Simpler and More Efficient Rank Estimation for Side-Channel Security Assessment

Cezary Glowacz[1], Vincent Grosso[2], Romain Poussier[2],
Joachim Schüth[1], François-Xavier Standaert[2].

[1] T-Systems GEI GmbH, Security Consulting & Engineering, Bonn, Germany.
[2] ICTEAM/ELEN/Crypto Group, Université catholique de Louvain, Belgium.

Abstract. Rank estimation algorithms allow analyzing the computational security of cryptographic keys for which adversaries have obtained partial information thanks to leakage or cryptanalysis. They are particularly useful in side-channel security evaluations, where the key is known by the evaluator but not reachable with exhaustive search. A first instance of such algorithms has been proposed at Eurocrypt 2013. In this paper, we propose a new tool for rank estimation that is conceptually simpler and much more efficient than this previous proposal. It allows approximating the key rank of (128-bit, 256-bit) symmetric keys with very tight bounds (i.e. with less than one bit of error), almost instantaneously and with limited memory. It also scales nicely to larger (e.g. asymmetric) key sizes, for which the previous algorithm was hardly applicable.

1 Introduction

Despite progresses in the analysis and understanding of side-channel attacks, empirical evaluations remain an essential ingredient in the security assessment of leaking devices. The main reason for this fact is that the leakage of cryptographic implementations is highly device-specific. This implies that the actual security level provided by ad hoc countermeasures such as masking (e.g. [2, 11] and related works) or shuffling (e.g. [6, 17] and related works) may depend on the underlying technology on which they are running (e.g. glitches in integrated circuits are an illustration of this concern [9]). In fact, even in leakage-resilient primitives that aim to prevent/mitigate side-channel attacks by cryptographic design, the need to bound/quantify the leakages in a rigorous way is an important ingredient for connecting formal analysis with concrete security levels (e.g. [4, 14]).

In this context, the usual strategy for an evaluation laboratory is to launch a set of popular attacks, and to determine whether the adversary can break the implementation (i.e. recover the key). The vast majority of these popular attacks are “divide-and-conquer” ones[1], where different pieces of a master key

[1] Including but not limited to Kocher et al.’s seminal Differential Power Analysis (DPA) [7], Brier et al.’s Correlation Power Analysis (CPA) [1], Chari et al.’s Template Attacks (TA) [3], Gierlichs et al.’s Mutual Information Analysis (MIA) [5] and Schindler et al.’s stochastic approach based on Linear Regression (LR) [12]. Following [8], we will use the term “standard DPAs” for those attacks.

are recovered independently, and then recombined via enumeration [10, 15]. But as recently observed by Veyrat-Charvillon, Gérard and Standaert at Eurocrypt 2013, such security evaluations are limited to the computational power of the evaluator [16]. This is typically a worrying situation since it sets a hard limit to the decision whether an implementation is "practically secure". For example, one could decide we have a practically secure AES implementation as soon as the number of keys to enumerate is beyond 2^{50}, but this does not provide any hint whether the concrete security level is 2^{51} or 2^{120}. The latter makes a significant difference in practice, especially in view of the possibility of improved measurement setups, signal processing, information extraction, ..., that usually has to be taken into account for any physical security evaluation, e.g. via larger security margins. As a consequence, the main contribution in [16] was to introduce a rank estimation algorithm which enables evaluators to (quite efficiently) approximate the security level of any implementation, by approximating the position of the master key in the list of 2^{128} possible ones provided by an attack (even if it is beyond enumeration power). This allowed, for the first time, to compute all the security metrics introduced in [13] and to summarize them into "security graphs" (i.e. plots of the adversary's success probability in function of the number of side-channel measurement and enumeration power, essentially).

Technically, the Eurocrypt 2013 algorithm essentially results from the time vs. memory tradeoff between depth-first and breadth-first search in a large data structure representing the key space. More precisely, since depth-first exploration of the key space is too computationally intensive, it rather exploits breadth-first search up to the memory limits of the computing device on which rank estimation is performed. This allows the algorithm to rapidly converge towards reasonably accurate bounds on the key rank. But of course, it implies that refining the bounds becomes exponentially difficult at some point, which may lead to limited accuracies in certain contexts (e.g. large key sizes, typically). Concretely, the representation of a side-channel attack's results also has a strong impact on the efficiency of the Eurocrypt 2013 rank estimation. For example in the AES case, representing a DPA outcome as 8 lists of size 2^{16} leads to more (time) efficient rank estimation than representing it as 16 lists of size 2^{8}. Using a (more memory consuming) representation with 5 lists of 2^{24} elements and one list of 2^{8} elements typically allowed bounds with approximately 10 bits of tightness[2] within seconds of computation, and bounds with approximately 5 bits of tightness within minutes of computation, for a 128-bit key leading to a post side-channel attack security level of 80 bits. Note that the time complexity of the latter rank estimation algorithm is dependent of the estimated security level (and 80-bit was the experimental worst-case in the 128-bit example of [16]). Summarizing, the Eurocrypt 2013 algorithm provides satisfying estimations of the key rank as long as the key size is limited (to symmetric key sizes, typically) and the tightness required by the evaluators can be left to a couple of bits.

[2] Measured with the log of the ratio between the upper and lower bounds.

In this paper, we provide an alternative rank estimation algorithm that enjoys simplicity and (much) improved (time and memory) efficiency. The algorithm essentially works in fours steps. First, we express the DPA outcome with lists of log probabilities (each list corresponding to a piece of key). Second, we compute the histograms of these log probabilities for all the lists, with a sufficient number of equally-sized bins. Third, we recursively compute the convolution between these histograms. Eventually, we approximate the security level from the last histogram as the number of keys having larger log probabilities than the correct one (that is known by the evaluator). Bounds can additionally be obtained by tracking the quantization errors (depending on the bin width). Besides its simplicity, this algorithm leads to bounds with less than one bit of tightness within seconds of computation (using the same computing platform as for the previous estimates). Furthermore, and contrary to the Eurocrypt 2013 algorithm, it nicely scales to larger key sizes and leads to rank estimations with good tightness for key sizes up to the ones considered in the asymmetric cryptographic setting.

We finally recall that the proposed algorithm is not limited to physical security evaluations, and is potentially useful in any cryptanalysis context where experiments are needed to validate an hypothetical attack model as well.

2 Background

2.1 Side-channel cryptanalysis

Details on how divide-and-conquer side-channel attacks actually extract information about the master key are not necessary for describing the rank estimation problem. For the rest of the paper, we only need to specify the DPA outcomes as follows. Say we target an n-bit master key k and cut it in $N_p = \frac{n}{b}$ pieces of b bits, next denoted as subkeys k_i (for simplicity, we assume that b divides n). The side-channel adversary uses the leakages corresponding to a set of q inputs $\mathcal{X}_q$ leading to a set of q leakages $\mathcal{L}_q$. As a result of the attack, he obtains N_p lists of probabilities $\Pr[k_i^*|\mathcal{X}_q, \mathcal{L}_q]$, where $i \in [1 : N_p]$ and k_i^* denotes a subkey candidate among the $N_k = 2^b$ possible ones. Note that TA and LR-based attacks indeed output such probabilities directly. For other (typically non-profiled) attacks such as DPA or CPA, a Bayesian extension can be used for this purpose [15].

2.2 Rank estimation

Concretely, each of the N_p lists of probabilities obtained by the divide-and-conquer adversary is typically small (i.e. easy to enumerate). So one can straightforwardly compute the rank of each subkey. The rank estimation problem is simply defined as the problem of estimating the master key rank based on the N_p lists $\Pr[k_i^*|\mathcal{X}_q, \mathcal{L}_q]$. Quite naturally, the problem is trivial when the attack is directly successful (i.e. when the master key is rated first). But it becomes tricky whenever this rank becomes larger. The solution in [16] was to organize the keys by sorting their subkeys according to the posterior probabilities provided by

DPA, and to represent them as a high-dimensional dataspace (with N_p dimensions). The full key space can then be partitioned in two volumes: one defined by the key candidates with probability higher than the correct key, one defined by the key candidates with probability lower than the correct key. Using this geometrical representation, the rank estimation problem can be stated as the one of finding bounds for these “‘higher” and “lower” volumes. It essentially works by carving volumes representing key candidates on each side of their boundary, in order to progressively refine the (lower and upper) bounds on the key rank. As mentioned in introduction, this approach is efficient as long as the carved volumes are large enough, and becomes computationally intensive afterwards.

3 Simpler and more efficient rank estimation

3.1 Algorithm specification

We first denote the lists of log probabilities obtained from the previously defined DPA outcomes as $LP_i = \log(\Pr[k_i^*|\mathcal{X}_q, \mathcal{L}_q])$, and the histograms (with N_{bin} equally-sized bins) corresponding to these lists as $H_i = \mathsf{hist}(LP_i, \text{bins})$. We further denote the convolution between two histograms as $\mathsf{conv}(H_i, H_j)$. From these notations, our rank estimation proposal is specified by Algorithm 1.

Algorithm 1 Rank estimation ($H_i, \log(\Pr[k|\mathcal{X}_q, \mathcal{L}_q])$).

```
initialization: $H_{\text{curr}} =$
```

3.2 Bounding the error

Let us assume two log probabilities $LP_1^{(j)}$ and $LP_2^{(j)}$ corresponding to the jth candidates in the lists LP_1 and LP_2. They are associated with two bins of central value $m_1^{(j)}$ and $m_2^{(j)}$ in the histograms H_1 and H_2. Whenever summing those log probabilities (as required to combine two lists of probabilities), it may happen that the central value of the bin corresponding to $LP_1^{(j)} + LP_2^{(j)}$ is different than $m_1^{(j)} + m_2^{(j)}$ (which corresponds to the approximated sum of log probabilities obtained from the convolution in Algorithm 1). This typically occurs if the distance between the log probabilities $LP_1^{(j)}, LP_2^{(j)}$ and their bins' central values $m_1^{(j)}, m_2^{(j)}$ is too large, as illustrated by the following numerical example.

Example 1. Take two lists $LP_1 = \{0, 0.02, 0.07, 0.11, 0.14, 0.16, 0.19, 0.3\}$ and $LP_2 = \{0.02, 0.02, 0.036, 0.04, 0.12, 0.19, 0.24, 0.29\}$. For $N_{\text{bin}} = 3$, it leads to a common binsize of $S_{\text{bin}} = 0.1$, and central values $\{0.05, 0.15, 0.25\}$. Hence, we obtain $H_1 = \{3, 4, 1\}$ and $H_2 = \{4, 2, 2\}$. The convolution $H_3 = \mathsf{conv}(H_1, H_2)$ is a histogram with $N_{\text{bin}} = 5$ and central values $\{0.1, 0.2, 0.3, 0.4, 0.5\}$, given by $H_3 = \{12, 22, 18, 10, 2\}$. As a result, the sum of log probabilities $LP_1^{(7)} + LP_2^{(8)}$ equals $0.19 + 0.29 = 0.48$ and should be placed in the bin with central value 0.5. Yet, since their corresponding central values are 0.15 and 0.25, the convolution approximates their sum within the bin of central value $0.15 + 0.25 = 0.4$.

In other words, the rank estimation accuracy is limited by quantization errors (of one bin in our example). Hopefully, we can bound the number of bins between the result of the convolution and the real sum of log probabilities as follows.

Proposition 1. *Let $\{LP_i\}_{i=1}^{N_p}$ be N_p lists of log probabilities with their jth elements denoted as $LP_i^{(j)}$ and set in the bins of central values $m_i^{(j)}$ of the corresponding histograms $\{H_i\}_{i=1}^{N_p}$. The quantization error (measured in bins) between $\sum_{i=1}^{N_p} LP_i^{(j)}$ (i.e. the actual sum of log probabilities) and $\sum_{i=1}^{N_p} m_i^{(j)}$ (i.e. the sum of the bins' central values corresponding to these log probabilities) is at most $\frac{N_p}{2}$.*

Proof. If S_{bin} is the binsize, the equation $\left|LP_i^{(j)} - m_i^{(j)}\right| \leq \frac{S_{\text{bin}}}{2}$ holds for each $i \in [1 : N_p]$. Hence, by summing over all the pieces, we obtain:

$$-\frac{S_{\text{bin}}}{2} \times N_p \leq \sum_{i=1}^{N_p} (LP_i^{(j)} - m_i^{(j)}) \leq \frac{S_{\text{bin}}}{2} \times N_p.$$

Hence, we also have:

$$\left|\sum_{i=1}^{N_p} LP_i^{(j)} - \sum_{i=1}^{N_p} m_i^{(j)}\right| \leq \frac{N_p}{2} \times S_{\text{bin}},$$

which limits the distance between $\sum_{i=1}^{N_p} LP_i^{(j)}$ and $\sum_{i=1}^{N_p} m_i^{(j)}$ to $\frac{N_p}{2}$ bins. □

Following, we can directly bound the estimated rank in Algorithm 1 with:

$$\text{rank\_lower\_bound} = \sum_{i=\mathsf{bin}(\log(\Pr[k|\mathcal{X}_q,\mathcal{L}_q]))+N_p}^{N_p\cdot N_{\text{bin}}-(N_p-1)} H_{\text{curr}}(i),$$

and:

$$\text{rank\_upper\_bound} = \sum_{i=\mathsf{bin}(\log(\Pr[k|\mathcal{X}_q,\mathcal{L}_q]))-N_p}^{N_p\cdot N_{\text{bin}}-(N_p-1)} H_{\text{curr}}(i),$$

where the N_p (rather than $\frac{N_p}{2}$) value comes from the fact that the distance limit holds for each list of log probabilities independently. Hence, a triangle inequality with $\sum_{i=1}^{N_p} m_i^{(j)}$ as origin gives us an interval of size $2 \times N_p$ bins around $\sum_{i=1}^{N_p} LP_i^{(j)}$.

4 Performance evaluation

In this section, we analyze the performances of Algorithm 1. For comparison purposes, we first use the same AES case study as Veyrat-Charvillon et al. We then extend our experiments to larger key sizes. Note that the functional correctness of our algorithm directly derives from the previous section. Yet, we tested its implementation by comparing our results with the ones obtained by enumeration for key ranks up to 2^{32}, and made sure that these results were consistent with the the ones obtained using the open source code of the Eurocrypt 2013 paper.

4.1 AES-128 case study

As in [16], we considered simulated attacks where the adversary is provided with 16 leakage samples of the shape $l_i = \mathsf{HW}(\mathsf{S}(x_i \oplus k_i)) + n_i$ for $i \in [1:16]$, where HW is the Hamming weight function, S is the AES S-box, k_i and x_i are the previously defined subkeys and corresponding plaintext bytes, and n_i is a Gaussian-distributed random noise. We then performed classical TAs using the noise variance and number of plaintexts as parameters, so that the adversary computes 16 lists of 256 posterior probabilities. As in the previous paper as well, the efficiency of the rank estimation algorithms was quite independent of the type of leakage exploited: the only influencing factor in our performance evaluations was the rank of the correct key candidate. For this purpose, we started by reproducing an experiment where we launched many independent attacks, with different security levels and increasing time complexities, and plotted the resulting bounds' tightness (defined in Footnote 1). The left (resp. right) part of Figure 1 contains the results of this experiment for the Eurocrypt 2013 algorithm[3] (resp. Algorithm 1). In both cases, they were obtained on a desktop

[3] Using 8 lists of size 2^{16} for illustration.

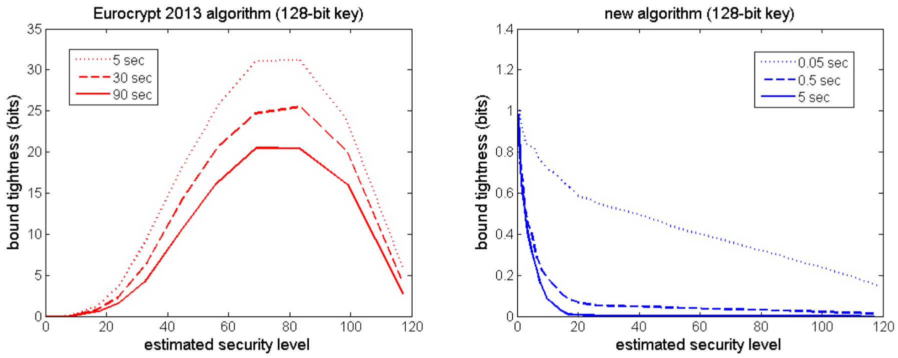

Fig. 1. Rank estimation tightness in function of the security level.

computer with an Intel i7 core, without any parallelization effort (details on the implementation of Algorithm 1 are in Appendix B). Two clear observations can be extracted from this figure. First, the security levels leading to the most complex rank estimations differ for the two algorithms (i.e. key ranks around 2^{80} are most challenging with the Eurocrypt 2013 algorithm, enumerable key ranks are the most challenging with ours). Second and most importantly, the new bounds are much tighter (less than one bit of distance between the bounds) and obtained much faster (in less than a second). For completeness, note that the experiments with 0.05 sec, 0.5 sec and 5 sec of computations in the right part of the figure respectively correspond to 5K, 50K and 500K bins.

In order to make the comparison even more explicit, we additionally provide the "convergence graphs" where the upper and lower bounds on the key rank are plotted in function of the time complexity. As clear from Figure 2, the convergence is incomparably faster with the histogram-based approach than with the Eurocrypt 2013 one. Additional results for other relevant security levels (namely $\approx$ 60-bit and $\approx$ 100-bit) are provided in Appendix, Figures 5 and 6.

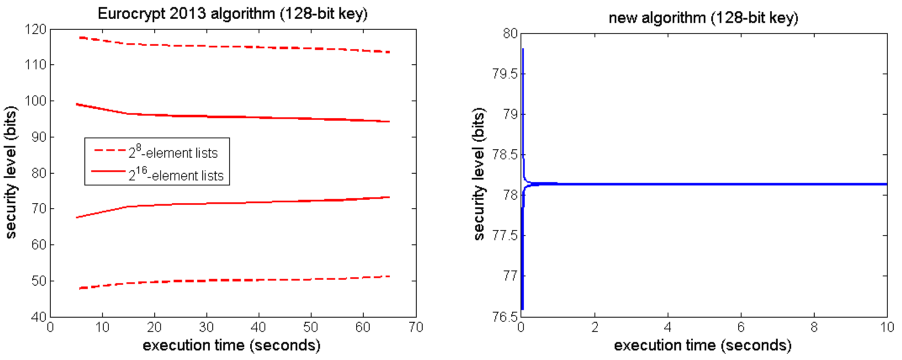

Fig. 2. Rank estimation convergence for an $\approx$ 80-bit security level.

4.2 Larger key sizes

In order to analyze situations with larger key sizes, we simply extended our AES simulated setting to more 8-bit pieces. Namely, we considered key sizes of 256, 512 and 1024 bits (i.e. $N_p = 32, 64, 128$). We omit the figure corresponding to the 256-bit case because it is extremely close to the 128-bit one, and represent the convergence graphs of the two latter cases in Figure 3. While the application of the Eurocrypt 2013 method hardly provides useful results on this context, the figure clearly exhibits that Algorithm 1 produces tight bounds within seconds of computation, even in this challenging case. Interestingly, the increase of execution time in the 1024-bit example mainly corresponds to the convolutions' cost that becomes significant as the number of bins increases (in $N_{\text{bin}} \log(N_{\text{bin}})$).

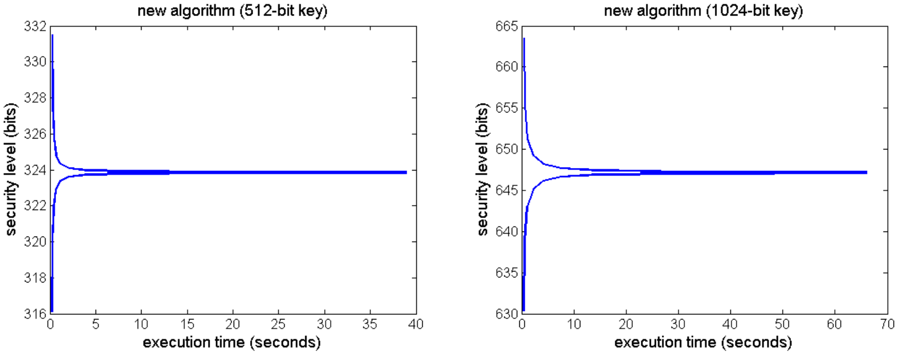

Fig. 3. Rank estimation convergence for 512- and 1024-bit keys.

Eventually and for completeness, we also provide graphs representing the bounds' tightness in function of the security level for these 512- and 1024-bit cases in Figure 4. They essentially confirm the observation already made in Figure 1 that the most challenging key ranks to estimate are the lower ones. Note that in these latter cases, the experiments with 0.1 sec (resp. 0.5) and 1 sec (resp. 5 sec) were performed with respectively 2K and 20K bins.

5 Conclusions

This paper provides a surprisingly simple alternative of rank estimation algorithm, that significantly outperforms the previous proposal from Eurocrypt 2013. It has natural applications in the field of side-channel cryptanalysis and is a tool of choice for evaluation laboratories willing to quantify the security level of a leaking implementation in a rigorous manner. More generally, it can also be useful in the evaluation of any cryptanalytic technique where the advantage gained is not sufficient for key recovery and not predictable by analytical means.

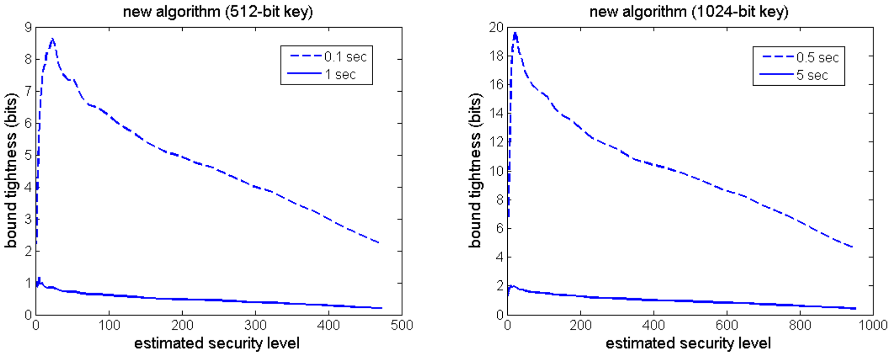

Fig. 4. Rank estimation tightness in function of the security level.

Acknowledgements. F.-X. Standaert is a research associate of the Belgian Fund for Scientific Research (FNRS-F.R.S.). This work has been funded in parts by the European Commission through the ERC project 280141 (CRASH).

References

1. Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2004.
2. Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Wiener [18], pages 398–412.
3. Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, volume 2523 of *Lecture Notes in Computer Science*, pages 13–28. Springer, 2002.
4. Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 293–302. IEEE Computer Society, 2008.
5. Benedikt Gierlichs, Lejla Batina, Pim Tuyls, and Bart Preneel. Mutual information analysis. In Elisabeth Oswald and Pankaj Rohatgi, editors, *Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings*, volume 5154 of *Lecture Notes in Computer Science*, pages 426–442. Springer, 2008.
6. Christoph Herbst, Elisabeth Oswald, and Stefan Mangard. An AES smart card implementation resistant to power analysis attacks. In Jianying Zhou, Moti Yung, and Feng Bao, editors, *Applied Cryptography and Network Security, 4th International Conference, ACNS 2006, Singapore, June 6-9, 2006, Proceedings*, volume 3989 of *Lecture Notes in Computer Science*, pages 239–252, 2006.

7. Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Wiener [18], pages 388–397.
8. Stefan Mangard, Elisabeth Oswald, and François-Xavier Standaert. One for all - all for one: unifying standard differential power analysis attacks. *IET Information Security*, 5(2):100–110, 2011.
9. Stefan Mangard, Thomas Popp, and Berndt M. Gammel. Side-channel leakage of masked CMOS gates. In Alfred Menezes, editor, *Topics in Cryptology - CT-RSA 2005, The Cryptographers' Track at the RSA Conference 2005, San Francisco, CA, USA, February 14-18, 2005, Proceedings*, volume 3376 of *Lecture Notes in Computer Science*, pages 351–365. Springer, 2005.
10. Jing Pan, Jasper G. J. van Woudenberg, Jerry den Hartog, and Marc F. Witteman. Improving DPA by peak distribution analysis. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, *Selected Areas in Cryptography - 17th International Workshop, SAC 2010, Waterloo, Ontario, Canada, August 12-13, 2010, Revised Selected Papers*, volume 6544 of *Lecture Notes in Computer Science*, pages 241–261. Springer, 2010.
11. Matthieu Rivain and Emmanuel Prouff. Provably secure higher-order masking of AES. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, volume 6225 of *Lecture Notes in Computer Science*, pages 413–427. Springer, 2010.
12. 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, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings*, volume 3659 of *Lecture Notes in Computer Science*, pages 30–46. Springer, 2005.
13. François-Xavier Standaert, Tal Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, volume 5479 of *Lecture Notes in Computer Science*, pages 443–461. Springer, 2009.
14. François-Xavier Standaert, Olivier Pereira, and Yu Yu. Leakage-resilient symmetric cryptography under empirically verifiable assumptions. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 335–352. Springer, 2013.
15. Nicolas Veyrat-Charvillon, Benoît Gérard, Mathieu Renauld, and François-Xavier Standaert. An optimal key enumeration algorithm and its application to side-channel attacks. In Lars R. Knudsen and Huapeng Wu, editors, *Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers*, volume 7707 of *Lecture Notes in Computer Science*, pages 390–406. Springer, 2012.
16. Nicolas Veyrat-Charvillon, Benoît Gérard, and François-Xavier Standaert. Security evaluations beyond computing power. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 126–141. Springer, 2013.

17. Nicolas Veyrat-Charvillon, Marcel Medwed, Stéphanie Kerckhof, and François-Xavier Standaert. Shuffling against side-channel attacks: A comprehensive study with cautionary note. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pages 740–757. Springer, 2012.
18. Michael J. Wiener, editor. *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*. Springer, 1999.

A Additional figures

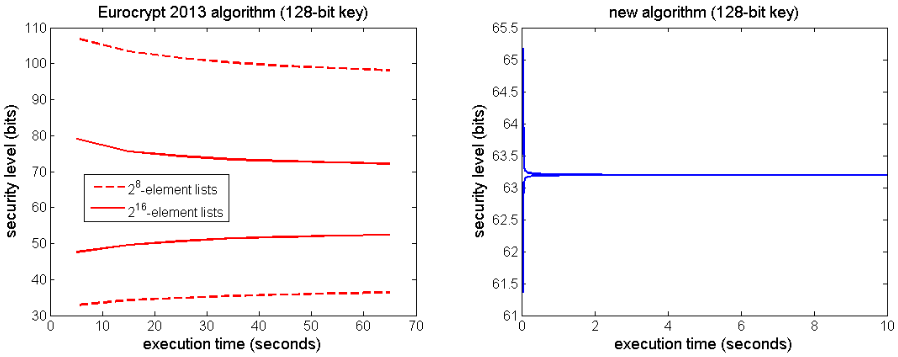

Fig. 5. Rank estimation convergence for an ≈ 60-bit security level.

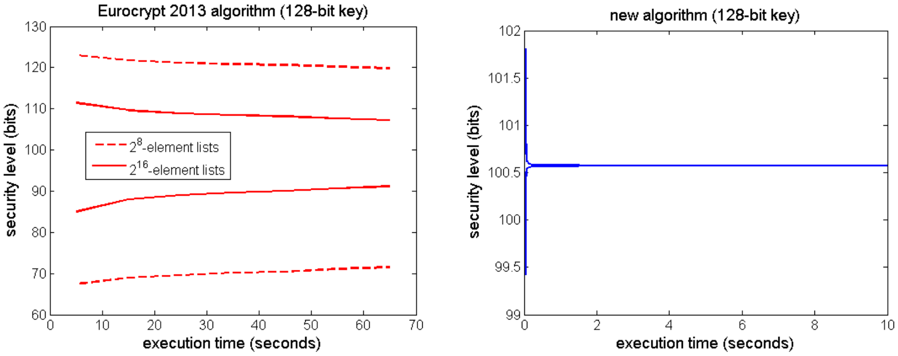

Fig. 6. Rank estimation convergence for an ≈ 100-bit security level.

B Implementation details

One additional advantage of Algorithm 1 is that is it straightforward to implement, in particular if efficient convolution algorithms for vectors of arbitrary precision integers are available out of the box, as in a number of mathematical programming languages. Our experiments were performed with Matlab scripts, which turn out to be sufficiently efficient for all the case studies we considered. Yet, we mention that if efficient convolutions algorithms are not available out of the box, they can easily be implemented using more readily available primitives. One possible approach is to use a mixture of floating point arithmetic and representation of large numbers according to the Chinese Remainder Theorem. For example, a set of moderately sized primes, like the 20 largest primes below 10000, is chosen, and each histogram is converted into a CRT representation by using the 20 integer vectors that are obtained by modular reduction with each of the 20 chosen primes. This particular choice of primes is suitable to represent numbers of up to 265 bits (i.e. for rank estimation of 256-bit keys). In this example, when two histograms are to be convoluted, 20 regular convolutions are computed, each one modulo the corresponding prime base. To speed up the computation, each of these 20 convolutions can be performed by multiplication in Fourier space with regular double precision floating point arithmetic, i.e. two FFTs, one element-wise multiplication of two complex vectors, and one inverse FFT. Since the exact result is known to consist of integer values, computational inaccuracies can unambiguously be removed by rounding to the nearest integers. In the CRT representation, the input values are bounded by the prime bases, and this sufficiently limits the requirements on floating point precision. After rounding to integer values, modular reductions to the corresponding prime bases are performed to obtain the CRT representation of the convolution result. The result can be left in CRT representation until all histograms (i.e. the histograms for all subkeys) have been convoluted. Only a single CRT back transform to large integers is required after all the histograms have been convoluted.