Gaussian Mixture Models for Higher-Order Side Channel Analysis*

Kerstin Lemke-Rust and Christof Paar

Horst Görtz Institute for IT Security
Ruhr University Bochum
44780 Bochum, Germany
{lemke,cpaar}@crypto.rub.de

Abstract. We introduce the use of multivariate Gaussian mixture models for enhancing higher-order side channel analysis on masked cryptographic implementations. Our contribution considers an adversary with incomplete knowledge at profiling, i.e., the adversary does not know random numbers used for masking. At profiling, the adversary observes a mixture probability density of the side channel leakage. However, the EM algorithm can provide estimates on the unknown parameters of the component densities using samples drawn from the mixture density. Practical results are presented and confirm the usefulness of Gaussian mixture models and the EM algorithm. Especially, success rates obtained by automatic classification based on the estimates of the EM algorithm are very close to success rates of template attacks.

Keywords: Side Channel Cryptanalysis, Higher-Order Analysis, Gaussian Mixture Models, EM Algorithm, Boolean Masking, Templates, Second-Order DPA.

1 Introduction

Since the paper of Kocher et al. [12] on Simple Power Analysis (SPA) and Differential Power Analysis (DPA) a great variety of similar implementation attacks and appropriate defenses has been proposed. For these kinds of attacks it is assumed that measurable observables depend on the internal state of a cryptographic algorithm. This impact is specific for each implementation and represents the *side channel*. Side channel attacks using instantaneous physical observables, e.g., the power consumption or electromagnetic radiation [12, 9] have to be mounted in the immediate vicinity of the device.

Besides univariate attacks such as DPA, multivariate analysis has been already adapted to side channel analysis by [5]. Multivariate analysis requires stronger assumptions on adversary's capabilities, i.e., it is assumed that the adversary can use a training device for learning probability density functions of

* Supported by the European Commission through the IST Contract IST-2002-507932 ECRYPT, the European Network of Excellence in Cryptology.

the observables. A template [5] is a multivariate Gaussian probability density function for one key dependent internal state of the implementation.

In response to side channel attacks designers of cryptographic implementations may include randomization techniques such as secret splitting or masking schemes, e.g., [4, 6]. These randomization techniques shall prevent from predicting any relevant bit in any cycle of the implementation. As result, statistical tests using physical observables at one instant cannot be assumed to be successful in key recovery. However, as already indicated by [12] high-order differential analysis can combine multiple instants from within one measurement trace.

Second-order DPA as proposed by [15, 22] uses again univariate statistics. It combines measurements at related time instants before statistics is applied. Related work on second-order DPA can also be found in [11, 19, 17, 21, 16, 1]. Except for [1] these contributions assume that the leakage of the cryptographic device corresponds to the Hamming weight model. Reference [1] acts on the different assumption that the adversary has access to an implementation with a biased random number generator at profiling.

An adversary with complete knowledge at profiling is able to build templates for all possible combinations of keys and random masks. At key recovery, the adversary then evaluates a mixture of densities for each key dependent internal state [20].

It is still an open research question whether an adversary with incomplete knowledge at profiling is capable of mounting a multivariate side channel analysis on unbiased masked cryptographic implementations. This paper provides a solution for this problem based on the use of Gaussian mixture models.

2 Our Model

We consider a two-stage side channel attack on a masked cryptographic implementation of a symmetric primitive, e.g., a block cipher. In the first stage of the attack, i.e., the *profiling stage*, the adversary aims at learning the data dependent probability density function (p.d.f.) of the side channel leakage emanating from the masked implementation at run-time. In the second stage, i.e., the *key recovery stage*, the adversary applies statistics gained from the profiling stage in order to recover an unknown secret key from the masked cryptographic implementation.

The cryptographic implementation of a symmetric primitive is assumed to apply a boolean masking scheme, i.e., the cryptographic key $k \in \{0,1\}^d$ is masked with an unpredictable uniformly distributed random number $y \in \{0,1\}^d$ that is internally generated by the cryptographic device. As result of masking, the internal state k is randomly mapped to $k \oplus y$ at run-time, i.e, one random representation of the overall parameter space. Therefore, internal states are no longer predictable by solely guessing on the key k thereby preventing both single-order simple and differential side channel attacks.

Higher-order analysis, however, considers both multiple internal states and multiple side channel observations of each internal state. Though our algorithms are also applicable for multiple internal states, in this contribution we restrict to two internal states, i.e., y and $y \oplus k$ for simple side channel attacks and y and $y \oplus k \oplus x$ for differential side channel attacks with $x \in \{0,1\}^d$ being a known random number. It is assumed that the mask y is freshly generated and used only once.[1]

Let $\boldsymbol{I}(x,k,y) = (I_1, \ldots, I_m)^T$ be an m-dimensional side channel observable with $\boldsymbol{i}(x,k,y) = (i_1, \ldots, i_m)^T$ representing one particular measurement outcome of $\boldsymbol{I}(x,k,y)$. Each vectorial sample includes some hidden physical leakage on the two internal states y and $y \oplus k \oplus x$.

We make the following assumptions regarding the side channel adversary $\mathcal{A}$.

- *Adversary's input of the profiling stage:* $\mathcal{A}$ is given N vectorial samples $\boldsymbol{i}(x,k,y)$ produced from the measurement setup $\mathcal{M}$ during run-time of the implementation of the cryptographic primitive $\mathcal{P}$ operating on random numbers x, k, and y.
- *Adversary's a-priori knowledge in the profiling stage:* $\mathcal{A}$ knows input $x \in \{0,1\}^d$ and key $k \in \{0,1\}^d$ that was processed by $\mathcal{P}$ at each of the N samples.
- *Adversary's output of the profiling stage:* $\mathcal{A}$ outputs a multivariate p.d.f. $f^{(x,k)}$ of the side channel leakage for each pair of (x,k).
- *Adversary's input of the key recovery stage:* $\mathcal{A}$ is given N° vectorial samples $\boldsymbol{i}(x,k^\circ,y)$ produced from the measurement setup $\mathcal{M}$ during run-time of the implementation of the cryptographic primitive $\mathcal{P}$ operating on a fixed key k° and random numbers x and y.
- *Adversary's a-priori knowledge in the key recovery stage:* $\mathcal{A}$ knows x that was processed by $\mathcal{P}$ at each of the N° samples. $\mathcal{A}$ knows the multivariate p.d.f.s $f^{(x,k)}$ for the side channel leakage for each pair of (x,k) from the profiling stage.
- *Adversary's output of the key recovery stage:* $\mathcal{A}$'s output is a key guess k^*.
- *Adversary's success at the key recovery stage:* $\mathcal{A}$ is successful if $k^* = k^\circ$. If key recovery is repeated multiple times the success rate of the adversary is the percentage of correct key guesses.[2]

One may think of $\mathcal{A}$ being an administrative user who is able to load test keys into one instance of a set of identical cryptographic devices and to run the cryptographic primitive $\mathcal{P}$. As common in side channel attacks $\mathcal{A}$ has physical access to the cryptographic device. $\mathcal{A}$ does not know the details of the cryptographic implementation and $\mathcal{A}$ is not able to modify or tamper with the cryptographic implementation. Further, $\mathcal{A}$ is not assumed to have any a-priori knowledge on the physical leakage function of $\mathcal{P}$, i.e., the impact of internal states on the side channel leakage[3].

[1] Note that weak masking schemes may re-use the mask in subsequent iterations of, e.g., a round function. In such a case the use of multiple internal states may be favorable.

[2] Note that the success rate also depends on N and N°.

[3] Because of that, $\mathcal{A}$ is not restricted to any specific leakage models such as the Hamming weight model.

It is assumed that the measurement vector $\boldsymbol{z} := \boldsymbol{i}(x, k, y) \in \mathbb{R}^m$ is distributed according to an m-variate Gaussian density

$$\mathcal{N}(\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^m |\boldsymbol{\Sigma}|}} \exp\left[-\frac{1}{2}(\boldsymbol{z} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{z} - \boldsymbol{\mu})\right] \tag{1}$$

where $\boldsymbol{\mu}$ is the mean vector, $\boldsymbol{\Sigma}$ the covariance matrix of the normally distributed random variable $\boldsymbol{Z}$ with $\boldsymbol{\Sigma} = (\sigma_{uv})_{1 \le u,v \le m}$ and $\sigma_{uv} := \mathbb{E}(Z_u Z_v) - \mathbb{E}(Z_u)\mathbb{E}(Z_v)$, $|\boldsymbol{\Sigma}|$ denotes the determinant of $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}^{-1}$ its inverse. A Gaussian distribution is completely determined by its parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Note that both parameters can depend on the data (x, k, y), therefore enabling side channel leakage.

3 Gaussian Mixture Models

In the profiling stage $\mathcal{A}$ determines the multivariate p.d.f. of $\boldsymbol{i}(x, k, Y)$ for each combination of (x, k) and the random variable Y, i.e., in total 2^{2d} p.d.f.s. In practice, one may argue that this number can be reduced to 2^d p.d.f.s characterizing $\boldsymbol{i}(x \oplus k, Y)$.

For each (x, k) $\mathcal{A}$ observes a *mixture p.d.f.*

$$f(\boldsymbol{z}, \theta^{(x,k)}) = \sum_{j=0}^{2^d-1} \alpha_j^{(x,k)} \mathcal{N}(\boldsymbol{z}, \boldsymbol{\mu}_j^{(x,k)}, \boldsymbol{\Sigma}_j^{(x,k)}) \tag{2}$$

that consists of 2^d m-variate Gaussian *component p.d.f.s* $\mathcal{N}(\boldsymbol{z}, \boldsymbol{\mu}_j^{(x,k)}, \boldsymbol{\Sigma}_j^{(x,k)})$. Herein, j denotes the mask with α_j being the probability to indeed observe mask j. The α_j satisfy

$$\alpha_j^{(x,k)} \ge 0, j = 0, \ldots, 2^d - 1, \text{ and } \sum_{j=0}^{2^d-1} \alpha_j^{(x,k)} = 1. \tag{3}$$

A Gaussian mixture p.d.f. is completely defined by parameters

$$\theta^{(x,k)} = \left(\alpha_0^{(x,k)}, \boldsymbol{\mu}_0^{(x,k)}, \boldsymbol{\Sigma}_0^{(x,k)}, \ldots, \alpha_{2^d-1}^{(x,k)}, \boldsymbol{\mu}_{2^d-1}^{(x,k)}, \boldsymbol{\Sigma}_{2^d-1}^{(x,k)}\right) . \tag{4}$$

Example 1. Fig. 1 provides an illustration of the mixing of p.d.f.s considering $x, k, y \in \{0, 1\}$ that was generated from measurement samples for $x \oplus k = 0$. It can be seen that separating the distributions for $y = 0$ and $y = 1$ from the mixed distribution is not a trivial problem as both p.d.f.s significantly overlap.

Finite mixture models are well known from cluster analysis and pattern recognition [7, 13, 3, 18, 8]. In a typical problem, features from known observations have to be learnt and statistical classifiers have to be trained by using means of similarity. These classifiers are then available for recognition of unknown observations. This two-stage procedure is very similar to applying a two-stage side

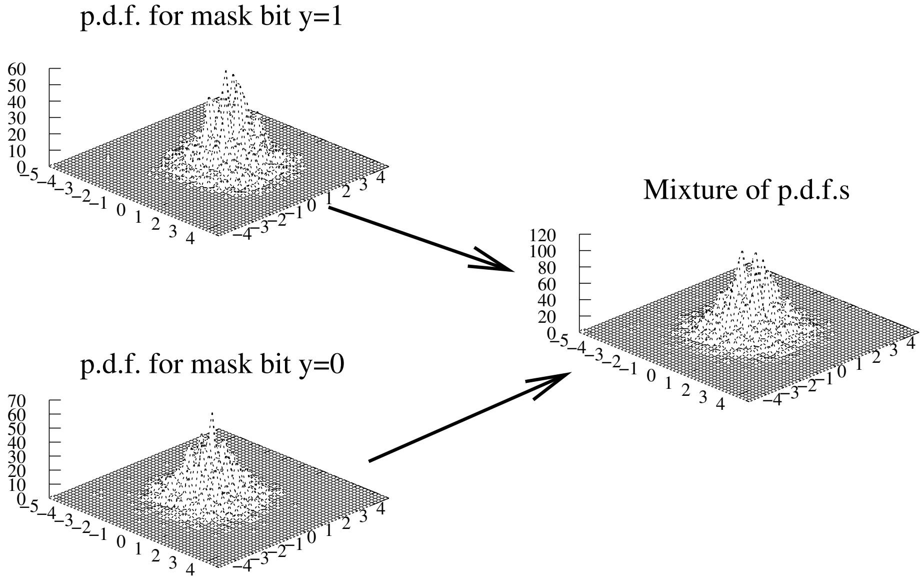

Fig. 1. Two-dimensional p.d.f.s extracted from experimental data. The x and y coordinates represent the measurement outcomes at two instants t_1 (y leaks) and t_2 ($y \oplus k \oplus x$ leaks). The plot on the right shows a mixture of p.d.f.s as it can be recognized by $\mathcal{A}$ at profiling. A more powerful adversary knowing masks y at profiling can determine the original two p.d.f.s on the left side. The measurement outcomes were initially standardized with $z_i := (z_i - \mu_i)/s_i$ wherein μ_i is the mean value and s_i the standard deviation for each scalar component of $\boldsymbol{z}_i$.

channel attack. In more detail, a powerful adversary in the position of building templates is given labelled samples and complete knowledge about processed data. This context is also known as *supervised learning*. Such a powerful adversary knowing y, e.g., the developer of the cryptographic implementation, can build m-variate Gaussian densities $\mathcal{N}(\boldsymbol{z}, \boldsymbol{\mu}^{(x,k,y)}, \boldsymbol{\Sigma}^{(x,k,y)})$ for each tuple (x, k, y), i.e., 2^{3d} templates. Accordingly to $\mathcal{A}$, it may be assumed that this powerful adversary can also manage with 2^{2d} templates $\mathcal{N}(\boldsymbol{z}, \boldsymbol{\mu}^{(x \oplus k,y)}, \boldsymbol{\Sigma}^{(x \oplus k,y)})$. The adversary $\mathcal{A}$ considered in this contribution, however, observes the system response on input patterns, but has incomplete knowledge about the internal state of the system, especially $\mathcal{A}$ does not know any labels of samples. This problem of *unsupervised learning* is the more difficult one.

The problem statement for $\mathcal{A}$ is at follows. Given an observation of a mixture of $f(\boldsymbol{z}, \theta^{(x,k)})$ in (2) estimate the parameters in (4) for the observed multi-modal[4]. p.d.f..

Some side information make the estimation easier for $\mathcal{A}$ if compared to other problems of pattern recognition:

[4] A density is said to be multi-modal if it includes several local maxima.

– The number of component p.d.f.s is known to be 2^d.
– The component p.d.f.s are uniformly distributed in an efficient masking scheme:

$$\alpha_j^{(x,k)} \approx 2^{-d} \tag{5}$$

Further, $\mathcal{A}$ does not need to identify the labels of the component p.d.f.s for key recovery, cf. Section 3.2.

This contribution considers four different variants for use at high-order side channel analysis. Three variants come from assumptions in order to reduce the number of unknown parameters in this scheme.

– Variant 1: The list of free parameters (4) is reduced to

$$\theta^{(x,k)} = \left(\boldsymbol{\mu}_0^{(x,k)}, \ldots, \boldsymbol{\mu}_{2^d-1}^{(x,k)}\right) . \tag{6}$$

– Variant 2: The list of free parameters (4) is reduced to

$$\theta^{(x,k)} = \left(\alpha_0^{(x,k)}, \boldsymbol{\mu}_0^{(x,k)}, \ldots, \alpha_{2^d-1}^{(x,k)}, \boldsymbol{\mu}_{2^d-1}^{(x,k)}\right) . \tag{7}$$

– Variant 3: The list of free parameters (4) is reduced to

$$\theta^{(x,k)} = \left(\alpha_0^{(x,k)}, \boldsymbol{\mu}_0^{(x,k)}, \ldots, \alpha_{2^d-1}^{(x,k)}, \boldsymbol{\mu}_{2^d-1}^{(x,k)}, \boldsymbol{\Sigma}^{(x,k)}\right) \tag{8}$$

wherein $\boldsymbol{\Sigma}$ denotes one common covariance matrix.
– Variant 4: All parameters are unknown. The list of parameters is given in (4).

Table 1. Number of free parameters in the Gaussian mixture model.

Variant	$\alpha_j^{(x,k)}$	$\boldsymbol{\mu}_j^{(x,k)}$	$\boldsymbol{\Sigma}_j^{(x,k)}$ or $\boldsymbol{\Sigma}^{(x,k)}$	Total
1	$\times$	$2^d m$	$\times$	$2^d m$
2	$2^d - 1$	$2^d m$	$\times$	$2^d(1+m) - 1$
3	$2^d - 1$	$2^d m$	$(m^2+m)/2$	$2^d(1+m) + (m+m^2)/2 - 1$
4	$2^d - 1$	$2^d m$	$2^d(m^2+m)/2$	$2^d(1+3m/2+m^2/2) - 1$

Example 2. If $d = 1$ and $m = 2$ (smallest reasonable mixture) the number of free parameters is 4 for Variant 1, 5 for Variant 2, 8 for Variant 3, and 11 for Variant 4.

Note that the estimation of component p.d.f.s is required for each (x, k), respectively for each $(x \oplus k)$. For the estimation of the component densities, the number of available measurements at profiling is on average reduced to

$$N^{(x,k)} \approx \frac{N}{2^{2d}} \tag{9}$$

for the characterization of $\boldsymbol{i}(x, k, Y)$ and to

$$N^{(x \oplus k)} \approx \frac{N}{2^d} \tag{10}$$

considering $\boldsymbol{i}(x \oplus k, Y)$.

Example 3. If $d = 1$ one obtains $N^{(x,k)} \approx \frac{N}{4}$ and $N^{(x \oplus k)} \approx \frac{N}{2}$. However, if $d = 8$ this yields to $N^{(x,k)} \approx \frac{N}{2^{16}}$ and $N^{(x \oplus k)} \approx \frac{N}{2^8}$, thereby drastically reducing the number of measurements that are available for the estimation of component p.d.f.s for each (x, k).

3.1 The EM Algorithm

For the estimation of the free parameters we propose to use the expectation-maximization (EM) algorithm that is based on a maximum-likelihood estimation and most favorable for practical applications [14, 7, 18].

The likelihood function is the product of $f(\boldsymbol{z}_1, \theta^{(x,k)}) \cdot f(\boldsymbol{z}_2, \theta^{(x,k)}) \cdot \dots \cdot f(\boldsymbol{z}_{N^{(x,k)}}, \theta^{(x,k)})$. This likelihood function is aimed to be maximized regarding the free parameters for each variant under the constraints of (3). For practical purposes one evaluates the logarithmic likelihood function

$$\mathcal{L}^{(x,k)} := \sum_{i=1}^{N^{(x,k)}} \ln f(\boldsymbol{z}_i, \theta^{(x,k)}) = \sum_{i=1}^{N^{(x,k)}} \ln \left(\sum_{j=0}^{2^d-1} \alpha_j^{(x,k)} \mathcal{N}(\boldsymbol{z}_i, \boldsymbol{\mu}_j^{(x,k)}, \boldsymbol{\Sigma}_j^{(x,k)}) \right) . \tag{11}$$

We treat the additional constraint of (5) as a soft constraint for Variant 2, Variant 3, and Variant 4, i.e., the deviation of the parameters is controlled as part of the estimation process and estimations with high deviations from (5) as result of the EM Algorithm are withdrawn.

The EM algorithm is an iterative algorithm that requires initial values for the set of parameters $\alpha_j^{(x,k)}$, $\boldsymbol{\mu}_j^{(x,k)}$ and $\boldsymbol{\Sigma}_j^{(x,k)}$. We follow the recommendation of [3] to initialize $\boldsymbol{\Sigma}_j^{(x,k)}$ with the identity map $\mathbf{I}$ on $\mathbb{R}^m$. For $\alpha_j^{(x,k)}$ we choose a uniform distribution as in (5), and the initial value of $\boldsymbol{\mu}_j^{(x,k)}$ is determined by randomly selecting a start value in a given interval for each scalar component of $\boldsymbol{\mu}_j^{(x,k)}$. Each estimation process is stopped if the maximization of (11) by using the estimators $\hat{\theta}^{(x,k)}$ of the $(l+1)$-th iteration converges if compared to the estimated parameters $\hat{\theta}^{(x,k)}$ of the l-th iteration [3, 18]. For the convergence one evaluates whether the growth of (11) is smaller than a pre-defined threshold, e.g., $\epsilon = 10^{-6}$, after each iteration. As the estimation process outcomes depend on the initialization, the EM algorithm is repeated with many random initialization values for $\boldsymbol{\mu}_j^{(x,k)}$ and the estimated parameters leading to the maximum likelihood in (11) are finally selected as EM estimates.

Application to Variant 4: Each iteration includes the Expectation Step (E-Step), the Maximization Step (M-Step), and the computation of (11) to check for convergence of the estimated parameters [3, 18, 7].

Expectation Step (E-Step):

$$\alpha_{jn} := \frac{\hat{\alpha}_j^{(x,k)}\, \mathcal{N}(\boldsymbol{z}_n, \hat{\boldsymbol{\mu}}_j^{(x,k)}, \hat{\boldsymbol{\Sigma}}_j^{(x,k)})}{\sum_{i=0}^{2^d-1} \hat{\alpha}_i^{(x,k)}\, \mathcal{N}(\boldsymbol{z}_n, \hat{\boldsymbol{\mu}}_i^{(x,k)}, \hat{\boldsymbol{\Sigma}}_i^{(x,k)})} \tag{12}$$

Maximization Step (M-Step):

$$\hat{\alpha}_j^{(x,k)} = \frac{1}{N^{(x,k)}} \sum_{n=1}^{N^{(x,k)}} \alpha_{jn} \tag{13}$$

$$\hat{\boldsymbol{\mu}}_j^{(x,k)} = \frac{1}{\sum_{n=1}^{N^{(x,k)}} \alpha_{jn}} \sum_{n=1}^{N^{(x,k)}} \alpha_{jn} \boldsymbol{z}_n \tag{14}$$

$$\hat{\boldsymbol{\Sigma}}_j^{(x,k)} = \frac{1}{\sum_{n=1}^{N^{(x,k)}} \alpha_{jn}} \sum_{n=1}^{N^{(x,k)}} \alpha_{jn} \left(\boldsymbol{z}_n - \hat{\boldsymbol{\mu}}_j^{(x,k)}\right) \left(\boldsymbol{z}_n - \hat{\boldsymbol{\mu}}_j^{(x,k)}\right)^T \tag{15}$$

Application to Variant 3 If the same covariance matrix $\boldsymbol{\Sigma}$ is used for all component p.d.f.s equations (12) and (15) are modified to (16) and (17), respectively [3].

$$\alpha_{jn} := \frac{\hat{\alpha}_j^{(x,k)}\, \mathcal{N}(\boldsymbol{z}_n, \hat{\boldsymbol{\mu}}_j^{(x,k)}, \hat{\boldsymbol{\Sigma}}^{(x,k)})}{\sum_{i=0}^{2^d-1} \hat{\alpha}_i^{(x,k)}\, \mathcal{N}(\boldsymbol{z}_n, \hat{\boldsymbol{\mu}}_i^{(x,k)}, \hat{\boldsymbol{\Sigma}}^{(x,k)})} \tag{16}$$

$$\hat{\boldsymbol{\Sigma}}^{(x,k)} = \frac{1}{N^{(x,k)}} \sum_{n=1}^{N^{(x,k)}} \sum_{j=0}^{2^d-1} \alpha_{jn} \left(\boldsymbol{z}_n - \hat{\boldsymbol{\mu}}_j^{(x,k)}\right) \left(\boldsymbol{z}_n - \hat{\boldsymbol{\mu}}_j^{(x,k)}\right)^T \tag{17}$$

Application to Variant 2 This variant replaces (12) with (18) in the E-Step and uses (13) and (14) in the M-Step.

$$\alpha_{jn} := \frac{\hat{\alpha}_j^{(x,k)}\, \mathcal{N}(\boldsymbol{z}_n, \hat{\boldsymbol{\mu}}_j^{(x,k)}, \boldsymbol{\Sigma}^{(x,k)})}{\sum_{i=0}^{2^d-1} \hat{\alpha}_i^{(x,k)}\, \mathcal{N}(\boldsymbol{z}_n, \hat{\boldsymbol{\mu}}_i^{(x,k)}, \boldsymbol{\Sigma}^{(x,k)})} \tag{18}$$

Application to Variant 1 This variant replaces (12) with (19) in the E-Step and uses solely (14) in the M-Step [7].

$$\alpha_{jn} := \frac{\alpha_j^{(x,k)}\, \mathcal{N}(\boldsymbol{z}_n, \hat{\boldsymbol{\mu}}_j^{(x,k)}, \boldsymbol{\Sigma}^{(x,k)})}{\sum_{i=0}^{2^d-1} \alpha_i^{(x,k)}\, \mathcal{N}(\boldsymbol{z}_n, \hat{\boldsymbol{\mu}}_i^{(x,k)}, \boldsymbol{\Sigma}^{(x,k)})} \tag{19}$$

3.2 Key Recovery

Key recovery is applied at the same implementation that is now loaded with a fixed unknown key k°. Given the 2^d component p.d.f.s $\mathcal{N}(\boldsymbol{z}, \boldsymbol{\mu}_j^{(x,k)}, \boldsymbol{\Sigma}_j^{(x,k)})$ with the associated probabilities $\alpha_j^{(x,k)}$ the adversary computes

$$\mathcal{L}_k := \sum_{i=1}^{N^\circ} \ln f(\boldsymbol{z}_i | k, x_i) = \sum_{i=1}^{N^\circ} \ln \left(\sum_{j=0}^{2^d-1} \alpha_j^{(x_i,k)} \mathcal{N}(\boldsymbol{z}_i, \boldsymbol{\mu}_j^{(x_i,k)}, \boldsymbol{\Sigma}_j^{(x_i,k)}) \right) \tag{20}$$

for each of the 2^d key hypotheses k using known $x_i \in \{0,1\}^d$ and decides in favour of that key hypothesis k^* that leads to the maximum likelihood:

$$k^* := \arg\max_k \mathcal{L}_k \ . \tag{21}$$

Note that the decision strategy for key recovery in a template attack is done in almost the same manner, just by replacing the estimated component p.d.f.s with the 'true' component p.d.f.s, i.e., with the templates in (20).

4 Experimental Case Study

For the experimental evaluation we consider the simplest reasonable case, i.e., a two-dimensional ($d = 1, m = 2$) setting. Samples were obtained by measuring the power consumption of an 8 bit microprocessor AT90S8515 while running a boolean masking scheme. All random numbers x, k, and y are known so that the results of the EM Algorithm (unsupervised learning) can be compared with the use of templates (supervised learning).

We selected two instants (for the selection process see Section 4.1) of the vectorial measurement sample. Instant t_1 leaks side channel information on bit y and at t_2 one finds side channel leakage on bit $y \oplus k \oplus x$. This scenario is identical to the one introduced by Messerges for second-order DPA [15].

We assume that two conditional p.d.f.s $f^{(x \oplus k)}$ on $\boldsymbol{i}(x \oplus k, Y)$ are sufficient for the characterization problem instead of four conditional p.d.f.s $f^{(x,k)}$ on $\boldsymbol{i}(x, k, Y)$. In a template attack, the four resulting conditional-state p.d.f.s for all possible combinations of $(x \oplus k, y)$ are identifiable and illustrated in Fig. 2. Fig. 3 shows the two mixed-state p.d.f.s for $x \oplus k$ as they can be observed by $\mathcal{A}$ due to its incomplete knowledge.

The EM algorithm was applied to the two mixed states for $x \oplus k$. In Table 2 the estimated parameters as result of the profiling stage are summarized for the template algorithm and the four variants of the EM algorithm introduced in Section 3. It can be seen that the results of the estimated parameters of the EM algorithm depend on the specific variant. Table 2 shows that Variant 1 and Variant 2 of the EM algorithm lead to quite similar results, the component p.d.f.s are made of concentric circles in these cases. Also the results of Variant 3 and Variant 4 are quite similar, however, the results form ellipsoids with different parameters compared to the use of templates. Obviously, different parameter settings can produce similar probability distributions.

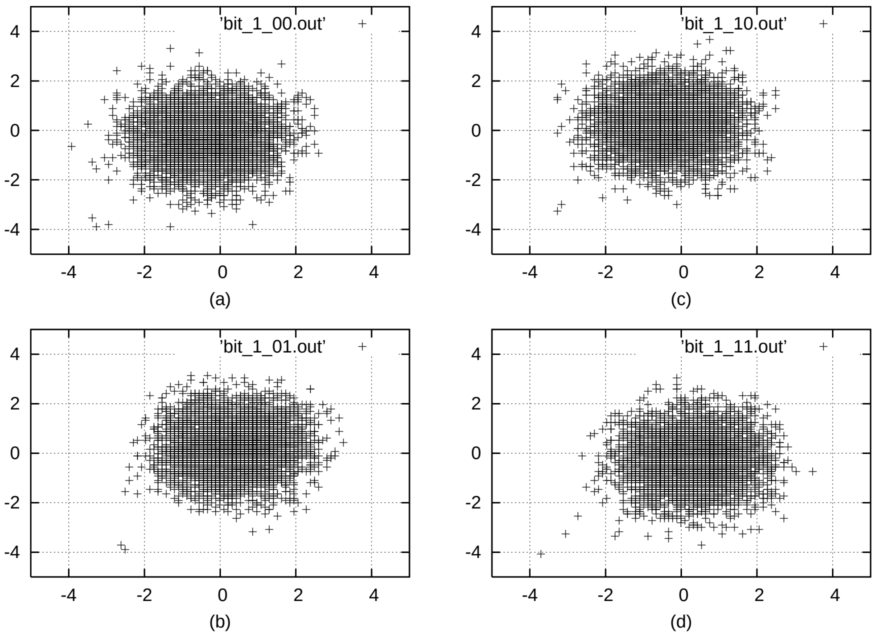

Fig. 2. Empirical component p.d.f.s for all four different combinations of bit y and bit $k \oplus x$. The x-axis gives the normalized measurement values at instant t_1 (y leaks) and the y-axis shows the normalized measurement values at instant t_2 ($y \oplus k \oplus x$ leaks). The distribution is shown for $k \oplus x = y = 0$ in (a), for $k \oplus x = 0$ and $y = 1$ in (b), for $k \oplus x = 1$ and $y = 0$ in (c), and for $k \oplus x = y = 1$ in (d). One can recognize shifts of the probability densities: to the left in (a) and (c), to the right in (b) and (d), to the top in b) and (c) and to the bottom in (a) and (d).

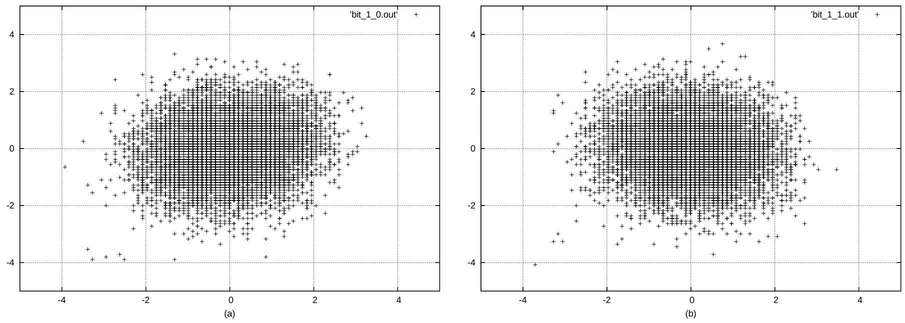

Fig. 3. Empirical mixed p.d.f.s for the two different values of bit $x \oplus k$ using the same data as in Fig. 2. It is $x \oplus k = 0$ in (a) and $x \oplus k = 1$ in (b). In (a) one can recognize a slight rotation of the distribution to the left and in (b) a slight rotation of the distribution to the right which is an indication of a mixture.

Table 2. Estimated parameters for the Gaussian component p.d.f.s by building templates and applying the EM algorithm. The terms μ_1 and μ_2 denote the estimated mean value of the leakage at instant t_1 and t_2, respectively, σ_{11}, σ_{22}, and $\sigma_{12} = \sigma_{21}$ are the estimated entries of the covariance matrix. The samples were normalized before statistics was applied. It was $N = 20,000$ for the profiling stage.

$x \oplus k$	y	μ_1	μ_2	σ_{11}	σ_{22}	$\sigma_{12} = \sigma_{21}$
Templates						
0	0	-0.343609	-0.264896	0.890693	0.929354	0.027368
0	1	0.363384	0.258210	0.849087	0.890358	0.046014
1	0	-0.353654	0.255177	0.885363	0.943963	0.042504
1	1	0.349743	-0.267222	0.877618	0.965020	0.062675
EM Algorithm, Variant 1						
$x \oplus k$	component no. j	μ_1	μ_2	σ_{11}	σ_{22}	$\sigma_{12} = \sigma_{21}$
0	0	-0.228378	-0.222345	1.0	1.0	0.0
0	1	0.252548	0.218852	1.0	1.0	0.0
1	0	0.152021	-0.158530	1.0	1.0	0.0
1	1	-0.173202	0.166899	1.0	1.0	0.0
EM Algorithm, Variant 2						
$x \oplus k$	component no. j	μ_1	μ_2	σ_{11}	σ_{22}	$\sigma_{12} = \sigma_{21}$
0	0	0.234364	0.203658	1.0	1.0	0.0
0	1	-0.249685	-0.243648	1.0	1.0	0.0
1	0	0.163380	-0.170083	1.0	1.0	0.0
1	1	-0.162391	0.156246	1.0	1.0	0.0
EM Algorithm, Variant 3						
$x \oplus k$	component no. j	μ_1	μ_2	σ_{11}	σ_{22}	$\sigma_{12} = \sigma_{21}$
0	0	-0.579599	0.066857	0.680029	0.973863	0.165637
0	1	0.543903	-0.063088	0.680029	0.973863	0.165637
1	0	-0.634439	0.133956	0.653078	0.977233	0.009398
1	1	0.529548	-0.108166	0.653078	0.977233	0.009398
EM Algorithm, Variant 4						
$x \oplus k$	component no. j	μ_1	μ_2	σ_{11}	σ_{22}	$\sigma_{12} = \sigma_{21}$
0	0	0.625019	-0.019519	0.636527	0.926563	0.143675
0	1	-0.520327	0.013980	0.695991	1.022322	0.134543
1	0	0.610178	-0.093003	0.610554	0.937405	-0.025781
1	1	-0.549292	0.088531	0.695000	1.024076	0.006803

Though not explicitly stated in Table 2 also second-order DPA requires a profiling stage to recover the sign of the leakage signal for each instant unless a further assumption is made that the adversary knows this sign, e.g., because the sign of a side channel leakage portion is predictable.[5]

Key Recovery Efficiency The decision strategy of Section 3.2 is applied here for the key hypotheses $k \in \{0, 1\}$. For $d = 1$ (20) simplifies to

$$\mathcal{L}_0 := \sum_{i=1}^{N^\circ} \ln \left(0.5 \cdot \mathcal{N}(\boldsymbol{z}_i, \boldsymbol{\mu}_0^{x_i}, \boldsymbol{\Sigma}_0^{x_i}) + 0.5 \cdot \mathcal{N}(\boldsymbol{z}_i, \boldsymbol{\mu}_1^{x_i}, \boldsymbol{\Sigma}_1^{x_i})\right) \quad \text{and} \tag{22}$$

$$\mathcal{L}_1 := \sum_{i=1}^{N^\circ} \ln \left(0.5 \cdot \mathcal{N}(\boldsymbol{z}_i, \boldsymbol{\mu}_0^{\neg x_i}, \boldsymbol{\Sigma}_0^{\neg x_i}) + 0.5 \cdot \mathcal{N}(\boldsymbol{z}_i, \boldsymbol{\mu}_1^{\neg x_i}, \boldsymbol{\Sigma}_1^{\neg x_i})\right) \tag{23}$$

assuming a uniform distribution of $y_i \in \{0, 1\}$.

Table 3. Success rate at key recovery by using the estimated p.d.f.s for the different methodical approaches. All samples were normalized before applying statistics.

N°	Templates	EM Algorithm				Second-Order DPA
		Variant 1	Variant 2	Variant 3	Variant 4	
10	58.17 %	58.77 %	58.60 %	59.00 %	58.49 %	54.84 %
20	62.82 %	61.63 %	61.73 %	61.06 %	62.26 %	56.74 %
50	68.43 %	67.90 %	67.81 %	68.51 %	68.26 %	61.67 %
100	75.33 %	74.59 %	74.19 %	74.80 %	74.52 %	67.46 %
200	83.85 %	81.22 %	81.51 %	81.92 %	83.13 %	73.93 %
400	91.59 %	89.52 %	89.36 %	91.07 %	91.05 %	81.89 %
600	95.88 %	93.57 %	93.51 %	94.65 %	95.33 %	86.89 %
800	97.86 %	96.75 %	96.02 %	97.16 %	97.39 %	89.77 %
1000	98.88 %	98.09 %	97.73 %	98.44 %	98.68 %	92.77 %
1500	99.74 %	99.52 %	99.52 %	99.71 %	99.68 %	96.60 %
2000	99.94 %	99.91 %	99.86 %	99.95 %	99.95 %	98.44 %

Success rates were empirically determined by applying the 2-variate Gaussian p.d.f.s of Table 2. For second-order DPA the correlation coefficient of $x_i \oplus k$ and $|z_{i,0} - z_{i,1}|$, i.e., the absolute difference of the two scalar components of $\boldsymbol{z}_i$ is computed as suggested by Messerges [15]. Results are presented in Table 3. One can observe that the key recovery efficiency of EM estimates is very close to templates. Further, there are only small decreases in the success rate for the variants based on a reduced set of free parameters. Another result of Table 3 is

[5] The microcontroller used in this case study does not follow the Hamming weight model. Therefore, the sign of the side channel leakage at each instant has to be examined in advance.

that using second-order DPA one needs about twice the number of samples for a comparable success rate.

4.1 Further Directions

Higher-Order Analysis: This experimental case study considers the simplest two-dimensional case for higher-order side channel analysis, but this may also be the only applicable case on an efficient masking scheme, especially in hardware. The use of higher dimensions leads to an increase in the number of unknown parameters for the component p.d.f.s. We expect that an increase of m, i.e., the number of instants considered in the multivariate p.d.f. can significantly improve the success rates for key recovery. Increasing d results in two drawbacks: (i) the number of free parameters increases exponentially (see Table 1) and (ii) the number of measurements that are usable for an estimation decreases exponentially (see (9) and (10)). The benefit of an improved signal-to-noise ratio due to a higher number of predicted bits may be therefore thwarted. A similar consideration holds for templates, i.e., a certain minimum number of measurements is required for a sufficient characterization of the multivariate side channel leakage [10].

How to find relevant instants without knowing the masks: First of all, for $m = 2$ the EM algorithm is applicable at all combinations of instants to check for significantly different component p.d.f.s. If successful at multiple combinations the EM algorithm can be reapplied in order to determine component p.d.f.s with $m > 2$. Further, for fixed parameters (x, k), the empirical variance of the sample may indicate time instants where internal random numbers are used. Another possibility to reduce the dimensions of the vectorial sample is principal component analysis [2]. Second-order DPA [15, 22] may also help to identify suitable points in time.

5 Conclusion

This contribution introduces the use of multivariate Gaussian mixture models for enhancing higher-order side channel analysis on masked cryptographic implementations. The proposed EM algorithm is applicable if an adversary does not have access to masks used during profiling and provides estimates on the component p.d.f.s. For a single-bit second-order setting it has been shown that the attained efficiency in key recovery is very close to templates and clearly better than the efficiency of second-order DPA.

As already outlined in previous contributions masking may not be sufficient to secure cryptographic implementations. Beyond it, this contribution highlights that even adversaries with incomplete knowledge at profiling can acquire appropriate multivariate estimates on component probability densities. Auxiliary countermeasures to decrease the signal-to-noise ratio of the side channel leakage should be definitively foreseen. The effectiveness of these combined countermeasures can be tested by building templates or applying the EM algorithm to mixture densities.

References

1. Dakshi Agrawal, Josyula R. Rao, Pankaj Rohatgi, and Kai Schramm. Templates as Master Keys. In Josyula R. Rao and Berk Sunar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2005*, volume 3659 of *LNCS*, pages 15–29. Springer, 2005.
2. Cédric Archambeau, Eric Peeters, François-Xavier Standaert, and Jean-Jacques Quisquater. Template Attacks in Principal Subspaces. In Louis Goubin and Mitsuru Matsui, editors, *Cryptographic Hardware and Embedded Systems - CHES 2006*, volume 4249 of *LNCS*, pages 1–14. Springer, 2006.
3. Hans Hermann Bock. *Automatische Klassifikation: Theoretische und praktische Methoden zur Gruppierung und Strukturierung von Daten (Cluster-Analyse)*. Vandenhoeck & Ruprecht, 1974.
4. Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankay Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In M. Wiener, editor, *Advances in Cryptology – CRYPTO '99*, volume 1666 of *LNCS*, pages 398–412. Springer, 1999.
5. Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template Attacks. In B. S. Kaliski, Ç Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems*, volume 2523 of *LNCS*, pages 13–28. Springer, 2003.
6. Jean-Sébastian Coron and Louis Goubin. On Boolean and Arithmetic Masking against Differential Power Analysis. In Ç.K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2000*, volume 1965 of *LNCS*, pages 231–237. Springer, 2000.
7. Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification.* John Wiley & Sons, 2001.
8. Mario A.T. Figueiredo and Anil K. Jain. Unsupervised Learning of Finite Mixture Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:381–396, 2002.
9. Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic Analysis: Concrete Results. In Ç Koç, D. Naccache, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2001*, volume 2162 of *LNCS*, pages 251–261. Springer-Verlag, 2001.
10. Benedikt Gierlichs, Kerstin Lemke-Rust, and Christof Paar. Templates vs. Stochastic Methods. In Louis Goubin and Mitsuru Matsui, editors, *Cryptographic Hardware and Embedded Systems - CHES 2006*, volume 4249 of *LNCS*, pages 15–29. Springer, 2006.
11. Marc Joye, Pascal Paillier, and Berry Schoenmakers. On Second-Order Differential Power Analysis. In Josyula R. Rao and Berk Sunar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2005*, volume 3659 of *LNCS*, pages 293–308. Springer, 2005.
12. Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In M. Wiener, editor, *Advances in Cryptology – CRYPTO '99*, volume 1666 of *LNCS*, pages 388–397. Springer, 1999.

13. Geoffrey McLachlan and David Peel. *Finite Mixture Models.* John Wiley & Sons, 2000.
14. Geoffrey J. McLachlan and Thriyambakam Krishnan. *The EM Algorithm and Extensions.* John Wiley & Sons, 1997.
15. Thomas S. Messerges. Using Second-Order Power Analysis to Attack DPA Resistant Software. In Ç.K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2000*, volume 1965 of *LNCS*, pages 238–251. Springer, 2000.
16. Elisabeth Oswald and Stefan Mangard. Template Attacks on Masking – Resistance is Futile. In Masayuki Abe, editor, *Topics in Cryptology – CT-RSA 2007, The Cryptographers' Track at the RSA Conference 2007*, volume 4377 of *LNCS*, pages 243–256. Springer, 2006.
17. Elisabeth Oswald, Stefan Mangard, Christoph Herbst, and Stefan Tillich. Practical Second-Order DPA Attacks for Masked Smart Card Implementations of Block Ciphers. In David Pointcheval, editor, *Topics in Cryptology – CT-RSA 2006, The Cryptographers' Track at the RSA Conference 2006*, volume 3860 of *LNCS*, pages 192–207. Springer, 2006.
18. Pekka Paalanen, Joni-Kristian Kämäräinen, Jarmo Ilonen, and Heikki Kälviäinen. Feature Representation and Discrimination Based on Gaussian Mixture Model Probability Densities – Practices and Algorithms. Technical report, Lappeenranta University of Technology, 2005. Available from: `http://www2.lat.fi/~jkamarai/publications/downloads/laitosrap95.pdf`.
19. Eric Peeters, François-Xavier Standaert, Nicolas Donckers, and Jean-Jacques Quisquater. Improved Higher-Order Side-Channel Attacks with FPGA Experiments. In Josyula R. Rao and Berk Sunar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2005*, volume 3659 of *LNCS*, pages 309–323. Springer, 2005.
20. Werner Schindler, Kerstin Lemke, and Christof Paar. A Stochastic Model for Differential Side Channel Cryptanalysis. In Josyula R. Rao and Berk Sunar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2005*, volume 3659 of *LNCS*, pages 30–46. Springer, 2005.
21. Kai Schramm and Christof Paar. Higher Order Masking of the AES. In David Pointcheval, editor, *Topics in Cryptology – CT-RSA 2006, The Cryptographers' Track at the RSA Conference 2006*, volume 3860 of *LNCS*, pages 208–225. Springer, 2006.
22. Jason Waddle and David Wagner. Towards Efficient Second-Order Power Analysis. In M. Joye and J.-J. Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004*, volume 3156 of *LNCS*, pages 1–15. Springer, 2004.