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

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

[1] Fraunhofer Research Institution 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] Infineon Technologies AG, Munich, Germany,
stefan.mangard@infineon.com

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 because of cryptographic protocols or side-channel countermeasures such as blinding. We propose a new class of algorithms, i.e. unsupervised cluster classification algorithms, to attack cryptographic exponentiations and recover secret exponents *without any prior profiling or heuristic leakage models*. Not requiring profiling is a significant advantage to attackers. In fact, 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 improve the signal-to-noise ratio of available leakage. We present empirical results from attacking an elliptic curve scalar multiplication and exploit location-based leakage from high-resolution electromagnetic field measurements without prior profiling. Individual measurements lead to a sufficiently low remaining brute-force complexity of the secret exponent. An errorless recovery of the exponent is achieved after a combination of few measurements.

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 using a secret exponent or elliptic curve scalar multiplications using a secret scalar. In both cases, essentially the same exponentiation algorithms are employed to serially process exponents. In DSA or ECDSA, the exponent is different for every execution, e.g., chosen randomly as ephemeral secret. RSA uses the same exponent multiple times, but exponent blinding [14] is often used as a countermeasure against side-channel analysis to make the exponent different for every execution. Hence, in all cases, side-channel attackers may only exploit single executions to recover a secret exponent. To prevent SPA and timing attacks [14] the operation sequences during the serial processing of the exponent 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 [4, 19, 13, 21]. This may for instance be location-based leakage [11], address bit leakage [13], or operation-dependent leakage, e.g., when square and multiply operations can be distinguished [4].

We propose to specifically take advantage of cluster classification algorithms [8] to exploit single-execution leakage and to recover secret exponents *without any prior profiling 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 not executing the exponentiation with public inputs on the same cryptographic engine as the private operation. Segments of the exponentiation which correspond to different exponent bits or digits are classified in an unsupervised way to find similar segments. This 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 classification depends on the available Signal-to-Noise Ratio (SNR) of the exploited leakage signal. As an important property, clustering algorithms allow for a straight-forward way to combine simultaneous side-channel measurements of single executions to increase the SNR of the exploited leakage. Such multiple measurements have to be simultaneous because the secret exponent changes in every execution. As another advantage, clustering algorithms allow to determine posterior probabilities for classified bits. Hence, if only a part of the secret is classified correctly, an attacker may brute-force bits with low posterior probabilities. This allows to significantly reduce the secret's entropy even if a complete recovery is impossible.

In an empirical study, we demonstrate the proposed attack and exploit the location-based single-execution leakage [11] of an FPGA-based implementation of an elliptic curve scalar multiplication. We employ high-resolution measurements of the electromagnetic field as a side-channel and select measurement positions without prior profiling. Nonetheless we demonstrate that the attack reduces the entropy of the secret scalar to a sufficiently low level. Furthermore, we show that a combination of few measurements reduces the remaining entropy of the secret to zero, hence leading to a complete recovery of the scalar.

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 successful 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.

Other Side-Channel Attacks on Exponentiations Schindler and Itoh [19] presented an attack against blinded exponentiation algorithms which uses multiple executions. A general single-execution leakage of exponent bits and exploitation thereof is assumed. Our contribution presents a complement rather than an alternative to Schindler and Itoh's attack since we propose cluster classification algorithms as a measure to improve the exploitation of such single-execution leakages. If the exponent can be recovered from a single-execution with our attack the method of Schindler and Itoh is not needed. Walter [21] describes a single-execution side-channel attack on m-ary ($m > 2$) sliding window exponentiation algorithms. He recognizes pre-computed multiplier values in segments of the digit-wise exponentiation and uses a proprietary 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 this 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 a heuristically tuned algorithm. Our approach can be extended to a wide range of exponentiation algorithms and exploit arbitrary single-execution leakages of independent exponent bits or digits.

There are published side-channel attacks on exponentiations based on the correlation coefficient. Messerges et al. [17] first mention cross-correlation of measurement segments. Amiel et al. [2] and Clavier et al. [6] correlate heuristic leakage models from fixed multiplier values with the measurement to recover the exponent. Witteman et al. [22] present an SPA attack on the square-and-multiply-always algorithm by cross-correlating measurements of consecutive operations sharing the same input values. Perin et al. [18] exploit bit-dependent differences in exponentiation algorithms using measurements of electromagnetic fields. However, they require averaging of multiple measurements in their practical results and simply subtract exponentiation segments from each other to recover information. No method to automatically derive the key without heuristic intervention is mentioned. Contrarily, we employ well-researched algorithms instead of heuristically tuned ones and are able to exploit *arbitrary* single-execution leakages. Instead of the correlation coefficient as a measure of similarity which only compares linear relations while disregarding the comparison of absolute values, thus, obviously disregarding contained information, we are able to use the Euclidean distance since we are independent of heuristic leakage models.

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. Lemke-Rust and Paar [15] propose a *profiled* multi-execution attack against masked implementations 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.

Combination of Measurements The 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. 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. [20] and Elaabid et al. [9] extend Correlation-based differential Power Analysis (CPA) [5] to combine simultaneous measurements by combining the correlation coefficients using a product [9] or sum [20]. 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.

3.1 Single-Execution Side-Channel Leakage of Exponentiations

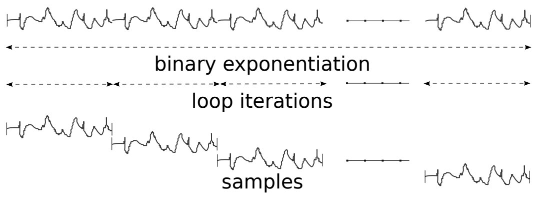


Fig. 1. Segmenting a side-channel measurement of an exponentiation into samples

The common property of all 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. 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 can be collected from one execution of an exponentiation as *single-execution side-channel leakage.*

Figure 1 abstractly depicts a side-channel measurement of a timing-safe binary exponentiation algorithm. 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.

3.2 Segmenting Side-Channel Measurements of Exponentiations

A side-channel measurement trace vector $\boldsymbol{t} = (t_1, \dots, t_l)$ of an exponentiation contains l measurement values t_x and covers the entire execution. Binary algorithms process n bits during this time. To exploit the single-execution leakage of n independent bits, the trace is cut into n multivariate samples $\boldsymbol{t}_i = (t_{(1+(i-1)\frac{l}{n})}, \dots, t_{(i\frac{l}{n})}),\ \ 1 \le i \le n$ of equal length $\frac{l}{n}$ where each sample then corresponds to one bit. Figure 1 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 cross-correlation of trace parts.

3.3 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. However, we found that *unsupervised cluster classification algorithms* such as *k-means clustering* [8] can be used to find partitions effectively. *We propose to use such algorithms for single-execution side-channel attacks on exponentiation algorithms without prior profiling. 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.

Algorithm 1 Unsupervised k-means clustering algorithm [8]

```
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 $c_i \in [1..k],\ 1 \leq i \leq n$

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

The choice of a clustering algorithm depends on the assumed shape of the clusters, hence the distribution of samples within clusters. We decided to employ a simple model of cluster distributions and assume that all variables 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. Algorithm 1 picks two random samples $\boldsymbol{t}_i$ as means and iteratively improves the classification by minimizing the *sum-of-squared-error* criterion until the result is stable. The k-means algorithm is usually executed multiple times and the best result in terms of the cluster criterion is selected finally.

If simplified models and the corresponding algorithms do not lead to satisfying results, models with more parameters must be used. The *expectation-maximization clustering algorithm* correspondingly provides more degrees of freedom in the model.

3.4 Brute-Force Complexity to Handle Classification Errors

If an attacker is unable to recover the entire exponent correctly, at least one sample is misclassified by the algorithm. Clustering algorithms allow to derive posterior class-membership probabilities [8] 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 can approach misclassification by brute-forcing the classification of samples with low posterior probabilities. A straight-forward approach is to iteratively consider an increasing number of samples with lowest posterior probabilities and brute-force their classification until all erroneous samples are included, thus, a correct classification is achieved. Given that m equals this number of samples in the final range of samples, an attacker proceeded iteratively and increased the number of included bits i starting from 1 until m was reached. The required brute-force complexity to handle classification errors can, thus, be given as an upper bound by using the sum formula of geometric series. Including the brute-forcing of 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 equals 0 for $m = 0$. *This means that even if the exponent is not recovered entirely, the entropy can be reduced significantly which is a significant advantage over previous methods which do not provide a mechanism to cope with errors in the recovery of the secret.*

3.5 Combining Side-Channel Measurements

The success of single-execution attacks on exponentiation algorithms generally suffers from low Signal-to-Noise Ratios (SNR)s of the exploited leakage [19, 4]. Countermeasures aim at reducing the SNR by introducing superficial noise or reducing the leakage signal. In the context of clustering algorithms in side-channel analysis, we assess the SNR as the proportion of the exploited signal leakage to the sum of switching noise and measurement noise. Hence, we define the SNR as the logarithm of the quotient of the squared difference of estimated cluster means $\boldsymbol{\mu}_A$ and $\boldsymbol{\mu}_B$ and the sum of the variances σ_A^2 and σ_B^2 of the two clusters, as in (1).

$$\mathrm{SNR}(\boldsymbol{\mu}_A, \boldsymbol{\mu}_B, \sigma_A^2, \sigma_B^2) = 10 * \log\left(\frac{(\boldsymbol{\mu}_A - \boldsymbol{\mu}_B)^2}{(\sigma_A^2 + \sigma_B^2)}\right) \mathrm{dB} \tag{1}$$

Averaging repeated measurements with equal input values is a simple example for an approach to increase the SNR. But this is not feasible if the secret changes in every execution which is the case for cryptographic exponentiations. However, 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 increase the SNR of clustering-based single-execution attacks through combining the contained information from multiple, simultaneous side-channel measurements.*

The estimation of cluster distributions, i.e. distribution parameters, could be improved by using samples from multiple executions with different secret exponents. Such estimated parameters may improve clustering-based attacks even though attacks only exploit measurements from a single execution.

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 [11] revealed by high-resolution measurements of the electromagnetic field [12]. Following the principle that our attack is non-profiled, we do not use any prior knowledge to find measurement positions with high SNR of this leakage. Instead, we make use of the fact that our method allows to combine simultaneous measurements and increase SNR by combining the leakage from multiple locations.

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 [16] which is a binary exponentiation algorithm and is, therefore, eligible for our attack. The algorithm processes 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 [7] as a countermeasure against differential power analysis. However, the design exhibits location-based information leakage [11] because it uses working registers depending on the value of the processed scalar bit and no protection mechanism against this is included. We exploit this leakage using high-resolution electromagnetic field measurements.

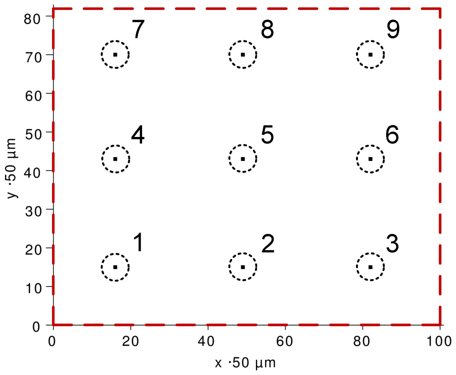


Fig. 2. FPGA die area as dashed rectangle with array of marked measurement positions

The plastic package on the backside of the FPGA was removed to enable measurements close to the die surface. Backside access generally requires less practical effort in case of plastic or smartcard packages. 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 SNR of the detected location-based leakage depends on the measurement position on the surface of the die [11]. Since our attack is non-profiled, we are unable to find a position with high SNR through prior profiling. Instead, we choose measurement positions by pure 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. The dashed rectangle depicts the surface of the FPGA die which measures $\approx 5000 * 4000$ µm.

We perform the attack on those individual measurements. Further, we exploit the fact that our attack allows a straight-forward combination of measurements to increase the SNR. Since the attacked scalar is changed in every execution, those measurements must be recorded simultaneously. Simultaneous measurements could be recorded with an array of electromagnetic probes [20]. However, we only have one measurement probe of the same kind. Hence, to simulate the case of an array probe, we move this one probe to the marked positions and repeat 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 are 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.

4.2 Clustering Individual Measurements

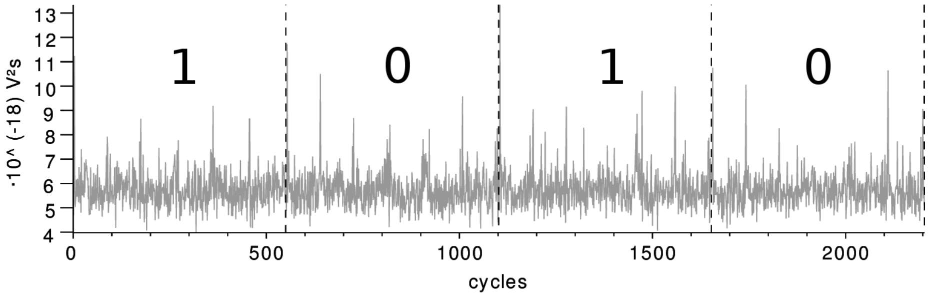


Fig. 3. Four samples (14 to 17) from the compressed measurement at position 3

We first perform the clustering attack on individual measurements. Hence, we segment 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 illustration 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 not clearly visible.

We attack the individual measurements by employing the unsupervised k-*means* clustering algorithm Alg. 1 to classify the samples in two clusters as described in Sect. 3.3. We assess the result by computing the remaining brute-force complexity required to recover the entirely correct scalar after clustering as described in Sect. 3.4. Figure 4 depicts this brute-force complexity for every individual measurement position according to Fig. 2. It is obvious, that none of the measurements contains enough SNR of the exploited location-based leakage for an entirely correct classification, thus, recovery of the secret scalar. *However, e.g., position* 8 *exhibits a brute-force complexity of only* 22 *bits which is clearly acceptable for a realistic attacker. This clearly demonstrates the capabilities of unsupervised cluster classification as a non-profiled single-execution attack on exponentiation algorithms to exploit single-execution leakage.*

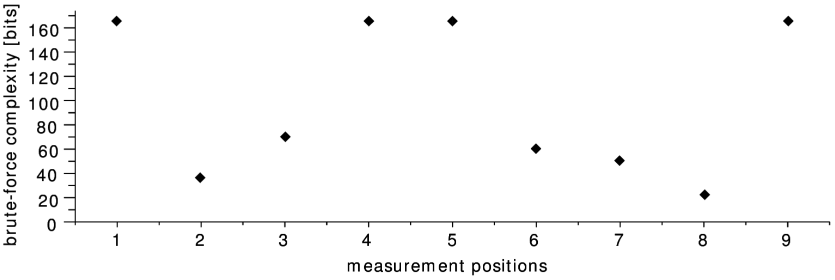


Fig. 4. Remaining brute-force complexity after clustering *individual measurements*

4.3 Clustering Combined Measurements

The results from clustering individual measurements lead to remaining brute-force complexities greater than zero. 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. 3.5 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. This clearly demonstrates the advantage of combining measurements for attacking exponentiation algorithms using unsupervised clustering algorithms.*

4.4 Discussion and SNR

Table 1 summarizes the derived remaining brute-force complexity values for all individual measurements as well as for combined measurements (denoted as '**all**'). 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 lead to largely incorrect results. Possible reasons for this are: an insufficient SNR of the exploited leakage, outlier samples, or that the specific clustering algorithm is inappropriate since the assumed model of cluster distributions does not fit.

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

measurement positions	1	2	3	4	5	6	7	8	9	**all**
SNR [dB]	9.3	8.9	11.1	7.0	12.2	11.2	11.6	10.7	10.0	**16.1**

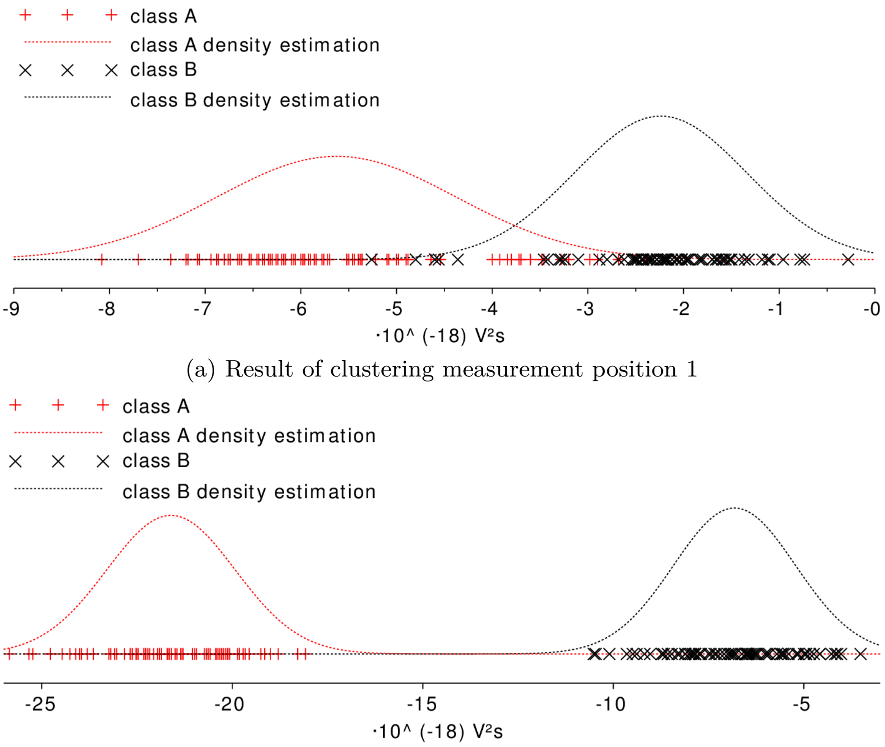
Table 2. SNR in dB for individual and combined measurements

Using the known scalar we derive the SNR contained in individual and combined measurements as in (1) and summarize the results in Tab. 2. *It can be observed that the SNR after a combination of measurements is significantly higher, i.e.* 16.1 *dB than in case of single measurements.*

The comparison of SNR values in Tab. 2 to brute-force complexity values in Tab. 1 from individual measurements leads to a less evident result. Position 5 e.g., exhibits a higher SNR than position 8 while the brute-force complexity for position 5 is 165 contrary to position 8, which only exhibits 22 bits. We explain this by assuming that the model of cluster distributions did not fit the leakage at this measurement position. A clustering algorithm with more parameters of freedom, e.g., the expectation-maximization algorithm, may exploit the SNR more effectively and lead to better classification results.

4.5 Illustration of Gain Through Combination of Measurements

Figure 5(a) and Fig. 5(b) *demonstrate the advantage of combining measurements* in an illustrative way. Figure 5(a) visually represents the result of clustering the measurement at position number 1. The clustering algorithm outputs two cluster means $\boldsymbol{\mu}_A$ and $\boldsymbol{\mu}_B$ and samples are classified according to a separation plane in the middle between those means. For the illustration of this clustering result, we projected all multivariate samples $\boldsymbol{t}_i$ (multi-dimensional) onto a line (one-dimensional) 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). Many samples are across the wrong side of the half distance between the two distributions which corresponds to the separation plane. These classification errors are expected when considering the values from Tab. 1.



(a) Result of clustering measurement position 1

(b) Result of clustering 9 combined measurements

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

Figure 5(b) depicts a similar linear projection after a clustering of 9 combined measurements. *It can clearly be observed, that the separation of the two classes is significantly improved by the combination of measurements.*

4.6 Countermeasures

Generally, all methods which reduce the SNR of arbitrary single-execution leakage, either by reducing the signal, or increasing the noise level, make attacks more difficult since the attacker relies on a single, or a few simultaneous measurements at best. Location-based single-execution leakage as it is exploited in this practical attack can specifically be prevented by randomizing variable locations [11], by balancing registers and their signal paths, or by locating them in an interleaved way that they cannot be distinguished [10].

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 which is a significant advantage for attackers.* In a practical evaluation we successfully recover the secret scalar from an FPGA-based ECC implementation. Individual measurements of the electromagnetic field lead to sufficiently low remaining brute-force complexities. Additionally, we demonstrate the advantage of combining simultaneous measurements which is straight-forward for clustering-based attacks. We conclude that attackers who exploit high-resolution measurements of the electromagnetic field, do not have to find measurement positions through profiling in this case because they are able to combine leakage information from multiple, simultaneous measurements.

References

1. Agrawal, D., Rao, J., Rohatgi, P.: Multi-channel attacks. In: Walter, C., Koç, C., 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. 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)
5. 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)
6. 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)
7. 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)
8. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification (2nd Edition). Wiley-Interscience, 2 edn. (Nov 2001)
9. 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)
10. 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)

11. 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)
12. 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)
13. 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)
14. 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)
15. 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)
16. 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)
17. 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)
18. 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 (march 2012)
19. Schindler, W., Itoh, K.: Exponent blinding does not always lift (partial) SPA resistance to higher-level security. In: Lopez, J., Tsudik, G. (eds.) Applied Cryptography and Network Security, Lecture Notes in Computer Science, vol. 6715, pp. 73–90. Springer Berlin / Heidelberg (2011)
20. 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)
21. Walter, C.: Sliding windows succumbs to big mac attack. In: Koç, C., 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)
22. 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)