# Leakage Squeezing Countermeasure against High-Order Attacks

Houssem Maghrebi, Sylvain Guilley, Jean-Luc Danger



# Leakage Squeezing Countermeasure Against High-Order Attacks

Houssem MAGHREBI, Sylvain GUILLEY and Jean-Luc DANGER.

TELECOM-ParisTech, Crypto Group,
37/39 rue Dareau, 75 634 PARIS Cedex 13, France.

**Abstract.** In the recent years, side channel attacks have been widely investigated. In particular, second order attacks (2O-attacks) have been improved and successfully applied to break many masked implementations. In this context we propose a new concept to hinder attacks of all order: instead of injecting more entropy, we make the most of a single-mask entropy. With specially crafted bijections instantiated on the mask path, we manage to reduce the inter-class variance (method we call "leakage squeezing") so that the leakage distributions become almost independent from the processed data. We present two options for this countermeasure. The first one is based on a recoded memory with a size squared w.r.t. the unprotected requirement, whilst the second one is an enhancement alleviating the requirement for a large memory. We theoretically prove the robustness of those implementations and practically evaluate their security improvements. This is attested by a robustness evaluation based on an information theoretic framework and by a 2O-DPA, an EPA and a multi-variate mutual information analysis (MMIA) attack metric. As opposed to software-oriented 3O-DPA-proof countermeasures that seriously impact the performances, our is hardware-oriented and keeps a complexity similar to that of a standard 2O-attack countermeasure with an almost untouched throughput, which is a predominant feature in computing-intensive applications.

**Keywords:** Higher-Order Differential Power Analysis, Variance-based Power Attack (VPA), Multi-variate Mutual Information Analysis (MMIA), Masking Countermeasure, Leakage Squeezing, FPGA.

## 1 Introduction

During the last ten years, a lot of effort has been dedicated towards the research about side-channel attacks [1, 10] and the development of corresponding countermeasures. In particular, there have been many endeavors to develop effective countermeasures against differential power analysis (DPA) [11] attacks.

Amongst the two major countermeasures against DPA, hiding and masking, the latter is certainly the least complex to implement when applied at the algorithmic level. The idea of masking the intermediate values inside a cryptographic algorithm has been suggested in several papers [2, 4, 12] as a possible countermeasure to power analysis attacks. Masking ensures that every single variable

is masked with at least one random value so that a classical (first order) DPA attack cannot be successfully carried out anymore. However other attacks, such as the Higher Order DPA attacks [19, 20, 24], exist that can defeat masking.

In fact, masking can be defeated if the attacker knows how to combine the leakages corresponding to the masked data and its mask. This is known as second-order, or more generally higher-order, power analysis (abridged 2O-DPA and HO-DPA) and was originally suggested by Thomas S. Messerges in [19]. Investigating 2O-DPA, however, is of major importance for practitioners as it remains a good alternative that is powerful enough to break real-life, DPA-protected security products.

The attacker is allowed to profile the leakage in order to exhibit a relationship between the statistical distribution of the leakage and the value of a sensitive variable. Once this relationship is determined, the likelihood of key guesses is estimated given the distribution of the leakage. Such attacks are based on the same principle as the template attacks introduced by Suresh Chari *et al.* in [5]. These attacks have been successfully applied by Éric Peeters *et al.* in [20] to break some masked implementations more efficiently than any combining 2O-DPA. Moreover, Houssem Maghrebi *et al.* in [14] proposed a 2O-DPA based on variance analysis, called Variance Power Analysis (VPA), which is powerful enough to practically break a masked DES implemented in an FPGA. More recently, a generic multi-variate attack called MMIA has been introduced by Benedikt Gierlichs *et al.* [8] to attack high-order countermeasures. Therefore, there is a need for countermeasures thwarting 2O-DPA in particular and HO-DPA in general. We describe in the present paper a methodology to squeeze the leakage distributions so that any partitioning becomes almost indistinguishable.

The paper is organized as follows. Section 2 presents the state-of-the-art of first order masking and describes its weaknesses against 2O-DPA. The description of the concept of leakage squeezing is provided in section 3. The section 4 presents two variants of implementations and includes the experimental results about the complexity and robustness evaluation. Finally, section 5 concludes the paper and opens some perspectives.

## 2 State of the art

### 2.1 First Order Masking Overview

Let us consider the masked DES studied at UCL [23], whose principle is illustrated in Fig. 1. This algorithmic masking associates a mask $ML, MR$ to the plaintext $L, R$.

At each round $i \in [1:16]$ an intermediate mask $ML_i$, $MR_i$ is calculated in parallel with the intermediate cipher word $L_i$, $R_i$. If we let apart the expansion $E$ and the permutation $P$, the DES round function $f$ is implemented in a masked way by using a set of functions $S$ and a set of functions $S'$:

$$\begin{cases} \text{masked data: } S(x_m \oplus k) = S(x \oplus m \oplus k) = S(x \oplus k) \oplus m' \,, \\ \text{mask} \qquad\qquad m' = S'(x_m \oplus k, m) = S'(x \oplus m \oplus k, m) \,. \end{cases} \tag{1}$$

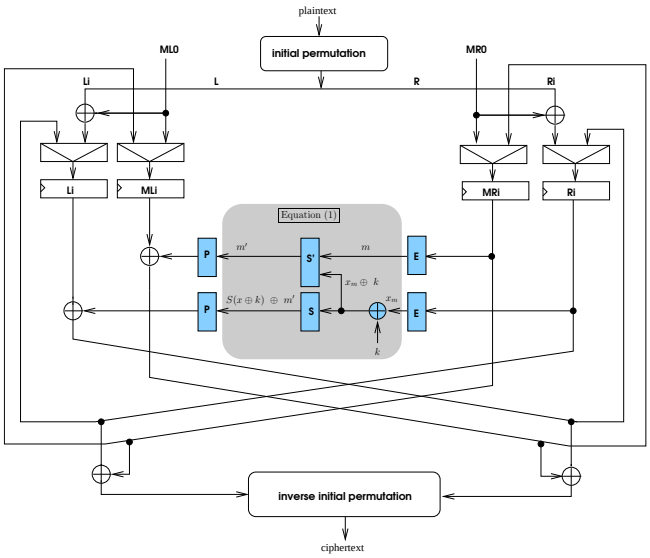



**Fig. 1.** ROM Masked DES.

The variable $m'$ is a new mask reusable for the next round. The set of functions $S$ contains the traditional S-boxes applied on masked intermediate words. The size of each $S$ is 64 words of 4 bits when implemented with a ROM. $S'$ is a new table which has a much greater ROM size of $4K$ words of 4 bits, as there are two input words of 6 bits.

The two operations of Eq. (1) can be executed sequentially, as in software. In hardware, they can be executed simultaneously. We call it "zero-offset" masking, and it will be our case of study in the rest of this article.

### 2.2 Vulnerability of the Masking against 1O-Attacks

It has been reported in [17] that first order DPA could be conducted on masked circuits. As investigated in [18], it happens that the leakage does not come from the registers, but from the combinational parts of the design. This logic is susceptible to produce glitches, whose appearance can be correlated with unmasked data during the internal demasking of the variables.

In this article, we reduce the number of glitches by confining the sensitive combinational logic in ROMs. The same approach has already been suggested in other papers, such as [9, §IV.1] Although this is not formally a guarantee that sensitive glitches disappear, we benefit all the same from the low-power design of the memory blocks that suppresses most of the non-functional activity.

For the proposed countermeasure to be evaluated clearly, we focus the rest of the article on the protection of registers: we assume a toggle count leakage

model (*aka* Hamming distance model), and we consider only attacks targeting this model.

### 2.3 Vulnerability of the Masking against 2O-Attacks

Implementations were studied to thwart attacks of high order, as that of Mehdi-Laurent Akkar [3] which uses constant masks. However to obtain an important robustness the price to be paid is a strong increase of the complexity. As illustration, it has been demonstrated by Jiqiang Lv in [13] that the DES algorithm requires at least three different masks and six additional S-boxes for every S-box to be resistant against high order attacks using this method. Another method, such as that used by François-Xavier Standaert *et al.* [23], consists in recomputing a new mask in every iteration at the same time as S-box, as Fig. 1 shows for the DES algorithm. The masked variable $x \oplus m$ of the register $R$ is associated in every round with a new mask $m$ stemming from the register $M$. So at the end of a round the variable $x \oplus m$ is transformed in $S'(x) \oplus m$ and the new mask $m'$ which is calculated according to $m$ and $x \oplus m$ by means of new S-box $S'$. This method offers a good compromise of complexity because it associates only a new S-box $S'$ with every existing S-box $S$.

This implementation remains subject to the 2O-DPA of Éric Peeters [20]. The figure 2 represents the S-box implementation $S'$ in ROM. For reasons of simplicity the figure disregards the expansion and permutation functions appropriate for the DES algorithm. The so-called "zero-offset" HO-DPA attack of

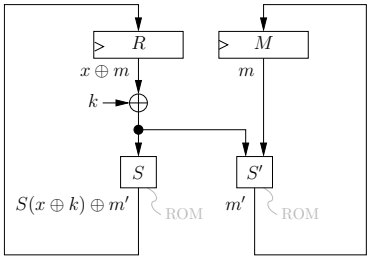



**Fig. 2.** Masked DES using two paths, implemented with ROM.

Éric Peeters [20] concerns variables $x \oplus m$ and $m$ which are stored in $R$ and $M$ registers. The principle consists in studying the distributions of the activity at the register outputs for various values of $x$. In CMOS logic, a model of activity, noted $A$, can be the Hamming distance, noted $HD$, between two consecutive words:

$$\begin{aligned} A(x \oplus m, m) &\doteq HD(x \oplus m, S(x \oplus k) \oplus m') + HD(m, m') \\ &= HW(x \oplus S(x \oplus k) \oplus m \oplus m') + HW(m \oplus m') \\ &= HW(\Delta(x) \oplus \Delta(m)) + HW(\Delta(m)) \,, \end{aligned}$$

where $HW$ corresponds to the Hamming Weight and $\Delta$ is the difference between two consecutive values of a register output:

$$\Delta(x) \doteq x \oplus S(x \oplus k) \text{ and } \Delta(m) \doteq m \oplus m'. \tag{2}$$

If $x$ and $m$ fit on a single bit, the corresponding activity is $2 \cdot HW(\Delta(m))$ if $HW(\Delta(x)) = 0$, whereas if $HW(\Delta(x)) = 1$, the corresponding activity is $HW(\overline{\Delta(m)}) + HW(\Delta(m))$ and is thus constantly equal to 1. The knowledge of the consumption distributions for every $HW(\Delta(x))$ values allows to build the HO-DPA attack by observing the consumption distributions and by comparing them with the predicted activity for a key hypothesis included in $x$.

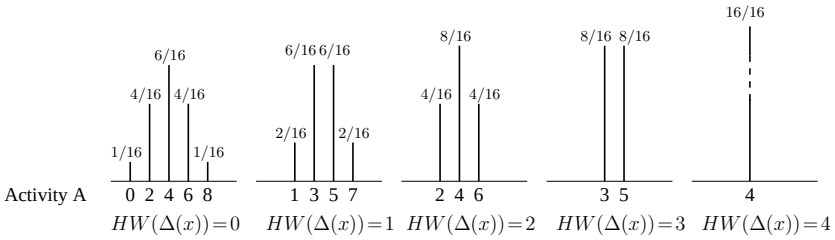



**Fig. 3.** Ideal (*i.e.* noise-free) probability density functions (pdf) corresponding to the five possible values of $HW(\Delta(x))$ without 2O-DPA protection [14].

Considering 4-bit registers, there are five possible distributions depending on the $HW(\Delta(x))$ values. They are shown in Fig. 3. It appears a clear difference between the five distributions, which could be exploited by a HO-DPA attack.

In [20], Éric Peeters proposed an improved higher-order technique to bypass the masking countermeasure. It is based on the efficient use of the statistical distributions of the power consumption described in figure 3 and it consists in computing the maximum likelihood of key guesses. Another alternative is to take advantage of the fact that the distributions showed in figure 3 all have the same mean value and only differ in their variances. This fact allows to understand the origin of previous attacks, as the one in [14], so-called Variance-based Power Attack where it is proposed to compute the difference of variance between the five possible distributions depending on the secret state of the implementation $HW(\Delta(x))$ values. This attack is quite efficient on "zero-offset" implementation and requires a reasonable number of traces (200K) [14]. Moreover, in [15] a novel approach to information-theoretic HO attacks, called the Entropy-based Power Analysis (EPA) was introduced using a weighted sum of conditional entropies as a distinguisher. It is designed to ease the distinguisability between hypotheses on candidate keys by computing the difference of conditional entropies between the distributions. Moreover, a novel approach, Multivariate Mutual Information Analysis MMIA, was proposed in [8]. This attack works in software masking but has never been applied on zero-offset implementations.

Therefore, there is a need for countermeasures thwarting 2O-DPA in particular and HO-DPA in general, by balancing the leakage distributions described in figure 3 so that any partitioning becomes almost indistinguishable whatever the secret state $HW(\Delta(x))$.

## 3 Proposed Masking Method for "Leakage Squeezing"

Indeed, we implement the S-boxes in (synchronous) ROMs of FPGA, which are much less if not totally immune to spurious glitching activity. We have checked that with a standard masking scheme, 2O-DPA succeeds but not 1O-DPA [14]. Also, unlike other initiatives, we do not attempt to add extra masks to increment the order $n$ of resistance against $n^{\text{th}}$-order DPA; our philosophy has been to stick with one sole mask, but to adapt the masking scheme and the leakage function. This approach is deliberately pragmatic and tightly linked to a specific leakage model, namely the "transition count" model, which has been experimentally verified for registers in FPGAs and ASICs. Such a methodology is of high practical interest for practitioners, because some theoretically backed countermeasures have been shown to present vulnerabilities and because most of them are almost impossible to implement in throughput-driven circuits due to excessive overhead.

### 3.1 Masking Principle

The "leakage squeezing" approach is not a countermeasure dedicated only to fight 2O-Attacks (for instance by making the distribution second order indiscernible, but by opening the door to an attack of still higher order). Instead, it consists in making the overall leakage indiscernible in order to reduce the information leakage provided by the countermeasure, thereby anticipating any adversarial strategy. The principle is somehow similar to static power balancing countermeasures (information hiding, with dual-rail for instance [16, Chp. 7]): this methodology is also attack-agnostic.

Following this philosophy, we do not concentrate on a particular characteristic of the squeezed leakages (such as the $n$th momentum) but instead consider a global metric.

The principle consists mainly in making the activity of the register storing the mask $m$ independent from the activity of the register containing the masked variable $x \oplus m$. A second action is to use ROMs for the implementations in order to avoid or at least strongly reduce the glitching activity. The first point is that if the variable $x$ does not influence the consumption distributions for the variable and the mask register, we obtain similar (and ideally identical) distributions for every $HW(\Delta(x))$ values, and as a result it is not possible any more to mount a successful 2O-DPA as that of Peeters [20] or the VPA attack [14].

The similarity between the five consumption distributions can be made by modifying the structure of the mask path without touching the path of the masked variable.

A simple approach consists in modifying the mask $m$ by using a bijective transformation $B$ before storing $B(m)$ in the mask register $M$. It is shown in Fig. 4. Indeed, the presence of $\Delta(m)$ twice in the leakage function, (Eqn. (2)), tends to reduce the effect of the masking countermeasure as the two terms compensate partially so that there remains a residual dependency in $HW(\Delta(x))$. To decorrelate those two terms, we need a Boolean function that implements good confusion, namely an S-box $B$. The activity of the variables $x \oplus m$ and $B(m)$ should be ideally decorrelated. This activity of the registers $R$ and $M$ is expressed by:

$$\begin{aligned} A_B &= HW(x \oplus S(x \oplus k) \oplus m \oplus m') + HW(B(m) \oplus B(m')) \\ &= HW[\Delta(x) \oplus \Delta(m)] + HW[\Delta B(m)]\,. \end{aligned} \tag{3}$$

With the bijection, the leakage (Eqn. (3)) is squeezed because $\Delta(m)$ and $\Delta(B(m))$ do not cancel as easily as previously.

The bijection and its inverse can be implemented as internal encodings in a table. The figure 4 describes a hardware architecture, where the registers $R$ and $M$ are protected against Hamming distance attack via a squeezing of their leakage. The rest of the schematic is combinational logic : either gates or memory blocks.

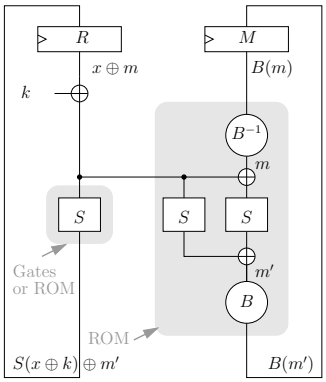



**Fig. 4.** Mask path with bijections for "Leakage squeezing".

By choosing the appropriate bijection $B$ we can obtain very close distributions which should not allow the adversary to take advantage of the residual mismatches.

### 3.2 Formal Security Assessment and Motivation for some Bijections

In order to evaluate the information revealed by the squeezing countermeasure, we follow the information theoretic approach suggested in [22]. Namely we com-

pute the mutual information between the sensitive variable $k$ and the leakage function $A_B$ of Eqn. (3).

In our experiments, we will consequently assume that the leakage is affected by some Gaussian noise. Thus, the physical observations are represented by a variable : $O = A_B + \mathcal{N}(0, \sigma^2)$.

For comparison purposes, we compute the mutual information value ($I(k; O)$) as proposed in [22] for several bijection functions. The lower the mutual information, the better the countermeasure.

The mutual information is represented in figure 5 for the different bijections, in function of the signal-to-noise ratio ($SNR = 10 \cdot \log_{10} \frac{\epsilon^2}{\sigma^2}$), where $\epsilon$ and $\sigma$ respectively denote the standard deviation of the signal and the noise emanated from the implementation.

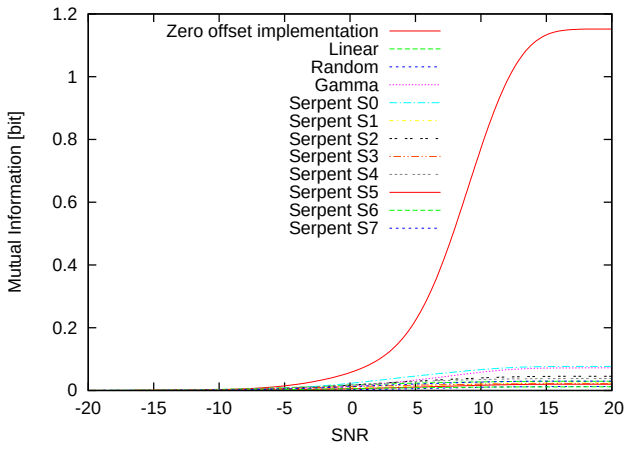



**Fig. 5.** Mutual information for some bijections.

These results demonstrate the information leakage reduction implied by the use of bijections functions. The linear function already decreases significantly the mutual information. Then, the non-linear functions still achieve a better improvement. It appears that Serpent S-boxes are leaking less than the randomly generated bijection or than the Gamma function of Noekeon. This justifies the use of the S-boxes crafted for strong symmetric algorithms.

This first analysis allows us to observe that the gain is high when the leakage squeezing is applied, because the mutual information is almost zero whatever the SNR. On the other hand, these results justify the best choice of the bijection to be used in our implementation. Indeed, the knowledge of this bijection (that can even be made public) is of no help for the attacker since the mask is unknown. Therefore, in all the cases, we assume a partitioning according to $HW(\Delta(x))$, that is independent of $B$.

# 4 Experiments on Masked DES Implementations

In this section, we apply the principle of leakage squeezing introduced in section 3.1 to DES. It requires an adaptation since its round function is more elaborate than $x \mapsto S(x \oplus k)$. Also, it is unrealistic to use 32-bits bijections. Therefore, we show how to split the bijection $B$ (refer to Fig. 4) into smaller bijections. Two implementations are proposed: a ROM based architecture and a simpler structure called "Universal S-box Masking" (USM).

## 4.1 ROM Implementation

For DES we can use eight different bijections[1], denoted $B_1$, one for each S-box. To further protect the new mask $m'$, we compose the DES parts by using external encodings with bijections $B_2$, for instance:

$$\underbrace{B_1^{-1} \circ E \circ S \circ P \circ \texttt{XOR}(L) \circ B_1}_{\text{ROM}} = \underbrace{B_1^{-1} \circ E \circ S \circ B_2}_{\text{ROM}} \circ P \circ \underbrace{B_2^{-1} \circ \texttt{XOR}(L) \circ B_1}_{\text{LUT network}}, \tag{4}$$

where $B_1$ and $B_2$ are 4-bit bijections, $E$, $S$, $P$ and $\texttt{XOR}(L)$, respectively the Expansion, S-Box, Permutation and Left part recombination of the DES algorithm. As the expansion $E$ needs 6 bits, specific care has to be taken for the 4-bit bijections. This point is discussed further.

This principle of internal encodings has already been proposed by Chow *et al.* in [6] in the context of white box cryptography. This protection method has already been attacked for the DES and for the AES. However these attacks should not apply for the mask path as it is random and consequently no values can be imposed at the table inputs.

The general ROM implementation is given in figure 6. With respect to figure 4, the intermediate data (*e.g.* Sboxes output) have been protected by the same strategy, so as to provide a seamless "squeezing" throughout the combinational logic.

The bijection $B_2$ is constrained to be a *xor* operation with a constant, as the permutation $P$ on 32 bits causes the ROM output bits to be split for the next round. The implementation of the mixing $L$ with the left part can be done by a Look Up Table (LUT) network in FPGAs rather than a ROM in order to reduce the complexity. This requires that the bijections are a set of three 2-bit bijections to take advantage of LUT having 4 inputs (LUT4) in FPGAs, or two 3-bit bijections if LUT6 are available.

If we compare this implementation to the one proposed in [23] and described in figure 1, we have the same ROM complexity which is of eight $2^{12}$ words of four bits.

[1] The same bijection can be reused eight times without compromising the security.

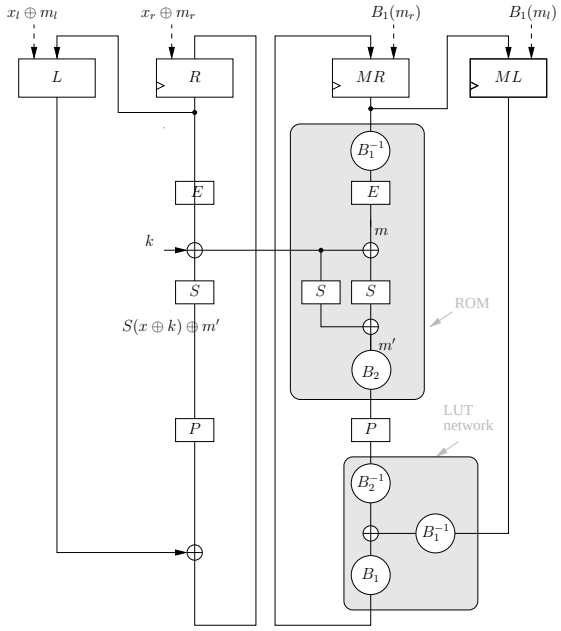



**Fig. 6.** Leakage squeezing of DES with a masked ROM implementation.

### 4.2 USM Implementation

The ROM implementation can be replaced by a more simple structure which is the Universal S-box Masking (USM) studied in [14]. This implementation presents some security weaknesses as discussed in [14]; the weakness can be exploited successfully by a classical CPA. If we apply function compositions as for Eq. (4) with new bijection encodings, the CPA and second order DPA attacks could be thwarted. Figure 7 illustrates the mask path of DES with USM implementation taking advantage of the "leakage squeezing" method. It is made up four stages which can be protected by using bijections $B_1$, $B_2$, $B_3$ and $B_4$. All the bijection are on four bits except $B_2$ which is on six bits.

Every stage can be implemented by a set of LUT networks or a ROM.

The bijection $B_4$ is constrained to be a *xor* operation with a constant, as the permutation $P$ on 32 bits causes the output bits to be split.

All the stages can be implemented with a LUT network based on sets of 2-bit bijections. The second stage with the S-box could also be implemented in a small $64 \times 4$ ROM.

In this stage the mask $m$ is xored with the masked data $x \oplus k \oplus m$ and the expansion $E$ is performed as 6 bits of masks are considered.

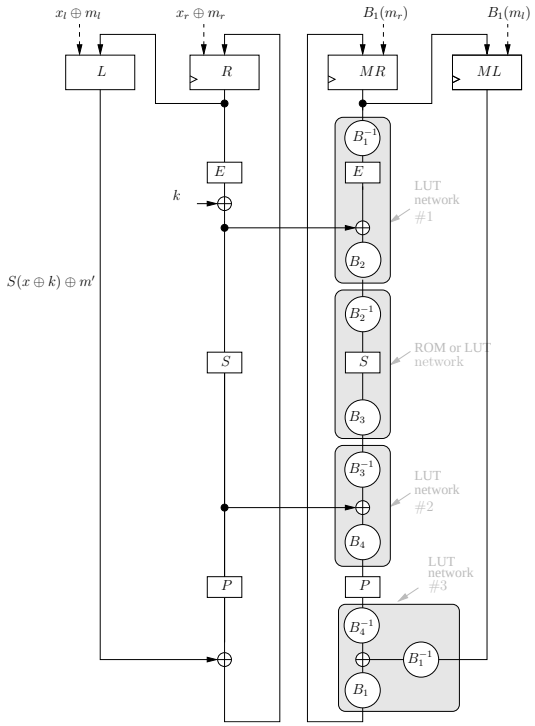



**Fig. 7.** Leakage squeezing of DES with a masked USM implementation.

### 4.3 Complexity and Throughput Results

The proposed implementations have been tested in a StratixII FPGA which is based on Adaptative LUT Module (ALM) cell. They have been compared with non protected DES, masked ROM and masked USM implementations without any leakage squeezing.

The table 1 summarizes the memories needed for each implementation and the estimated throughput.

These results show that the leakage squeezing method on hardware implementations has little impact on complexity and speed compared with software implementation against HO-DPA [21]. Moreover the USM implementation is particularly efficient as it avoids the use of large ROMs while keeping a high throughput.

In order to validate our implementations, we conduct in the next sections an evaluation of the leakages resulting from the leakage squeezing implementation. In [22], a theoretical framework was consequently introduced and suggests analyzing side-channel attacks with a combination of information theoretic and security metrics. These metrics respectively aim at evaluating the amount of in-

**Table 1.** Complexity and speed results. "l. s." denotes the "leakage squeezing" countermeasure.

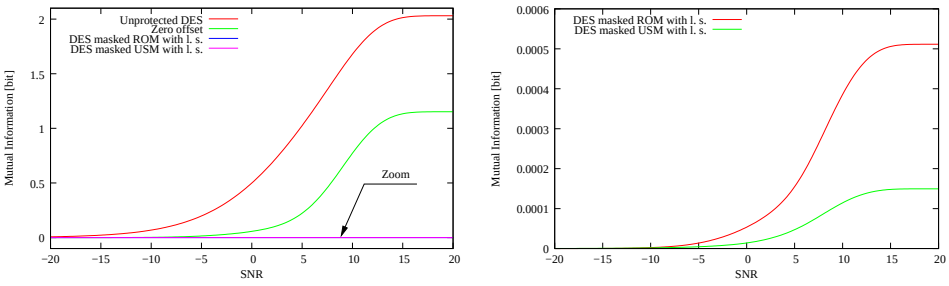
| Implementation | ALMs | Block mem-ory [bit] | M4Ks | Throughput [Mbit/s] |
|---|---|---|---|---|
| **Unprotected DES (*reference*)** | 276 | 0 | 0 | 929.4 |
| **DES masked USM** | 447 | 0 | 0 | 689.1 |
| **DES masked ROM** | 366 | 131072 | 32 | 398.4 |
| **DES masked ROM with l. s.** | 408 | 131072 | 32 | 320.8 |
| **DES masked USM with l. s.** | 488 | 0 | 0 | 582.8 |

formation provided by a leaking implementation and the possibility to turn this information into a successful key recovery.

### 4.4 Information-Theoretic Evaluation of the Proposed Solutions

As it was suggested in [22], we computed the mutual information between the secret state $k$ and the leakage function in the Hamming weight model with Gaussian noise for our two implementations and the others for comparison purposes.

Figure 8 (a) shows the mutual information values obtained for each kind of leakage with respect to an increasing noise standard deviation over [0.1, 10] (*i.e.* an increasing SNR over [-20 , 20])




(a) Comparison (b) Zoom

**Fig. 8.** Mutual information metric computed on several DES implementations.

These results demonstrate the information leakage reduction implied by the use of the leakage squeezing technique. As expected, the two implementations based on leakage squeezing leak less information than the zero offset implementation and the unprotected DES for all SNRs. The somewhat surprising conclusion of our experiments is that the mutual information is almost zero which proves the robustness of this technique. In figure 8 (b), we zoom on the evolution of

the mutual information in the case of the implementations based on the leakage squeezing technique in order to make a comparison between them.

We clearly see that when the SNR increases the mutual information for the USM implementation tend asymptotically to the value $1e^{-4}$ bit and remains below the mutual information leaked in the case of the ROM implementation (*i.e.* $5e^{-4}$) and then is the most robust implementation.

We can explain this results by the fact that the leakage squeezing techniques, (*i.e.* by applying bijection), aim at balancing the leakage distributions described in figure 3 so that any partitioning becomes almost distinguishable whatever the secret state $HW(\Delta(x))$ and as a consequence the information leakage is reduced. We showed in figure 9 the five possible values of $HW(\Delta(x))$ for the USM implementation with the squeezing leakage technique using the sixteenth serpent S-Box (*i.e.* proved to be the most appropriate bijection, see subsection 3.2). These distributions are clearly identical.

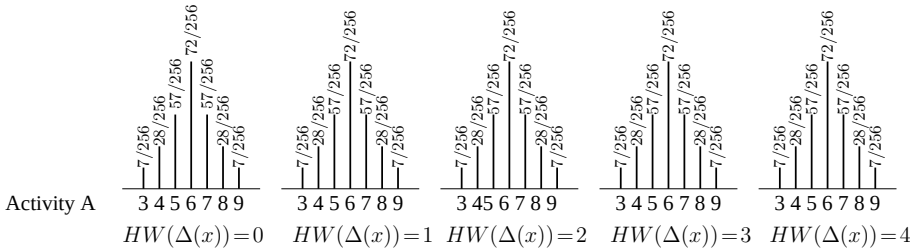



**Fig. 9.** Probability density functions (pdf) corresponding to the five possible values of $HW(\Delta(x))$ with Leakage Squeezing protection.

### 4.5 Evaluation of the Implementations against 2O-Attacks

After the information theoretic evaluation, the second step to evaluate the robustness of a leaking device is the security evaluation using various distinguishers to see how the information leakages translate into success rate under different assumptions.

First, we applied several side-channel distinguishers to leakage measurements simulated in the Hamming weight model with Gaussian noise. We not only applied (HO)-DPA, but also other kinds of attacks, namely MMIA. We chose to test these three side channel distinguishers against different kinds of masking, firstly because they are the most widely used in the literature, and secondly because they represent a brand spectrum of adversary capabilities.

Afterward, we performed these attacks against real power consumption measurements of our FPGA implementations in order to check them in a real-world context.

For each scenario, we acquired a set of 25,000 power consumption traces using random masks and plaintexts. We performed the first order success rate as in [22].

We showed in figure 10 our experimental results also for these attacks on the "zero offset" hardware implementation used here for comparison purposes with our hardware solution based on the leakage squeezing technique.

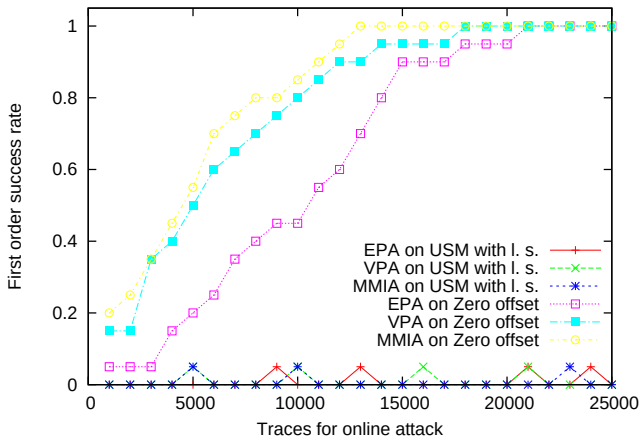



**Fig. 10.** First order success rate of 3 distinguishers, FPGA implementation.

We can see that the attacks based on various distinguishers perform well in the case of the "zero offset" implementation. About 5,000 traces suffice to achieve a success rate of 50% and starting from about 13,000 traces the MMIA attack reveals the correct key with success rate of 100%. The VPA and EPA attack perform well also. For the EPA, the success rates stay well above 50% even when using 11,000 measurements, but eventually reaches success rate of 95% using 18,000 traces.

For our proposed countermeasure, the attacks perform worse. The success rates stay under 10% even when using 25,000 measurements.

We conclude that the experiments on a real circuit shows the evidence of benefit of our countermeasure since it leaks little information which are not exploited by the adversary to mount a successful attack.

## 5 Conclusion and Perspectives

Second order DPA attacks not only allow to theoretically invalidate some countermeasures, but can break them in practice. We presented in this paper a method called "leakage squeezing" which aims at balancing the power consumption distribution on hardware masked implementations. This method consists in

using bijective encodings composed of functional operations and implemented in ROMs or LUT networks. Two implementations have been proposed and evaluated. They provide a great robustness against 2O-DPA (VPA, EPA) and MMIA as none of the subkeys have been guessed using 25k traces. The robustness is corroborated by an information theoretic analysis of the leakage. Moreover the performances decrease in terms of complexity and speed are very limited, which is particularly true for the USM implementation which does not require large memories.

The main perspective of this work is to compare our countermeasure based fundamentally on Boolean masking with others solutions such the affine masking [7] scheme which also provides good performance-security against HO attacks.

## Acknowledgment

This work has been supported by the french Agency "Agence Nationale de la Recherche" in the frame of the SECRESOC project ANR-09-SEGI-013.

## References

1. D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi. The EM Side-Channel(s). In *CHES*, volume 2523 of *LNCS*, pages 29–45. Springer, 2002.
2. M.-L. Akkar and C. Giraud. An Implementation of DES and AES Secure against Some Attacks. In LNCS, editor, *Proceedings of CHES'01*, volume 2162 of *LNCS*, pages 309–318. Springer, May 2001. Paris, France.
3. M.-L. Akkar and L. Goubin. A generic protection against High-order differential Power Analysis. In LNCS, editor, *Proceedings of FSE'03*, volume 2887 of *LNCS*, pages 192–205. Springer, 2003. Berlin, Germany.
4. S. Chari, C. S. Jutla, J. R. Rao, and P. Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In *CRYPTO*, volume 1666 of *LNCS*. Springer, August 15-19 1999. Santa Barbara, CA, USA. ISBN: 3-540-66347-9.
5. S. Chari, J. R. Rao, and P. Rohatgi. Template Attacks. In *CHES*, volume 2523 of *LNCS*, pages 13–28. Springer, August 2002. San Francisco Bay (Redwood City), USA.
6. S. Chow, P. A. Eisen, H. Johnson, and P. C. van Oorschot. A White-Box DES Implementation for DRM Applications. In *Security and Privacy in Digital Rights Management, ACM CCS-9 Workshop, DRM 2002*, volume 2696 of *LNCS*, pages 1–15. Springer, 2002.
7. G. Fumaroli, A. Martinelli, E. Prouff, and M. Rivain. Affine masking against higher-order side channel analysis. Cryptology ePrint Archive, Report 2010/523, 2010. `http://eprint.iacr.org/2010/523`. To be published at SAC'2010 (PDF).
8. B. Gierlichs, L. Batina, B. Preneel, and I. Verbauwhede. Revisiting Higher-Order DPA Attacks: Multivariate Mutual Information Analysis. In *CT-RSA*, volume 5985 of *LNCS*, pages 221–234. Springer, March 1-5 2010. San Francisco, CA, USA.
9. C. Kim, M. Schläffer, and S. Moon. Differential Side Channel Analysis Attacks on FPGA Implementations of ARIA. *ETRI Journal*, 30(2):315–325, 2008. DOI: 10.4218/etrij.08.0107.0167.

10. P. C. Kocher, J. Jaffe, and B. Jun. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In *Proceedings of CRYPTO'96*, volume 1109 of *LNCS*, pages 104–113. Springer-Verlag, 1996. (PDF).
11. P. C. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In *CRYPTO*, volume 1666 of *LNCS*, pages pp 388–397. Springer, 1999.
12. Louis Goubin and Jacques Patarin. DES and Differential Power Analysis - The "Duplication" Method, 1999.
13. J. Lv and Y. Han. Enhanced DES implementation secure against differential power analysis in smart-cards. In *Information Security and Privacy, 10th Australasian Conference*, volume 3574 of *LNCS*, pages 195–206, Brisbane, Australia, July 2005. Springer-Verlag.
14. H. Maghrebi, J.-L. Danger, F. Flament, and S. Guilley. Evaluation of Countermeasures Implementation Based on Boolean Masking to Thwart First and Second Order Side-Channel Attacks. In *SCS*, IEEE, pages 1–6, November 6–8 2009. Jerba, Tunisia. Complete version online: `http://hal.archives-ouvertes.fr/hal-00425523/en/`. DOI: 10.1109/ICSCS.2009.5412597.
15. H. Maghrebi, S. Guilley, J.-L. Danger, and F. Flament. Entropy-based Power Attack. In *HOST*, IEEE Computer Society, pages 1–6, June 13-14 2010. Anaheim Convention Center, Anaheim, CA, USA. DOI: 10.1109/HST.2010.5513124.
16. S. Mangard, E. Oswald, and T. Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer, December 2006. ISBN 0-387-30857-1, `http://www.dpabook.org/`.
17. S. Mangard, N. Pramstaller, and E. Oswald. Successfully Attacking Masked AES Hardware Implementations. In LNCS, editor, *Proceedings of CHES'05*, volume 3659 of *LNCS*, pages 157–171. Springer, August 29 – September 1 2005. Edinburgh, Scotland, UK.
18. S. Mangard and K. Schramm. Pinpointing the Side-Channel Leakage of Masked AES Hardware Implementations. In *CHES*, volume 4249 of *LNCS*, pages 76–90. Springer, October 10-13 2006. Yokohama, Japan.
19. T. S. Messerges. Using second-Order Power Analysis to Attack DPA resistant Software. In *CHES*, volume 1965 of *LNCS*, pages 71–77. Springer, August 17-18 2000. Worcester, MA, USA.
20. r. Peeters, F.-X. Standaert, N. Donckers, and J.-J. Quisquater. Improved Higher-Order Side-Channel Attacks With FPGA Experiments. In *CHES*, volume 3659 of *LNCS*, pages 309–323. Springer-Verlag, 2005. Edinburgh, UK.
21. M. Rivain, E. Prouff, and J. Doget. Higher-Order Masking and Shuffling for Software Implementations of Block Ciphers. In *CHES*, volume 5747 of *Lecture Notes in Computer Science*, pages 171–188. Springer, September 6-9 2009. Lausanne, Switzerland.
22. F.-X. Standaert, T. Malkin, and M. Yung. A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In *EUROCRYPT*, volume 5479 of *LNCS*, pages 443–461. Springer, April 26-30 2009. Cologne, Germany.
23. F.-X. Standaert, G. Rouvroy, and J.-J. Quisquater. FPGA Implementations of the DES and Triple-DES Masked Against Power Analysis Attacks. In *proceedings of FPL 2006*. IEEE, August 2006. Madrid, Spain.
24. J. Waddle and D. Wagner. Towards Efficient Second-Order Power Analysis. In *CHES*, volume 3156 of *LNCS*, pages 1–15. Springer, 2004. Cambridge, MA, USA. PDF.