Document authentication using 2D codes: Maximizing the decoding performance using statistical inference

Lamine Diong, Patrick Bas, Chloé Pelle, Wadih Sawaya

Document authentication using 2D codes: Maximizing the decoding performance using statistical inference

Mouhamadou L. Diong[1], Patrick Bas[1], Chloé Pelle[1], and Wadih Sawaya[2⋆]

[1] CNRS-LAGIS, Ecole Centrale de Lille, France firstname.lastname@ec-lille.fr
[2] LAGIS - Telecom-Lille, France wadih.sawaya@telecom-lille1.eu

Abstract. Authentication of printed documents using high resolution 2D codes relies on the fact that the printing process is considered as a Physical Unclonable Function used to guaranty the security of the authentication system. The 2D code is corrupted by the printing process in a non-invertible way by inducing decoding errors, and the gap between the bit error rate generated after the first and second printing processes enables to perform the authentication of the document. In this context, the adversary's goal is to minimize the amount of decoding errors obtained from the printed code in order to generate a forgery which can be considered as original. The goal of this paper is to maximize the decoding performance of the adversary by inferring the original code observing the printed one. After presenting the different kinds of features that can be derided from the 2D code (the scanner outputs, statistical moments, features derived from Principal Component Analysis and Partial Least Squares), we present the different classifiers that have been evaluated and show that the bit error rate decreases from 32% using the baseline decoding to 22% using appropriated features and classifiers.

1 Introduction

Fighting forgery and falsification constitutes a major challenge in various industrial sectors (Medicines, Documents, consumer goods, etc). Those issues are becoming increasingly critical with the fast development of global exchanges and internet. The development of digital devices such as digital camera, printer, scanner and copying-machines, also facilitates attacks from forgers. According to the Organization for Economic Co-operation and Development (OECD), international trade in counterfeit and pirated goods reached more than US $250 billion in 2009 [16]. According to the World Health Organization in 2005, more than 10 per cent of medicines on the global market are forgeries and this figure rises to nearly 25 per cent in developing countries [15]. To fight against fraud, the companies use to adopt authentication methods which consist in printing secret signatures (holograms, security inks...) on products to distinguish them from

⋆ This work was partly founded by the French National Research Agency program referenced ANR-10-CORD-019 under the Estampille project.

falsified ones. However, the solutions based on those signatures, are generally complex and therefore create heavy costs and constraints.
The authentication system that is studied in this paper has been firstly proposed in [14,13]; it proposes to use copy detection patterns represented as 2D codes in order to detect forged documents. The authentication mechanism is based on the property that the printing process can be considered as a Provably Unclonable Function because of the non-invertibility of the whole printing process. This non-invertibility is due to different factors such as the high resolution of the printer, the random organization of the fibers on the paper or the stochastic formation of the ink drop (or the toner powder) of printers.
Similar techniques exist for authenticating items using non-invertible 3D profiles created by later marks [17] or material singularities [8]. But the originality of the proposed system relies in the fact that the side-information (the 2D code) carries the output of the PUF (the printing process) and that no other helper information than the 2D code is needed to perform authentication. Using this system, an adversary that wants to copy the 2D code will have to perform a new print and scan process; and once decoded the forged 2D code will present more errors than the original one. Authentication will be performed by measuring the average number of decoding errors, the original codes creating an amount of errors significantly lower than copied ones.

1.1 Definition of the authentication system

This authentication process can be formally defined as follow. Let us consider the game (see figure 1) which involves one main communication channel - the print-and-scan process - and three players: the legal sender Alice, the legal receiver Bob and the adversary Eve. The figure 3 summarizes the different communication channels between the three players.

Alice sends, for instance a text document to Bob. Bob wants to verify the authenticity of this document using for example a binary graphical code (X_{Alice}) printed in grayscale (Y_{Alice}) on the document. The size of the code is arbitrary (100×100 for instance). The figure 1 shows an example of random graphical code that Alice can use. In this setting, this code is considered to be a secret key between Bob and Alice.

Once the code is printed, we obtain a grayscale code Y_{Alice} (see figure 2). The adversary Eve wants to produce a forged document with a graphical code Y_{Eve}. She wants also that the legal receiver accepts her code as if it comes from Alice. Therefore, her goal is to make Y_{Eve} statistically as close as possible to Y_{Alice}. On the other side, the receiver Bob wants to build an authentication system T which discriminates between a document coming either from Alice or from Eve.

We can consider that Alice is a passive player and the security game is between Eve and Bob. We merge the printing process and the scanner into the main channel. We use a classical methodology for security in order to try to find the "worst case attack" performed by Eve and to evaluate the authentication system associated to this attack.

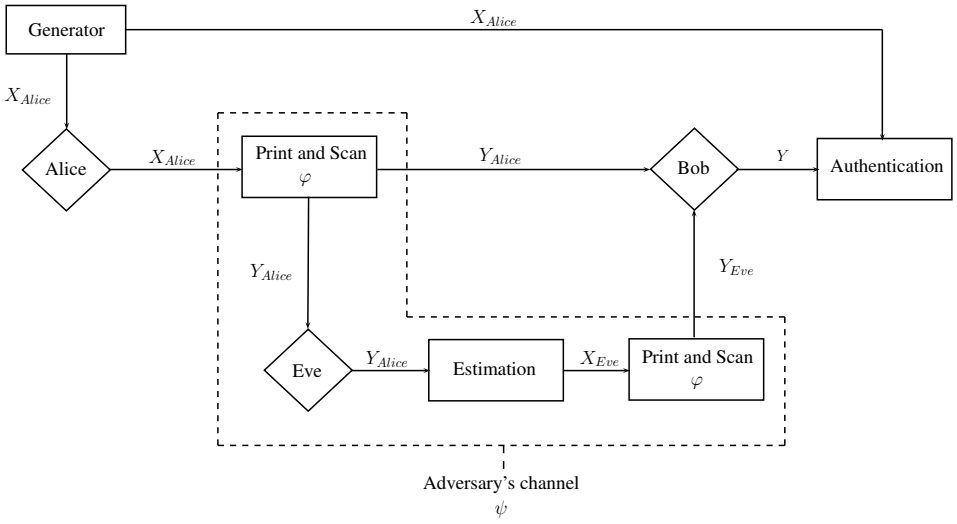


Fig. 1. The different communication channels.

The goal of Bob is to build a authentication system T whose response will enable to decide between hypothesis H_0 (the code received Y is accepted) or hypothesis H_1 (the code Y is rejected). One possible solution consists in building an estimation function G_{Bob} $Y \xrightarrow{G_{Bob}} \widehat{X}$ and to compute the error estimation $\varepsilon\,(\,Y \,|\, X_{Alice}) = G_{Bob}(Y) - X_{Alice}$. The authentication test T is achieved after choosing a certain threshold η:

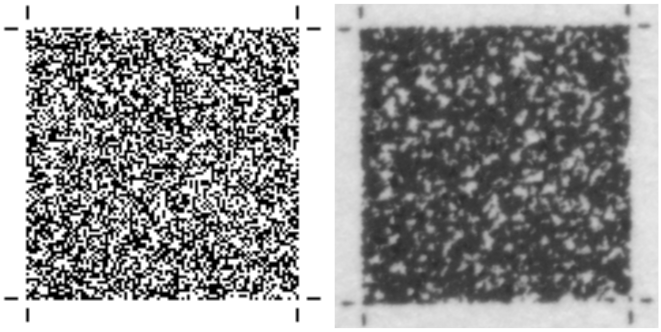
$$\begin{cases} H_0 : Y = Y_{Alice} & \text{if } \varepsilon(Y \,|\, X_{Alice}) \leq \eta \\ H_1 : Y = Y_{Eve} & \text{else.} \end{cases} \tag{1}$$

The choice of η should be driven by two constraints:

1. We want to accept as much as possible the codes coming from the legal sender Alice; this constraint corresponds to the minimization of the Probability of False Alarm (P_{fa} or Probability of detecting a genuine document as a copy):$P_{fa} = \mathbb{P}(\varepsilon(Y \,|\, X_{Alice}) > \eta \,|\, Y = Y_{Alice})$.
2. The second error consists in detecting as false the codes coming from Eve; this constraint corresponds to the minimization of the Probability of Non Detection (P_{nd} or Probability of detecting a copy as genuine document):

$$P_{nd} = \mathbb{P}(\varepsilon(Y \,|\, X_{Alice}) \leq \eta \,|\, Y = Y_{Eve}). \tag{2}$$

This authentication system is based on the fact that there is no reversible degradation after printing, we can replace the error estimation by any norm or function that reflects these phenomena. In order to perform a security analysis using this basic authentication system and to evaluate the potential attack of the adversary, we assume that Bob and Eve have exactly the same tools (printer, scanner,



(a) (b)

Fig. 2. (a) Graphical code before printing (X). (b) Graphical code after printing (Y) (The segments around the corners are onlt used for synchronization purposes).

software for acquisition). Eve and Bob have at their disposal noisy samples of printed images, important computation capacities and the graphical code Y_{Alice} printed by Alice. The only differences are that (1) Bob knows the original code X_{Alice} and Eve does not and (2) that Eve uses a more advanced decoder than Bob. This second assumption enables to evaluate the risk taken by Bob if he overestimates the security of its PUF. Table 1 summarizes these different assumptions.

Tools	Eve	Bob
Acquisition tool	Same than Bob's	Scanner
Printer	Same than Alice's	Same than Alice's
Authentication Method	-	Estimation + Hypothesis testing
Y_{Alice}	Yes	Yes
X_{Alice}	No	Yes
Decoder	Advanced	Baseline

Table 1. Assumptions for the game between Eve and Bob.

1.2 Adversary's options

The main goal of the opponent Eve is to reproduce what she observes as precisely as possible. In fact, ideally, this accuracy should be such that the legal receiver cannot distinguish the codes coming from the legal sender and those from the opponent. We study mathematically the different implications of this formulation of the problem. Given (see also Fig. 2):

– X_{Alice} (and X_{Eve}) the binary code that Alice (respectively Eve) sends through the print and scan channel,

- Y_{Alice} (and Y_{Eve}) the printed grayscale code obtained from X_{Alice} (respectively X_{Eve}),
- φ the print and scan channel,
- G_{Eve} the estimation function built by the opponent Eve,
- ψ the adversary channel, composed of two print and scan channel and a decoding function.

Under those notations, the main channel (i.e. legal channel from Alice to Bob) consists in one print and scan step:

$$X_{Alice} \xrightarrow{\varphi} Y_{Alice} \ ,$$

while the adversary channel:

$$X_{Alice} \xrightarrow{\psi} Y_{Eve} \ ,$$

consists in two print and scan steps and one estimation step between:

$$X_{Alice} \xrightarrow{\varphi} Y_{Alice} \xrightarrow{G_{Eve}} X_{Eve} \xrightarrow{\varphi} Y_{Eve}.$$

So then, the adversary's channel corresponds mathematically to:

$$\psi = \varphi \circ G_{Eve} \circ \varphi. \tag{3}$$

The following equation summarizes the ideal goal of the adversary:

$$\varphi(X_{Alice}) = \psi(X_{Alice}). \tag{4}$$

If we have indeed this equality, the two channels are identical (in fact, they produce identical results). Using the expression in eq.(3), we can rewrite the problem as:

$$\varphi(X_{Alice}) = (\varphi \circ G_{Eve}) \circ \varphi(X_{Alice}) = \varphi \circ (G_{Eve} \circ \varphi)(X_{Alice}). \tag{5}$$

We can deduct from this last expression that if we have an estimation function G_{Eve} such as:

$$\varphi \circ G_{Eve} = Id, \tag{6}$$

or:

$$G_{Eve} \circ \varphi = Id, \tag{7}$$

(where Id is the identity function), in both cases the goal is reached. Now, we need to specify what these two expressions mean and how to build G_{Eve} from them. We now detail the two types of solutions using this specification.

Minimization of the "copy" error: eq. (6) corresponds to the design of G_{Eve} such as:

$$\varphi(G_{Eve}(Y_{Alice})) = Y_{Alice}. \tag{8}$$

In practice, the print and scan process is highly stochastic and non-linear, so we cannot solve the problem analytically. To tackle numerically the problem, we need to transform it into a minimization problem. Given $a, b \mapsto \| a - b \|_\alpha$, $\| . \|_\alpha$ is an arbitrary norm (Minimum Square error, Bit error Rate if binary values case...); the problem in eq. (8), becomes:

$$G_{Eve} = argmin \| \varphi(G_{Eve}(Y_{Alice})) - Y_{Alice} \|_\alpha, \tag{9}$$

which is an optimization problem. But since $Y_{Eve} = \varphi(G_{Eve}(Y_{Alice}))$, the expression becomes simply:

$$G_{Eve} = argmin \| Y_{Eve} - Y_{Alice} \|_\alpha, \tag{10}$$

which corresponds to minimizing the copy error**.** The goal here is to design a code $X_{Eve} = G_{Eve}(Y_{Alice})$ that allows us to reproduce the observation Y_{Alice} without using the original code X_{Alice}. In fact, in order to solve the equation (8), we do not need X_{Alice}but we need a model $\widehat{\varphi}$ of the print and scan channel.

Several studies explored this solution in the field of document degradations: [6] used in the context of bar codes a hidden Markov process for the stochastic modeling,[19]uses a nonlinear model with additive noise dependent to the input in the same context, [10] provides a text degradation a model using flipping probabilities and morphological filtering.

Minimization of the decoding error: Another alternative to solve eq. (4) is to consider eq. (7) which corresponds to the building of G_{Eve} such as:

$$G_{Eve}(\varphi(X_{Alice})) = X_{Alice}. \tag{11}$$

Using the norm defined in subsection 1.2:

$$G_{Eve} = argmin \| G_{Eve}(\varphi(X_{Alice})) - X_{Alice} \|_\alpha, \tag{12}$$

But since $Y_{Alice} = \varphi(X_{Alice})$, the expression becomes:

$$G_{Eve} = argmin \| G_{Eve}(Y_{Alice}) - X_{Alice} \|_\alpha, \tag{13}$$

which corresponds to the minimization of the decoding error. For this solution the adversary Eve tries to retrieve the original code, but since she does not know X_{Alice}, Eve needs to infer the decoding function G_{Eve} using arbitrary codes X_i and arbitrary samples Y_i coming from the printing process.

Contrary to the first solution, the second solution is not well studied in this specific domain. However its efficiency has been proved for a wide set of applications dealing with complex empirical data (cf. [4]). In this paper, we adoptthe second method (minimization of the estimation error) and we use statistical inference methods, especially supervised classification to build the decoding function.

2 Maximizing the decoding performance

2.1 Practical setup

To constitute the database, we printed 100 random binary codes (size: 100×100 dots) with 50% of black dots. The printer used is a laser printer (Dell 2350dn). The acquisition of the printed codes were done using a high resolution scanner (Canon CanoScan 9000F). The main channel is constituted by the printer, the scanner and the codes extraction algorithm which perform various treatments on the code. The printing and scanning conditions are the following:

- The Resolution of the printer is set to 600dpi (native resolution of the printer);
- The intensity of the printer is set to 8 (out of 10),
- The Quality of the printing is set to "raw".
- The Resolution of the scanner is set to 9600dpi (highest resolution).

With those conditions, the output obtained is a grayscale image of size: 1500×1500 pixels. We now show the design of the decoding function under these conditions.

2.2 Local specification of G_{Eve}

Let X the 100×100 binary code before printing; and Y the 1500×1500 grayscale code obtained after printing and scanning. The goal here is to find an decoding function G_{Eve} such as:

$$\widehat{X} = G_{Eve}(Y). \tag{14}$$

Because of the dimension of the codes (X is described by 10,000 dots while Y is described by 2,250,000 pixels), writing directly a functional form for G_{Eve} is hardly conceivable. The solution adopted here is to consider the local evolution within the codes**.** X is in fact a collection of dots, each dot located at position $(i\,,\,j)$ is characterized by its binary value $x_{i,j}$. Let $y_{i,j}$ a vector of $\mathbb{R}^{225}$, corresponding to the 15×15 high resolution printed image of $x_{i,j}$. We locally specify the estimator G_{Eve} by a function g such as:

$$\forall i, j, \quad \widehat{x}_{i,j} = g(y_{i,j}). \tag{15}$$

The input is a vector of $\mathbb{R}^{225}$ while the output is binary. Therefore, we can use a basic threshold function to specify the local estimator g. We call this estimator the baseline decoder and we assume in the sequel that it is used by the authentication system. In the next subsection, we present a more efficient design using supervised classification. This specification, however introduce several additional biases. Firstly, we ignore the interactions within the dots after the printing; in fact the information about $x_{i,j}$ is spread within the scanned image. Secondly, we assume that we can estimate each dot independently while the dots printed interact strongly. To partially attenuate these effects, we assume that by taking

in account the printed image of dots in a 3×3 neighborhood of $x_{i,j}$, we capture the relevant information. If we call $u_{i,j}$ the vector obtained:

$$u_{i,j} = [y_{i-1,j-1}\ ,\ \dots\ ,\ y_{i,j}\ ,\ \dots\ ,\ y_{i+1,j+1}], \tag{16}$$

The dimension of $u_{i,j}$ is $225 \times 3 \times 3 = 2025$. The local estimator becomes:

$$\forall i, j, \quad \widehat{x}_{i,j} = g(u_{i,j}). \tag{17}$$

Using the local and contextual specification, we transform our problem into finding a decoding function from $\mathbb{R}^{2025}$ to $\{0, 1\}$.

2.3 Supervised classification

We present here the tools that have been used to infer the original code. Given:

- $t \in T^d$ a vector of d structural characteristics also called features, which summarizes a given observation;
- $c(t) \in \{c_1, c_2, ..., c_k\}$ a characteristic about the observation we want to identify (in our case $x_{i,j}$);

We assume that each observation is obtain by an i.i.d. sampling from an unknown distribution $p(t)$. The problem of classification consists in building a function δ that outputs a class to each features vector:

$$\delta : \quad t \longmapsto \widehat{c}. \tag{18}$$

using a finite sequence of data called training set:

$$D = \{(t_1\ ,\ c_1)\ ,\ (t_2\ ,\ c_2)\ ,\ \dots\ ,\ (t_m\ ,\ c_m)\} \tag{19}$$

Statistical decision theory provides a solution which consists in partitioning the input space according to each class by using decision boundaries which separate the classes. For binary classification (k = 2) for instance, if $\mathcal{F} : f(t) = 0$ is a decision boundary and if we encode the classes such as: $c_1 = 1$ and $c_2 = -1$, we have:

$$\delta(t) = sign(f(t)). \tag{20}$$

Classification algorithms or classifiers require:

- A category of boundaries (linear, quadratic, nonlinear and nonparametric...);
- A loss function (misclassification, exponential...) to penalize misclassifications;
- A regularization term to limit overfitting (i.e. dependance to the training set) .

So then, the classifier choose the boundary that minimize the sum of these two terms. To evaluate classifiers, the solution consists in generating a new sequence D_{test} :

$$D_{test} = \{(t_1\,,\,c_1)\,,\,(t_2\,,\,c_2)\,,\,...\,,\,(t_{m'}\,,\,c_{m'})\}, \tag{21}$$

and to evaluate the classifiers on it by calculating the generalization or prediction error rate:

$$\widehat{Err}_g = \frac{1}{m} \sum_{(t,c)\in D_{test}} |c - \delta(t)|. \tag{22}$$

Theoretically, the size of D_{test} should be "infinite" (i.e. sufficiently large in practice) to cover the whole distribution $p(t)$. In general, D_{test} is not large enough; therefore, other estimates as cross-validation (K-fold and Leave-one-out) and bootstrap validation are computed. Since its output is binary, the local estimator g is in fact a classifier. In this work, we compare 5 classifiers :

- Three linear classifiers: Linear Discriminant Analysis, Naive Bayes, Logistic Regression,
- Two nonlinear classifiers: Quadratic Discriminant Analysis , Support Vector Machine.

[2,7] provides full description of these methods. They are widely used supervised classification techniques and achieve good performances in general.

2.4 Feature extraction

The dimension of the feature vector chosen (2025) can constitute a serious issue for classification. In fact, in high dimensions, the vectors are far from each others and finding good boundaries becomes increasingly difficult. The number of samples required increases exponentially with the dimension. This problem is well known as the "curse of dimensionality" and to break it, we need to represent all the information with less features. This operation is called feature extraction in statistical learning, it consists in concentrating the information in privileged directions with minimal loss. We tested three methods:

- **Statistics**: we summarize the 2025 features by taking the 4 first moments (mean, variance, skewness, kurtosis) for the 3×3 context after printing which give us 36 features. These new features are linear and nonlinear functions of the 2025 features. The moments summarizes the spatial distribution of the 9 images. They can be completed with the median, the quartiles, the min or the max.
- **Principal Components Analysis** (PCA); PCA performs linear projection to lower dimensional space. The new features obtained are decorrelated and ranked according to their variance. The percentage of variance preserved during the projection is a measure of the quantity of information saved during the projection.

– **Partial Least square Regression** (PLS), PLS is very similar with PCA; but PLS take in account the preservation of the variance of the targets simultaneously in its projection.

The statistics does not give a criteria to evaluate the quantity of information loss, while PCA and PLS methods provide the percentage of variance captured according to the number of variables kept. That percentage gives a possible criteria to select the dimension of the new space. We defined various group of features for our classification task using these feature extraction methods (see also Fig. 3):

– The first set F1 is constituted by the $15 \times 15 = 225$ pixels of a dot printed (cf. 2.2);
– F2 is constituted by the pixels of the central dot printed and by those of the neighboring dots in its 3×3 neighborhood (so then, we have $3 \times 3 \times 15 \times 15 = 2025$ features);
– F3 is constituted by Statistical Moments obtained from each 15×15 printed image of a dot printed and by those in its 3×3 context; we add moments of 4 crossover blocks to capture the transitions between the dots; we have then 52 features;
– F4 is constituted by PCA features deduced from F2; we retained 200 first features using the ratio of variance of the input explained; those features explained 99% of the variance;
– F5 is constituted by PLS features deduced from F2; we retained 500 first features using the ratio of variance of the input and the target explained; those features explained 99% of the variance.

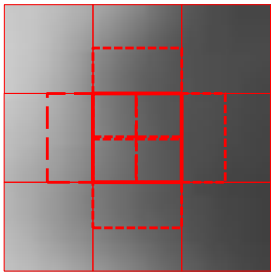


Fig. 3. The different 15×15 blocks: the bold one is used to build F1, the set of 9 blocks in solid line to build F2, F4 and F5, and the dashed blocks are also used to compute moments (F3).

3 Results

To test the methods selected, we used the 50 graphical codes kept in 2.1. We use 5 codes as a training set; which give us $5 \times 10,000 = 50,000$ training samples.

The classifiers are afterward tested on the rest of the codes (45 codes). For each code ($= 10,000$ examples) we compute a Bit Error Rate. We assume that Bob's decoder uses a basic thresholding as a baseline method. It consists in averaging each scanned dots, and choosing an optimal threshold between 0 to 255. The baseline approach enables to obtain a BER of 32 % with a standard deviation of 1.6% on the testing set. We compare now this naive approach with respect to one used by the adversary. Table 3 and Fig. 5 depict an overview of the results for the different feature sets and classification tools.

3.1 Using F1 and F2 (raw inputs with and without neighborhood)

The table (a1) and the boxplot (b1) shows results for F1. LDA and logistic regression provides the best results when F1 is used. However, the boxplot shows variability according to the images tested. In fact, we encounter this effect for all set of features. QDA is very good in training, but produces bad predictions. This is typically overfitting. Naive Bayes classifier gives a robust result i.e. with less variability but is less accurate than LDA and Log. Reg.; as for SVM, its average performance can be explained by its sensitivity to irrelevant and non-weighted variables. However, because LDA and Log. Regression implicitly weight the variables in several directions, they are more robust to those features.

When F2 are used (cf. (a2) and (b2)), there isa degradation of the performance of the classifiers and a general trend to overfit except for the Naive Bayes classifier. The extension to 2025 features add irrelevant variables, which increases the effect of the "curse of dimensionality".

3.2 Using F3 (moments)

Using those features improves the BER for all classifiers, especially SVM (22.1%) which now is the most accurate among the ones tested. However, LDA and logistic regression (22.6%) manage to give very close results than those obtained with SVM. QDA avoid overfitting but it is still outperformed by the others. Naive Bayes does not show any improvement compare to the results obtained using F2.

3.3 Using F4 and F5 (PCA and PLS)

The results with PCA are very close to those obtained with the moments. The exception is Naive Bayes which gives results even better than LDA. The possible explanation is the fact that Continuous Naive Bayes assumes independent features. Therefore, the covariance matrix of the features is forced to be diagonal. In the previous representations, this assumption is violated. So then, Naive Bayes assumption results in loss of possible discriminative loss. Since PCA provide uncorrelated features, the covariance is really diagonal this time. Therefore, Naive Bayes is equivalent to LDA in this context. The other exception is QDA which is overfitting again.

We obtained similar results (cf. a5, b5) using PLS features; the exception is SVM. In fact, we suspect a default in the hyperparameters setting. Since they are set by cross-validation (best error rate among a set of values using a small testing set), it is more likely that the range of values chosen should be extended.

3.4 Summary

Moments provides good results in classification followed by features generates from PCA (see Table 3 and Fig. 5). As for the classifiers, the linear ones assure good results compare to SVM.

Another observation is that the mean BER is in general than 22%. That constitutes an empirical lower bound about the amount of information that we can retrieve using the methods tested.

3.5 Impacts on authentication

Classification allows a BER of 22% while the baseline method give us 32%. We now evaluate the impact of this gain on the authentication system described in section 1.1. To do so, we perform a second printing considering 2 cases: (1) the adversary Eve uses the same baseline method as the decoder and (2) Eve uses the LDA classifier with moments as features (F3). The reprinted code or copy is called Y_{Eve}^{Bas} in case (1) and Y_{Eve}^{LDA} in case (2). We assume also (cf. table 1) that Bob uses the baseline method to perform the hypothesis test. Under those assumptions, we obtained in table 2 the results when Bob decodes authentic and copied codes.

Printed code	Mean BER	Std BER
Y_{Alice}	32%	1.6%
Y_{Eve}^{LDA}	38.7%	1.2%
Y_{Eve}^{Bas}	40%	0.5%

Table 2. BER obtained by Bob using baseline

We may observe that the results of the decoding of Y_{Eve}^{Bas} and $Y_{Eve}^{LDA}$are very close. The significant gain that the adversary obtains in recovering the original code gives a slight improvement for the copy. However, the baseline method is more sensitive to print and scan instabilities; for instance when there is a significant variation of the quantity of ink per code, the copy performed using baseline method can reach for the darkest codes a BER close to 50% while LDA copy manages to give a BER under 42%.

To have a better understanding of the differences between the two decoding methods, we compute the Receiver Operating Characteristic (ROC) curve for each of them (fig. 4). To do so, we assume that the distributions of the BER after one and two printing are Gaussian. We use the results in table 2 for the

parameters of the distribution. By varying the threshold of the hypothesis test, we compute analytically the true and false positive rate (respectively TPR and FPR).

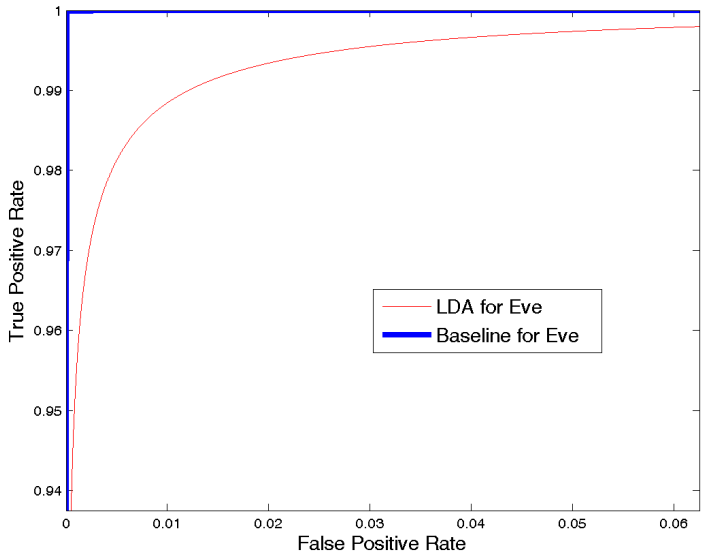
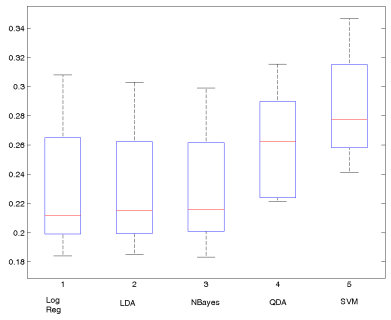


Fig. 4. ROC curve for each method of copy

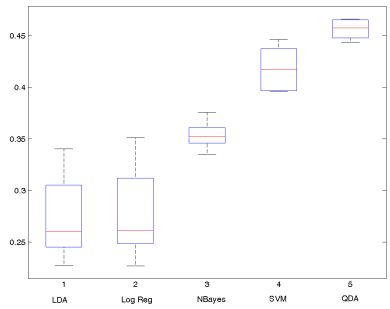
The figure 4 shows that the authentication is more difficult when the adversary uses LDA because the ROC curve in this case is closer to the diagonal than when the adversary uses the baseline decoder. This is not only due to a better average performance obtained when decoding with LDA, but also due to the stronger variability of this method.

4 Conclusion

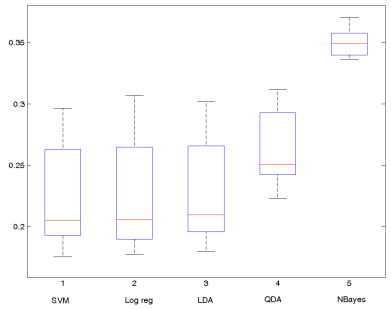
This paper proposes to assess the security of a PUF based authentication system which uses the printing process as a non-invertible function. The security analysis has been carried out using a "black box" strategy where we try to infer the inverse of the physical system from a set of observations without modeling the printing process itself. This approach enables to already show that the adversary can improve the recovery of the original code with respect to a naive decoding



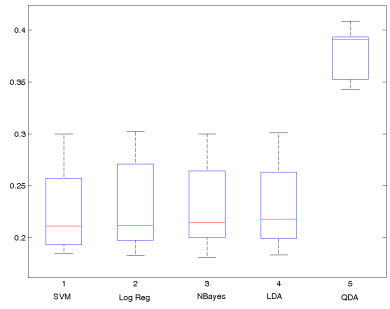
b1) F1 (225 features)



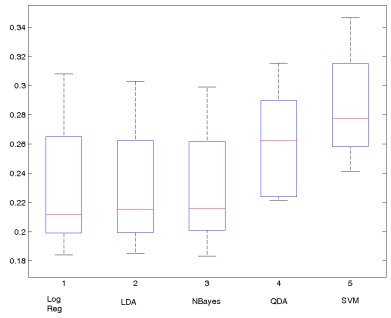
b2) F2 (2025 features)



b3) F3 (Moments)



b4) F4 (200 PCA features)



b5) F5 (500 PLS features).

Fig. 5. Boxplots of the BER per image for different feature sets. On each box, the central mark in red is the median of the BER. Under the lower edge of the box, we have 25% of the cases. Under the upper edge we have 75%. The whiskers extend to the most extreme data points not considered as outliers.

Algorithm	Mean BER	Std
LDA	24.4%	3.9%
QDA	32.6%	3%
Naive Bayes	31.3%	2.4%
Logistic regression	24.3%	4.2%
Kmeans + SVM	27.6%	3.9%

a1) F1 (225 features),

Algorithm	Mean error	Std
LDA	27.3%	3.8%
QDA	45.7%	0.9%
Naive Bayes	35.4%	1.2%
Logistic regression	27.6%	4%
Kmeans + SVM	41.8%	2%

a2) F2 (2025 features),

Algorithm	Mean error	Std
LDA	22.6%	4.1%
QDA	26%	2.9%
Naive Bayes	35%	1.2%
Logistic regression	22.6%	4.6%
Kmeans + SVM	22.1%	3.9%

a3) F3 (Moments),

Algorithm	Mean Error	Std
LDA	22.9%	4%
QDA	37.7%	2.4%
Naive Bayes	22.8%	4.1%
Logistic regression	22.9%	4.4%
Kmeans + SVM	22.6%	4.2%

a4) F4 (200 PCA features),

Algorithm	Mean Error	Std
LDA	22.9%	4.1%
QDA	25.9%	3.6%
Naive Bayes	22.9%	4%
Logistic regression	22.9%	4.3%
Kmeans + SVM	28.5%	3.5%

a5) F5 (500 PLS features).

Table 3. Bit Error Rates w.r.t. different feature sets.

by a substantial amount (the BER drops from 32% to 22%) but with a higher dispersion (std BER of 4% instead of 1.6%). That gain allows him to perform copies which are more difficult to detect and our primary results indicate that the performances of the authentication system are considerably affected. Further works will try to perform better decoding using structural and/or prior information about the 2D codes to improve our inference. We will also study the impact of the bias and variance of the decoder on the whole authentication system.

References

1. Amiri, S., Jamzad, M.: An algorithm for modeling print and scan operations used for watermarking. In: Kim, H.J., Katzenbeisser, S., Ho, A. (eds.) Digital Watermarking. Lecture Notes in Computer Science, Springer Berlin / Heidelberg (2009)
2. Bishop, C.: Pattern recognition and machine learning, vol. 4. springer New York (2006)
3. Borges, P., Mayer, J., Izquierdo, E.: Document image processing for paper side communications. Multimedia, IEEE Transactions on 10(7), 1277–1287 (2008)
4. Breiman, L.: Statistical modeling: The two cultures (with comments and a rejoinder by the author). Statistical Science 16(3), 199–231 (2001)
5. Davy, M., Tourneret, J.: Generative supervised classification using dirichlet process priors. Pattern Analysis and Machine Intelligence, IEEE Transactions on 32(10), 1781–1794 (2010)

6. Dridi, N., Delignon, Y., Sawaya, W., Septier, F.: Blind detection of severely blurred 1d barcode. In: GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference. pp. 1–5. IEEE (2010)
7. Friedman, J., Hastie, T., Tibshirani, R.: The elements of statistical learning. Springer Series in Statistics (2009)
8. Gassend, B., Clarke, D., Van Dijk, M., Devadas, S.: Controlled physical random functions. In: Computer Security Applications Conference, 2002. Proceedings. 18th Annual. pp. 149–160. IEEE (2002)
9. Guyon, I.: Feature extraction: foundations and applications, vol. 207. Springer Verlag (2006)
10. Kanungo, T., Haralick, R., Baird, H., Stuezle, W., Madigan, D.: A statistical, nonparametric methodology for document degradation model validation. Pattern Analysis and Machine Intelligence, IEEE Transactions on 22(11), 1209–1223 (2000)
11. Lai, L., El Gamal, H., Poor, H.: Authentication over noisy channels. Information Theory, IEEE Transactions on 55(2), 906–916 (2009)
12. Maurer, U.: Authentication theory and hypothesis testing. Information Theory, IEEE Transactions on 46(4), 1350–1356 (2000)
13. Picard, J., Vielhauer, C., Thorwirth, N.: Towards fraud-proof id documents using multiple data hiding technologies and biometrics. SPIE Proceedings–Electronic Imaging, Security and Watermarking of Multimedia Contents VI pp. 123–234 (2004)
14. PICARD, J., ZHAO, J.: Improved techniques for detecting, analyzing, and using visible authentication patterns (Jul 28 2005), wO Patent WO/2005/067,586
15. Press, W.: Global congress addresses international counterfeits threat immediate action required to combat threat to finance/health (2005), "http://www.wcoomd.org/press/default.aspx?lid=1&id=22"
16. Press, W.: Counterfeiting and piracy endangers global economic recovery, say global congress leaders (2009), "http://www.wcoomd.org/press/default.aspx?lid=1&id=201"
17. Shariati, S., Standaert, F., Jacques, L., Macq, B., Salhi, M., Antoine, P.: Random profiles of laser marks. In: Proceedings of the 31st WIC Symposium on Information Theory in the Benelux (2010)
18. Solanki, K., Madhow, U., Manjunath, B., Chandrasekaran, S., El-Khalil, I.: Print and scan'resilient data hiding in images. Information Forensics and Security, IEEE Transactions on 1(4), 464–478 (2006)
19. Villan, R., Voloshynovskiy, S., Koval, O., Pun, T.: Multilevel 2 d bar codes: toward high-capacity storage modules for multimedia security and management. In: Proc. SPIE. vol. 5681, pp. 453–464 (2005)
20. Wells, R., Vongkunghae, A., Yi, J.: A signal processing model for laser print engines. In: IECON 02 [Industrial Electronics Society, IEEE 2002 28th Annual Conference of the]. vol. 2, pp. 1514–1519. IEEE (2002)
21. Yu, L., Sun, S., et al.: Print-and-scan model and the watermarking countermeasure. I