identified were:

- Switching ON and OFF the entire system, creating a change in the phase of the optical signal.
- Changes in the orientation of the camera between 5º and 15º degrees in relation to the target.
- Changes in alignment of the PUF token in the mask.

However, in [11] it is stressed that token alignment is a key factor in the implementation of optical PUF devices. In fact, even a small difference in the positioning of the PUF token in the mask can generate an entirely different speckle pattern. This is because the structures of the paper function as a chaotic system. This means that any variability introduced in this way will most likely render the acquisition unusable, as authentication of the token will not be possible. Because of this, only the two first sources of intra-variability were employed in the acquisition of the dataset.

Two different datasets were obtained. In the first obtained dataset, four different tokens were utilized. For each PUF token, certain rules were followed during the acquisition process:

- Switching OFF and ON the system between all acquisitions.
- Obtaining 50 acquisitions with a static camera.
- Obtaining 50 acquisitions while varying the camera orientation between 5º and 15º degrees in relation to the target.

The difference between camera positioning/orientation is exemplified in image 3.2.

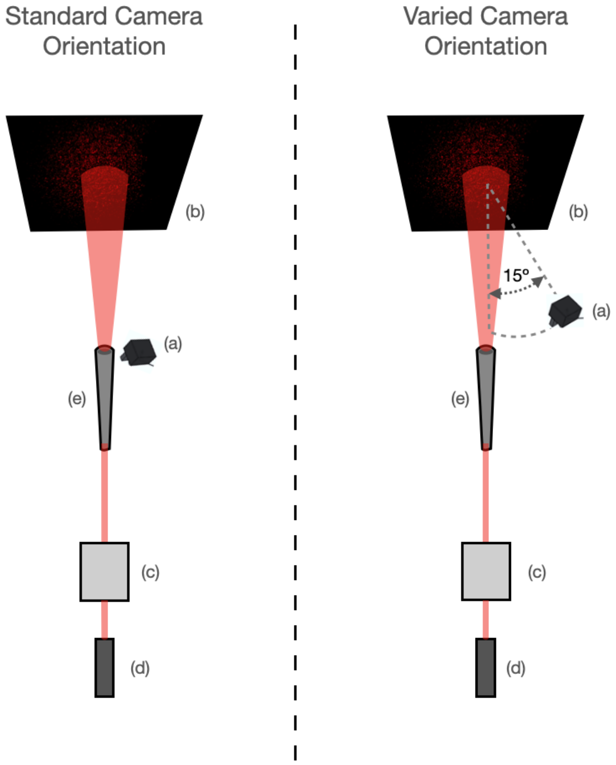


Figure 3.2: Exemplification of the variance introduced by changing the positioning/orientation of the camera; (a) Camera; (b) Target; (c) PUF Token; (e) Beam Expander;

This translates into a dataset with 400 images. A sample speckle pattern image of this dataset A is shown in figure 3.3

After testing was done with the described dataset A, some limitations needed to be addressed. Particularly, an effort was made to aid the cropping of the black rectangle. To do this, a new dataset B

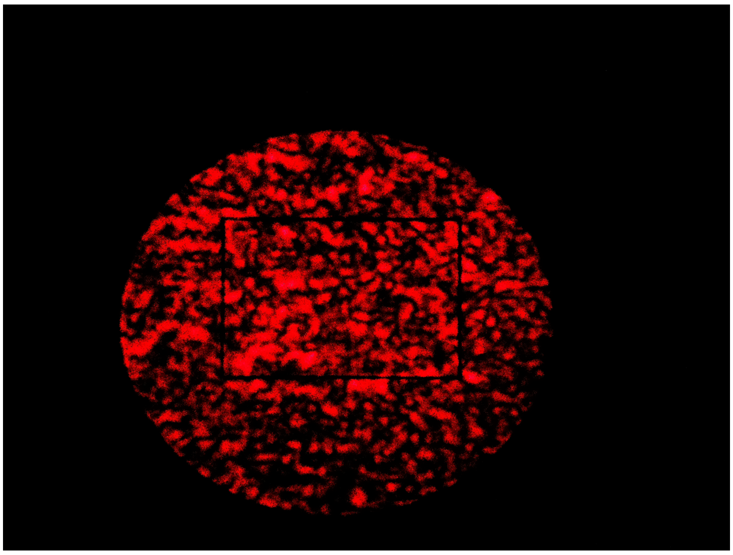


Figure 3.3: Sample image belonging to the first dataset A.

Table 3.1: Overview of the obtained datasets.

Dataset	PUF Token	Conditions	Notes
A	1AA	Standard camera orientation	Standard target
	1AB	Different camera orientations	
	1BA	Standard camera orientation	
	1BB	Different camera orientations	
B	2AA	Standard camera orientation	Target with black tape to reflect only the contents of the rectangle
	2AB	Different camera orientations	
	2BA	Standard camera orientation	
	2BB	Different camera orientations	

was obtained. A black non-reflective tape was used on the outside of the black rectangle. This means that the camera only captures the content on its inside. The same acquisition rules employed in the first dataset were also used. This translates into a dataset with 400 images. A sample speckle pattern image of this dataset is shown in figure 3.4.

With both datasets combined, 800 speckle pattern images are available to test the implemented authentication algorithms that are described over the next chapter. An overview of the described datasets is presented in table 3.1

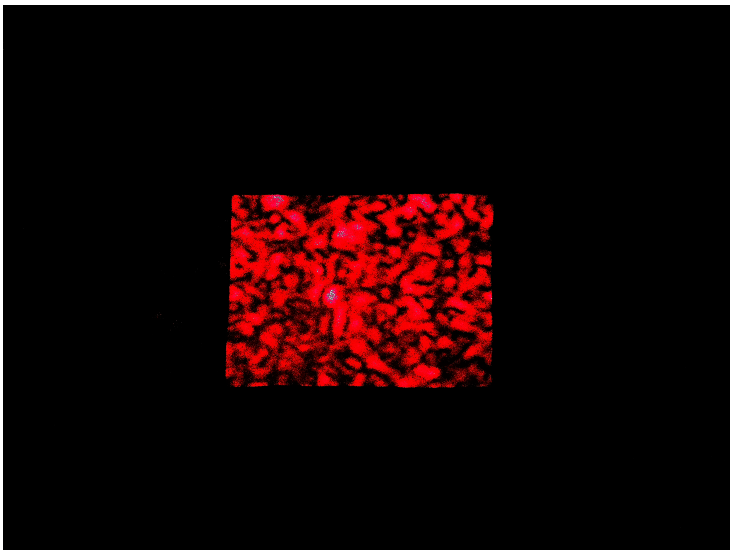


Figure 3.4: Sample image belonging to the second dataset, where black tape is utilized to aid in the cropping of the speckle patterns.

4

Proposed Approach for Authenticating a PUF

Contents

4.1 Introduction 32
4.2 Pre-Processing 36
4.3 Feature Extraction and Hash Generation 42
4.4 Classification 44
4.5 Pipelines of the Proposed Approaches 45
4.6 Discussion of the Proposed Approaches 50

4.1 Introduction

This chapter covers the different implementations that were developed and analysed for generating hashes and authenticating the PUF tokens. First, in this introductory section, the considered use cases for the authentication with the PUF system are discussed. Potential application environments need to be discussed so that any developed solution is tailored to them. The proposed approaches are also discussed, along with some important characteristics that they should have in order to perform adequately in a PUF context.

The remainder of this chapter is structured in the logical order of a perceptual hashing based authentication algorithm, which has 5 main stages, as it is shown in section 4.1.2:

- Image Acquisition
- Pre-Processing
- Feature Extraction
- Quantization
- Classification

After this, a section detailing the pipelines of the different proposed approaches is presented.

From the methods covered in Chapter 2, a few were selected in each stage. In the pre-processing phase, multiple methods were tested, namely ORB and the Hough transform. The Gabor transform is also employed. For feature extraction, a decision was made to implement and compare the DCT and PCA. As for classification of the resulting hashs, multiple machine learning classifiers are also utilized.

4.1.1 Desired Features in an Authentication Algorithm for PUFs

The main application scenario considered, aside from cryptographic key generation, is utilizing the PUF system to authenticate important objects/documents.

There are certain requirements that should be met when developing a perceptual hashing algorithm with the intent of authenticating speckle pattern images from a PUF system. These characteristics largely vary depending on the use scenario the PUF system is going to be used, and on the PUF system itself. For example, a system where the camera is static and there is no illumination variability does not require the same features in a hashing algorithm as a system where the camera is not static and illumination defects may occur. A scenario where the time it takes to authenticate a PUF token is not one of the main concerns will also require different qualities in the hashing algorithm implemented.

Taking into consideration the PUF system used, there are two main application scenarios to be considered:

- A system with an integrated camera and computer, where registration and authentication time are not the main concern. The main objective is to provide very accurate authentication results. This could happen in the context of, for example, authenticating important documents or utilizing a PUF to gain entry into a highly secure system.
- A system where the camera and computer are not inherently part of it. In this scenario, computational time of the implemented algorithm is considerably more important. This could happen in the context of, for example, acquiring a speckle pattern image, with a smartphone, to authenticate a piece of equipment that is tagged with a PUF.

In general, and considering the PUF system used in this thesis where the camera is not static and the speckle pattern is projected in a target, the desired qualities in a hashing algorithm [7] should be:

- The algorithm must be computationally **efficient**. Depending on the use case, part of the algorithm will run multiple times during registration, so it should be fast. Efficiency is an essential characteristic that should have a large weight in deciding which hashing approaches fit best each use case.
- The algorithm must offer **sufficient distinguishability** in order to allow the authentication of the tokens. Ideally, some distance metric should be maximized when comparing two speckle patterns from different PUF tokens and zero when comparing two speckle patterns derived from the same PUF token.
- The algorithm should be **susceptible to mathematical analysis**, allowing the exploration and characterization of its properties and performance through analytic expressions.
- The algorithm needs to be **insensitive to global changes in ambient light level**. In other words, the algorithm shouldn't have a dc response. Because the camera isn't in a controlled environment, there is no guarantee that the average intensity of the pixels in the acquired images will remain constant over time.
- The algorithm should be **insensitive to small changes in speckle patterns** that lead to token misregistration or misauthentication. Speckle patterns are highly sensitive to changes in positioning of the PUF token due to light being reflected towards a vast number of random directions inside the structures of the optical diffuser. If the token moves, these directions change along with pattern itself. The algorithm should accommodate, to a certain degree, these small changes in the speckle patterns.
- The algorithm must be flexible enough to **accommodate any changes in the positioning and scale of the speckle pattern**. Because the camera is not fixed, the speckle pattern can suffer

warping and changes in positioning when acquiring an image, which should be accounted for during the perceptual hashing phase of the system.

4.1.2 Proposed Approach

Each proposed approach can be divided into 5 distinct modules, which are represented in figure 4.1.

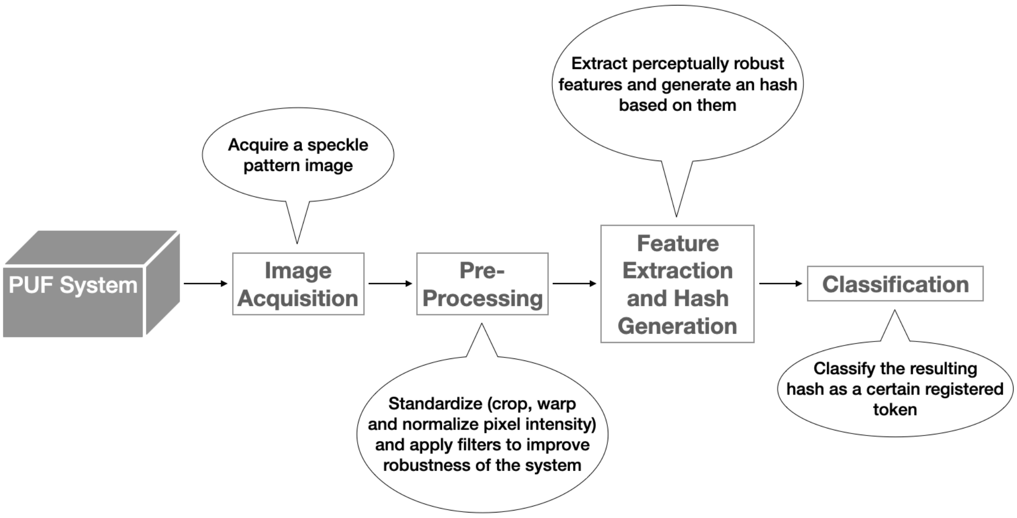

Extract perceptually robust features and generate an hash based on them
Acquire a speckle pattern image
PUF System
Image Acquisition
Pre-Processing
Feature Extraction and Hash Generation
Classification
Standardize (crop, warp and normalize pixel intensity) and apply filters to improve robustness of the system
Classify the resulting hash as a certain registered token

Figure 4.1: Diagram and description of each module in a perceptual hashing based authentication algorithm.

It is important to note that an authentication system has two distinct operational settings: a registration phase and an authentication phase. The registration phase encapsulates the procedure of registering a new item (a new PUF token) in the system. In this case, it consists of perceptually analysing a speckle pattern image and generating an hash value. This value is then stored. The authentication phase consists of the same process of analysing a new speckle pattern image and generating an hash value for it. However, once this is done, the hash value is compared to all other previously registered hashes and consequently classified as a certain PUF token. This process is presented in figure 4.2. In short, the registration phase does not utilize the classification module.

For this study, it is intended to understand which methods of authentication best suit each use case of the system. Particularly, comparing a DCT based approach (an invariant feature transform based method, which is data independent) to a PCA based approach (a dimension reduction based method, and thus a method that is dependent on the data used for training the system). These feature extraction

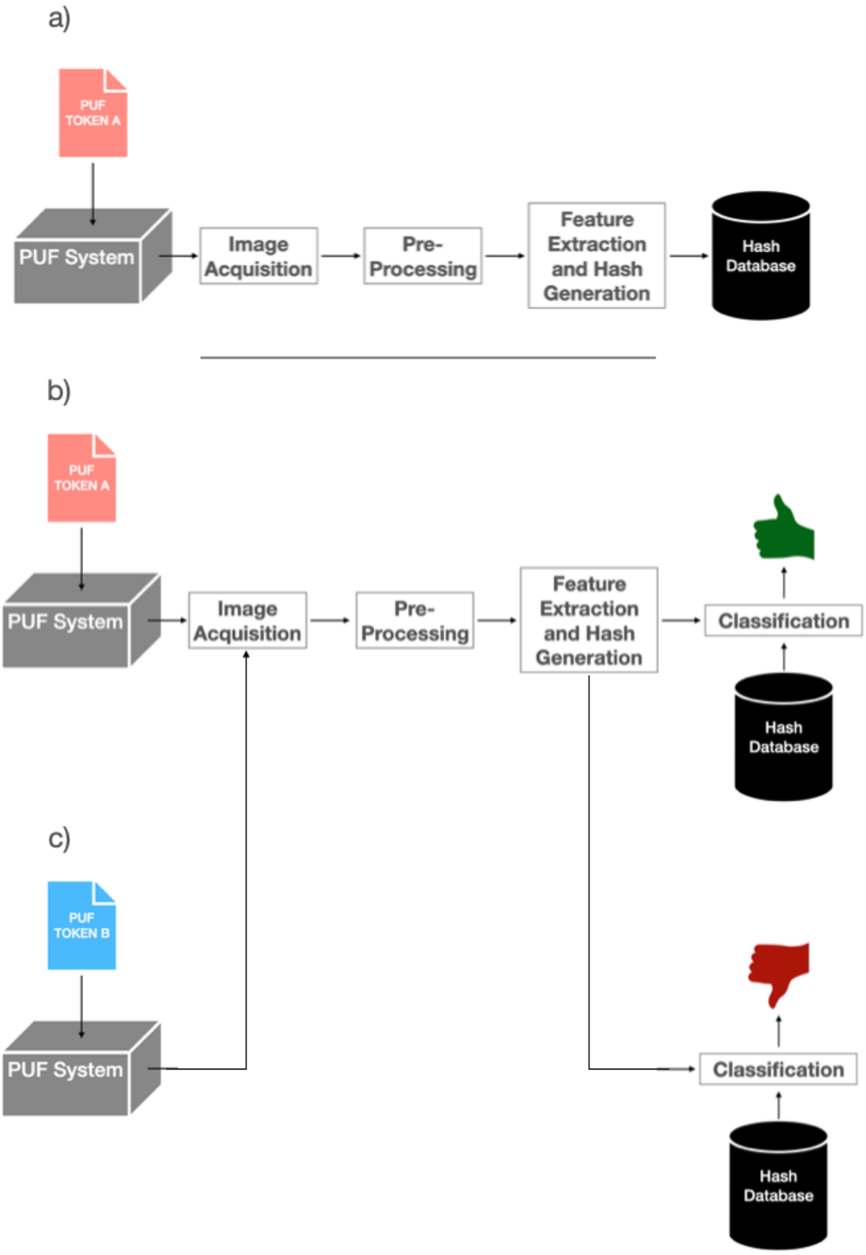


Figure 4.2: (a) Registration phase of token A. A speckle pattern image of token A is acquired and an hash is generated and stored in a database; (b) Authentication phase of token A. A speckle pattern image of token A is acquired, an hash is generated and compared to all other hashes in the database. Because it has been previously registered, the authentication is correct; (c) Authentication phase for an unregistered token. A speckle pattern image of token B is acquired, an hash is generated and compared to all other hashes in the database. Because this token has not yet been registered, the authentication process fails;

techniques were chosen for comparison because one is data independent (DCT) and the other is data dependent (PCA). In other words, the PCA depends on data that is used for training the system, while the DCT does not require training. In this way, two very distinct approaches are tested. Each of these feature extraction methods is then paired and tested alongside different classifying strategies: a simple hamming distance threshold classification or a machine learning classifier (like Support Vector Machines). This modular process of testing how the different feature extraction and classification methods work together, in this context, gives us four distinct approaches, which are represented in Table 4.1.

Table 4.1: Authentication approaches implemented and tested.

Approach	Feature Extraction Method	Classification Method
Approach 1	DCT	Hamming Distance Classification
Approach 2	DCT	Machine Learning Classifier
Approach 3	PCA	Hamming Distance Classification
Approach 4	PCA	Machine Learning Classifier

In the following sections we will take a look at the pipeline of each of these approaches. In both the DCT and PCA based approaches, a pre-processing stage is utilized to test filtering techniques to improve robustness and discrimination of the system. Gabor filters and mean filters were tested and compared.

4.2 Pre-Processing

The pre-processing module is always present in the proposed solutions, as it contributes decisively to improve their robustness. This module has four steps:

- Conversion to grayscale.
- Perspective warping and cropping of the speckle pattern images.
- Image standardization, which includes image resizing and normalizing the intensity of all pixels.
- Gabor filtering.

In a first step, the raw input image is first converted to grayscale. The essential semantic information resides in the luminance component, so the others can be discarded. Discarding color information also makes the hashing process faster and allow the matching between identical speckle patterns that may have slightly altered color spaces [29]. After this, the images are warped and cropped. This process is described in detail in Section 4.2.1.

The image is then resized to 64x64 pixels, which is important for subsequent image matching operations. Resulting speckle pattern hashes will now match similar ones regardless their original dimensions and aspect ration. The smaller the images are resized to, the faster the algorithms will compute the hash values. However, this comes at the cost of discrimination. Different image sizes were tested. The matching capability of the algorithm and the time it took to hash all the images of different sizes is presented in Section 5.2.2.

After this, the intensity of each pixel of the image is normalized following a correction ratio. This makes each image have all pixels in a range of intensities from 0 to 255. Because there are variations in illumination settings in the environment, as well as in the camera itself, it is important to equalize all speckle pattern images to improve the robustness of the system.

Finally, Gabor filtering is applied. The filtering process is described in detail in Section 4.2.2. The pre-processing stage is presented in Figure 4.3 as a block diagram.

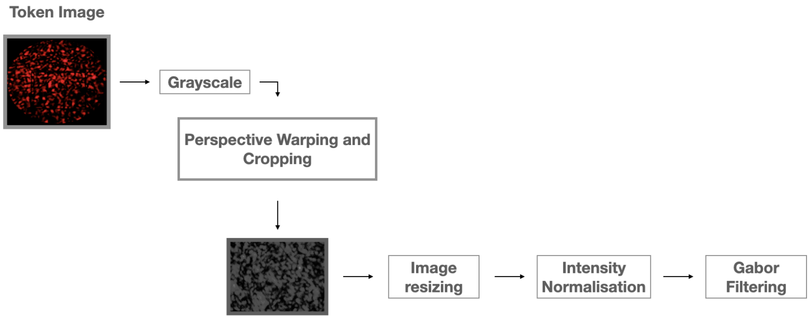


Figure 4.3: Pre-processing diagram all implementations.

4.2.1 Warping and Cropping Speckle Pattern Images

The captured images contain some background, besides the speckle image. And, although a squared region of interest is considered for authentication comparison, the variable positioning of the camera may lead to distortions in the acquired images. Therefore, cropping and warping of the obtained images, to select only the region of interest and in the desired rectangular format, are the first operation to be performed towards improving the system's authentication performance.

The process of cropping and warping the images can be structured into the following main steps:

- Detecting the black rectangle (possibly utilizing the Hough Transform, ORB, or the pixel intensity values)

- Cropping the contents, to only keep the contents inside the black rectangle

- Warping (using a set of points between the distorted and a default images, for finding the appropriate transformation parameters)

For applications like this, where we want to discriminate a geometric shape in our image, the Hough transform would be a relevant option to test. Any longer line that would exist in the speckle pattern, like the side of a rectangle, would be differentiated. However, a problem arises when we start to analyse the speckle pattern images. Because there is a large number of very similar points, and the speckle pattern slightly distorts the sides of the rectangle, the results are not satisfactory. In figure 4.4 an example usage of Canny Edge Detection combined with the Hough Transform [30] in a speckle pattern image is presented. The lines discriminated by the Hough transform appear in red. It is clear that there aren't any significant larger lines detected.

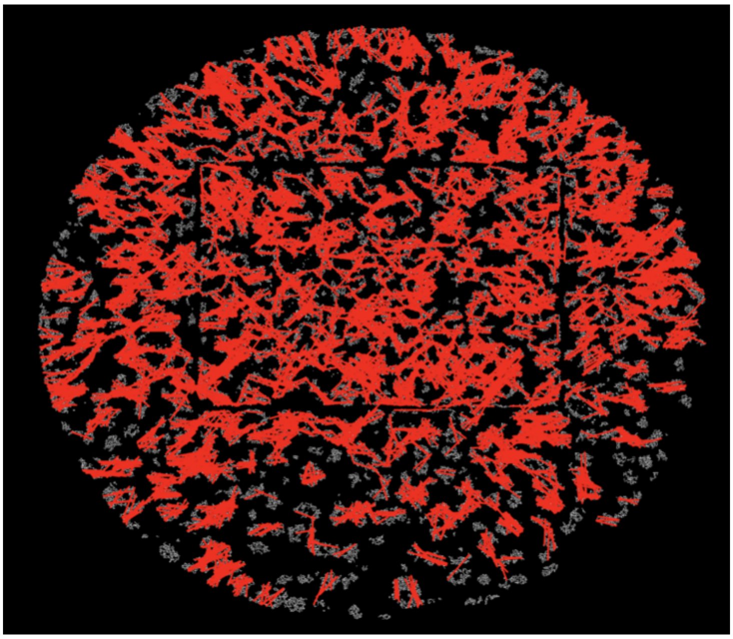


Figure 4.4: Hough transform combined with Canny edge detection.

For the first dataset, where black non-reflective tape is not used, cropping is done manually by selection of the area between the pixel coordinates that more or less coincide with the black rectangle. This lack of discrimination of the black rectangle became a problem. In fact, the manual cropping process takes a considerable ammount of time, making it impractical in the scope of an authentication algorithm. However, dataset B was obtained with these problems in mind. With the non-reflective black tape around the rectangle, it can easily be discriminated from the rest of the image, allowing for an automatic cropping process. To accomplish this, the image is converted to binary using a thresholding operation. This improves the accuracy of contour discrimination. Contours can be described simply as a curve joining all the continuous points, along a boundary, having the same intensity. Afterwards, a straight bounding rectangle that encompasses the largest contour can be computed (this is done with OpenCV's *boundingRect()* function). The results are presented in figure 4.5. The images can then be cropped to the coordinates of this rectangle.

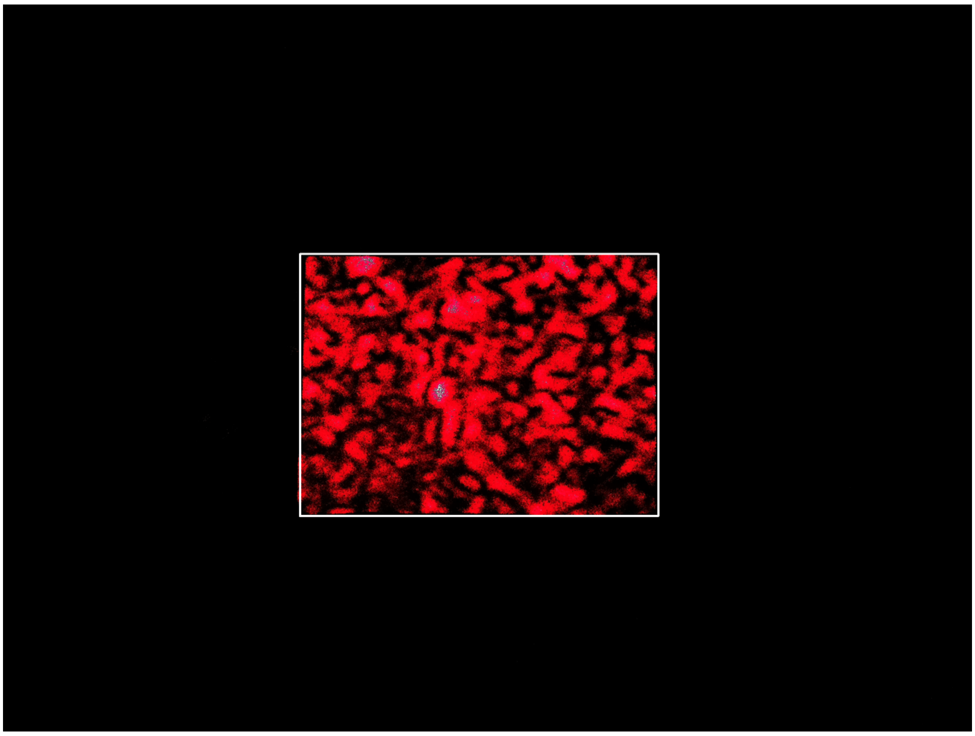


Figure 4.5: Bounding rectangle (in white) computed that encompasses the entire region of interest of the speckle pattern image.

In terms of warping, different approaches can be taken.

Feature point detection algorithms like ORB [4] or SIFT [17] are candidates to be tested. ORB was implemented and tested. ORB features are invariant to rotation, scale and to limited affine changes, making it an appropriate tool to use in this scenario. Because there are a lot of very similar points in speckle patterns, the descriptor pairs may not be correct. A large subset of these key point pairs, with a line connecting them is shown in Figure 4.6. It is clear that the majority of these matches is wrong, given that the lines are not horizontal.

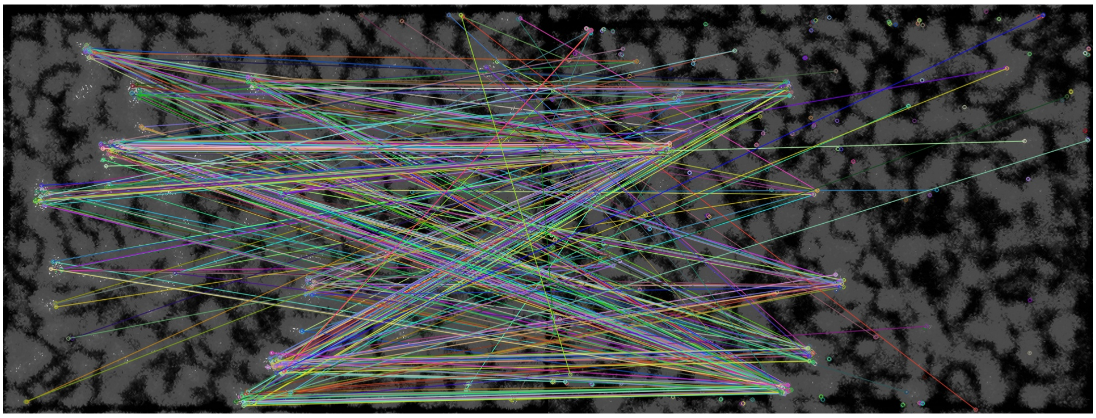


Figure 4.6: Matched key points by ORB between images with different camera positions.

To find the correspondences that allow calculating the warping parameters between the distorted image and the desired rectangle, we can use the key points obtained by ORB. However, as we have seen, not all points will lead to a correctly warped image, as there are many incorrectly calculated matches.

We assume that there is a subset of all found matches that will lead to a satisfactory warped image. By ordering all matches in terms of their hamming distance, in an ascending order, the best ones can be selected using a ratio. This ratio can be, for example, 20% of the top matches. Because we assume that there is a subset of the best matches that will lead to satisfactory results, this ratio can be iterated until we find it. This method utilizes the DCT to obtain a hash value for both images and compare their hamming distances. We iterate through the different ratio values, warp the distorted image, apply the DCT and calculate the hash values. There is a ratio value that will minimize the hamming distance between these hashes. When this happens, we assume that the image was correctly warped.

In RBrder to remove the need to iterate through all possible ratio values, when the hamming distance between the hash values is below a certain threshold we assume that the distorted image was correctly warped. The proposed warping system architecture is presented in Figure 4.7. One way to improve the performance of this algorithm is to keep only the ORB key points that are closest to the corners of the image.

It is important to note that this warping algorithm is, in a way, an authentication method as well. The algorithm could be applied to each token that is registered and stored in our system. Here we would have to store each speckle pattern image. In each algorithm run (for each unauthenticated image - registered image pair), the lowest hamming distance is kept. The image is then authenticated as the

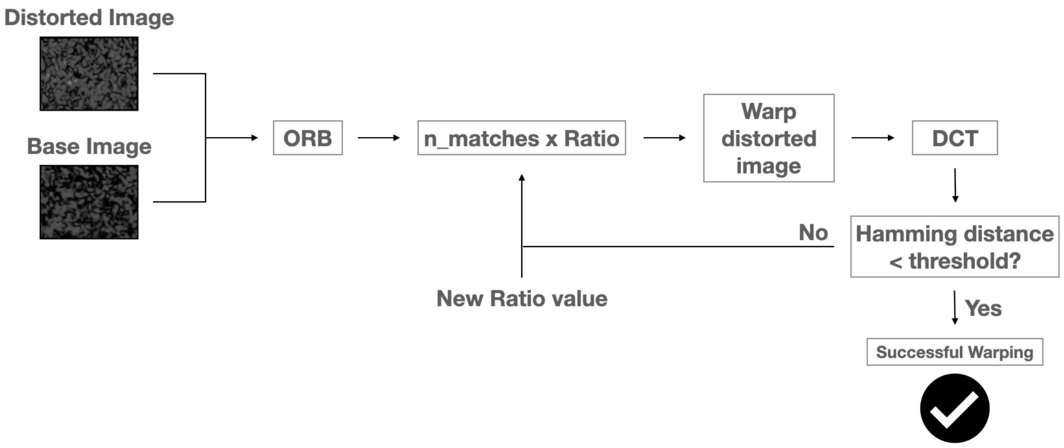


Figure 4.7: Diagram of the warping algorithm implemented, utilizing ORB and DCT.

token that provided the lowest hamming distance, if that hamming distance is below a certain threshold (this prevents an image of a token that isn't yet registered in the system to be wrongfully authenticated). This warping algorithm could be combined with the approaches that will be presented over the following sections to improve robustness and authentication performance.

4.2.2 Gabor Filtering

There are multiple image processing schemes devised around the application of Gabor kernels for texture and pattern analysis. Filtering can improve the robustness of the final algorithm by minimizing the impact of small distortions and variability in speckle pattern images. Kernels with different frequencies were tested. These results are presented in section 5.2.2.B. The final implementation utilizes a single Gabor kernel with a 45º degree diagonal orientation. As it was shown in [7], diagonally orientated Gabor kernels emphasize diagonal features in the speckle pattern, reducing the effect of unwanted horizontal or vertical movement in the token image. A comparison of a diagonal kernel with an horizontal kernel is shown in figure 4.8. The same kernels were applied on two images of the same token: one with the camera in a standard position and another with the camera in a non-standard position.

One of the ways intra-variability is introduced in the system is by altering the position of the camera, which mainly warps and moves the speckle pattern along an horizontal axis. This step should improve the performance of the algorithms.

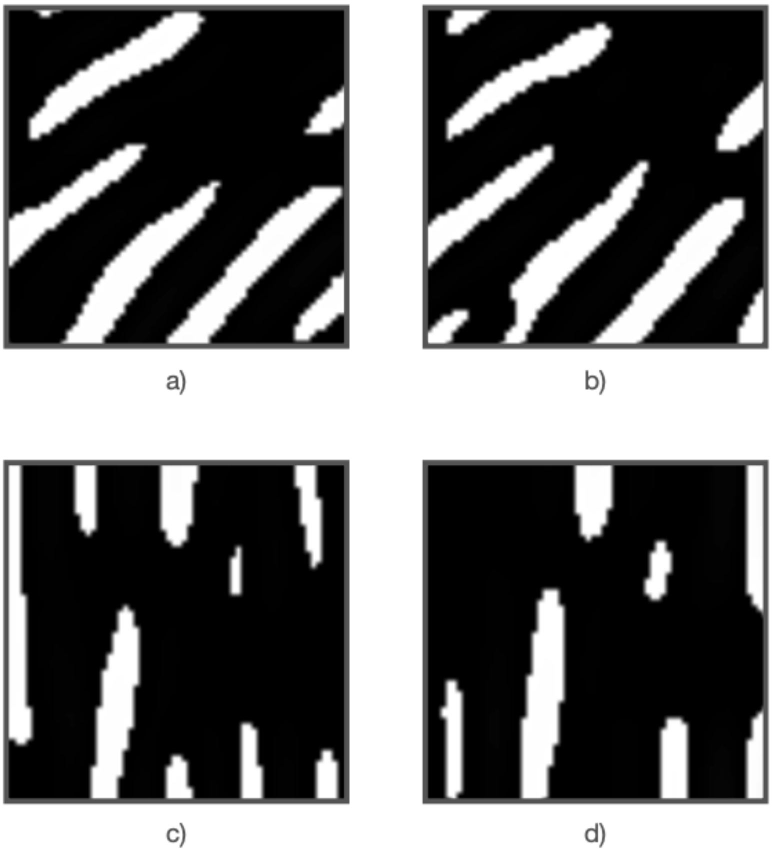


Figure 4.8: Comparison of Gabor kernels with a diagonal orientation and an horizontal orientation. A larger variation can be seen between images filtered with an horizontal kernel: a) 45º Gabor Kernel with standard camera position; b) 45º Gabor Kernel with different camera position; c) 180º Gabor Kernel with standard camera position; d) 180ª Gabor Kernel with different camera position;

4.3 Feature Extraction and Hash Generation

4.3.1 DCT for Feature Extraction

In this section we will take a look at the pipelines of the different implemented algorithms that use the DCT for feature extraction.

The DCT is applied, resulting in the DCT coefficients matrix. To obtain a one-dimensional array, a zig-zag type scan is utilized, as the DCT tends to compact most of the energy in the first coefficients of the matrix, as it is possible to see in figure 4.9. As it has been previously described, the upper left corner tends to concentrate the majority of the low-frequency components of the image, as well as the DC component in the first entry.

Figure 4.9: Example of a DCT matrix obtained from a 64x64 resized speckle pattern. Most of the energy tends to be compacted in the first coefficients of the matrix, i.e. the upper left corner.

The DC coefficient is discarded as it does not translate any discriminating information from the images. This results in a vector of coefficients, which will be the hash value of the image. Because the initial coefficients hold the majority of the image's energy, the final hash does not need to contain all coefficients given by the DCT. More coefficients in the final hash will translate more of the image's data. However, the hashes will take up more space in memory and hash matching algorithms, i.e. the classifiers will take longer to compute the results. From the research done in [11], for this type of PUF token, a 64 bit hash provides sufficient discrimination between different paper tokens. Hash sizes of 64 bits are also common in cryptographic studies [31], [32]. Because of this, the DCT based algorithms generate hashes with 64 bits.

Finally, the resulting array is quantized into the final hash, so that the sequence can be normalized into a binary form. The conditions for the quantization are

$$h_i = \begin{cases} 0, & \text{if } C_i < m \\ 1, & \text{if } C_i \geq m \end{cases} \tag{4.1}$$

where h_i is the hash value, C_i is the coefficient of the array and m is the mean of all the coefficients. So any coefficients that are above the median value are declared to 1, and any below as 0. Quantization is necessary for hamming distance calculation. There are several quantization schemes, some more complex than others. The applied mean based quantization is taken from [29], which showed good results in conjunction with the DCT.

4.3.2 PCA for Feature Extraction

As it was said in previous sections, Principal Component Analysis is a data dependent method of feature extraction. This means that it needs a training set to extract predominant patterns from. After the token images are pre-processed, they are reshaped into a vector with one dimension. This vector is then appended to a larger matrix tha1t contains all other vectors from reshaped images of tokens already registered in the system. This matrix serves as a vector space for other token images to be projected upon [33], returning weighted coefficients, that allow for their classification.

The resulting hash is then quantized in the same way as in the DCT based approaches.

4.4 Classification

There are two different means of classification utilized in this dissertation. One is a normalized hamming distance threshold based classification and the other consists of machine learning classifiers.

Because a normalized hamming distance threshold based classification algorithm is data independent, it is interesting to compare to other data dependent methods, like supervised learning techniques. The hamming distance obtained from two images of the same PUF token is denominated intra-HD, while the one obtained from two images of different PUF tokens is denominated inter-HD. In these types of classification problems, the HDs are represented in a histogram and can be described by a Gaussian probability density function. The mean (μ) and variance (σ^2) values of this Gaussian distribution function can be utilized to evaluate the robustness of the system. More distant mean values of the intra-HDs and inter-HDs translate in a better classification capacity. Lower variance also makes a correct classification more likely.

Utilizing a hamming distance threshold for classification means that, in short, if the hamming distance between two hashes is below a certain threshold, they are considered to be from the same PUF token. This threshold can be calculated with the mean values of the intra and inter Hamming distances, as well as their variances [34]:

$$\lambda_{opt} = \frac{(\mu_a\sigma_b^2 - \mu_b\sigma_a^2) \pm \sqrt{(\mu_a\sigma_b^2 - \mu_b\sigma_a^2)^2 - (\sigma_a^2 - \sigma_b^2)(\mu_b^2\sigma_a^2 - \mu_a\sigma_b^2 - 2\sigma_a^2\sigma_b^2 \ln \frac{p_b\sigma_a}{p_a\sigma_b})}}{(\sigma_a^2 - \sigma_b^2)} \tag{4.2}$$

p_a and p_b are the error probabilities associated with the intra-HD and the inter-HD distributions. In this problem, these values are assumed to be 0.5, as there is 50% probability of considering a hamming distance to be intra or inter. The error probability associated with this threshold can also be calculated:

$$PoE(\lambda) = \frac{p_a}{\sqrt{2\pi\sigma_a^2}} \int_{-\infty}^{d_{opt}} \exp -\frac{(x-\mu_a)^2}{2\sigma_a^2} dx + \frac{p_b}{\sqrt{2\pi\sigma_b^2}} \int_{-\infty}^{d_{opt}} \exp -\frac{(x-\mu_b)^2}{2\sigma_b^2} dx \tag{4.3}$$

From Equation (4.3) the False Positive Rate, which is the sum of the probabilities of misclassifying a different token, and the False Negative Rate, which is the sum of the probabilities of misclassifying an incorrect token, can be obtained.

Because of intra-variability, there will almost always be a certain degree of difference between all hashes. With this approach, after a feature vector is generated from a feature extraction method, there is a need for it to be binary, to then calculate the hamming distances between hashes. Like it was described in the previous section, hash vectors are quantized into binary form after the DCT or PCA is applied. On the other hand, when using machine learning classifiers, there is no need for the quantization of the generated hashes.

Four different machine learning classifiers were selected to holistically understand what works best. Each of these supervised learning techniques can be organized in different categories [5]. These machine learning classifiers and their organization are presented in table 4.2.

Table 4.2: Overview and categorization of the machine learning classifiers tested in the implemented algorithms.

Method	Category
Support vector machine (SVM)	Linear Model
K-nearest neighbors	Non-parametric model
Decision tree	Non-metric model
Random forest	Mixed method

Each of these methods treat the input dataset in distinct ways, possibly posing different benefits and downsides.

4.5 Pipelines of the Proposed Approaches

4.5.1 DCT Based Approaches

It is important to detail how the previously described steps fit in an application with a registration and an authentication phase. Here approach 1 (DCT based hashing + hamming distance classification) and approach 2 (DCT based hashing + machine learning classifier), presented in Table 4.1, are considered.

The pipeline for approach 1 is presented in images 4.10 and 4.11. In the registration phase, a token image is taken and a hash is generated, which is then stored in a database.

In the authentication phase the process is similar. A new token image is taken and a hash is generated with the same parameters used in the registration phase. This hash is then compared to every other hash in the database via hamming distance. If the hamming distance of two certain hash values

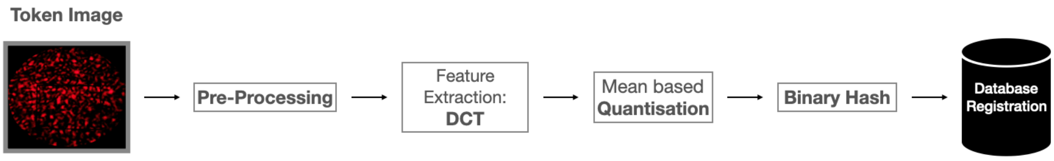


Figure 4.10: Registration Phase - Approach 1: Pipeline for the registration of new tokens with a hamming distance based classification.

is bellow a certain threshold, the token is authenticated. This works on the basis that hash values from different tokens have a lot more differences in their binary string when compared to hash values from the same token.

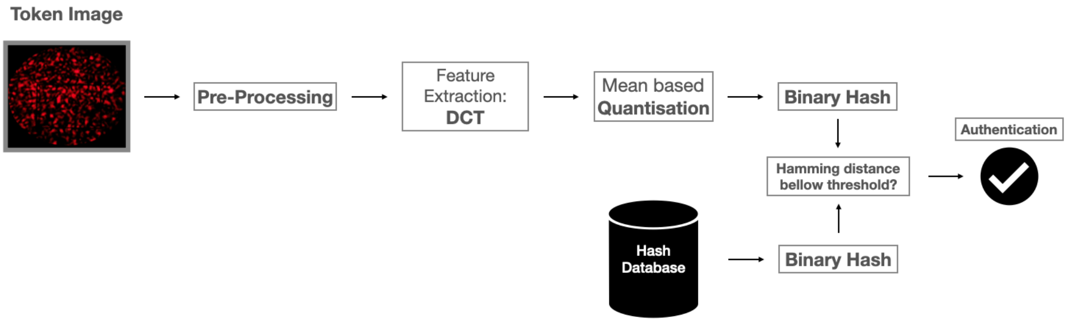


Figure 4.11: Authentication Phase - Approach 1: Pipeline for the authentication of tokens against registered tokens in a database with a hamming distance based classification.

The pipeline for approach 2 is presented in images 4.12 and 4.13. Note that here, when a certain token is registered, multiple speckle pattern images must be provided. This is because a classifier needs to be trained to later properly authenticate any other hash value it receives in the authentication phase. The more token images supplied in this phase, the better the authentication performance will be (up to a certain degree, or else overfitting might become an issue). Because a machine learning classifier is used, there is no need for the quantization of the hash values, which allows for the preservation of some information that would be lost through binarization.

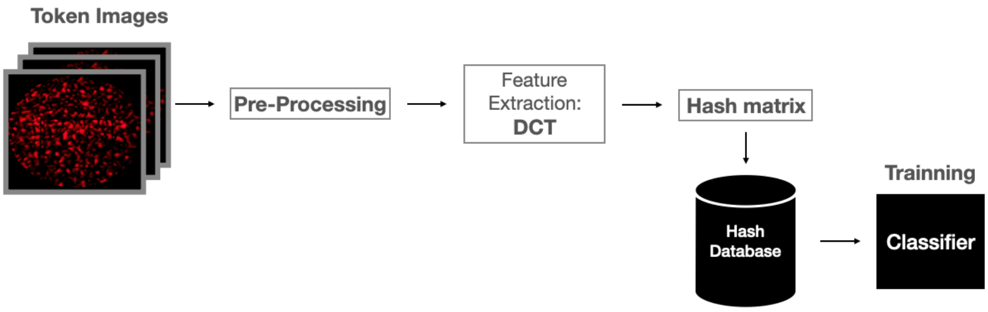


Figure 4.12: Registration Phase - Approach 2: Pipeline for the registration of new tokens utilizing the DCT and a Classifier.

The authentication phase of approach 2 is similar to approach 1 except the generated hash value is not quantized and it is classified through a machine learning classifier. Several of these classifiers were tested and the results obtained are presented in section **??**.

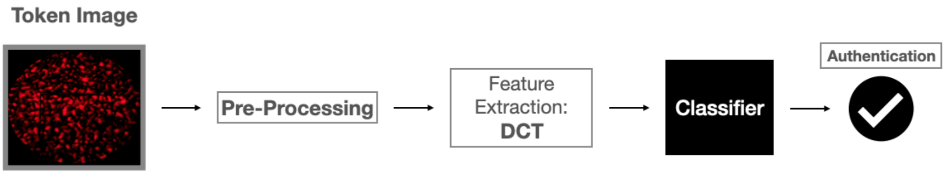


Figure 4.13: Authentication Phase - Approach 2: Pipeline for the authentication of tokens against other registered tokens in a database utilizing the DCT and a Classifier.

4.5.2 PCA Based Approaches

In this section the pipelines of the different implemented algorithms that use the PCA for feature extraction are presented.

First approach number three from Table 4.1, where a hamming distance based classification is employed, should be discussed. PCA requires multiple training images to create a vector space on which new unauthenticated images will be projected upon [33]. Because of this, the pipeline for the algorithm using a hamming distance threshold based classification becomes slightly more complicated. Not only is there a need for multiple speckle pattern images to register a certain PUF token, but another prototype image of the same speckle pattern is required to generate a hash value. This prototype image will be projected on to the created vector space, and the weights of the principal components will be quantized into a binary hash. The quantization scheme used, as it was previously mentioned, is the same as in the DCT based implementations. The registration phase is presented in figure 4.14.

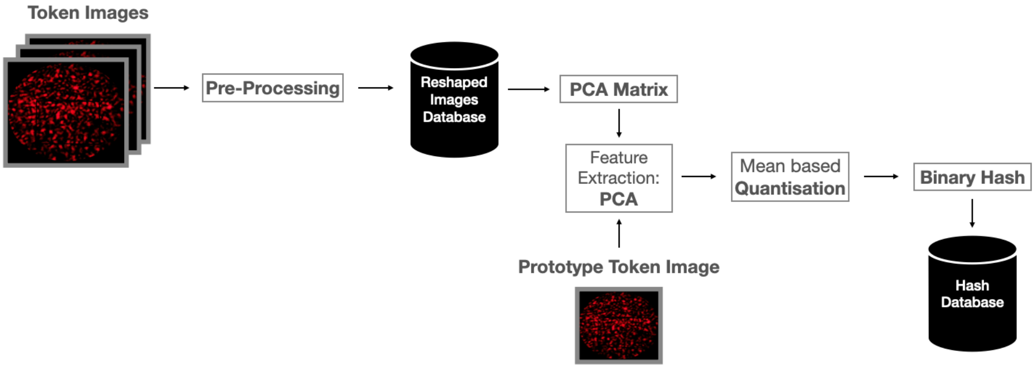


Figure 4.14: Registration Phase - Approach 3: Registration phase for the PCA based implementation using a hash hamming distance threshold as classification. Multiple token images are used to create a matrix on which PCA is applied and a final token image is used to generate an actual hash value for the PUF token.

In the authentication phase, it is necessary to pull data from databases on two distinct instances, as illustrated in figure 4.15. Firstly, after pre-processing, the PCA transformed image matrix is necessary for feature extraction. In short, an hash value is generated in the same way as in the registration phase for the prototype image. Mean based quantization is also done. Once the hash value of the unauthenticated token image is calculated, it is then compared, via hamming distance, to every other hash that is already registered in the system. If a certain hamming distance between two hashes is below a certain threshold, they are considered to be from the same PUF token.

Next, approach number four, where classifiers are used to authenticate token images, is considered. The same concept remains, where multiple speckle pattern images are necessary to register a new token in the system. Once these images have been pre-processed and reshaped as vectors, they are appended to a larger matrix that contains all images already registered in the system. The PCA is then computed on this matrix and the initial pre-processed images are projected on the generated space, in order to extract predominant features in them. The resulting hashes are then stored in a database and

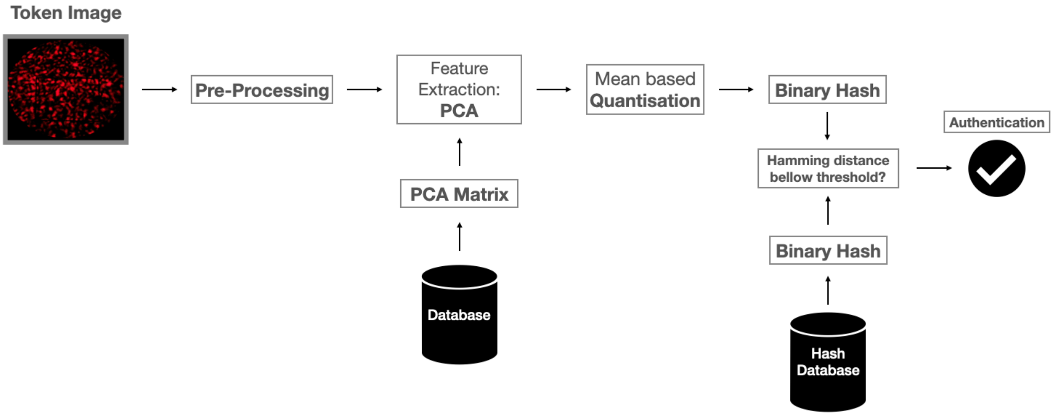


Figure 4.15: Authentication Phase - Approach 3: Authentication phase for the PCA based implementation using a hash hamming distance threshold as classification. Data needs to be pulled from databases in two separate instances during the authentication process.

used to train a classifier. Note that, a machine learning classifier is used in this implementation, there is no need for binarizing the resulting hashes, which in this context will prove to show significantly better results. The registration phase is shown in figure 4.16.

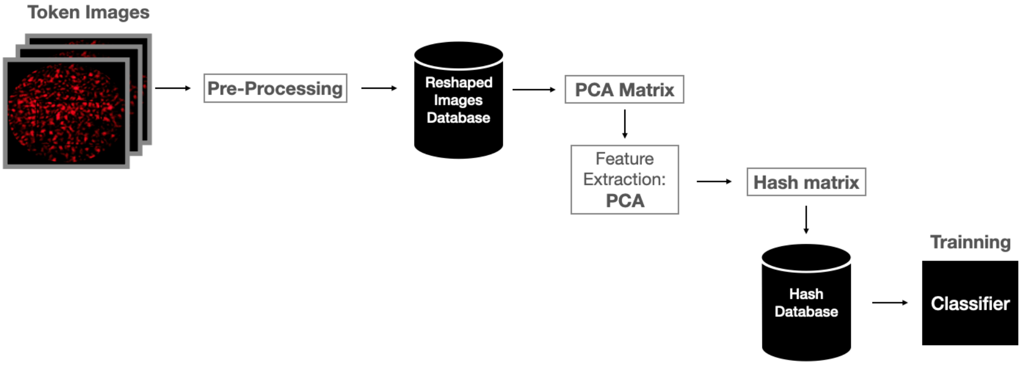


Figure 4.16: Registration Phase - Approach 4: Registration phase for the PCA based implementation using a classifier. Multiple token images are used to create a matrix on which PCA is applied and the generated hashes are used to train a classifier.

In the authentication phase, the features from the pre-processed token images are extracted using the PCA matrix created and continuously updated during registration phases. The generated hash is then given has input to the previously trained classifier and, if the token has already been registered in the system, the authentication is successful. A diagram containing this process is shown in figure 4.17.

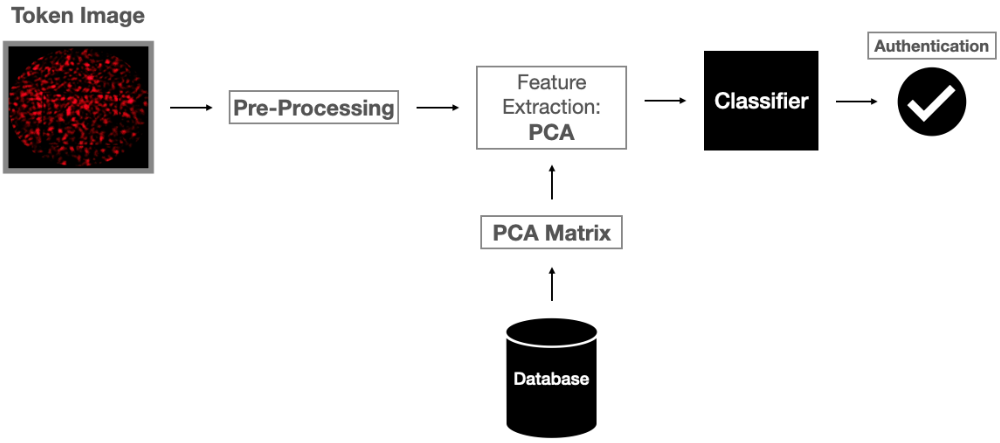


Figure 4.17: Authentication Phase - Approach 4: Authentication phase for the PCA based implementation using a classifier.

4.6 Discussion of the Proposed Approaches

It is clear that there are inherent differences between the proposed authentication methods. These differences can be selective in the sense that a method would work in a certain application scenario, but not in another. An objective analysis of the advantages and drawbacks of all implemented approaches is important to aid their selection for specific purposes. The main application scenario, aside from cryptographic key generation, is utilizing the PUF system to authenticate important objects/documents. However, in what circumstances and environment this authentication process occurs is what makes some methods better than others. Not only in terms of speed and performance, but also in terms of inherent concepts of the implementations. For example, some approaches have features that could be problematic in a security perspective, i.e. storing the full speckle pattern image in a database.

4.6.1 Advantages and Drawbacks: Gabor filtering

Gabor filtering is an important step in these implementations to improve their robustness and discrimination, allowing for a better authentication performance. It is also utilized in all proposed approaches. Gabor kernels can emphasize features that are more robust through different acquisitions of the same token. These are, in this specific type of token (paper tokens), diagonal features of the speckle pattern.

However, as we have seen in chapter 2, it also has it´s drawbacks. Gabor Kernels have an inherent orientation, making the filtering process sensitive to rotation attacks. One main drawback derived from this fact is that, in certain types of PUFs, unwanted rotation of the token is a common problem. In paper based PUFs though, rotation of the token is less likely when compared to, for example, optical waveguide based PUFs. If the algorithms implemented in this Thesis were to be applied in these types of PUFs, the rotation sensitivity of Gabor filters would be an important topic to study. There have been various proposed implementations of rotation invariant Gabor filters, like [24], [35] or [36], which could be combined with the proposed approaches of this present work.

4.6.2 Advantages and Drawbacks: DCT versus PCA Based Approaches for the PUF Scenario

The clear main advantage of the DCT based approaches is that there is no need to store the actual speckle pattern images. This means that there is only one database to store the already computed hash values, which obviously take up significantly less space than entire speckle pattern images. This is because the DCT is data-independent, and does not need a dataset to be defined against. The PCA needs to be defined with respect to a certain dataset, which would need to be stored in another database. When a new token image is registered, its speckle pattern images are stored in this database and the PCA is computed again. This makes the PCA based approaches require significantly more memory than the DCT based ones. It is important to note, however, that we only need to keep adding new speckle pattern images to the matrix, on which PCA is computed on, until we have an eigenspace large enough to faithfully describe any new speckle pattern that is not yet registered in the system.

5

Experimental Results

Contents

5.1 Introduction 54

5.2 Experimental Setup 54

5.3 Performance Results with Hamming Distance Based Classification 59

5.4 Performance Results with Machine Learning Classifiers 67

5.5 Discussion of the Results 71

5.1 Introduction

After implementing the previously described algorithms, it is essential to analyse how they perform with both obtained datasets. This chapter covers the results obtained for each of the implemented methods with different classification approaches.

First, a brief overview of the experimental setup is presented. In this section, the metrics utilized to evaluate the results are discussed, as well as the parameter tuning that was performed to tailor the developed algorithms to the obtained datasets. The computational resources utilized are also presented.

Afterwards, the obtained results are presented and discussed. A brief discussion of possible applications of each method, in the light of the obtained results, is also done.

5.2 Experimental Setup

In this section, the experimental setup for the testing of the implemented algorithms is presented. In regards to data used for testing, as it was previously described in Section 3.3, two datasets (dataset A and dataset B) were considered. Each dataset utilizes 4 different PUF tokens and contains 400 images (100 for each token under different conditions).

It is important to note that all methods (except the PCA and machine learning classifiers) utilized in these algorithms are from the OpenCV Python package. The PCA implementation and machine learning classifiers utilized belong to Scikit Learns's Python package

5.2.1 Evaluation Metrics

First, it is important to define the evaluation metrics utilized in this chapter. In general, the classification measures obtained in these types of problems are defined in a confusion matrix [37]. An example of a confusion matrix for a binary classification problem is shown in table 5.1.

Table 5.1: Confusion matrix for a binary classification problem.

	Predicted Class	
True Class	Positive	Negative
Positive	TP	FN
Negative	FP	TN

Where the concepts true positive (TP), false negative (FN), false positive (FP) and true negative (TN) can be explained as:

- True positive means it was correctly predicted as positive.
- False negative means it was incorrectly predicted as negative.

- False positive means it was incorrectly predicted as positive.
- True negative means it was correctly predicted as negative.

The accuracy rate (Acc), which can be derived from the confusion matrix, is a widely used and practical evaluation metric. It evaluates the performance of the classifier by means of its percentage of correct predictions. The Acc is computed as in Equation (5.1).

$$Acc = \frac{TN + TP}{FN + FP + TN + TP} \tag{5.1}$$

The error rate (Ecc) is another useful metric, which translates the percentage of incorrect predictions done by the classifier. Both the Acc and Ecc are general measurements that can be extended to multi-class classification problems. The Ecc is computed as in Equation (5.2).

$$Ecc = \frac{FN + FP}{FN + FP + TN + TP} = 1 - Acc \tag{5.2}$$

To evaluate the performance of each proposed approach, the Acc and Ecc are utilized.

5.2.2 Parameter Tuning

Several different tools are used in the implementations of the proposed solutions. Each of these methods and processes have parameters that need to be tailored to their application environment. In this section, the choices for what parameters to use for each process will be discussed.

5.2.2.A Speckle Pattern Image Resizing

First, the size to which images should be resized to should be discussed. This choice, specifically for authentication purposes, can be troublesome in certain applications. On one hand, keeping large images can provide better discrimination and robustness. However, it comes at the cost of drastically increasing computational times. On the other hand, reducing image sizes can negatively impact authentication results, but authentication and registration times are much lower. In the end, an adjustment should be done taking into consideration the application scenario, with the intent of minimizing the computational effort while achieving the desired levels of discrimination and robustness of the algorithm.

In this analysis, image sizes will be chosen with the aim of keeping processing times low, while achieving satisfactory authentication results. Based on some past research [29], from which the DCT based proposed approaches are based on, resizing images to 32x32 pixels seems to work well. When it comes to the PCA based approaches, it has been shown by Yuen et al. [38] and also in [39] that a face image with a resolution of 16×16 is enough for authentication using PCA. Because of this, we consider the DCT based approaches as the bottleneck in image size selection. With this in consideration, for the tests to decide what image size to use, the DCT is utilized. Image sizes from 16x16 to 512x512 are examined. The process consists of the following steps:

1. Resize all images to a specific size from the following set: [16x16, 32x32, 64x64, 128x128, 256x256, 512x512]

2. Use approach 1, described in Section 4.3, without the Gabor filtering module, to calculate a hash value in a subset of the dataset. To reduce the computational burden, only 5 images of each token are used for this purpose. The reason Gabor filtering is not used at this stage is because the filters need to be tailored to each image size, and therefore image size selection should be done before analysing which Gabor kernels work best.

3. Compare the hamming distance of each image with every other image in the subset of the dataset.

4. Store the mean of the hamming distance values for each image against images of the same token and the mean of the hamming distance values for each image against images different tokens.

5. Store the time taken to complete this process.

6. Repeat the process with different image sizes.

The most suitable image size is then selected with the purpose of maximizing the difference between the mean inter-hamming distance (between images of different tokens) and the mean intra-hamming distance (between images of the same token), as well as the computational time taken. This is plotted, for each image size, in Figure 5.1.

In this figure, each dot is the average result for a specific image size. On the y axis we have the difference between the mean inter hamming distance and the mean intra hamming distance. On the x axis the time it took to process the entire subset of the dataset is displayed. It is clear that resizing images to 16x16 or 32x32 provides the worst performance in terms of authentication capability and resizing to 512x512 takes considerably longer to process. The optimal image size will be the one closest to the ”elbow” of the blue line, i.e. 64x64. Resizing images to 64x64 provides nearly as much authentication capability as bigger image sizes and is just a little slower than smaller image sizes.

It is important to decide on the image size to be used for all the proposed algorithms, in order to select the optimal setting of the Gabor filters, as the results obtained with a specific Gabor kernel are dependent on image size.

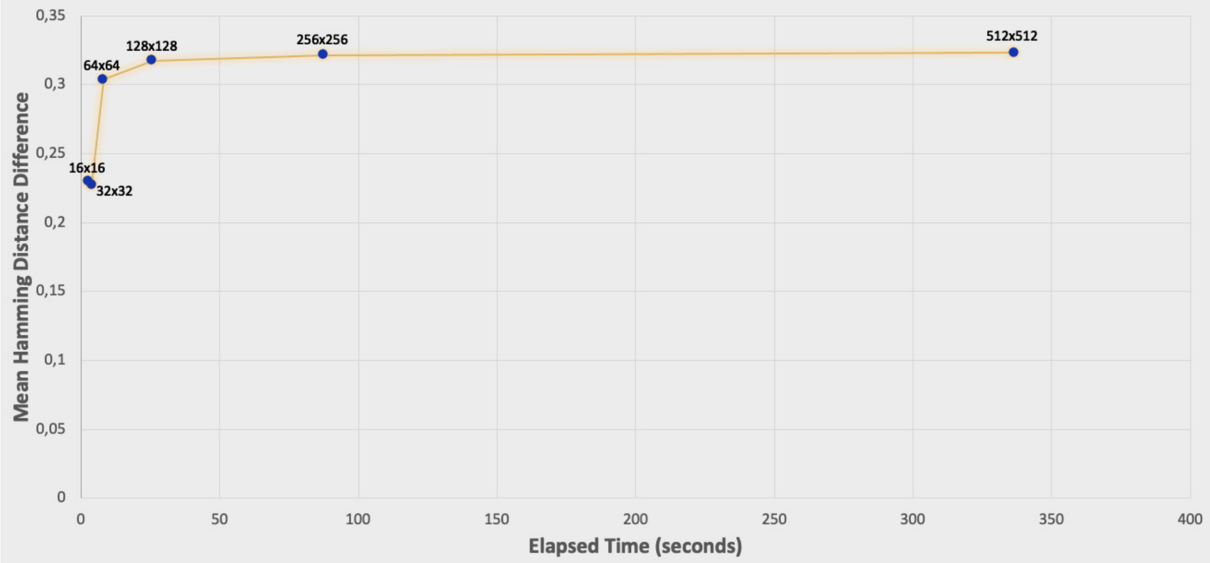


Figure 5.1: Impact of resizing the speckle pattern images to different sizes in terms of computational time taken and hamming distance difference between the same tokens and different tokens.

5.2.2.B Gabor Kernel Parameters

Once the image size is selected, the Gabor kernel utilized should be discussed. When generating Gabor kernels, three parameters have a larger impact, in the resulting filtered image, when compared to the others. These are the wavelength (or frequency), orientation and the scale of the kernel [40]. A limited number of Gabor kernels in a subset of the full parameter space is considered to reduce the computational cost of this analysis. For instance, the orientation of the Gabor kernel will be kept at $\frac{\pi}{4}$ relative to the x axis. As explained in the previous chapter, this orientation reduces the impact of unwanted vertical and horizontal changes in camera positioning. The size of the kernel will also be set at 3x3, as it is one of the most common choices [41]. This reduces computational time, which is especially important when training machine learning classifiers and calculating the PCA. This means that the only variable left to be selected is the frequency of the kernel.

Gabor kernels, as described in Section 2.3, have a real and imaginary part. In [7] it was shown how the imaginary part of Gabor kernels are less affected by changes on the ambient light level of the speckle image. Because there are bound to be small fluctuations due to either changes in the lighting of the environment or the power of the laser itself, it is useful to just consider the imaginary part when filtering the speckle pattern images.

The process to select which Gabor filter best applies to dataset A is the following:

1. Select a specific frequency and generate a Gabor kernel accordingly.

2. Apply the generated Gabor kernel to a subset of the entire dataset (to reduce computational burden, only 5 images of each token are used) and utilize approach 1 from section 4.3 to calculate a hash value.

3. Compare the hamming distance of each image with every other image in the subset of the dataset.

4. Store the mean of the inter hamming distance values and the mean of the intra hamming distance values.

5. Repeat the process with a different frequency.

The Gabor filter which best applies to this specific type of speckle pattern image will be the one that maximizes the difference between the mean inter-hamming distance and the mean intra-hamming distance. It is important to note that all images were previously resized to 64x64 pixels. These results are shown in Figure 5.2.

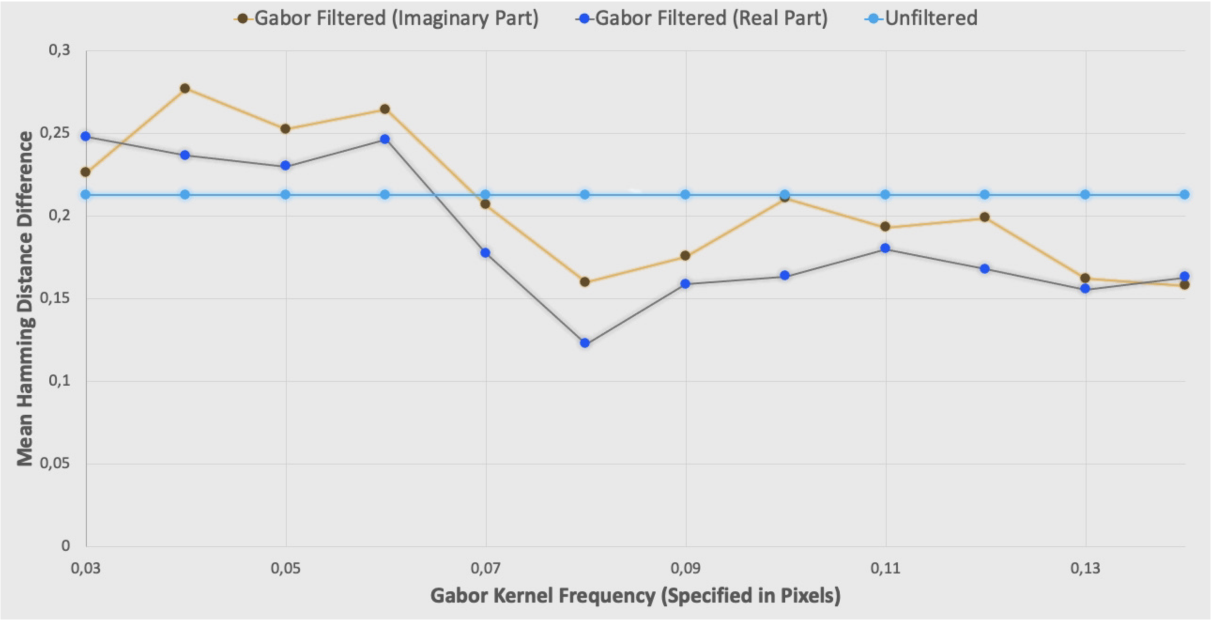


Figure 5.2: Impact of Gabor filter frequency in authentication performance, comparing the mean hamming distance difference. It is possible to see the benefit of considering only the imaginary part of the Gabor kernels.

From Figure 5.2, the benefit of utilizing only the imaginary response from Gabor filtering is perceivable. Namely, selecting a spatial frequency of the harmonic function from 0.04 to 0.06 (specified in pixels) achieves the largest increase in discrimination of the speckle patterns. Because of this, the Gabor kernel selected for use in the pre-processing stage of the algorithms has the parameters summarized in Table 5.2.

Table 5.2: Main characteristics of the Gabor kernel used in the pre-processing stage of the implemented algorithms.

Gabor Kernel	
Characteristic	Value
Size	3x3
Orientation	$\frac{\pi}{4}$
Frequency	0.04

5.3 Performance Results with Hamming Distance Based Classification

In this section, the authentication results obtained with each proposed approach and a hamming distance based classification are presented and discussed.

5.3.1 Testing with Dataset A

This section reports the tests conducted using both the DCT and the PCA based algorithms in combination with a normalized hamming distance based classification. Systems performing well in terms of discrimination and authentication should achieve low intra-HDs and high inter-HDs.

5.3.1.A DCT based algorithm

The DCT based algorithm is tested both with and without the Gabor filter. To help visualize the difference between computed hashes from the same token and from different tokens, the hamming distance values can be associated with a certain color. Results obtained from the DCT based approach with Gabor filtering are presented in Figure 5.3. There are 100 photos for each PUF token (50 with standard camera positioning and 50 with varied camera positioning). The green blocks are intra-HD (from the same tokens) and the rest is inter-HDs (from different tokens). However, there are some yellow/red lines going through some green blocks. This can be due to a larger variability in the image acquisition environment that altered the obtained speckle pattern, resulting in a less than ideal hash value. These substandard hashes reduce the authentication performance. Nevertheless, despite these yellow/red lines, from Figure 5.3 it is possible to understand that the algorithm is performing adequately, having mostly green intra-HDs and yellow/red inter-HDs

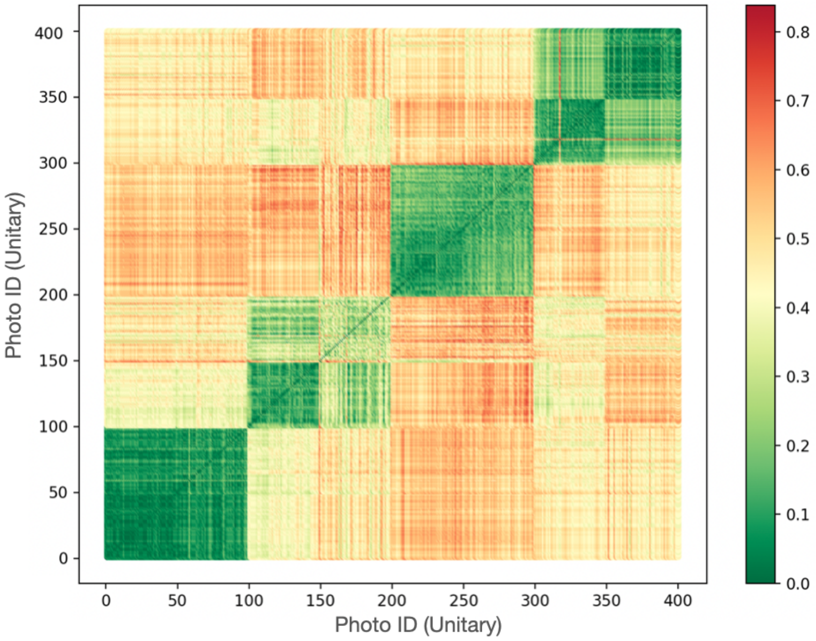


Figure 5.3: Map of the normalized hamming distances, obtained with the DCT based algorithm on dataset A. Higher hamming distances are associated with a progressively redder colour. Lower hamming distances are associated with a green colour.

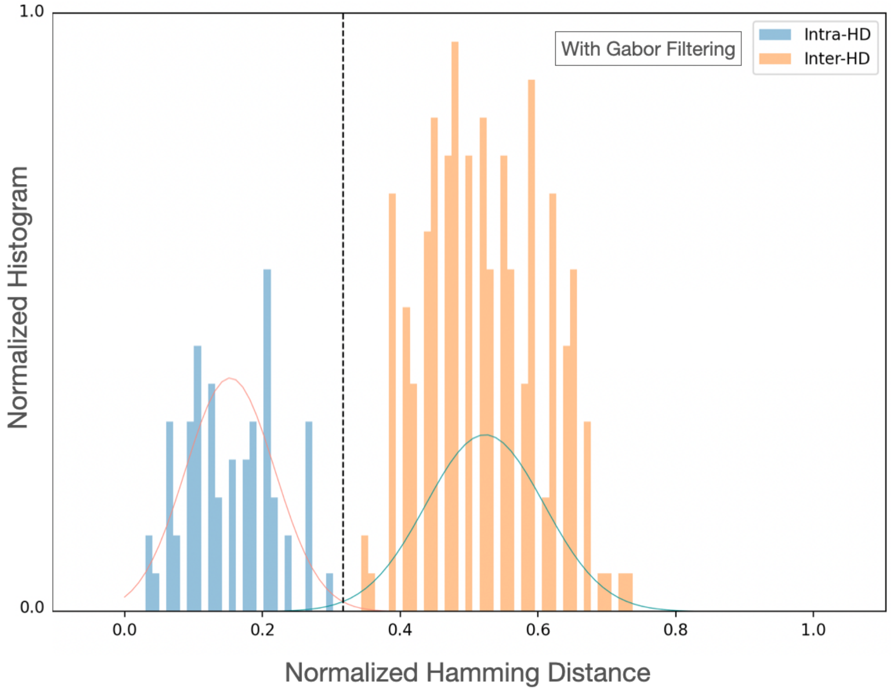
Table 5.3: Metrics obtained from the hamming distance histograms of the DCT based approach on dataset A.

Algorithm	$\mu_{inter-HD} - \mu_{intra-HD}$	$\mu_{intra-HD}$	$\mu_{inter-HD}$	$\sigma_{intra-HD}$	$\sigma_{inter-HD}$
With Gabor Filtering	0.317	0.153	0.523	0.064	0.085
Without Gabor Filtering	0.285	0.138	0.455	0.066	0.081

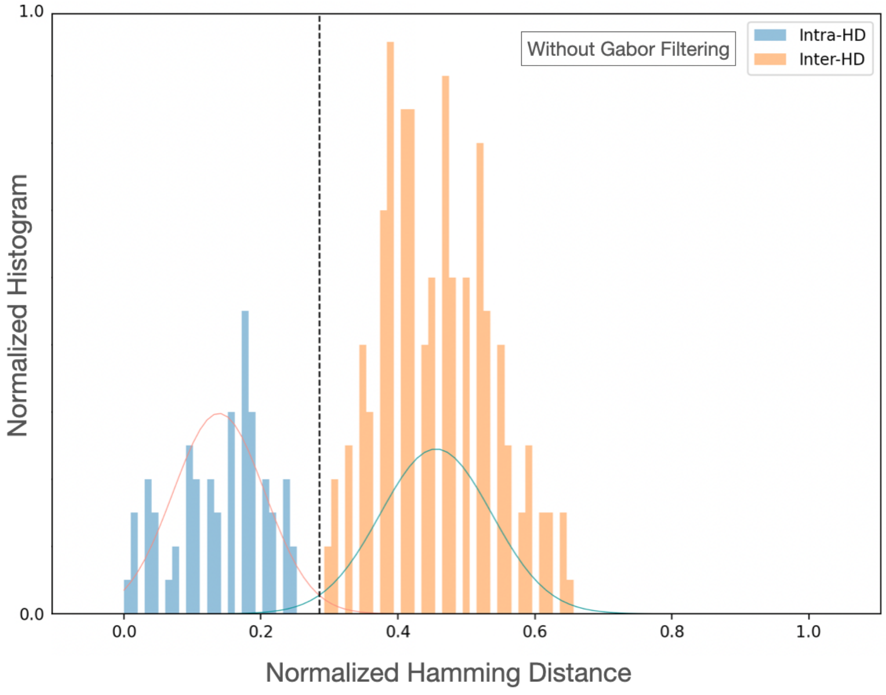
The hamming distances obtained by comparing the hashes generated with the dataset can also be plotted using an histogram. The resulting metrics are utilized for performance evaluation, as previously described. To better evaluate the impact of Gabor filtering, the algorithms were tested both with and without it (the pre-processing pipeline was kept the same, except for Gabor filtering removal). It is also important to note that the algorithms generate 64 bit hashes, as previously mentioned in section 4.3.1, and that only the imaginary part resulting from Gabor filtering is kept. The obtained histograms are presented in Figure 5.4. The adapted normal distribution curves are also plotted. The optimal decision threshold, which was discussed in Section 4.4, was also computed and plotted as a vertical dotted line in the histograms.

It is important to analyse the specific metrics of each normal distribution to accurately evaluate the performance of each algorithm. These values are presented in Table 5.3.

Lower variance (σ) values are beneficial, as well as a higher difference between the mean of the inter-HDs and intra-HDs ($\mu_{inter-HD}$ and $\mu_{intra-HD}$). Both algorithms performed adequately, achieving



(a)



(b)

Figure 5.4: (a) Histogram obtained with the DCT based algorithm and Gabor filtering on dataset A; (b) Histogram obtained with the DCT based algorithm without Gabor filtering on dataset A;

Table 5.4: Metrics obtained from the hamming distance histograms of the PCA based approach on dataset A.

Algorithm	$\mu_{inter-HD} - \mu_{intra-HD}$	$\mu_{intra-HD}$	$\mu_{inter-HD}$	$\sigma_{intra-HD}$	$\sigma_{inter-HD}$
With Gabor Filtering	0.324	0.156	0.480	0.059	0.071
Without Gabor Filtering	0.182	0.345	0.527	0.177	0.138

a clear distinction between PUFs obtained from different tokens and overcoming the inevitable intra-variability that is always introduced in acquistions.

In this setting, both algorithms correctly classified all speckle pattern images. From the obtained results, however, it is possible to see that Gabor filtering provided a slight edge in performance. Although the variance values are similar, the difference between mean values is higher with the algorithm that employed Gabor filtering.

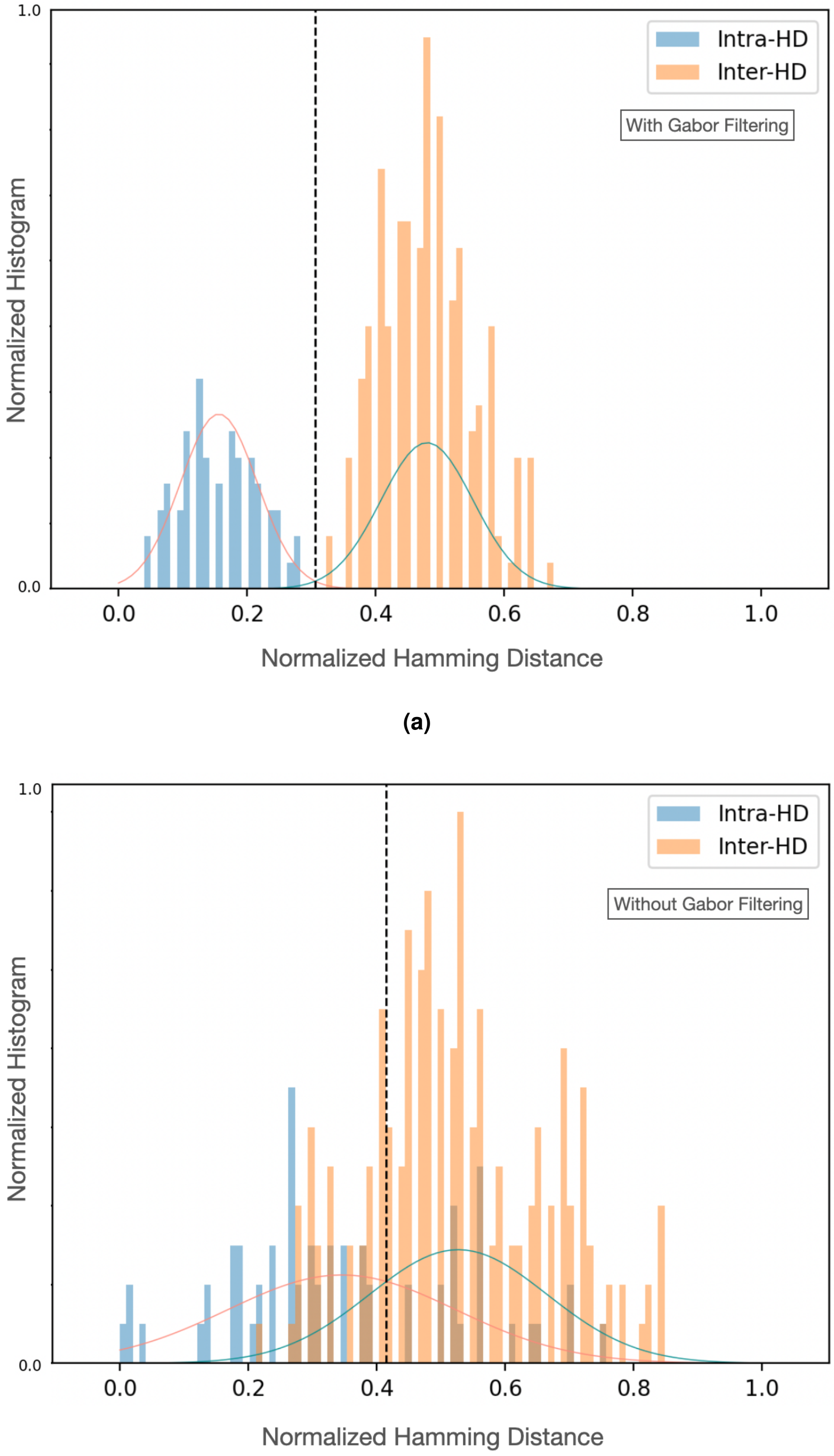
5.3.1.B PCA based algorithm

The PCA based algorithm was also tested with dataset A under the same settings as the DCT based algorithms. The resulting histograms are presented in Figure 5.5.

These results are obtained by utilizing a 64 bit hash (utilizing 64 principal components), which is the same hash length as for the DCT based algorithms. This allows for a more accurate comparison between the two methods. For the generation of the PCA matrix, on which new images are projected upon, 20 images from each token are utilized (10 images with standard camera orientation and 10 images with varied camera orientations). The remainder of the dataset is then utilized for testing. It is important to note that, with the algorithm that utilizes Gabor filtering, 10 of these 20 images are the resulting real part and the other 10 are the resulting imaginary part derived from the Gabor kernel. When new images are projected on the PCA matrix, only the imaginary part of the Gabor filtering is utilized. This combination provided superior classification performance. This is likely due to an increase of the eigenspace generated, allowing more faithful descriptions of new speckle pattern images.

From Figure 5.5, it is possible to see that Gabor filtering had a larger impact in the classification capacity of the algorithm. In Table 5.4, the specific metrics of the histograms are presented.

The PCA based algorithm that did not utilize Gabor filtering performed significantly worse, having an Acc score of 76,5% (the other algorithms showed a clear distinction between intra-HDs and inter-HDs, which translates into an Acc of 100%). It is important to point out that the quantization method utilized likely results in a large amount of data loss, as each hash coefficient is binarized only with relation to the mean of the entire hash.



(a)

(b)

Figure 5.5: (a) Histogram obtained with the PCA based algorithm and Gabor filtering on dataset A; (b) Histogram obtained with the PCA based algorithm without Gabor filtering on dataset A;

5.3.2 Testing with Dataset B

The same testing process performed on dataset A was followed for the tests with dataset B. The changes performed during the acquisition process of dataset B were done with the intention of aiding the automation of cropping the raw speckle patterns images. The output of the pre-processing stage of the algorithms should be similar. Dataset B should allow the proposed algorithms to achieve a slight increase in robustness, as the automatic image cropping procedure is expected to avoid some imprecision that may result when doing manual cropping.

5.3.2.A DCT based algorithm

The obtained results when applying the DCT based algorithm are presented in Figure 5.6, and the metrics associated with the plotted histograms are presented in Table 5.5.

Table 5.5: Metrics obtained from the hamming distance histograms of the DCT based approach on dataset B.

Algorithm	$\mu_{inter-HD} - \mu_{intra-HD}$	$\mu_{intra-HD}$	$\mu_{inter-HD}$	$\sigma_{intra-HD}$	$\sigma_{inter-HD}$
With Gabor Filtering	0.340	0.152	0.492	0.056	0.052
Without Gabor Filtering	0.372	0.093	0.465	0.042	0.041

While in the previous dataset Gabor filtering showed an improvement in classification performance, in this dataset the same is not observed. Although both algorithms performed at 100% Acc, Gabor filtering resulted in a slight decrease the metrics presented in Table 5.5. This emphasizes the fact that, although beneficial, Gabor filtering is largely sensitive to the image acquisition environment. Since the Gabor kernel was tuned with dataset A, its performance might improve if tuned again for the conditions of dataset B. However, since the achieved accuracy was still 100%, it was considered that the previous setup can be used for both datasets.

5.3.2.B PCA based algorithm

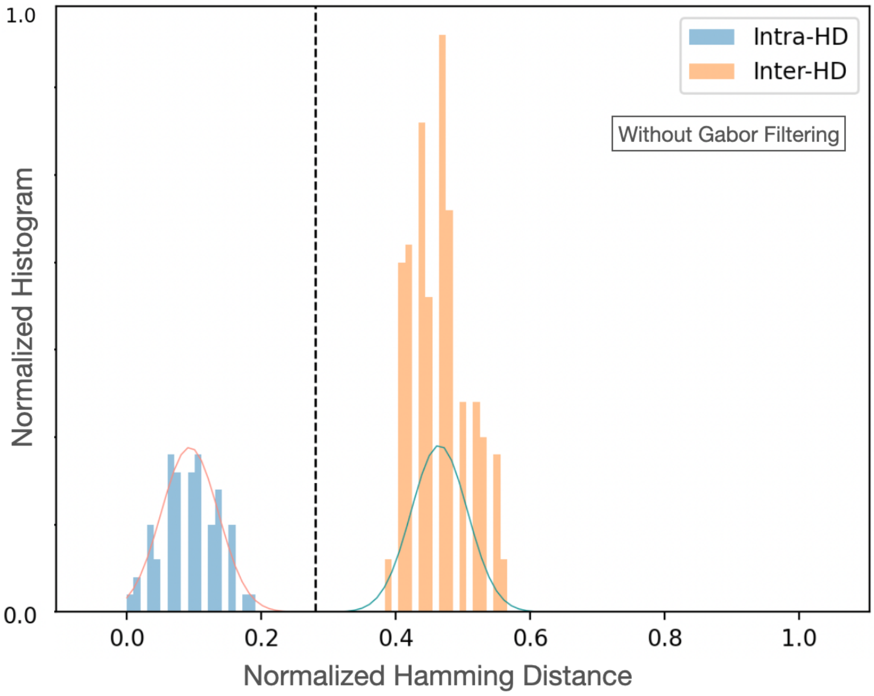
The PCA based algorithm was also tested with dataset B under the same settings as the DCT based algorithms. The obtained results, in the form of histograms, are presented in Figure 5.7.

The metrics associated with the plotted histograms are presented in table 5.6.

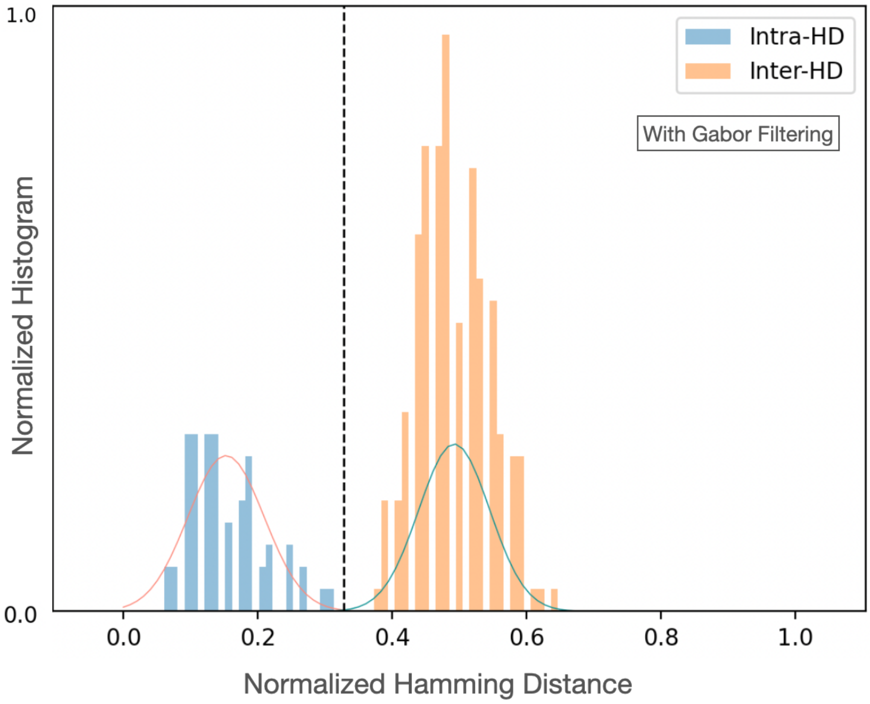
Table 5.6: Metrics obtained from the hamming distance histograms of the PCA based approach on dataset B.

Algorithm	$\mu_{inter-HD} - \mu_{intra-HD}$	$\mu_{intra-HD}$	$\mu_{inter-HD}$	$\sigma_{intra-HD}$	$\sigma_{inter-HD}$
With Gabor Filtering	0.161	0.375	0.536	0.103	0.080
Without Gabor Filtering	0.191	0.347	0.538	0.157	0.143

From the results we can conclude the PCA based algorithm does not provide satisfactory results in this scenario. Also, when using Gabor filtering a decrease in performance in dataset B when compared

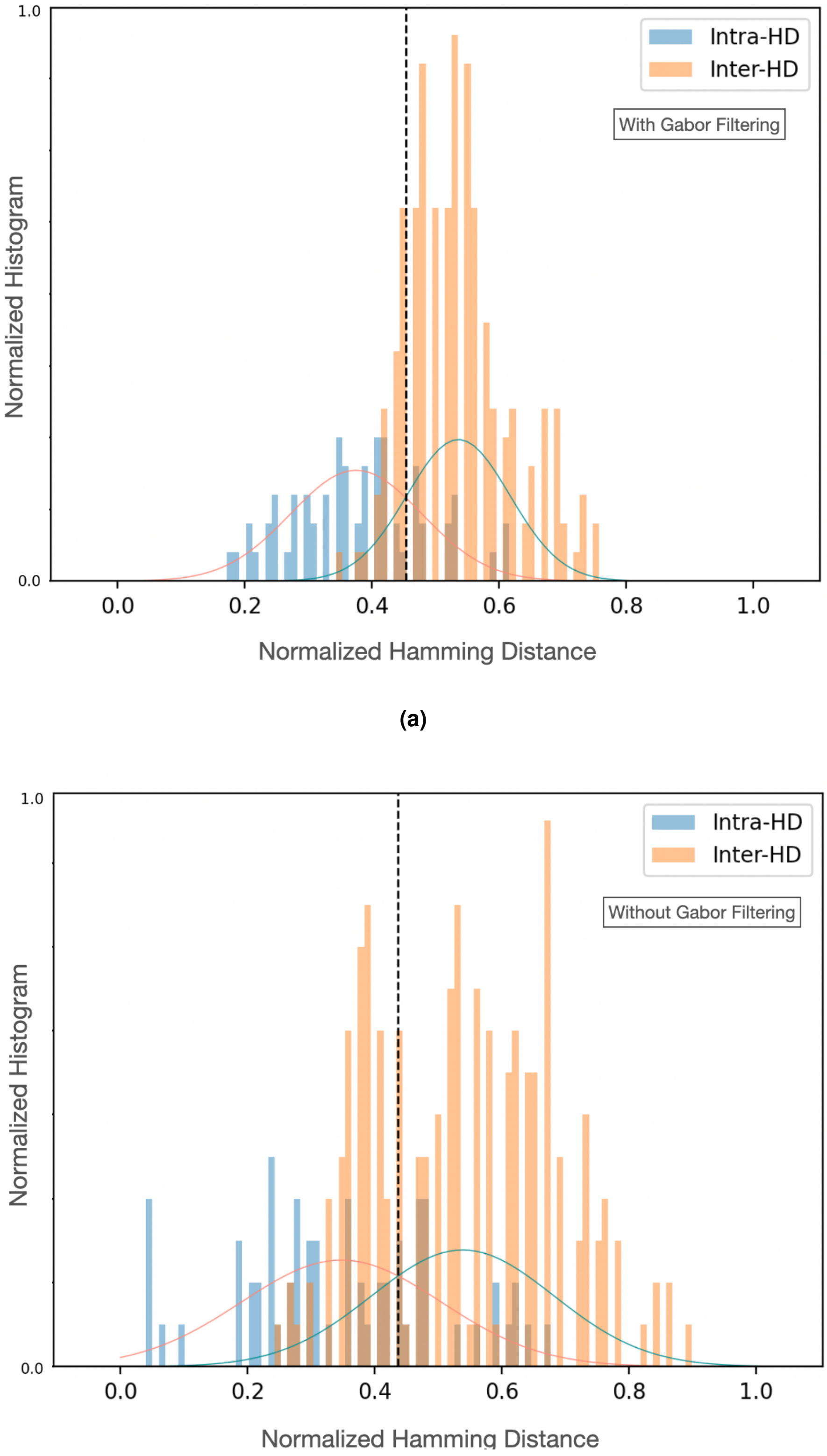


(a)



(b)

Figure 5.6: (a) Histogram obtained with the DCT based algorithm and Gabor filtering on dataset B; (b) Histogram obtained with the DCT based algorithm without Gabor filtering on dataset B;



(a)

(b)

Figure 5.7: (a) Histogram obtained with the PCA based algorithm and Gabor filtering on dataset B; (b) Histogram obtained with the PCA based algorithm without Gabor filtering on dataset B;

with dataset A was observed. The PCA based algorithm with Gabor filtering obtained an 86,3% Acc score, while the algorithm without Gabor filtering obtained a 69,9% 86,3% Acc score, showing that in terms of classification performance Gabor filtering still showed an improvement. This improvement should likely increase if the Gabor kernel were tuned to this specific dataset.

5.4 Performance Results with Machine Learning Classifiers

In this section, the authentication results obtained with each proposed approach and machine learning based classification are evaluated. The metrics discussed in the previous section are also utilized here for evaluation.

In the evaluation stage of the supervised machine learning methods, overfitting is an important concern to take into consideration [42]. A model that overfits the training data will fail to accurately fit the observed data on the test dataset. Since overfitting typically occurs when the amount of training data is limited, a common solution to this problem is called cross-validation [43]. In short, it consists of randomly partitioning the dataset into N mutually exclusive subsets, all of approximately equal size. One partition is kept for testing, while all the others are used to train the classifier. This process is iterated N times. Cross-validation is employed in this dissertation to prevent overfitting the data.

All four machine learning classifiers considered in section 4.4, as well as the cross-validation functions, are implemented using the Scikit Learn Python package (*sklearn*). To test the developed algorithms, the datasets are split into 10 subsets, with two of these subsets being the test set. This means that the classifiers are trained with 80% of the dataset and tested with the remaining 20% of data, translating into a 5-fold cross validation scheme.

5.4.1 Testing with Dataset A

This section reports the tests conducted using both the DCT and the PCA based algorithms in combination with machine learning classifiers. Systems performing well in terms of discrimination and authentication should achieve high accuracy rates in the classification of tokens.

5.4.1.A DCT based algorithm

The DCT based algorithm was tested both with and without Gabor filtering, to assess its impact in the overall performance of the system. To objectively study the performance of each machine learning classifier with the proposed algorithm, the previously described Acc and Err metrics are utilized.

The results obtained with the DCT based algorithm and the different classifiers are shown in Table 5.7.

Table 5.7: Authentication results with dataset A and the DCT based approach.

DCT Based Approach						
	with Gabor Filtering			without Gabor Filtering		
Classifier	Accuracy Rate (Acc)	Standard Deviation	Err	Acc	Standard Deviation	Err
SVM (Linear Kernel)	0.98	0.01	0.02	0.39	0.03	0.61
SVM (Polynomial Kernel)	0.86	0.06	0.14	0.28	0.04	0.72
Decision Tree	0.96	0.03	0.04	0.49	0.03	0.51
k-Nearest Neighbour	0.96	0.02	0.04	0.43	0.05	0.57
Random Forest	0.98	0.02	0.02	0.49	0.04	0.51

From the displayed results, it is possible to conclude that the proposed algorithm including Gabor filtering, with any of the tested machine learning classifiers, achieves satisfactory classification/authentication performance.

Both SVM (with a linear kernel) and Random Forest classifiers achieved the highest accuracy, being correct in 98% of the classifications. SVM with a polynomial kernel performed the worst. This might seem counter intuitive, as a polynomial kernel seems instinctively be better than a linear kernel, but the explanation is probably that overfitting is occurring because the kernel has a higher complexity. In that case the classifier excessively adapts the decision boundary to the training data and later struggles to correctly classify new data.

It is important to note, however, that these results are obtained with a somewhat restricted dataset of only 4 different tokens with 400 images. Were these algorithms tested with a larger dataset, the performance would likely decrease.

When Gabor filtering is removed, there is a significant decrease in the authentication performance. Under these conditions, no combination of algorithm + machine learning classifier achieves an accuracy higher than 50%. This reinforces the efficacy of using the selected Gabor kernel in the optical PUF scenario.

5.4.1.B PCA based algorithm

The PCA based algorithm was also tested on dataset A with and without Gabor filtering (the pre-processing pipeline was kept the same, except for the Gabor filter). The results are presented in Table 5.8.

It is possible to conclude that the PCA based approach, combined with any of the machine learning

Table 5.8: Authentication results with dataset A and the PCA based approach.

PCA Based Approach						
	With Gabor Filtering			Without Gabor Filtering		
Classifier	Acc	Standard Deviation	Error Rate (Ecc)	Acc	Standard Deviation	Ecc
SVM (Linear Kernel)	0.99	0.01	0.01	1.00	0.00	0.00
SVM (Polynomial Kernel)	0.90	0.03	0.10	0.75	0.11	0.25
Decision Tree	0.97	0.02	0.03	0.99	0.01	0.01
k-Nearest Neighbour	0.96	0.03	0.04	0.98	0.01	0.02
Random Forest	0.95	0.02	0.05	0.98	0.02	0.02

classifiers achieves a very good authentication performance. It is important to reiterate that these results might decrease if the dataset was larger, including more PUF tokens. Notably, the accuracy results obtained for the SVM with a linear kernel and without Gabor filtering would likely not be 100%.

In this result set, the difference between an algorithm with Gabor filtering and another without it is not as apparent. For four of the classifiers (all except the SVM with a polinomial kernel) the accuracy results seem to have slightly decreased when including Gabor filtering. This could be explained by the fact that these results are all very high, and in many of the cases the slight decrease with Gabor filtering is not meaningful. In fact, if the results obtained by employing the SVM with a polynomial kernel are analysed, a big increase in performance is seen by utilizing Gabor filtering. Here, the SVM with polynomial kernel is probably overfitting the training data, as previously observed for the results with the DCT based approach.

5.4.2 Testing with Dataset B

The same testing process done in dataset A, described in the previous subsection, was replicated in dataset B. The performance obtained with this dataset shouldn't be significantly different from the one obtained with dataset A.

5.4.2.A DCT based algorithm

The results obtained for the DCT based approach combined with machine learning classifiers are presented in Table 5.9.

The displayed results support the efficacy of this algorithm when applied to the considered PUF

Table 5.9: Authentication results with dataset B and the DCT based approach.

DCT Based Approach						
	With Gabor Filtering			Without Gabor Filtering		
Classifier	Acc	Standard Deviation	Ecc	Acc	Standard Deviation	Ecc
SVM (Linear Kernel)	0.99	0.01	0.01	0.89	0.03	0.11
SVM (Polynomial Kernel)	0.68	0.03	0.32	0.51	0.04	0.49
Decision Tree	0.95	0.02	0.05	0.90	0.03	0.10
k-Nearest Neighbour	0.97	0.02	0.03	0.89	0.05	0.11
Random Forest	0.97	0.02	0.03	0.88	0.02	0.12

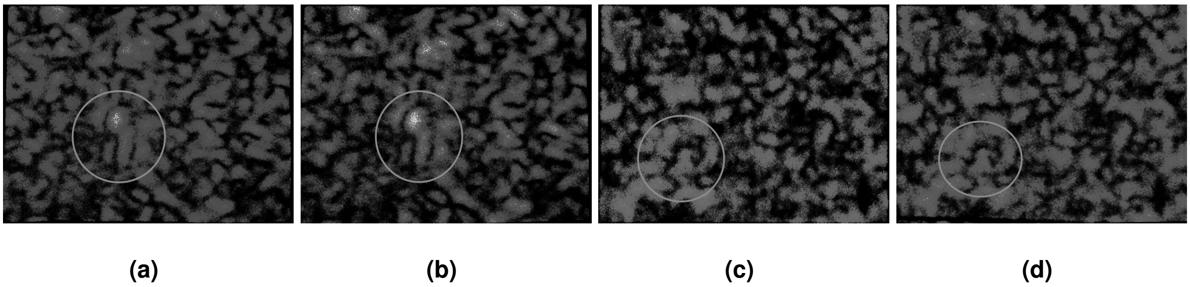


Figure 5.8: (a) Dataset B - Image (taken with standard camera position/orientation) after automatic cropping. Dimensions of the circled feature are approximately 150x245 pixels; (b) Dataset B - Image (taken with different camera position/orientation) after automatic cropping and warping. There is a slight variation in alignment and size of features. Dimensions of the circled feature are approximately 150x240 pixels; (c) Dataset A - Image (taken with standard camera position/orientation) after manual cropping. Dimensions of the circled feature are approximately 185x220 pixels; (d) Dataset A - Image (taken with different camera position/orientation) after manual cropping and automatic warping. There is a larger variation in alignment and size of features. Dimensions of the circled feature are approximately 155x200 pixels;

system, as they are satisfactory for most of the combinations considered.

Comparing to dataset A, there is a decrease in the difference of performance between the algorithms that employ Gabor filtering and the ones that don't. In fact, all the accuracy results obtained from algorithms without Gabor filtering increased with dataset B in comparison to dataset A. This could be due to the fact that the cropping and warping process was optimized. The speckle patterns are better aligned with each other and that could lead to superior comparison results. This slight improvement in alignment can be seen in the details presented in Figure 5.8.

In this scenario, however, not all accuracy results improved. In fact, the SVM with a polynomial kernel and Gabor filtering performed worse with dataset B than with dataset A. Here, the fact that overfitting is likely happening should be reiterated. With much less variability in the training data, the decision boundary is likely excessively adapted to the training data. When new images from the test set are classified, any small variation will likely lead to an incorrect classification, reducing the accuracy results.

5.4.2.B PCA based algorithm

The PCA based algorithm was also tested on dataset B, with and without Gabor filtering, in the same way as in dataset A . The results are displayed in Table 5.10.

Table 5.10: Authentication results with dataset B and the PCA based approach.

PCA Based Approach						
	With Gabor Filtering			Without Gabor Filtering		
Classifier	Acc	Standard Deviation	Ecc	Acc	Standard Deviation	Ecc
SVM (Linear Kernel)	0.99	0.01	0.01	1.00	0.03	0.00
SVM (Polynomial Kernel)	0.85	0.03	0.15	0.74	0.06	0.49
Decision Tree	0.98	0.01	0.02	0.99	0.02	0.10
k-Nearest Neighbour	0.97	0.02	0.03	0.99	0.05	0.01
Random Forest	0.94	0.02	0.04	0.97	0.02	0.12

In dataset B, the PCA based algorithm performs equally well. All accuracy results are similarly high. There is a slight decrease in performance with the SVM when utilizing a polynomial kernel. This is likely due to overfitting being exacerbated, as discussed earlier. Again, these accuracy results would likely decrease if the dataset contained images from more tokens.

5.5 Discussion of the Results

In Section 4.1.1, the possible application scenarios for this authentication system were discussed. The scenarios are again included here for convenience of the reader:

- **Scenario A**: A system with an integrated camera and computer, where registration and authentication time are not the main concern. The main objective is to provide very accurate authentication results. This could happen in the context of, for example, authenticating important documents or utilizing a PUF to gain entry into a highly secure system.
- **Scenario B**: A system where the camera and computer are not inherently part of it. In this scenario, computational time of the implemented algorithm and ease of registration of new tokens are

considerably more important. This could happen in the context of, for example, acquiring a speckle pattern image, with a smartphone, to authenticate a piece of equipment that is tagged with a PUF.

Taking into consideration the obtained results for each approach, it can be argued that some of the algorithms work best in certain application scenarios, while others work best in other application scenarios.

The DCT based approach showed good authentication performance when paired with a normalized hamming distance based classification. A clear distinction between intra-HDs and inter-HDs was achieved. Because this approach is entirely data independent, only one speckle pattern image from a certain token is necessary to register it in the system. This makes it more versatile in terms of possible application environments. Regarding the considered application scenarios, scenario B or other similar scenarios would benefit more from this approach.

The PCA based algorithm paired with a normalized hamming distance based classification showed worse authentication performance. However, when combined with machine learning classifiers, the PCA based approach showed remarkably good performance. The previously considered DCT based approach is data independent, and conversely this approach is data dependent. This means that it requires a certain amount of speckle pattern images to register a new token in the system. In fact, during testing, 80% of the datasets was used for training of the algorithm. This means 80 images per token for training (or registering a token), leaving the 20 remaining images for testing. Another point that should be discussed is that every time a new token is registered in the system, a new PCA matrix and a new classifier model need to be computed. Ideally, this is only necessary until a descriptive enough eigenspace from the PCA matrix is obtained, i.e.,when the information gathered from the registered tokens becomes sufficient to represent also previously unseen tokens. However, this can be a computationally intensive task, representing a drawback if having to be repeated every time a new token is registered in the system.

From these characteristics, which include higher authentication performance but an increased need for computational resources and data, it can be concluded that the PCA with machine learning classifiers approach would be best suited for scenario A, where registration and authentication time are not the main concern and the main objective is to provide very accurate authentication results.

6

Conclusions and Future Work

Contents

6.1 Conclusions 74

6.2 Future Work 75

6.1 Conclusions

In this work, it was demonstrated that a PUF system, utilizing low cost tracing paper tokens, can be utilized for authentication purposes, despite intra-variability between acquired speckle pattern images. Obtained speckle patterns from the same token can vary depending on acquisition environment and system alignment. Multiple algorithms were tested, following a modular approach, to assess what works best for this specific PUF system and it's application scenarios. This implementation consisted in: (i) Image acquisition; (ii) Pre-processing for normalization and filtering; (iii) Feature extraction and hash computation; (iv) Classification.

Image acquisition was done with the physical PUF system constructed by [11]. Two datasets were obtained. Both datasets were obtained with the purpose of simulating the intra-variability that would be expected from a regular use of the system. This included turning the system On and Off between acquisitions, creating a change in the phase of the optical signal, and varying the orientation and positioning of the camera in relation to the target. A second dataset was necessary due to limitations, with the first dataset, in automatic cropping of the region of interest of the speckle pattern images due to lack of sufficient discrimination information. By utilizing non-reflective black tape around the region of interest, it becomes much more discernible, allowing for automatic cropping. This is very important in the scope of a fully functioning authentication system, especially with possible application scenarios like smartphone-based authentication.

Pre-processing included cropping, warping, resizing and normalizing the pixel intensities. Cropping needed to be automatic, otherwise it would be a considerable bottleneck of the system. Warping was done via ORB, which showed a satisfactory performance. Different image sizes were tested. All images were resized to 64x64 as that size led to the best trade-off between authentication performance and computational time. The normalization of pixel intensities was necessary to correct any intra-variability in terms of illumination settings. Filtering was also done by utilizing Gabor kernels. Gabor filtering is fairly common in the context of Optical PUFs, and showed an improvement in authentication performance in some of the algorithms. The Gabor kernels utilized had a 45º orientation and only the imaginary part was considered. This improved the overall robustness of some of the algorithms by reducing the impact of image misalignment and illumination defects. The frequency of the Gabor kernel utilized was only calibrated to the first obtained dataset. This resulted in a loss of performance boost when the algorithms were tested with dataset B. This reinforces the idea that Gabor filtering is selective and that it should be tailored to the specific environment where the algorithm is run to maximize performance.

Feature extraction considered two distinct methods: the type-II DCT and PCA. These methods utilize very different mechanisms. A notable difference between them is that the DCT is data independent and PCA is data dependent. Depending on the type of classification, a quantization scheme was employed. This binarization of the resulting hashes showed good results when applied to the DCT. However, when utilized with PCA it led to data loss, which resulted in a decrease in authentication performance.

Classification was done either with a normalized hamming distance based classification or with machine learning classifiers. Here, again, one method is data independent while the other is data dependent. All classification methods considered showed good performance in authentication.

For an entirely data independent approach, the DCT with an hamming distance based classification showed the best results. It is to note that if the PCA based algorithm employed a more adequate quantization scheme, perhaps the results obtained would be matched. This data independent approach has the main benefit of not needing a large amount of data to register a certain token in the system. The PCA with machine learning classifiers showed the best authentication performance out of all the tested algorithms. This is a data dependent approach. The only downside to this approach is that multiple images of a certain PUF token are required to register it in the system and that each time a new token is registered, the PCA matrix needs to be re-generated.

6.2 Future Work

During this work, only the DCT and PCA were employed for feature extraction. It is recognized that many more algorithms are capable of performing image-based hash extraction. It would be beneficial to implement and test other algorithms to improve the robustness of the PUF system and more accurately assess what works best.

Although an hamming distance based classification approach was considered, Error Correction Code algorithms were not employed. They are widely used for this type of problems, as well as in the cryptographic and communication areas in general, to enhance the performance of authentication algorithms by decreasing the error obtained from intra-variability. This would make them a compelling tool to test an implement along side the algorithms proposed in this dissertation.

Bibliography

[1] C. Mesaritakis, M. Akriotou, A. Kapsalis, E. Grivas, C. Chaintoutis, T. Nikas, and D. Syvridis, "Physical Unclonable Function based on a Multi-Mode Optical Waveguide," *Scientific Reports*, vol. 8, no. 1, 2018.

[2] R. Arppe-Tabbara, M. Tabbara, and T. J. Sørensen, "Versatile and Validated Optical Authentication System Based on Physical Unclonable Functions," *ACS Applied Materials and Interfaces*, 2019.

[3] M. Plauth, W. Hagen, F. Feinbube, F. Eberhardt, L. Feinbube, and A. Polze, "Parallel implementation strategies for hierarchical non-uniform memory access systems by example of the scale-invariant feature transform algorithm," in *Proceedings - 2016 IEEE 30th International Parallel and Distributed Processing Symposium, IPDPS 2016*, 2016.

[4] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011.

[5] C. Wei-Lun, "Machine learning tutorial," *Aec-Apc*, 2011.

[6] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering*, 2019.

[7] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 5589, 2002.

[8] C. Herder, M. D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," 2014.

[9] Y. Ou and K. H. Rhee, "A survey on image hashing for image authentication," in *IEICE Transactions on Information and Systems*, vol. E93-D, no. 5, 2010.

[10] L. Du, A. T. Ho, and R. Cong, "Perceptual hashing for image authentication: A survey," *Signal Processing: Image Communication*, vol. 81, 2020.

[11] Tiago Filipe Santos Silvério, "Photonic Implementation of Physically Unclonable Functions," Ph.D. dissertation, Universidade de Aveiro, 2021.

[12] A. Sharma, L. Subramanian, and E. Brewer, “PaperSpeckle: Microscopic fingerprinting of paper,” in *Proceedings of the ACM Conference on Computer and Communications Security*, 2011.

[13] S. Shariati, F. X. Standaert, L. Jacques, and B. Macq, “Analysis and experimental evaluation of image-based PUFs,” *Journal of Cryptographic Engineering*, vol. 2, no. 3, 2012.

[14] Y. Lei, Y. Wang, and J. Huang, “Robust image hash in Radon transform domain for authentication,” *Signal Processing: Image Communication*, vol. 26, no. 6, 2011.

[15] D. Q. Nguyen, L. Weng, and B. Preneel, “Radon transform-based secure image hashing,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7025 LNCS, 2011.

[16] C. P. Yan, C. M. Pun, and X. C. Yuan, “Quaternion-Based Image Hashing for Adaptive Tampering Localization,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, 2016.

[17] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, 2004.

[18] X. Lv and Z. Jane Wang, “Perceptual image hashing based on shape contexts and local feature points,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, 2012.

[19] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded up robust features,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3951 LNCS, 2006.

[20] S. S. Kozat, R. Venkatesan, and M. K. Mihçak, “Robust perceptual image hashing via matrix invariants,” in *Proceedings - International Conference on Image Processing, ICIP*, vol. 2, 2004.

[21] L. Ghouti, “Robust perceptual color image hashing using quaternion singular value decomposition,” in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2014.

[22] Z. Huang and S. Liu, “Robustness and discrimination oriented hashing combining texture and invariant vector distance,” in *MM 2018 - Proceedings of the 2018 ACM Multimedia Conference*, 2018.

[23] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. Upper Saddle River, N.J: Prentice-Hall, Inc., 2007.

[24] Y. Li, Z. Lu, C. Zhu, and X. Niu, “Robust image hashing based on random gabor filtering and dithered lattice vector quantization,” *IEEE Transactions on Image Processing*, vol. 21, no. 4, 2012.

[25] J. P. Jones and L. A. Palmer, “An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex,” *Journal of Neurophysiology*, vol. 58, no. 6, 1987.

[26] Z. Tang, M. Ling, H. Yao, Z. Qian, X. Zhang, J. Zhang, and S. Xu, “Robust image hashing via random gabor filtering and DWT,” *Computers, Materials and Continua*, vol. 55, no. 2, 2018.

[27] G. Strang, “The discrete cosine transform,” *SIAM Review*, vol. 41, no. 1, 1999.

[28] U. Rührmair, C. Hilgers, and S. Urban, “Optical PUFs Reloaded,” *IACR Cryptology*, 2013.

[29] B. Coşkun and B. Sankur, “Video İşaretlerinin Algisal Dayanikli Kiyimi,” in *Proceedings of the IEEE 12th Signal Processing and Communications Applications Conference, SIU 2004*, 2004.

[30] D. R. Sona, R. C. Prayline, D. Dey, A. Jain, R. R. Das, and S. O. Olabiyisi, “A case study: Edge detection techniques using hough transform and canny edge algorithm,” *International Journal of Mechanical Engineering and Technology*, vol. 8, no. 10, 2017.

[31] R. Fitas, B. Rocha, V. Costa, and A. Sousa, “Design and comparison of image hashing methods: A case study on cork stopper unique identification,” *Journal of Imaging*, vol. 7, no. 3, 2021.

[32] C. Zauner, “Implementation and benchmarking of perceptual image hash functions,” *Master’s thesis, Upper Austria University of Applied . . .*, 2010.

[33] G. M. Zafaruddin and H. S. Fadewar, “Face recognition using eigenfaces,” in *Advances in Intelligent Systems and Computing*, 2018, vol. 810.

[34] J. F. Ramalho, L. C. António, S. F. Correia, L. S. Fu, A. S. Pinho, C. D. Brites, L. D. Carlos, P. S. André, and R. A. Ferreira, “Luminescent QR codes for smart labelling and sensing,” *Optics and Laser Technology*, vol. 101, 2018.

[35] J. Han and K. K. Ma, “Rotation-invariant and scale-invariant Gabor features for texture image retrieval,” *Image and Vision Computing*, vol. 25, no. 9, 2007.

[36] M. H. Rahman, M. R. Pickering, and M. R. Frater, “Scale and rotation invariant gabor features for texture retrieval,” in *Proceedings - 2011 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2011*, 2011.

[37] E. P. Costa, A. C. Carvalho, A. C. Lorena, and A. A. Freitas, “A review of performance evaluation measures for hierarchical classifiers,” in *AAAI Workshop - Technical Report*, vol. WS-07-05, 2007.

[38] P. C. Yuen, D. Q. Dai, and G. C. Feng, “Wavelet-based PCA for human face recognition,” *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation*, 1998.

[39] M. B. Ramalho, P. L. Correia, and L. D. Soares, “Hand-based multimodal identification system with secure biometric template storage,” *IET Computer Vision*, vol. 6, no. 3, 2012.

[40] P. Moreno, A. Bernardino, and J. Santos-Victor, “Gabor parameter selection for local feature detection,” in *Lecture Notes in Computer Science*, vol. 3522, no. I, 2005.

[41] S. Ozturk, U. Ozkaya, B. Akdemir, and L. Seyfi, “Convolution kernel size effect on convolutional neural network in histopathological image processing applications,” in *2018 International Symposium on Fundamentals of Electrical Engineering, ISFEE 2018*, 2018.

[42] X. Ying, “An Overview of Overfitting and its Solutions,” in *Journal of Physics: Conference Series*, vol. 1168, no. 2, 2019.

[43] D. Berrar, “Cross-validation,” in *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, 2018, vol. 1-3.