Behind the Scene of Side Channel Attacks in a nutshell

Made by : Tanguy Flégeau

On : 12/20

For : ISTIC – RENNES – Project SCA

Source : https://eprint.iacr.org/2013/794.pdf

Abstract : Since the introduction of side channel attacks in the nineties, a large amount of work has been devoted to their effectiveness and efficiency improvements. On the one side, general results and conclusions are drawn in theoretical frameworks, but the latter ones are often set in a too ideal context to capture the full complexity of an attack performed in real conditions. On the other side, practical improvements are proposed for specific contexts but the big picture is often put aside, which makes them difficult to adapt to different contexts. This paper tries to bridge the gap between both worlds. In this context we propose new ideas to improve the effectiveness and/or efficiency of the two considered attacks : stochastic and CPA attacks.

TABLE OF CONTENT

Introduction to SCA

Definition and context

In computer security, a side-channel attack is any attack based on information gained from the implementation of a computer system, rather than weaknesses in the implemented algorithm itself (e.g. cryptanalysis and software bugs). Timing information, power consumption, electromagnetic leaks or even sound can provide an extra source of information, which can be exploited. In reality, many powerful side-channel attacks are based on statistical methods which require a lot of datas. This means that the majority of SCA require technical knowledge of the internal operation of the system.

Nowadays, the rise of Web 2.0 applications and software-as-a-service has significantly raised the possibility of side-channel attacks on the web, even when transmissions between a web browser and server are encrypted. Even if the necessary skills are not easely accessible to everyone, the possibilities of access and manipulation of the systems are constantly increasing. This aspect is very important because it demonstrates that the physical security of the components is as important as the security on the software side.

Theoretical frameworks VS practical works

Since the seminal DPA (Differential Power Analysis), various Side Channel Attacks (SCA) have been proposed and improved. In order to to compare and classify them, theoretical frameworks have then been introduced. Their main purpose is to identify the attacks similarities and differences, and to exhibit contexts where one is better than another. They have laid the foundation for a general comparison and evaluation framework. In parallel, several practical works have addressed issues arising when applying an SCA in the real world (e.g. in an industrial context). Those works essentially attempt to fill the gap between the theoretical analysis of the attacks and their application in non-idealized contexts. The latter analyses are indeed usually dedicated to one specific attack running against a specific target device, which makes them hard to generalize.

The starting observation of our study is that side channel traces are never reduced to one point in practice, even when they rely on the manipulation of a single variable. In contrary, those traces are often composed of a large number of points (typically several thousands). In spite of the evidence of this observation, it is rarely taken into account when analysing the effectiveness of a side channel attack. Such an analysis is indeed frequently done under the assumption, sometimes implicit, that a small number of points of interest (POI) has already been extracted from the traces either by pattern matching, by dimension reduction or thanks to a previous successful attack. However, the two first categories of techniques are not yet perfect and, after reduction, the traces are often still composed of several points in practice. The third technique allows for interesting analyses, but it does not correspond to a real attack context. Moreover, the best POI for one attack type may not be so good for another one. Eventually, we come to a situation where attacks are analysed in a (uni-dimensional) context which does not fit with the (multi-dimensional) reality faced by the attack practitioners.

Another interesting issue when dealing with a large number of high dimensional traces is the reduction of the computational complexity. Here again, some works have investigated the use of parallel computing to decrease the data processing time but their goal was not to diminish the algorithmic complexity of the attacks.

Practical Evaluations of SCA attacks

Stochastic attacks

Linear regression attacks (a.k.a. stochastic attacks) have been introduced in 2005. Initially, they were presented with a profiling step and were viewed as an alternative to the template attacks. They have been used to analyse/model the deterministic part of the information leakage for complex circuits. LRA are known to be a robust methods as they make use of independent bits leakage. This leakage assumption is more general than Hamming weight/ Hamming distance model used in correlation power attack (CPA). Even if it can be applied in the same context as CPA, this attack has a weaker assumption on the device behavior. As a matter of fact, all those analyses assume that the side-channel traces are composed of a single leakage point, where it's generally high dimensional in real contexts. This is the same issue between theoretical frameworks and practical works we exposed before.

The strength of the LRA is its ability to adapt to the instantaneous leakage. However, this adaptability is also its weakness as it makes difficult to compare the different instantaneous results at the right time. Concretly, the issue with this kind of attack is that the comparaison of the instantaneous maximum scores can be checked at a specific time where the correct key is not ranked first. This explains why the total maximisation approach sometimes fails in returning the correct key candidate. This also refers to the fact that side channel attacks are not always adequate for certain practical cases.

To build a better rule than the total maximisation test, we respectively plotted in the third and fourth traces of Figure 2 the mean (plain green trace) and the variance (plain red trace) of the instantaneous scores (i.e. the values $\mu(u) = 2^{-8} \sum_k R[k][u]$ and $\sigma(u) = 2^{-8} \sum_k (R[k][u] - \mu(u))^2$ with **u** denoting the time coordinate in abscissa). For each time, we also plotted in black dashed line, the maximum score **max(u) = maxk(R[k][u])**. It may be observed that the correct key is ranked first at the time **u** when the distance **max(u) − μ(u)** is large and **σ(u)** is small. The third (red) trace and the fourth (gray) trace aim at supporting this claim. Eventually, they suggest us the following pre-processing before comparing the instantaneous attack results: for each leakage coordinate, center the maximum of the coefficients of determination and divide it by their standard deviation. The resulting scoring is plotted in the fifth (magenta) trace, where it can been checked that the maximum is indeed achieved for the correct key. Also, further concerns about LRA's algorithm efficiency exist but wont be explained in this document.

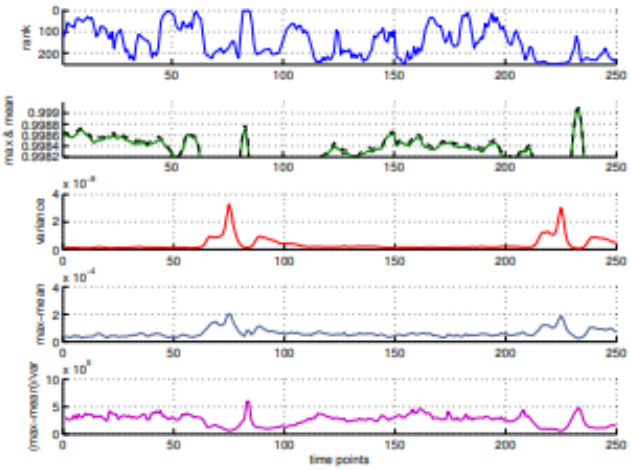

Figure 1 : LRA : Scores Statistics

Correlation Power analysis attacks

Correlation Power Analysis was proposed in 2004. The idea is to use PCC (Pearson's correlation coefficient) as a distinguisher instead of the difference of means test like for LRA. CPA is certainly nowadays the most popular side channel technique as it proved itself (combined with classical leakage models like the Hamming weight and Hamming distance) very efficient on an overwhelming majority of hardware architectures. CPA with PCC distinguisher and Hamming techniques even alows for better detection of 'gost peaks' than the classical DPA and LRA. Gost peaks are responsible for false positives and negatives in your results. Subsequent works on CPA effectiveness essentially consisted in applying signal filtering techniques or pre-processing on the leakage traces as it's already doing pretty well. Concerning the algorithmic efficiency, very few works dealt with improving this part. In fact, published works on this subject have essentially addressed the issue of processing the CPA on multicore CPU or GPU. However, it seems to be possible to re-writing CPA in a partitioning way, thus allowing for a significant efficiency gain.

In a correlation power analysis attack, the adversary computes, for each plaintext **xi** and each sub-key hypothesis **k**, an hypotheses tuple **zi** such that **zi = F(xi , k)**. Then, he chooses a model function and applies it to each hypothesis which leads to the construction of a new set of predictions. This latter set of hypotheses is then compared to the set of leakages to assess on the likelihood of **k** as a candidate for one of the sub-keys **k**. The core principle of a CPA is to make the comparison by estimating the linear correlation between **H** and each coordinate **L** independently. We give bellow the pseudo-code corresponding to the CPA attack discussed previously.

In the previous section, we showed that LRA attacks can be improved with a normalisation strategy. It may seem natural to try it in a CPA context. Unfortunately, CPA is too different than LRA and this strategy doesn't help improving it. However, the fact that the same constant model **m** is used for each instantaneous attack (which was not the case for the LRA), implies that two correlation coefficients corresponding to two different coordinates of the vectors are directly comparable (since they are quantifying the linear correlation between the leakages and a common model). Therefore the most likely hypothesis corresponds to the greatest value of the correlation coefficient/score when the set of hypotheses **(zi)i** is tested with respect to each coordinate **u** of the vectors. This seems to be the main reason why CPA is far better than LRA and DPA (in most cases).

Input : a set of d-dimensional leakages $(\overrightarrow{\ell}_i)_{i\leqslant N}$ and the corresponding plaintexts $(x_i)_{i\leqslant N}$, a model function $\mathtt{m}(\cdot)$
Output: A candidate sub-key $\hat{k}$

/* Leakage mean and variance processing */
for $i = 0$ **to** $N-1$ **do**
  $\mu_{\overrightarrow{\mathbf{L}}} = \mu_{\overrightarrow{\mathbf{L}}} + \overrightarrow{\ell}_i$
  $\sigma_{\overrightarrow{\mathbf{L}}} = \sigma_{\overrightarrow{\mathbf{L}}} + \overrightarrow{\ell}_i^2$
$\mu_{\overrightarrow{\mathbf{L}}} = 1/N \cdot \mu_{\overrightarrow{\mathbf{L}}}$
$\mathrm{var}_{\overrightarrow{\mathbf{L}}} = 1/N \cdot \sigma_{\overrightarrow{\mathbf{L}}} - \mu_{\overrightarrow{\mathbf{L}}}^2$
/* Hypotheses mean and variance processing */
for $\hat{k} = 0$ **to** $2^n - 1$ **do**
  for $i = 0$ **to** $N-1$ **do**
    $z \leftarrow \mathtt{m}(F(x_i, \hat{k}))$
    $\mu_{\hat{k}} = \mu_{\hat{k}} + z$
    $\sigma_{\hat{k}} = \sigma_{\hat{k}} + z^2$
  $\mu_{\hat{k}} = \mu_{\hat{k}}/N$
  $\mathrm{var}_{\hat{k}} = \sigma_k/N - \mu_k^2$
/* Correlations processing */
for $\hat{k} = 0$ **to** $2^n - 1$ **do**
  /* Test hypothesis $\hat{k}$ for all leakage coordinates */
  for $u = 0$ **to** $d-1$ **do**
    /* Instantaneous attack (at time u) */
    $\mathrm{cov} = 0$
    for $i = 0$ **to** $N-1$ **do**
      $\mathrm{cov} = \mathrm{cov} + \mathtt{m}(F(x_i, \hat{k})) \times \overrightarrow{\ell}_i[u]$
    $\mathrm{cor}[\hat{k}][u] = (1/N \times \mathrm{cov} - \mu_{\hat{k}} \times \mu_{\overrightarrow{\mathbf{L}}}[u])/\sqrt{\mathrm{var}_{\hat{k}} \times \mathrm{var}_{\overrightarrow{\mathbf{L}}}[u]}$
/* Most likely candidate selections */
$\mathrm{candidate} = \mathrm{argmax}_{\hat{k}}(\max_u \mathrm{cor}[\hat{k}][u])$
return candidate

Figure 2 : CPA algorithm

Conclusions and Outlook

In this document we explained what's behind the scene of side channel attacks and in what context they apply. Also, we highlight that even if many working side channel attacks already exist, they are not as efficient as they could be. This can be explained by a few points. On one hand, the fact that SCA are often very specific to a system and can't generalise well. This is the theoretical frameworks vs practical works problem witch can be resolved/dimmed simply by making less assumptions on the information leakage, or/and by trying to get around the approximation of the pattern matching and dimension reduction steps. One the other hand, efficiency can be improved with optimised algorithms that runs in parrallel (GPU prefered) or via a more specific approach/implementation that fit the targeted system. In that case, the theoretical vs practical (or general vs specific) problem is still present, therefore alowing for better performance in this specific context, but still poorer performance than with a highly optimized general-purpose attack.

To conclude, as Side channel attacks progress, they become less and less specific and more and more powerfull. They can run faster with less complexity, and increase their accuracy to avoid false positives and false negatives during the feature extraction and detection processes. Nowadays, two types of attacks/technics seems to stand out from the pack (CPA put aside): TA (Template attack) and DL (Deep Learning) which seems to be most promising candidates for side channel attacks. They both have pros and cons depending on the context of the attacks. Surprisingly, TA improved with Principal Components Analysis (PCA) and normalization, honorably makes the grade versus the latest DL methods which demand more calculation power. Also, both approaches face high difficulties against static targets such as secret data transfers or key schedule. At the end of the day, the benefit of DL techniques stands in the better resistance of CNN to misalignment and to the general vs specific problem.