Clustering Algorithms for Non-Profiled Single-Execution Attacks on Exponentiations

Johann Heyszl[1], Andreas Ibing[2], Stefan Mangard[3⋆], Fabrizio De Santis[2,4], and Georg Sigl[2]

[1] Fraunhofer Institute AISEC, Munich, Germany
johann.heyszl@aisec.fraunhofer.de
[2] Technische Universität München, Munich, Germany
andreas.ibing@in.tum.de, desantis@tum.de, sigl@tum.de
[3] Graz University of Technology, Graz, Austria
stefan.mangard@iaik.tugraz.at
[4] Infineon Technologies AG, Munich, Germany

Abstract. Most implementations of public key cryptography employ exponentiation algorithms. Side-channel attacks on secret exponents are typically bound to the leakage of single executions due to cryptographic protocols or side-channel countermeasures such as blinding. We propose for the first time, to use a well-established class of algorithms, i.e. unsupervised cluster classification algorithms such as the k-means algorithm to attack cryptographic exponentiations and recover secret exponents *without any prior profiling, manual tuning or leakage models*. Not requiring profiling is of significant advantage to attackers, as are well-established algorithms. The proposed non-profiled single-execution attack is able to exploit any available single-execution leakage and provides a straight-forward option to combine simultaneous measurements to increase the available leakage. We present empirical results from attacking an FPGA-based elliptic curve scalar multiplication using the k-means clustering algorithm and successfully exploit location-based leakage from high-resolution electromagnetic field measurements to achieve a low remaining brute-force complexity of the secret exponent. A simulated multi-channel measurement even enables an error-free recovery of the exponent.

Keywords: Exponentiation, side-channel attack, non-profiled, single-execution, unsupervised clustering, simultaneous measurements, EM.

1 Introduction

The main computations in public key cryptosystems are modular exponentiations with secret exponents or elliptic curve scalar multiplications with secret scalars. In both cases, the same exponentiation algorithms are employed to serially process exponents. In DSA or ECDSA, the exponents are different for

⋆ This research has been conducted while working for Infineon Technologies AG, Munich, Germany

every execution, e.g., chosen randomly as ephemeral secrets. RSA employs the same exponent multiple times, but exponent blinding [15] is often used as a countermeasure against side-channel analysis to use different exponents in every execution. Hence, side-channel attackers may only exploit *single executions* to recover a secret exponent. To prevent conventional SPA and timing attacks [15] the operation sequences during the serial processing of the exponents are rendered as homogeneous as possible. Algorithms like the square-and-multiply(-always), double-and-add(-always) or the Montgomery ladder algorithm are examples with constant operation sequences. However, a certain amount of side-channel leakage during single executions, i.e., single-execution leakage, about serially and independently processed bits or digits during the exponentiation cannot be prevented in many cases [5, 21, 14, 24]. This may for instance be location-based leakage [12], address bit leakage [14], or operation-dependent leakage, e.g., when square and multiply operations can be distinguished [5].

We propose to specifically take advantage of well-established cluster classification algorithms [9] in general and the k-means algorithm for example to exploit any of such single-execution leakage and to recover secret exponents *without any prior profiling, manual tuning or heuristic leakage models. It is of significant advantage for an attacker if no profiling is required* because profiling can easily be prevented by using e.g., exponent blinding in the implementation or by executing the accessible exponentiation with public inputs on a different cryptographic engine as the private operation. Segments of the exponentiation which correspond to different exponent bits or digits are classified to find similar segments in an unsupervised way and by using algorithms from the *well-researched field of pattern classification*. This is contrary to previous attempts which use individual algorithms. An unsupervised classification equals the recovery of a secret exponent. Unsupervised clustering is generally useful in side-channel analysis when profiling information is not available and an exhaustive partitioning is computationally infeasible. The success of a correct classification of the exponent bits depends on the amount of available leakage signal in a certain measurement. Clustering algorithms further allow to determine posterior probabilities for classified bits. Hence, if only a part of the secret exponent is classified correctly, an attacker may brute-force bits with low posterior probabilities first. This enables a *straight-forward approach to cope with erroneous bits and allows to significantly reduce the secret's entropy*, thus, brute-force complexity, even if a complete recovery is impossible. The only way for an attacker to gather more leakage is to perform simultaneous measurements in multiple channels because attackers are *not* able to collect measurements from repeated executions since exponents change in every execution. Clustering algorithms allow for a straight-forward approach to *combine such simultaneous side-channel measurements*.

In an empirical study, we demonstrate the proposed attack and exploit the location-based single-execution leakage [12] of an FPGA-based implementation of an elliptic curve scalar multiplication using the k-means clustering algorithm. We employ high-resolution measurements of the electromagnetic field and select measurement positions without prior profiling. The main result from our

practical experiments is that the proposed method successfully reduces the remaining brute-force complexity of the secret scalar to a well-acceptable level in two out of nine cases. Additionally, we show how a combination of simultaneous measurements leads to a complete recovery of the scalar in a simulated setting.

Related work is discussed in Sect. 2. We present the non-profiled clustering attack on exponentiation algorithms in Sect. 3. In Sect. 4, we describe our practical evaluation of the attack and discuss countermeasures. Conclusions are provided in Sect. 5.

2 Related Work

In the following, we present related work in three aspects of this contribution: other attacks on exponentiation algorithms, previous applications of cluster analysis, and combination of measurements.

On Side-Channel Attacks on Exponentiations Schindler and Itoh [21] present an attack against *multiple blinded* executions of exponentiation algorithms assuming that a single execution does not provide enough leakage. Our contribution presents a complement rather than an alternative to Schindler and Itoh's attack since we propose cluster classification algorithms as a *single execution attack* and means to improve the exploitation of any single-execution leakage. If our attack does not lead to correct exponents, Schindler and Itoh's attack can be used on top of it. Walter [24] describes a single-execution side-channel attack on m-ary ($m > 2$) sliding window exponentiation algorithms. He recognizes precomputed multiplier values in segments of the digit-wise exponentiation and uses his own algorithm to scan through the segments in one single pass and partition them into buckets according to their pair-wise similarity. While the main idea of our contribution is similar to the one described by Walter, we propose to employ unsupervised cluster classification algorithms which have been thoroughly researched in other statistical applications instead of using an individual algorithm which has not been investigated by the respective scientific community. In this way, our approach can be extended to a wide range of exponentiation algorithms and exploit any available kind of single-execution leakage of independent exponent bits or digits.

There are many published side-channel attacks on exponentiations based on the correlation coefficient. Messerges et al. [19] first mention cross-correlation of measurement segments to compare them and then perform a classification based on manually tuned thresholds. Witteman et al. [25] present an SPA attack on the square-and-multiply-always algorithm by cross-correlating measurements of consecutive operations sharing the same input values. From our view, using a correlation coefficient as a measure of similarity only incorporates linear relations while disregarding absolute values, thus, contained information. Hence, it is only meaningful in cases when absolute values are of different scales such as when comparing heuristic models of power consumption to actual measurements or when comparing measurements from different setups. Amiel et al. [2]

and Clavier et al. [7] correlate heuristic leakage models from fixed multiplier values with the measurement to recover the exponent. Perin et al. [20] exploit bit-dependent differences in exponentiation algorithms using measurements of electromagnetic fields. However, they require averaging of multiple measurements in their practical results, which is infeasible in realistic circumstances. Algorithmically they simply subtract exponentiation segments from each other and use manually tuned thresholds to recover information. Hence, and contrary to us, all those approaches require a manual tuning of thresholds and, in part, heuristic leakage models as well as ad-hoc algorithms. Our approach using well-established algorithms provides an algorithmic advantage compared to them. Furthermore, we use the Euclidean distance instead of the correlation coefficient as a similarity metric to incorporate the maximum amount of contained information when comparing segments of the same measurement.

On Previous Applications of Cluster Analysis in SCA There are previous contributions which mention cluster analysis in the context of side-channel analysis. Batina et al. [3] propose Differential Cluster Analysis (DCA) as an extension to DPA. Instead of a difference-of-means test as in classic DPA, a cluster criterion is used as statistical distinguisher. However, they do *not use unsupervised cluster classification algorithms*. In [4, 18], this work is extended by considering PCA. Lemke-Rust and Paar [16] propose a *profiled* multi-execution attack against masked implementations of symmetric algorithms using the expectation-maximization clustering algorithm and a training set for the estimation of the clusters. In a profiled setting, they estimate mixture densities of clusters for known key values and unknown mask values using multiple executions. Contrarily, our approach is a *non-profiled* attack.

On the Combination of Measurements A combination of simultaneous measurements can generally improve the success of side-channel attacks. Agrawal et al. [1] combine simultaneous measurements of the power consumption and electromagnetic field for profiled template attacks. Standaert and Archambeau [23] extend this and apply Principal Component Analysis (PCA) and Fisher's Linear Discriminant Analysis (LDA) to reduce the data dimensionality for template attacks. They also present a simple approach to combine simultaneous measurements for classic Differential Power Analysis (DPA) by treating measurements from different channels jointly. Souissi et al. [22] and Elaabid et al. [10] extend Correlation-based differential Power Analysis (CPA) [6] to combine simultaneous measurements by using products [10] or sums [22] of correlation coefficients. Contrary to previous contributions, our approach presents a way of combining measurements for a non-profiled single-execution attack.

3 Non-Profiled Clustering to Attack Exponentiations

When attacking exponentiation algorithms used in public key cryptography, only a single execution is available to an attacker to recover a secret exponent because of cryptographic protocols or protection against side-channel analysis.

In the following subsections we first describe the term single-execution leakage and how measurement traces are segmented into samples for classification. As a main part, we describe how to apply unsupervised clustering algorithms for a non-profiled and non manually-tuned attack. For the case that the attack is not entirely successful due to insufficient single-execution leakage, we describe an approach to cope with classification errors to achieve low remaining brute-force complexities nonetheless. Finally, we describe how to use multiple simultaneous measurements to gather more leakage.

3.1 Single-Execution Side-Channel Leakage of Exponentiations

The common property of all popular exponentiation algorithms, e.g., binary, m-ary, or sliding window exponentiations is that the computation is segmented and performed in a loop. In every segment, the same operations are repeated to process independent bits or digits of the exponent. (If the operations would be different and depending on exponent bits, the implementation would be prone to conventional SPA and timing attacks [15].) We use the case of binary exponentiations which process the exponent bit-wise for our explanations. The square-and-multiply-always algorithm for instance repeatedly either performs a square-and-multiply, or a square-and-dummy-multiply operation, depending on each processed bit. Such repeated operations share similarities for equal bits. Depending on the implementation and included countermeasures, different side-channels can be exploited to detect such similarities. We refer to the side-channel information about different bits which is leaked in single executions of exponentiations as *single-execution side-channel leakage*. Our approach is able to exploit any kind of such single-execution leakage.

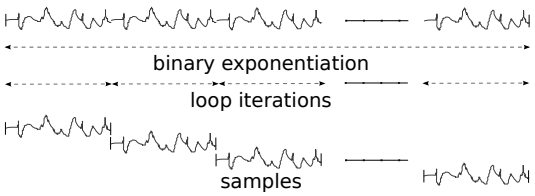


Fig. 1. Segmenting a side-channel measurement of an exponentiation into samples

Figure 1 abstractly depicts a side-channel measurement of a timing-safe binary exponentiation algorithm in the upper part. The observed computation consists of a loop with multiple iterations of constant timing which correspond to single exponent bits. The algorithm could e.g. be a square-and-multiply-always, double-and-add-always, or Montgomery ladder algorithm. Such a side-channel measurement trace vector $\boldsymbol{t} = (t_1, \ldots, t_l)$ of an exponentiation contains l measurement values t_x and covers the entire execution. Binary algorithms process n bits during this time in total. To exploit the single-execution leakage of n independent bits, the trace is cut into n multivariate samples $\boldsymbol{t}_i = (t_{(1+(i-1)\frac{l}{n})}, \ldots, t_{(i\frac{l}{n})})$,

$1 \leq i \leq n$ of equal length $\frac{l}{n}$ where each sample then corresponds to one bit. Figure 1 also depicts an abstract example for how a side-channel measurement is cut into samples. The segmentation borders can e.g. be derived from visual inspection or comparison of shifted trace parts.

3.2 Clustering of Samples Reveals the Secret without Profiling

The multivariate samples $\boldsymbol{t}_i$ contain the leakage of independent, secret exponent bits. Hence, the samples belong to *one of two* classes, i.e., ω_A and ω_B. (When attacking m-ary, or sliding window exponentiation algorithms, m classes are expected.) All side-channel measurements are affected by normally distributed measurement- and switching noise. Therefore, samples within classes $\omega_j,\ j \in \{A, B\}$ are normally distributed around means $\boldsymbol{\mu}_j$. The distance between these means $\boldsymbol{\mu}_j$ is caused by the exploited single-execution leakage. Hence, the distribution of samples $\boldsymbol{t}_i$ in two classes ω_A and ω_B can be described as $p(\boldsymbol{t}_i|\omega_A) \sim \mathcal{N}(\boldsymbol{\mu}_A, \boldsymbol{\Sigma}_A)$ and $p(\boldsymbol{t}_i|\omega_B) \sim \mathcal{N}(\boldsymbol{\mu}_B, \boldsymbol{\Sigma}_B)$.

The correct partition of samples $\boldsymbol{t}_i$ into classes ω_A and ω_B is unknown to the attacker. The number of possible partitions equals 2^n for binary exponentiations with n bit exponents. Testing all possible partitions equals brute-forcing a secret and is computationally infeasible for realistic exponent sizes. Template attacks find these classifications through matching against *templates which are found in a profiling phase.* Other related work use cross-correlation and *manually tuned thresholds* as well as individual and ad-hoc algorithms.

However, we found that *well-researched unsupervised cluster classification algorithms* such as *k-means clustering* [9] can be used to find partitions effectively and *without any manual methods or prior profiling. Hence, we propose to use such algorithms for single-execution side-channel attacks on exponentiation algorithms. Finding a correct partition, or classification, equals the recovery of the secret exponent.* If the correct partition is found, there are only two possibilities to assign the bit values 0 and 1 to two classes ω_A and ω_B, hence, to recover the secret exponent.

The choice of a clustering algorithm depends on the shape of the clusters, hence the distribution of samples within clusters. We decided to start with a simple model of cluster distributions and assume that *all variables (dimensions) within the multivariate samples* $\boldsymbol{t}_i$ *are independent* and exhibit equal variances σ^2 *within the two classes.* Hence, the distribution of both classes ω_A and ω_B can be described as $p(\boldsymbol{t}_i|\omega_j) \sim \mathcal{N}(\boldsymbol{\mu}_j, \sigma^2\boldsymbol{I}),\ j \in \{A, B\}$. The *optimal* classification algorithm *under these assumptions* is the *k-means clustering algorithm* which is depicted in Alg. 1. It uses the *Euclidean distance* as a similarity metric and estimates k cluster means $\boldsymbol{\mu}_j,\ j \in \{1, k\}$. In the case of binary algorithms, k equals 2 and two classes ω_A and ω_B are expected.

Initially, k random samples $\boldsymbol{t}_i$ are randomly selected as means and the remaining samples are classified according to shortest Euclidean distance. Then, in iterations, new means are computed within each class, and the classification according to shortest Euclidean distance is repeated until the classification is stable in subsequent iterations. The k-means algorithm is usually executed multiple

Algorithm 1 Unsupervised k-means clustering algorithm [9]

input: samples $\boldsymbol{t}_i,\ 1 \leq i \leq n$, number of clusters k
output: cluster means $\boldsymbol{\mu}_j,\ 1 \leq j \leq k$ and classification $c_i \in [1..k],\ 1 \leq i \leq n$

1: initialize by picking k random samples $\boldsymbol{t}_i$ as start values for $\boldsymbol{\mu}_j,\ 1 \leq j \leq k$
2: **repeat**
3: assign samples $\boldsymbol{t}_i$ to classes $c_i \in [1..k]$ from shortest distance to $\boldsymbol{\mu}_j,\ 1 \leq j \leq k$
4: compute new $\boldsymbol{\mu}'_j$ as mean of all samples $\boldsymbol{t}_i$ with $c_i = j$
5: **until** $\boldsymbol{\mu}'_j = \boldsymbol{\mu}_j\ \forall\ j$, assign $\boldsymbol{\mu}_j$ new values $\boldsymbol{\mu}'_j$ and repeat

times and the best result in terms of a *sum-of-squared-error* criterion is finally selected in order to prevent the algorithm from getting stuck in local maxima.

Clustering algorithms essentially estimate cluster parameters to perform classifications. This estimation of clusters could be improved by using more samples from multiple executions in a first step, even though the secret would then be different in every execution. The actual attack would then be performed in a second step and certainly only target a single execution.

Decorrelation and Reduction of Dimensions If the samples derived from measurements do not comply with the model which is required for the application of k-means (described above), the results will be worse than theoretically possible. The k-means algorithm assumes statistical independence of dimensions (variables) in the samples, thus, uncorrelated noise influences. However, subsequent measurement values of the power consumption possibly contain the same switching noise influence. One way to handle this is to employ the *expectation-maximization clustering algorithm* which provides more degrees of freedom in such cases (because it also models the covariance between variables). However, it requires a significant overhead in computation. Alternatively, if necessary, this can be coped with by employing Principal Component Analysis (PCA) [9]. PCA performs a projection into a lower dimensional, *orthogonal* space by maximizing the variance in the samples. Hence, the remaining dimensions are uncorrelated. (As a drawback, this is performed without regard of cluster distributions or cluster discriminants which could possibly lead to inferior results.) PCA can certainly also be used to reduce the amount of dimensions in the samples $\boldsymbol{t}_i$ for computational reasons.

3.3 Brute-Force Complexity to Handle Classification Errors

If a recovered exponent cannot be verified as being entirely correct, at least one sample (bit) is misclassified by the algorithm. We propose a way to cope with such situations. Clustering algorithms allow to derive posterior class-membership probabilities [9] for all samples $\boldsymbol{t}_i$ along with their classification. For instance when employing the *k-means* clustering algorithm, samples which are classified into class ω_A and are close to the separating plane between ω_A and ω_B have a low posterior probability of belonging to class ω_A. An attacker may approach misclassifications by brute-forcing samples with low posterior probabilities first.

A straight-forward approach is to iteratively consider an increasing range of samples i with the lowest posterior probabilities and brute-force their classification until all erroneous samples are included in this range, thus, a correct classification is achieved. Given that m equals the final number, which the attacker certainly does not know from the beginning, he would proceed iteratively and increase the number of included bits i starting from 1 until m is reached. The required brute-force complexity to handle classification errors can, thus, be given as an upper bound by using the sum formula of a geometric series. Including the mandatory step of brute-forcing the classes-to-bit-values assignment (A and B to 0 and 1), this required brute-force complexity equals $2 \times \sum_{i=1}^{m} 2^i = 2^{m+1+1} - 2$ for $m > 0$ and can be defined as 2 for $m = 0$ (classification entirely correct; one out of two trial for correct class-to-bit-value assignment). *This means that even if the exponent is not recovered entirely, the entropy can be significantly reduced which is a significant advantage over previous methods which do not provide such a mechanism to cope with errors during an attack.*

3.4 Combining Side-Channel Measurements

The success of single-execution attacks on exponentiation algorithms generally suffers from insufficient leakage [21, 5]. Countermeasures introduce superficial noise to decrease the signal-to-noise ratio of the leakage or aim at reducing the leakage signal directly. Averaging repeated measurements with equal input values is a simple example for an approach to decrease such noise. But this is not feasible if the secret changes in every execution which is the case for most cryptographic exponentiations. Hence, simultaneous measurements are the only way for an attacker to increase the gathered side-channel leakage. Clustering algorithms allow to combine simultaneous side-channel measurements in a straight-forward way. This is achieved by generating multivariate samples using values from all measurements. As an example, samples $\mathbf{t}_i^1$ from measurement 1 are combined with samples $\mathbf{t}_i^2$ from measurement 2 leading to combined samples $\mathbf{t}_i^{\text{combined}} = (\mathbf{t}_i^1, \mathbf{t}_i^2)$. This improves the classification, if the new measurements contain additional leakage information. *Hence, we propose to improve clustering-based single-execution attacks through combining the contained information from multiple, simultaneous side-channel measurements.*

4 Practical Evaluation

In this section, we practically demonstrate our proposed attack against an FPGA-based ECC implementation. As a single-execution side-channel leakage, we exploit location-based leakage [12] revealed by high-resolution measurements of the electromagnetic field [13]. Following the principle that our attack is non-profiled, we do not use any prior knowledge to find measurement positions with high leakage.

4.1 Design-Under-Test and Measurement Setup

Our target is an implementation of an elliptic curve scalar multiplication configured into a *Xilinx Spartan-3 (XC3S200)* FPGA. It gets affine x- and y-coordinates of a base point P and a scalar d as input and returns affine x- and y-coordinates of the resulting point $d \cdot P$. The result is computed using the Montgomery ladder algorithm presented by López and Dahab [17] which is a binary exponentiation algorithm processing a 163 bit scalar bitwise in a uniform operation sequence. This prevents timing-based single-execution leakage. The projective coordinates of the input point are randomized [8] as a countermeasure against differential power analysis. However, the design exhibits location-based information leakage [12] because it uses working registers depending on the value of the processed scalar bit and no protection mechanism against this is included. For these reasons, the design is eligible for our attack and we exploit this single-execution leakage using high-resolution electromagnetic field measurements.

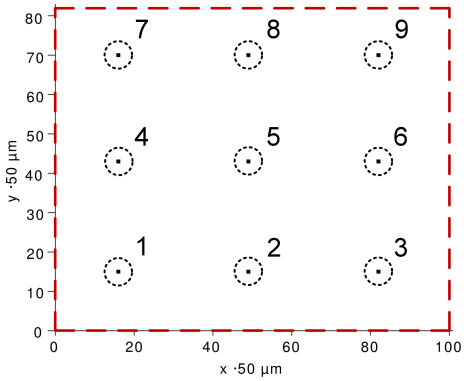


Fig. 2. FPGA die area as dashed rectangle with array of marked measurement positions

Backside access to integrated circuit dies generally requires less practical effort in case of plastic or smartcard packages. The plastic package on the backside of the FPGA was removed mechanically to enable measurements close to the die surface. We use an inductive near-field probe with a 100 µm resolution, built-in 30 dB amplifier, and external 30 dB amplifier (both with a noise figure of 4.5 dB). The detected location-based leakage depends on the measurement position on the surface of the die [12]. Since our attack is non-profiled, we are unable to find a position with high leakage through prior profiling. Instead, we choose measurement positions by geometrical means. Fig. 2 shows those 9 positions marked with circles and annotated with numbers. They are organized in an 3 by 3 array with 1.5 mm distance in x- and y-direction. These geometries seem feasible for an actual array of electromagnetic probes [22]. The dashed rectangle depicts the surface of the FPGA die which measures $\approx 5000 * 4000$ µm. We performed the attack on those measurements.

Furthermore, we demonstrate a combination of simultaneous measurements to increase the leakage in a simulated setting. Since we did not have an array probe or multiple probes at hand, we simulated this by moving one probe to the marked positions and repeated the measurement with exactly equal processed values. Hence, we prevent the device from changing the exponent and random numbers during repeated executions. While this simplification is not exactly the same as simultaneously using multiple probes, we are convinced that the results are still conclusive.

All measurements were recorded at a sampling rate of 5 GS/s and compressed by using the sum of squared values in every clock cycle (V^2s) to reduce the amount of data and computation complexity during clustering. Through synchronization of the oscilloscope and the function generator, we prevent frequency jitter and drift in the measurements.

4.2 Clustering Individual Measurements

We performed our clustering-based attack on individual measurements. Hence, we segmented every measurement into multivariate samples $\boldsymbol{t}_i$. Each sample contains 551 compressed values of 551 clock cycles during which one exponent bit is processed. Figure 3 depicts a cut-out of four consecutive samples (14 to 17) from the measurement at position 3 for illustrative purposes. The borders of the samples are depicted as vertical dashed lines after every 551 cycles. The exponent bit values which are processed in the segments are annotated, however, the corresponding single-execution leakage is not clearly visible.

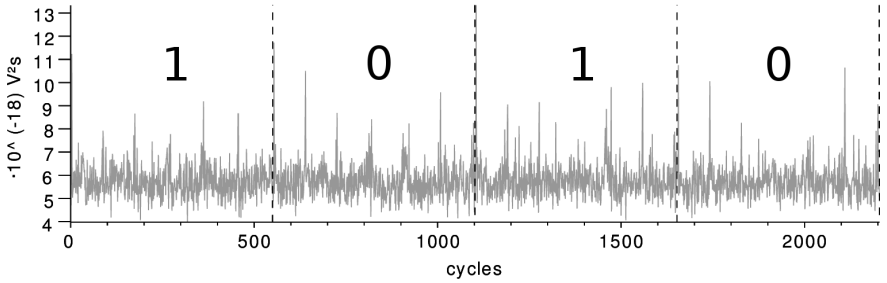


Fig. 3. Four samples (14 to 17) from the compressed measurement at position 3

We attacked the individual measurements by employing the unsupervised *k-means* clustering algorithm Alg. 1 to classify the samples in two clusters as described in Sect. 3.2. The runtime on a regular PC was neglegible and in the range of seconds. We assess the quality of the result by computing the remaining brute-force complexity required to recover the entirely correct scalar after clustering as described in Sect. 3.3. Figure 4 depicts the resulting brute-force complexity for every individual measurement position according to Fig. 2 and Tab. 1 displays them in tabular form (columns marked with '1' to '9').

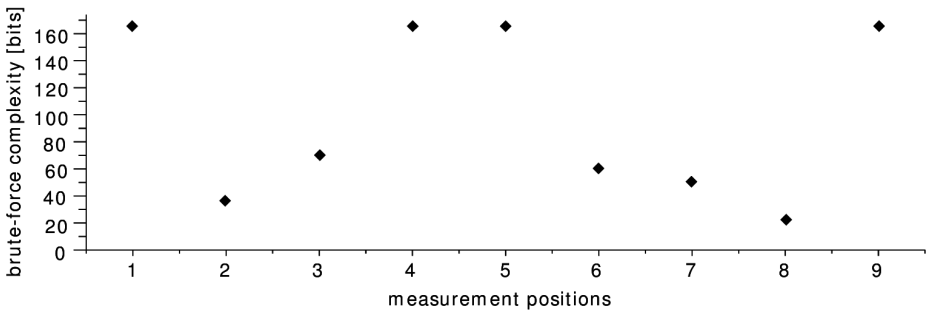


Fig. 4. Remaining brute-force complexity after clustering *individual measurements*

measurement positions	1	2	3	4	5	6	7	8	9	**all**
brute-force complexity [bits]	165	37	70	165	165	60	51	22	165	**0**

Table 1. Brute-force complexity after clustering single and combined measurements

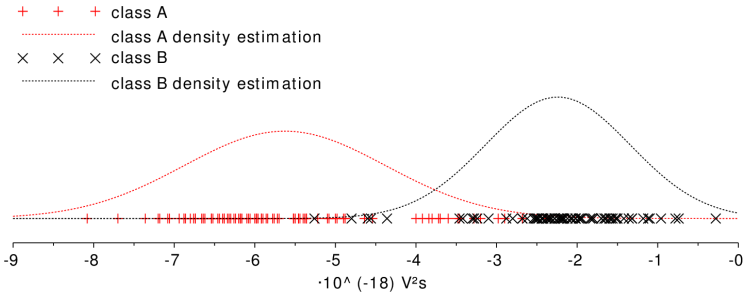
As a main result of our practical study, we are able to report that in two out of nine cases, for the measurements at position 8 *and* 2*, the remaining brute-force complexity (*22 *and* 37 *bits) is clearly within a practical reach.* An attacker could, thus, repeat a measurement at different positions, perform the attack including the incremental brute-force and eventually be successful with a high probability. *This clearly demonstrates the capabilities of unsupervised cluster classification as a non-profiled single-execution attack on exponentiation algorithms to exploit single-execution leakage.*

Positions 1, 4, 5 and 9 lead to a brute-force complexity of 165 bits which is the maximum value ($163+1+1$ bits) indicating that the clustering algorithm led to largely incorrect results. Possible reasons for this are insufficient signal-to-noise ratios of the exploited leakage, outlier samples, or that the k-means algorithm is insufficient since the assumed model of cluster distributions does not fit. (An influence of one bit of some internal ALU operation for the separation of two clusters is impossible since each sample contains many ALU operations with different data.)

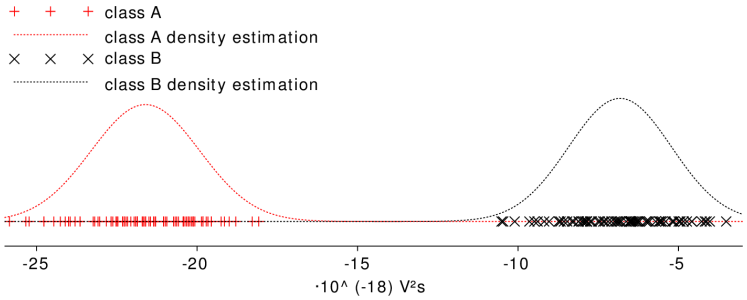
4.3 Clustering Combined Measurements

The results from clustering individual measurements lead to remaining brute-force complexities greater than zero and in seven out of nine cases beyond limits for practical brute-force. As a second step, we demonstrate how simultaneous side-channel measurements can be combined to reduce the remaining brute-force complexity, hence, improve the attack. We combined the measurements as described in Sect. 3.4 and repeated the *k-means* clustering. *As an important result we report, that the classification then leads to a remaining brute-force complexity of zero, denoted as '**all**' in Tab. 1. This clearly demonstrates the*

advantage of combining measurements for attacking exponentiation algorithms using unsupervised clustering algorithms.



(a) Result of clustering measurement position 3



(b) Result of clustering 9 combined measurements

Fig. 5. Visual representation of clustering results to show gain of combination

Figure 5(a) and Fig. 5(b) *demonstrate the advantage of combining measurements* in an more illustrative way. Figure 5(a) visually represents the **result after clustering** the *single measurement at position number* 1. The clustering algorithm output two cluster means $\boldsymbol{\mu}_A$ and $\boldsymbol{\mu}_B$ and samples which are classified according to a separation plane in the middle between those means (equals classification according to shortest Euclidean distance). For the illustration of this clustering result, we projected all multivariate samples $\boldsymbol{t}_i$ (multi-dimensional) onto a line (one-dimensional) which extends through both cluster means. As such, the resulting single values per sample are linear combinations of all vector dimensions according to the weighting factors determined by the clustering result. After this projection, the two cluster distributions become clearly observable. For the illustration, we use the correct scalar to mark the samples according to their proper class membership. Additionally, we estimate the two assumed Gaussian distributions and depict two curves, denoted as *class A/B density estimation*. It is obvious that the two distributions overlap in Fig. 5(a) which means that there have been misclassifications. Many samples are across the wrong side of the half distance between the two distributions which corre-

sponds to the separation plane. This leads to the high brute-force complexity reported in Tab. 1.

Figure 5(b) depicts a similar linear projection of the **result after clustering** of 9 combined measurements. *It can be observed clearly, that the separation of the two classes is significantly improved by the combination of measurements which also complies with the brute-force complexity of* 0 *reported in Tab. 1.*

4.4 Countermeasures

Generally, all methods which reduce the signal-to-noise ratio of arbitrary single-execution leakage, either by reducing the single-execution leakage signal, or increasing the noise level, make our attack more difficult since the attacker is limited in the number of measurements he can record simultaneously. There is no dedicated other countermeasure except for such general ones.

Location-based single-execution leakage as it is exploited in this practical attack can specifically be prevented by randomizing variable locations [12], by balancing registers and their signal paths, or by locating them in an interleaved way that they cannot be distinguished [11].

5 Conclusion

We demonstrate that unsupervised clustering algorithms are powerful for attacking a wide range of exponentiation algorithms in single-execution settings and *without any prior profiling or manually tuned thresholds, which is of significant advantage for attackers.* Instead of individual ad-hoc algorithms we propose to use well-research cluster classification algorithms. Any available single-execution side-channel leakage can be exploited.

In a practical evaluation we successfully recover the secret scalar from an FPGA-based ECC implementation. Individual measurements of the electromagnetic field partly lead to sufficiently low remaining brute-force complexities. By performing the attack including the incremental brute-force at several positions, the attacker might get successful with a realistic effort. Additionally, we provide evidence for the advantage of combining simultaneous measurements. This means that instead of finding specifically good measurement positions, an attacker might simply combine leakage information from multiple simultaneous measurements.

Acknowledgements. This work was partly funded by the German Federal Ministry of Education and Research in the project SIBASE through grant number 01IS13020.

References

1. Agrawal, D., Rao, J., Rohatgi, P.: Multi-channel attacks. In: Walter, C., Koç, Ç., Paar, C. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2003.

Lecture Notes in Computer Science, vol. 2779, pp. 2–16. Springer Berlin / Heidelberg (2003)
2. Amiel, F., Feix, B., Villegas, K.: Power analysis for secret recovering and reverse engineering of public key algorithms. In: Adams, C., Miri, A., Wiener, M. (eds.) Selected Areas in Cryptography, Lecture Notes in Computer Science, vol. 4876, pp. 110–125. Springer Berlin Heidelberg (2007)
3. Batina, L., Gierlichs, B., Lemke-Rust, K.: Differential cluster analysis. In: Clavier, C., Gaj, K. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2009. Lecture Notes in Computer Science, vol. 5747, pp. 112–127. Springer Berlin / Heidelberg (2009)
4. Batina, L., Hogenboom, J., Woudenberg, J.: Getting more from PCA: First results of using principal component analysis for extensive power analysis. In: Dunkelman, O. (ed.) Topics in Cryptology - CT-RSA 2012, Lecture Notes in Computer Science, vol. 7178, pp. 383–397. Springer Berlin Heidelberg (2012)
5. Bauer, S.: Attacking exponent blinding in RSA without CRT. In: Schindler, W., Huss, S. (eds.) Constructive Side-Channel Analysis and Secure Design, Lecture Notes in Computer Science, vol. 7275, pp. 82–88. Springer Berlin / Heidelberg (2012)
6. Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: Joye, M., Quisquater, J.J. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2004. Lecture Notes in Computer Science, vol. 3156, pp. 135–152. Springer Berlin / Heidelberg (2004)
7. Clavier, C., Feix, B., Gagnerot, G., Roussellet, M., Verneuil, V.: Horizontal correlation analysis on exponentiation. In: Soriano, M., Qing, S., López, J. (eds.) Information and Communications Security, Lecture Notes in Computer Science, vol. 6476, pp. 46–61. Springer Berlin Heidelberg (2010)
8. Coron, J.S.: Resistance against differential power analysis for elliptic curve cryptosystems. In: CHES '99: Proceedings of the First International Workshop on Cryptographic Hardware and Embedded Systems. pp. 292–302. Springer-Verlag, London, UK (1999)
9. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification (2nd Edition). Wiley-Interscience, 2 edn. (Nov 2001)
10. Elaabid, M., Meynard, O., Guilley, S., Danger, J.L.: Combined side-channel attacks. In: Chung, Y., Yung, M. (eds.) Information Security Applications. Lecture Notes in Computer Science, vol. 6513, pp. 175–190. Springer Berlin / Heidelberg (2011)
11. He, W., de la Torre, E., Riesgo, T.: An interleaved EPE-immune PA-DPL structure for resisting concentrated em side channel attacks on fpga implementation. In: Schindler, W., Huss, S. (eds.) Constructive Side-Channel Analysis and Secure Design. Lecture Notes in Computer Science, vol. 7275, pp. 39–53. Springer Berlin / Heidelberg (2012)
12. Heyszl, J., Mangard, S., Heinz, B., Stumpf, F., Sigl, G.: Localized electromagnetic analysis of cryptographic implementations. In: Dunkelman, O. (ed.) Topics in Cryptology - CT-RSA 2012. Lecture Notes in Computer Science, vol. 7178, pp. 231–244. Springer Berlin / Heidelberg (2012)
13. Heyszl, J., Merli, D., Heinz, B., De Santis, F., Sigl, G.: Strengths and limitations of high-resolution electromagnetic field measurements for side-channel analysis. In: Mangard, S. (ed.) Smart Card Research and Advanced Applications. Lecture Notes in Computer Science, Springer Berlin Heidelberg (2012)

14. Itoh, K., Izu, T., Takenaka, M.: Address-bit differential power analysis of cryptographic schemes OK-ECDH and OK-ECDSA. In: Cryptographic Hardware and Embedded Systems - CHES 2002. Lecture Notes in Computer Science, vol. 2523, pp. 399–412. Springer Berlin / Heidelberg (2003)
15. Kocher, P.C.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology. pp. 104–113. CRYPTO '96, Springer-Verlag, London, UK (1996)
16. Lemke-Rust, K., Paar, C.: Gaussian mixture models for higher-order side channel analysis. In: Paillier, P., Verbauwhede, I. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2007, Lecture Notes in Computer Science, vol. 4727, pp. 14–27. Springer Berlin / Heidelberg (2007)
17. López, J., Dahab, R.: Fast multiplication on elliptic curves over GF(2m) without precomputation. In: CHES '99: Proceedings of the First International Workshop on Cryptographic Hardware and Embedded Systems. pp. 316–327. Springer-Verlag, London, UK (1999)
18. Mavroeidis, D., Batina, L., Laarhoven, T., Marchiori, E.: PCA, eigenvector localization and clustering for side-channel attacks on cryptographic hardware devices. In: Flach, P., Bie, T., Cristianini, N. (eds.) Machine Learning and Knowledge Discovery in Databases, Lecture Notes in Computer Science, vol. 7523, pp. 253–268. Springer Berlin Heidelberg (2012)
19. Messerges, T., Dabbish, E., Sloan, R.: Power analysis attacks of modular exponentiation in smartcards. In: Cryptographic Hardware and Embedded Systems. Lecture Notes in Computer Science, vol. 1717, pp. 724–724. Springer Berlin / Heidelberg (1999)
20. Perin, G., Torres, L., Benoit, P., Maurine, P.: Amplitude demodulation-based EM analysis of different RSA implementations. In: Design, Automation Test in Europe Conference Exhibition (DATE), 2012. pp. 1167 –1172 (Mar 2012)
21. Schindler, W., Itoh, K.: Exponent blinding does not always lift (partial) SPA resistance to higher-level security. In: López, J., Tsudik, G. (eds.) Applied Cryptography and Network Security, Lecture Notes in Computer Science, vol. 6715, pp. 73–90. Springer Berlin / Heidelberg (2011)
22. Souissi, Y., Bhasin, S., Guilley, S., Nassar, M., Danger, J.L.: Towards different flavors of combined side channel attacks. In: Dunkelman, O. (ed.) Topics in Cryptology - CT-RSA 2012. Lecture Notes in Computer Science, vol. 7178, pp. 245–259. Springer Berlin / Heidelberg (2012)
23. Standaert, F.X., Archambeau, C.: Using subspace-based template attacks to compare and combine power and electromagnetic information leakages. In: Oswald, E., Rohatgi, P. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2008. Lecture Notes in Computer Science, vol. 5154, pp. 411–425. Springer Berlin / Heidelberg (2008)
24. Walter, C.: Sliding windows succumbs to big mac attack. In: Koç, Ç., Naccache, D., Paar, C. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2001. Lecture Notes in Computer Science, vol. 2162, pp. 286–299. Springer Berlin / Heidelberg (2001)
25. Witteman, M., van Woudenberg, J., Menarini, F.: Defeating RSA multiply-always and message blinding countermeasures. In: Kiayias, A. (ed.) Topics in Cryptology - CT-RSA 2011. Lecture Notes in Computer Science, vol. 6558, pp. 77–88. Springer Berlin / Heidelberg (2011)