# First Principal Components Analysis: A New Side Channel Distinguisher

Youssef Souissi[1], Maxime Nassar[1,2], Sylvain Guilley[1], Jean-Luc Danger[1] and Florent Flament[1].

[1] TELECOM ParisTech, CNRS LTCI (UMR 5141),
46 rue Barrault
75 634 Paris Cedex, France.
[2] BULL TrustWay
Rue Jean Jaurès, B.P. 68
78 340 Les Clayes-sous-Bois, France.

**Abstract.** Side Channel Analysis (SCA) are of great concern since they have shown their efficiency in retrieving sensitive information from secure devices. In this paper we introduce First Principal Components Analysis (FPCA) which consists in evaluating the relevance of a partitioning using the projection on the first principal directions as a distinguisher. Indeed, FPCA is a novel application of the Principal Component Analysis (PCA). In SCA like Template attacks, PCA has been previously used as a pre-processing tool. The originality of FPCA is to use PCA no more as a preprocessing tool but as a distinguisher. We conducted all our experiments in real life context, using a recently introduced practice-oriented SCA evaluation framework. We show that FPCA is more performant than first-order SCA (DoM, DPA, CPA) when performed on unprotected DES architecture. Moreover, we outline that FPCA is still efficient on masked DES implementation, and show how it outperforms Variance Power Analysis (VPA) which is a known successful attack on such countermeasures.

**Keywords:**
Principal Component Analysis (PCA), Data Encryption Standard (DES), Side Channel Attacks (DoM, DPA, CPA, VPA), Masking countermeasures.

## 1 Introduction

Different forms of technologies, which require an adequate level of security, are extensively manipulated around the world. Any violation of such systems could lead to the loss of sensitive and personal information. In this context, Side Channel Analysis (SCA) pose a real threat to these technologies since they are non intrusive, low cost and easily mounted in practice [16]. Actually, SCA exploit the information leaked from cryptographic devices during the encryption or decryption process to extract the secret information referred to as *secret key*. This

information is retrieved by analysing the power consumption or the electromagnetic (EM) radiations of the device under attack. SCA are based on statistical computations to exhibit the secret. Indeed, the leaked information can be statistically modeled by a continuous random variable following an unknown or uncertain probability law $P_{law}$.

The main challenge of SCA is to make a sound estimation of $P_{law}$ relevant features without loss of information. The more accurate this estimation is, the greater the efficiency of SCA is. Basically, random variables are measured and analyzed in term of their statistical and probabilistic features [7]. In the case of SCA, calculations based on the first and second order statistics seem to be good ways to quantify the secret information. For instance, Differential Power Analysis (DPA) is mainly based on computations related to the first-order statistic, the "mean". Moreover, Variance Power Analysis (VPA) [18, 30] which is based on the variance, has shown its efficiency on masked implementations.

Recently, a new powerful variant of SCA so-called MIA [8] has been presented to the cryptographic community. This attack is based on mutual information theory which requires a reliable estimation of the probability density function of $P_{law}$. Basically, an accurate probabilistic measure, such as the entropy, describes better one random variable than other statistics [24]. However, the optimal accuracy is hardly achieved specially when the probability law is unknown. As a matter of fact, the probability density of an unknown law is quite difficult to properly estimate when the available data to be studied is limited [7]. Statisticians are used to calculate quantities easier to estimate. These quantities are the moments of a probability distribution like the mean, the variance or the kurtosis. By analogy to the cryptographic domain, statisticians are identified to attackers and the available data to power or EM consumption signals. Indeed, the attacker is often required to conduct its attack under certain constraints. Actually, according to the security levels as defined by Abraham et al. [2], secure devices could be classified into seven levels of security. According to each level, the attacker behaves in different manners. In the real life, the attacker has to perform the attack by considering the external environment of the device under attack which depends on the factory and the type of the circuit (FPGA, ASIC, ...). For instance, some security measures could be employed to limit the acquisition of power consumption signals (*traces*). Thus, the attacker would not be free to acquire as much traces as he wants. In addition to that, we believe that any cryptographic design could be attacked by exploiting its sensitivity against one chosen statistic, denoted by $CS$, that could be the mean, the variance or any other statistic describing one $P_{law}$. The higher the sensitivity is, the greater is the vulnerability of the implementation against attacks based on the considered $CS$. This is true since an ideal cryptographic implementation could not really exist, in accordance with the fact that real life application could not fit exactly the theory.

In this paper, we outline the way how Principal Component Analysis (PCA [12]) could be used to extract the value of the secret key. PCA is a multivariate data analytic technique [24, 26] that has found application in fields such as computer

vision [15,28], robotics [34], sociology and economics [27]. It is a way of identifying patterns in multidimensional data set, and visualising these data into a lower dimensional space, in order to highlight their similarities and differences. In the SCA techniques portfolio, PCA has already revealed its efficiency on Template attacks [23]. Basically, Template attacks are considered very powerful since they can break cryptographic implementations which security is dependent on the assumption that an attacker cannot obtain more than one or a limited number of side channel traces. Moreover, these attacks require that an attacker has access to a clone device on which he can perform trials to get trained. As described in [23], PCA improves the class of Template attacks by pre-processing the leakage traces before performing the attack on the real cryptographic device. Indeed, in the pre-processing phase the attacker builds templates in order to profile the clone device. Then, those templates are used to mount an attack on the real device. Our attack uses PCA no more as pre-processing tool but as a distinguisher. Moreover, it follows the usual steps of differential power analysis (DoM [20], DPA [5] or CPA [6]) that consists of only one phase and does not require a clone device for profiling, which makes the task of the attacker easier.

The rest of the paper is organized as follows. First, Section 2 attempts to give some elementary background that is required to understand the process of PCA. Second, this background knowledge is taken advantage of in section 3 to outline the way how PCA could be exploited to mount an efficient attack. This section goes through the different steps needed to perform the FPCA. Section 4 is devoted to experiments on unprotected and protected DES implementations. This section highlights the efficiency of FPCA by making a comparative analysis with existing attacks (DoM, DPA, CPA, VPA). The conclusions and perspectives are in section 5.

## 2 Principal Component Analysis: background knowledge

Let a data set of $M$ quantitative variables describing $N$ samples, arranged respectively in rows and columns. The goal of PCA is to ensure a better representation of the $N$ samples by describing the data set with a smaller number $M'$ of new variables. Technically speaking, PCA proposes to seek a new representation of the $N$ samples in a subspace of the initial space by defining $M'$ new variables which are linear combinations of the $M$ original variables, and that are called principal components. Generally speaking, reducing the number of variables used to describe data will lead to some loss of information. PCA operates in a way that makes this loss minimal. For PCA to work properly, the data set should be centred. PCA starts by computing the covariance matrix of the data set in order to find the eigenvectors and eigenvalues which permit the capture of the existing dispersion in variables. In other words, it makes a change of orthogonal reference frame, the new variables being replaced by the Principal Components which are totally characterized by the associations of the eigenvectors and eigenvalues. But, more importantly, these associations reveal the hidden dynamics of the data set. Determining this fact allows the attacker to discern which dynam-

ics are important and which are just redundant. The first component can be expected to account for a fairly large amount of the total variance. Each succeeding component will account for progressively smaller amounts of variance. In practice, the attacker sorts eigenvectors by their eigenvalues, from the highest to the lowest. This gives the components in order of significance. Most of the time, only few *M'* components account for meaningful amount of variance. Thus, only these first *M'* components will be retained. The decision on the number of the *M'* best components could be achieved by performing some deciding tests such as the Kaiser criterion, the scree test or the cumulative variance criteria [13].

## 3 FPCA: the attack process

In the SCA field, PCA has often been used as pre-processing tool to minimize the coding complexity by reducing the dimensionality of recorded traces [3, 29]. By contrast, our approach is different in the sense that PCA is used as an attack tool to retrieve the secret information. Indeed, FPCA uses the projection on the first principal components to tell good *secret key* candidates from incorrect ones. FPCA shares some key points with first-order SCA, differential and correlation power analysis (DPA and CPA). As stated before, FPCA does not require a detailed knowledge about the cryptographic device to be performed. It exploits data dependency of the power consumption of the device under attack. The main difference with first-order SCAs resides in the way to distinguish the behaviour of the good key hypothesis. In fact, we remind that each attack has its own statistical test, referred to as distinguisher [9, 30], which allows the attacker to detect the value of the secret key. In this context, FPCA comes with a new distinguisher for side channel analysis. In the rest of this section, we detail the different steps needed to perform a FPCA, while introducing our notations at the same time. One schematic description of FPCA attack is depicted in Fig. 1.

### 3.1 Preliminary preparation phase

This phase is common with differential and correlation power attacks. Suppose that $T$ power consumption traces are recorded while a cryptographic device is performing an encryption or a decryption operation. Collected traces are $L$-dimensional time vectors. The attacker chooses an intermediate result of the cryptographic algorithm that is processed by the cryptographic implementation. The intermediate value denoted by *f(d,k)* is a function that takes two parameters. The first parameter denoted by $d$ is a known data value that can be either the plain text or the cipher text. The number of data values is equal to $T$, the number of recorded traces. These known data values are represented by a vector $D_{vect} = (d_1, d_2, \ldots, d_D)$ of size $D$. The second parameter, denoted by $k$, is secret, hence unknown. Indeed, $k$ is a small part of the cryptographic key and can take $K$ possible values referred to as key hypotheses that we write as a vector $K_{vect} = (k_1, k_2, \ldots, k_K)$.

Thus, the *trace* can be written as a matrix of size $D \times L$. Given vectors $D_{vect}$ and $K_{vect}$, the attacker is able to compute, without difficulties, the hypothetical intermediate value *f(d,k)* for all $K$ key hypotheses and for all $T$ executed cryptographic operation. Then the attacker builds a matrix $V$ of size $K \times T$: $V_{i,j} = f(d_i, k_j)$ with $1 \leq i \leq T$ and $1 \leq j \leq K$. For each value $V_{i,j}$, the attacker computes a hypothetical power consumption value $h_{i,j}$ based on a power consumption model. The most commonly used power models are the Hamming distance (HD) and the Hamming weight (HW) [6]. $R$ being the number of possible values that the power consumption model could take, the traces are arranged in $X$ ($X \leq R$) different partitions for each key hypothesis $k_j$. We denote these partitions as a vector $P_{k_j} = (P_{k_j,1}, P_{k_j,2}, \ldots, P_{k_j,X})$ with $1 \leq j \leq K$. For instance, suppose that our power consumption model is the HD and that it can take integral values from 0 to 4: $HD = \{0, 1, 2, 3, 4\} = \{HD_i\}_{i=1}^{5}$. The trivial partitioning is to associate each $HD_i$ value to one partition. Thus $X = R = 5$. One other possibility is to build only $X$=3 partitions in this way : First partition for $HD > 2$, second for $HD = 2$ and third for $HD < 2$. Intuitively, the more accurate the used power model is, the better our description of the secret information will be. Many papers are dealing with the investigation of new power models and techniques for traces classification [1, 21]. The optimal choice of the power consumption model, including the partitioning process, is out of the scope of this paper. In what follows, our study will focus on the Hamming distance model as it is one of the most commonly used, and often one of the most efficient.

### 3.2 References computation

Once traces are arranged in $X$ partitions for each key hypothesis $k_j$, we propose to compute for each partition a statistical trace based on one $CS$ and referred to as $reference$. For instance, if $CS$ is the "mean" then the $reference$ would be the average of all traces that belong to the considered partition. Actually, the $X$ references of one key hypothesis $k_j$ will be used by PCA as criterions to highlight differences between the $X$ partitions. For references computation, we notice that the same $CS$ (the mean, the variance ...) is used for all partitions and for all key hypotheses $k_j$. One reference is an $L$-dimensional time vector. Thus we have one dataset of $X$ references, for each $k_j$. We denote this set by $V_{ref_{k_j}}$. In what follows, our study will focus on analysing each dataset $V_{ref_{k_j}}$ corresponding to each key hypothesis $k_j$. This analysis will allow the attacker to discriminate the behavior of the secret key with regards to all other key hypotheses. Moreover, it will reduce the computational complexity of the PCA step.

### 3.3 FPCA distinguisher

For one key hypothesis $k_j$, the dependencies between references are made more eligible by PCA, when the references are projected to the new axes system composed by the principal components. The PCA is used to analyze these dependencies by measuring the dispersion of the references in the new coordinate

space. Indeed, the larger the eigenvalue, denoted by $\lambda$, corresponding to one eigenvector is, the greater is the dispersion of the references on this eigenvector. As stated by equation (1), the total variance of one $V_{ref_{k_j}}$ is equal to the sum of all eigenvalues corresponding to all principal components:

$$V_{tot} = \sum_{j=1}^{L} \lambda_j \,. \tag{1}$$

Given a valid power consumption model and one $CS$, there are two cases to be discussed regarding the fluctuation of the total variance when increasing the number of recorded traces. The first case is the one for which the cryptographic implementation is not sensitive to the considered $CS$. In this case, PCA could not discriminate references of the secret key as well for the other key hypotheses.

The second case happens when the implementation is sensitive to the chosen $CS$. In this case, $V_{tot}$ related to $V_{ref_{k_{secret\ key}}}$ is getting high by increasing the number of recorded traces. This can be explained by the fact that the secret key partitioning is the one for which the references are the most different. Intuitively, for an infinity of traces, $V_{tot}$ converges towards the leakage value. By contrast, $V_{tot}$ corresponding to one false key approaches the zero value when increasing the number of traces. This is due to the fact that PCA is not able to discriminate the references.

In order to highlight the dispersion of the references related to the secret key with regards to false keys, we carried out an experiment on DES [19] power consumption traces that are made freely available on line, in the context of the first version of DPA Contest competition [33]. The DES algorithm that has been selected for the competition is unprotected and easily breakable by first-order SCA. More details about this implementation could be found in [11]. For this purpose, we fixed the "mean" as $CS$ and the Hamming distance as power consumption model. Fig. 2 shows the dispersion of references related to the secret key and one false key, when projected to the first and the second principal components. These principal components are the most significant given that they cover a high rate of the total variance so-called explained variance ($EV$). For the *m-th* principal component $PC_m$, this rate is defined by the following ratio:

$$EV(PC_m) = \lambda_m / V_{tot} \,,$$

where $\lambda_m$ is the eigenvalue corresponding to $PC_m$. For $m'$ principal components, we introduce the cumulative explained variance ($CEV$) that is defined by:

$$CEV(PC_1, \ldots, PC_{m'}) = (\sum_{i=1}^{m'} \lambda_i) / V_{tot} \,.$$

In practice, last principal components are usually considered to be related to the noise contribution and only few $m'$ components are retained for analysis.

The main idea behind using PCA is to reduce the dimensionality of power consumption traces in order to take account of the secret information for different

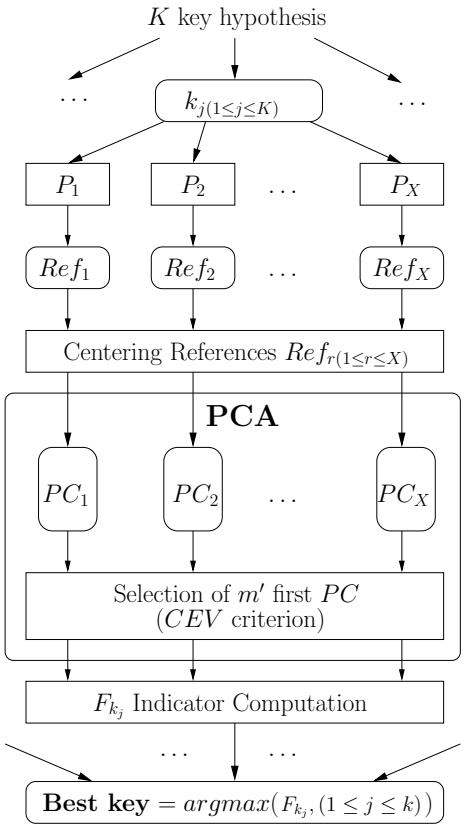


**Fig. 1.** FPCA description.

time samples and thus to properly exploit the leakage. For this purpose we used the cumulative variance criteria to extract the significant components. For instance, we keep only the $m'$ first components which explain more than 95% of the total variance, for each key hypothesis $k_j$.

Then, we propose to compute an indicator $F_{k_j}$ that is defined as follows:

$$F_{k_j}^{CS} = \sum_{m=1}^{m'} (\lambda_m \cdot \Big| \ h(W, C^m)\Big|) = \sum_{m=1}^{m'} (\lambda_m \cdot \Big|\sum_{i=1}^{X} (w_i \cdot c_i^m)\Big|)\,, \tag{2}$$

where $m'$ is the number of retained principal components, $\lambda_m$ is the eigenvalue corresponding to $PC_m$, $h$ is a linear combination function with $C^m = \{c_i^m\}_{i=1}^X$ is the centred coordinate vector of references when projected to $PC_m$ and $W = \{w_i\}_{i=1}^X$ is the associated weight vector. Actually, this indicator takes two factors into consideration: the dispersion and the position of references in the new system coordinate which is composed by the principal components. The dispersion is

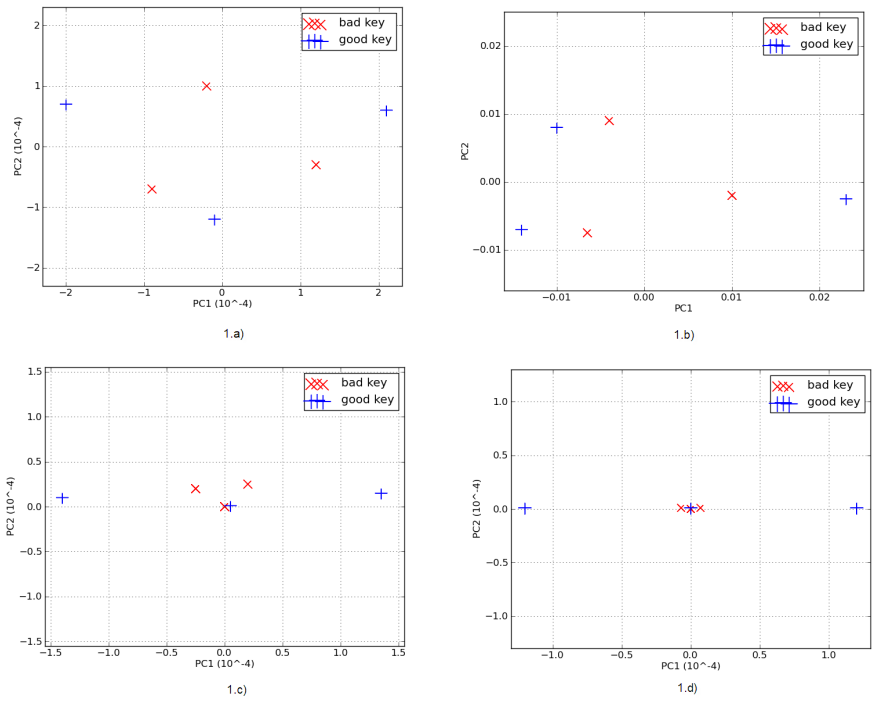


**Fig. 2.** References dispersion for different number of traces: (1.a) 100 traces,(1.b) 1000 traces, (1.c) 10000 traces, (1.d) 81000 traces.

quantified by the value of the eigenvalues $\lambda_m$ and the position by the vector of weights $W$. The best key guess corresponds to the highest value of $F_{k_j}$ regarding all key hypotheses ($argmax(F_{k_j})$).

One other alternative is to consider only the factor of dispersion. This is useful in the case that the position factor is unknown. In fact, the dispersion factor represents a global description of the leakage without the need of more detailed knowledge about the encryption process. The idea is that, if the key guess is correct, PCA applied to the different partitions should be able to explain a big proportion of the variance with only a few components. On the opposite, if the key guess is wrong, the power traces are sorted randomly, and PCA will need more components to explain the same proportion of variance. Thus, a reduced form of the indicator $F_{k_j}$ is deduced from equation (2) and defined as follows:

$$Red\_F_{k_j}^{CS} = \sum_{m=1}^{m'} (\lambda_m) \,.$$

This indicator can be used for Dual rail Precharge Logic (DPL) architectures, like WDDL [10], which aim at making the activity of the cryptographic process constant independently from the manipulated data. Statistically, the idea behind DPL is to force all references to have the same statistic and probabilistic features, for all made partitions. However, in the real life application, an ideal DPL implementation could not exist. In that sense, the reduced indicator $Red\_F_{k_j}^{CS}$ can exploit the leaked information without the knowledge of the position factor.

## 4 FPCA on DES implementations

All our experiments were conducted on real power consumption traces recorded from three different hardware implementations of DES coprocessor. The first architecture is the unprotected DES of DPA Contest. The second and the third ones deal with two masking styles: USM and Masked-ROM [18] DES implementation which are configured in an Altera Stratix II FPGA on the SASEBO-B evaluation board provided by the RCIS [25]. Moreover, we note that the length of acquired side channel traces covers only the first two rounds for all investigated DES implementations.

Following the recent advances concerning the comparison of univariate side-channel distinguishers [30], Standaert et al. proposed two evaluation metrics [31] to assess the performance of different attacks. On one hand, the first-order success rate expresses the probability that, given a pool of traces, the attack's best guess is the correct key. On the other hand, the guessing entropy measures the position of the correct key in a list of key hypotheses ranked by a distinguisher.

In this paper, we deal with DoM, DPA, CPA, and VPA attacks. These attacks have shown their efficiency to break cryptographic implementations. Moreover, they are the basis of new derived distinguishers like the Spearman's rank correlation [4]; the correlation concept of Kendall is also of potential interest. Recently, Gierlichs et al. have presented an article dealing with the comparison of many existing distinguishers related to the aforementioned attacks [9]. For the reason that we aim at making a reliable evaluation for our attack (FPCA), the rest of the paper deals with experiments on unprotected and masked implementations which have been the target of the mentioned attacks.

### 4.1 FPCA on unprotected DES

In order to mount a successful FPCA on unprotected DES we fixed the "mean" as $CS$, as it is shown that such implementation is very vulnerable against differential attacks which are generally based on the "mean" in their calculations. In fact, the leakage related to the mean is linearly correlated to the power consumption model $HD = \{0, 1, 2, 3, 4\}$. For this purpose, the weight vector $W$ can be defined as follows: $W = \{-2, -1, 0, +1, +2\}$. One other alternative is to consider the probability that one trace belongs to one partition according to one power consumption model. Hence $W = \{-0.25, -1, 0, +1, +0.25\}$. Results regarding attacks on unprotected DES implementation are depicted in Fig. 3

and Fig. 4. Indeed, the first-order success rate shows a superior performance of FPCA attack. This can be explained by the fact that DoM, CPA, and DPA are implicitly taking into account only the position factor relatively to our proposed attack. According to Fig. 4, FPCA needs around 160 traces to perform a successful attack. Unsurprisingly, the guessing entropy metric depicted in Fig. 3 is in accordance with the first success rate results. One note is that FPCA is able to distinguish the *secret key* at an early stage. In fact, only 30 traces are required to get the *secret key* in the top ten of the key hypotheses rank list.

### 4.2 FPCA on Masked DES

Basically, masking technique is considered to be a powerful countermeasure against SCA. Indeed, it aims at masking the intermediate values that occur during encryption and decryption process. Many masking schemes have been proposed to the cryptographic community for symmetric encryption algorithms (DES, AES, ...) [14, 22, 32]. Basically, they differ in term of hardware design complexity. But, they all aim at fulfilling the same goal by ensuring the resistance against first-order SCA like DPA and CPA. Statistically, an ideal masking implementation is one for which all references, for all made partitions, are the same when using the mean as $CS$. However, it has been proved that masking technique is still susceptible to first-order SCA as long as glitches problem remains not completely resolved [32]. For instance, authors in [18], have shown that one masked structure so-called "Universal Substitution boxes with Masking" (USM) is vulnerable against DPA. Moreover, masked implementations are not resistant against new variants of SCA like $VPA$ which is mainly based on the variance analysis. It is also shown that a full-fledged masked DES implementation using a ROM (Masked-ROM) is breakable by $VPA$ attack, in spite of its high resistance against first-order attacks. In what follows, we use the same power consumption model as described in [18] to perform the FPCA on USM and Masked-ROM DES implementations.

First, in order to make a fair evaluation for our attack on USM DES structure we kept the "mean" as $CS$ and we classified traces into five partitions for each key hypothesis $k_j$. For reasons of clarity, comparison is made between FPCA and DPA for which we realised the best performance with regards to DoM and CPA. Results are deduced from Fig. 5 and Fig. 6. Obviously, according to the first-order success rate metric shown in Fig. 6, FPCA is more efficient than DPA. Indeed, 15000 traces are needed for DPA to achieve a rate of 0.8. Whereas, for the same rate, FPCA attack requires only 10000 traces. The guessing entropy metric, is quite equivalent for both attacks. Second, we targeted a Masked-ROM DES implementation. For this purpose, we chose the variance as $CS$, as it has shown that such implementation is sensitive to VPA, which is based on variance analysis [17,18]. Fig. 7 shows that the guessing entropy curve, related to FPCA, approaches the best rank (the zero rank) more rapidly than VPA. Moreover, the success rate metric depicted in Fig. 8 reveals noticeable differences between both attacks.

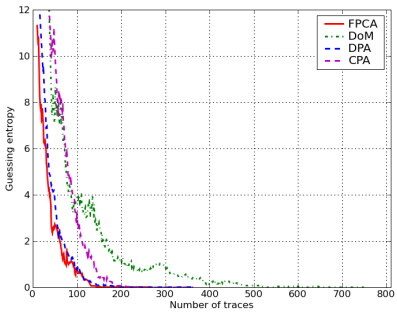


**Fig. 3.** Unprotected DES guessing entropy metric.

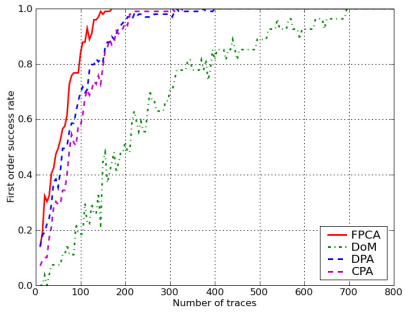


**Fig. 4.** Unprotected DES 1st-order success rate metric.

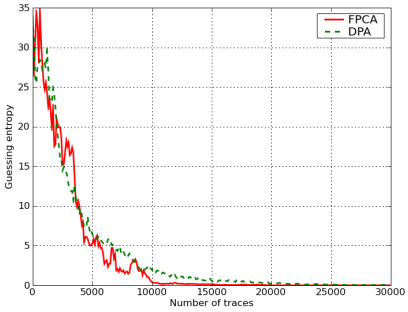


**Fig. 5.** USM DES guessing entropy metric.

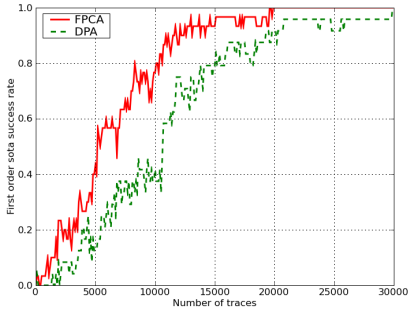


**Fig. 6.** USM DES 1st-order success rate metric.

## 5 Conclusion and outlooks

In this work we proposed a new variant of SCA called FPCA, which is mainly based on Principal Components Analysis (PCA), the powerful multivariate data analytic tool. We have shown the efficiency of FPCA on unprotected and protected cryptographic implementations. Moreover, we have empirically shown its superior performance with regards to existing attacks (DoM, DPA, CPA, VPA). Our future work consists in investigating new ways to improve FPCA. Actually, we are looking for new applications based on other multivariate data analytic tools such as the Linear Discriminant Analysis (LDA), PCA based Spearman correlation, Kernel PCA or Independent Component Analysis (ICA), which have been proposed as new alternatives to the basic PCA. One other possible key point that could be investigated is the improvement of our distinguisher by combining different $CS$ (the mean, the variance, the entropy ...), in order to make more eligible the description of the leakage related to the secret information.

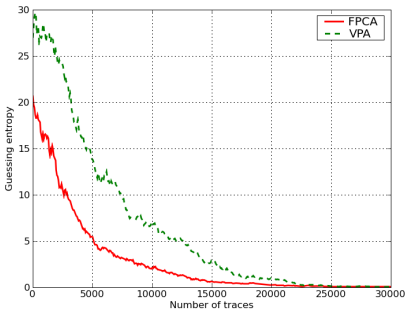


**Fig. 7.** Masked-ROM guessing entropy metric.

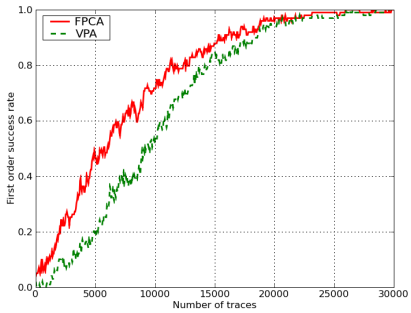


**Fig. 8.** Masked-ROM 1st-order success rate metric.

## References

1. M. A. E. Aabid, S. Guilley, and P. Hoogvorst. Template Attacks with a Power Model. Cryptology ePrint Archive, Report 2007/443, December 2007. `http://eprint.iacr.org/2007/443/`.
2. Abaraham, Dolan, Double, and Stevens. Transaction security system. *IBM Systems Journal*, 30(2):206–229, 1991.
3. C. Archambeau, É. Peeters, F.-X. Standaert, and J.-J. Quisquater. Template Attacks in Principal Subspaces. In *CHES*, volume 4249 of *LNCS*, pages 1–14. Springer, October 10-13 2006. Yokohama, Japan.
4. L. Batina, B. Gierlichs, and K. Lemke-Rust. Comparative Evaluation of Rank Correlation Based DPA on an AES Prototype Chip. In *ISC*, volume 5222 of *Lecture Notes in Computer Science*, pages 341–354. Springer, September 15-18 2008. Taipei, Taiwan.
5. R. Bevan and E. Knudsen. Ways to Enhance Differential Power Analysis. In *ICISC*, volume 2587 of *Lecture Notes in Computer Science*, pages 327–342. Springer, 2002.
6. É. Brier, C. Clavier, and F. Olivier. Correlation Power Analysis with a Leakage Model. In *CHES*, volume 3156 of *LNCS*, pages 16–29. Springer, August 11–13 2004. Cambridge, MA, USA.
7. H. David and H. N. Nagaraja. *Order Statistics*. Wiley.
8. B. Gierlichs, L. Batina, P. Tuyls, and B. Preneel. Mutual Information Analysis – A Generic Side-Channel Distinguisher. In E. Oswald and P. Rohatgi, editors, *Cryptographic Hardware and Embedded Systems - CHES 2008*, volume 5154 of *Lecture Notes in Computer Science*, pages 426–442, Washington DC, US, 2008. Springer-Verlag.
9. B. Gierlichs, E. De Mulder, B. Preneel, and I. Verbauwhede. Empirical comparison of side channel analysis distinguishers on DES in hardware. In IEEE, editor, *ECCTD. European Conference on Circuit Theory and Design*, pages 391–394, August 23-27 2009. Antalya, Turkey.
10. S. Guilley, S. Chaudhuri, L. Sauvage, P. Hoogvorst, R. Pacalet, and G. M. Bertoni. Security Evaluation of WDDL and SecLib Countermeasures against Power Attacks. *IEEE Transactions on Computers*, 57(11):1482–1497, nov 2008.

11. S. Guilley, P. Hoogvorst, and R. Pacalet. A Fast Pipelined Multi-Mode DES Architecture Operating in IP Representation. *Integration, The VLSI Journal*, 40(4):479–489, July 2007. DOI: 10.1016/j.vlsi.2006.06.004.
12. I. T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics, 2002. ISBN: 0387954422.
13. R. Khattree and D. N.Naik. *Multivariate data reduction and descrimination*. 2000.
14. I. Koichi, T. Masahiko, and T. Naoya. Encryption secured against DPA, June 10 2008. Fujitsu US Patent 7386130, `http://www.patentstorm.us/patents/7386130/fulltext.html`.
15. U. Kyungnam Kim Department of Computer Science University of Maryland. Face recognition using principal component analysis. Available online on 26 february 2002.
16. T.-H. Le, C. Canovas, and J. Clédière. An overview of side channel analysis attacks. In *ASIACCS*, pages 33–43. ASIAN ACM Symposium on Information, Computer and Communications Security, 2008. DOI: 10.1145/1368310.1368319. Tokyo, Japan.
17. Y. Li, K. Sakiyama, L. Batina, D. Nakatsu, and K. Ohta. Power Variance Analysis Breaks a Masked ASIC Implementation of AES. In *DATE'10*. IEEE Computer Society, March 8-12 2010. Dresden, Germany.
18. H. Maghrebi, J.-L. Danger, F. Flament, and S. Guilley. Evaluation of Countermeasures Implementation Based on Boolean Masking to Thwart First and Second Order Side-Channel Attacks. In *SCS*, IEEE, November 6–8 2009. Jerba, Tunisia. Complete version online: `http://hal.archives-ouvertes.fr/hal-00425523/en/`.
19. NIST/ITL/CSD. Data Encryption Standard. FIPS PUB 46-3, Oct 1999. `http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf`.
20. P. Kocher and J. Jaffe and B. Jun. Differential Power Analysis. In *Proceedings of CRYPTO'99*, volume 1666 of *LNCS*, pages 388–397. Springer-Verlag, 1999. (PDF).
21. É. Peeters, F.-X. Standaert, and J.-J. Quisquater. Power and electromagnetic analysis: Improved model, consequences and comparisons. *Integration, The VLSI Journal, special issue on* "Embedded Cryptographic Hardware", 40:52–60, January 2007. DOI: 10.1016/j.vlsi.2005.12.013.
22. T. Popp and S. Mangard. Masked Dual-Rail Pre-charge Logic: DPA-Resistance Without Routing Constraints. In LNCS, editor, *Proceedings of CHES'05*, volume 3659 of *LNCS*, pages 172–186. Springer, Sept 2005. Edinburgh, Scotland, UK.
23. C. Rechberger and E. Oswald. Practical Template Attacks. In *WISA*, volume 3325 of *LNCS*, pages 443–457. Springer, august 2004.
24. G. SAPORTA. *Probabilités analyse des données et statistiques*. 2008.
25. SASEBO board from the Japanese RCIS-AIST: `http://www.rcis.aist.go.jp/special/SASEBO/index-en.html`.
26. J. Shlens. A tutorial in Principal Component Analysis. Available online on 10 decembre 2005.
27. S.Kolenikov and G.Angeles. The use of discrete data in PCA for socio-economic status evaluation. Available online on 2 february 2005.
28. L. I. Smith. A tutorial in Principal Component Analysis. Available online on 26 february 2002.
29. F.-X. Standaert and C. Archambeau. Using Subspace-Based Template Attacks to Compare and Combine Power and Electromagnetic Information Leakages. In *CHES*, volume 5154 of *Lecture Notes in Computer Science*, pages 411–425. Springer, August 10–13 2008. Washington, D.C., USA.

30. F.-X. Standaert, B. Gierlichs, and I. Verbauwhede. Partition vs. comparison side-channel distinguishers: An empirical evaluation of statistical tests for univariate side-channel attacks against two unprotected cmos devices. In *ICISC*, pages 253–267, 2008.
31. F.-X. Standaert, T. Malkin, and M. Yung. A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 443–461. Springer, April 26-30 2009. Cologne, Germany.
32. K. S. Stefan Mangard. Pinpointing the Side-Channel Leakage of Masked AES Hardware Implementations. In *CHES*, volume 4249 of *LNCS*, pages 76–90. Springer, October 10-13 2006. Yokohama, Japan. DOI: 10.1007/11894063_7.
33. TELECOM ParisTech SEN research group. DPA Contest (1st edition), 2008–2009. `http://www.DPAcontest.org/`.
34. Zeng Guang Hou. PCA for data fusion and navigation of mobile robots. volume 3495 of *LNCS*, pages 610–611. Springer, 2005.