

Evaluation of Countermeasures Implementations Based on Boolean Masking to Thwart Side-Channel Attacks

Housseem Maghrebi, Jean-Luc Danger, Florent Flament, Sylvain Guilley, Laurent Sauvage
Département COMELEC
Institut TELECOM, TELECOM ParisTech, CNRS LTCI (UMR 5141), 46 rue Barrault,
75 634 Paris Cedex, France.
jean-luc.danger@telecom-paristech.fr

Abstract—This paper presents hardware implementations of a DES cryptoprocessor with masking countermeasures and their evaluation against side-channel attacks (SCAs) in FPGAs. The masking protection has been mainly studied from a theoretical viewpoint without any thorough test in a noisy real world design. In this study the masking countermeasure is tested with first-order and higher-order SCAs on a fully-fledged DES. Beside a classical implementation of the DES substitution boxes (S-Boxes) a simple structure called Universal Substitution boxes with Masking (USM) is proposed. It meets the constraint of low complexity as state-of-the-art masked S-Boxes are mostly built from large look-up tables or complex calculations with combinatorial logic gates. However attacks on USM has underlined some security weaknesses. ROM masked implementation exhibits greater robustness as it cannot be attacked with first-order DPA. Nevertheless any masking implementation remains sensitive to Higher-Order Differential Power Analysis (HO-DPA) as shown in a proposed attack, nicknamed VPA. This attack is based on a variance analysis of the observed power consumption and it clearly shows the vulnerabilities of masking countermeasures.

Index Terms—Side-channel attack, masking countermeasure, Higher-Order DPA, Variance-based Power Attack (VPA), FPGA.

I. INTRODUCTION

Amongst the two major countermeasures, hiding and masking, against side-channel attacks (SCAs), the latter is certainly the less complex to implement when applied at algorithmic level. In this last case, masking is performed on internal variables which are transformed into shares of masked variables and the mask itself. Software and hardware implementations can both take advantage of this countermeasure which has been largely studied [1], [2], [4], [7]. The hardware design consists in modifying the architecture at register transfer level which is very convenient as there is no extra-work at the place and route stage of the design flow. Masking implementations in hardware could lead to rather complex architectures in terms of number of operations or memories used as look-up tables (LUT) [8], [9]. This motivates the study of generic structures as the Universal S-Boxes with Masking (USM) which is proposed in this paper and which is by far less complex than ROM implementation.

The Differential Power Analysis (DPA) is generally based on activity prediction at the register stage. On masking implementation, the first order attacks that target registers fail in practice, which is in accordance with the theory [5]. Moreover by attacking combinatorial logic at the beginning of logic cones, some nets can be also attacked and exhibit some weaknesses. This is the goal of the “shallow attack” which is presented in this paper. The robustness of masking at word level could be sensitive to higher-order attacks [6], [9], [14] which takes advantage of multiple correlated variables activity. However the evaluations performed to prove the HO-DPA are incomplete as they are based on simulations or on a limited implementation of the algorithm. The proposed second-order attack studied here is called the Variance Power Analysis (VPA). It is carried out on a DES coprocessor which is part of a SoC programmed in an FPGA.

The paper is organized as follows. Sec. II presents the state-of-the-art of techniques for masking. The description of the USM and ROM architecture is provided in Sec. III along with the robustness evaluation of both countermeasures against the first-order attack. This section includes the proposed shallow attack. Sec. IV presents the second-order DPA with the proposed variance test. It provides experimental results against the masked ROM implementation. Finally, Sec. V concludes the paper and opens some perspectives.

II. STATE-OF-THE-ART

Power consumption analyses are based on the fact that a cryptographic device’s power consumption is correlated to its internal data it processes. Therefore the private key being manipulated by the device during an encryption can be guessed by analyzing passively the device’s power consumption.

The idea of the Boolean masking countermeasure [2], [4] is to mask the sensitive data by a XOR operation with a random word, in order to avoid the correlation between the cryptographic device’s power consumption and the data being processed. The sensitive data is then neither directly manipulated into the device nor stored unmasked.

The implementation of masking is simple when the function

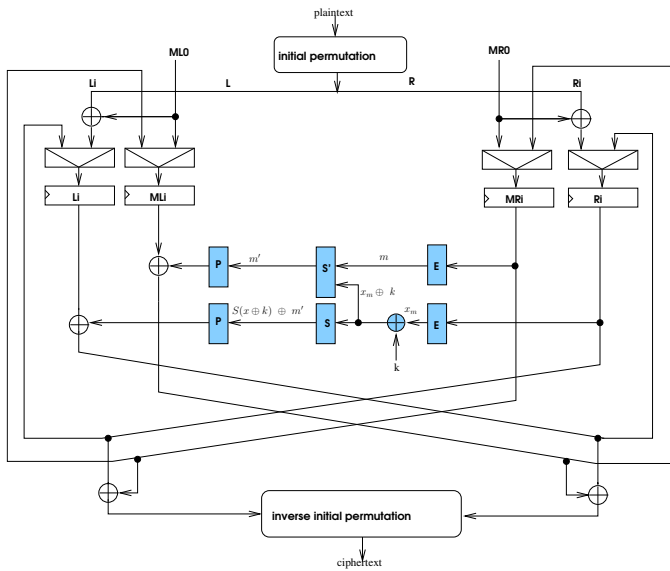


Fig. 1. Masked DES datapath.

f to be protected has the following linearity property w.r.t. group law θ :

$$\forall x, m, \quad f(x_m) = f(x \theta m) = f(x) \theta f(m). \quad (1)$$

The value of $f(x)$ can be reconstructed from the application of f on the masked variable x_m and on the mask m , hence the computation of $f(x)$ can be extracted at the very end of the algorithm. This avoids a direct leakage of information as $x \theta m$ and m are decorrelated with x . However non-linear operations such as substitution boxes, which are used in symmetrical cryptographic algorithms, do not respect this property. Hence masking implementations of algorithms using non-linear operations have to be customized to produce the same result as the one computed by their corresponding unmasked implementation.

A solution in hardware consists in using a two-path implementation, one for the masked variable and one for the mask itself, as proposed in [13] on a Data Encryption Standard (DES) example illustrated in Fig. 1. This algorithmic masking associates a mask ML, MR to the plaintext L, R .

At each round 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 another set of functions S' , defined as:

$$S(x \oplus k \oplus m) = S(x \oplus k) \oplus m', \quad (2)$$

$$S'(x \oplus k, m) = m'. \quad (3)$$

The S' function can be obtained using the corresponding S function twice and two XOR boxes.

S requires a 2^n words ROM whereas S' needs a 2^{2n} words ROM. For AES, masking can take advantage of the fact that the substitution boxes are calculated by using the inverse in $GF(2^8)$ as proposed in [1]. However this implementation is

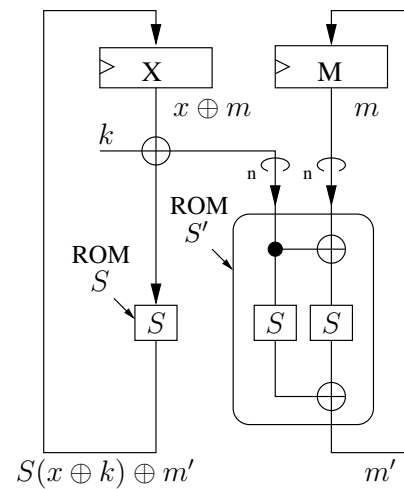


Fig. 2. Masked DES using two paths, implemented with ROMs.

very sensitive to zero-value attack [3] because the power consumption of the value 0 is never masked when multiplicative masking is used. Improvements have been proposed in [8] with a slight increase of complexity as it considers additive masking in sub-fields such that the zero-value attack is ineffective.

III. EVALUATION OF IMPLEMENTATIONS AGAINST FIRST-ORDER ATTACK

A. USM Implementation

The proposed Universal S-Boxes Masking (USM) implementation aims at reducing the size overhead needed by the naive masking countermeasure implementation.

The main idea is to replace large LUTs needed to compute some internal data by smaller logical blocks. More precisely, each non-linear function S of n -bit input words induces in the masked implementation a supplementary non-linear function S' of $2n$ -bit input words. Such non-linear functions can be implemented as LUTs into a ROM. The USM masked implementation avoids the use of such $2n$ -bit input words LUTs. The construction is illustrated on the Fig. 3 and shows that the S' function takes about the same size as the S function, thus having a masked implementation of the cryptographic algorithm about twice larger than the unprotected implementation. According to the previous section, the ROM masked implementation of the DES algorithm is about 65 times larger than the unprotected one, and the ROM masked implementation of the AES is about 257 times larger than the unprotected AES implementation.

When looking carefully at the architecture of the USM implementation in Fig. 3, we can see that the sensible data $x \oplus k$ circulates unprotected on the net between the xor and the S-Box on the right branch of our design. We made the implicit assumption that unprotected data on combinatorial nets would not be exploitable by a power consumption analysis, because such activity would not be synchronized enough as in the case of registers activity, cadenced by a global clock. Unfortunately,

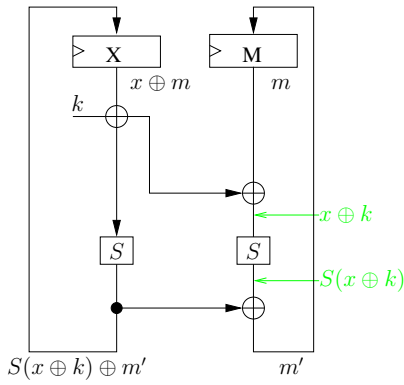


Fig. 3. Universal S-Box Masking implementation of a S-Box, with transiently unmasked values highlighted in green.

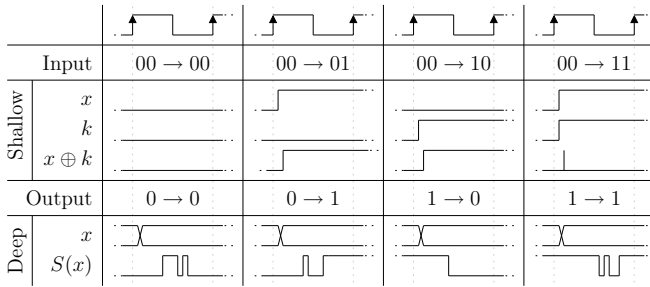


Fig. 4. Shallow versus deep logic typical activity.

as it will be shown in the next section, this assumption does not hold.

B. The Shallow Attack

Beside the attack on a targeted register, we devised a new attack called *Shallow Attack*. Whereas a classical power analysis correlates a device's power consumption to data stored into registers, the *Shallow Attack* is to correlate a device's power consumption with data on combinatorial nets.

In Fig. 4, two hypothetical situations of nets are depicted. The upper one corresponds to a shallow XOR gate, driven by two isochronous registers. The activity of the XOR gate is correlated to the input transitions. In the case $00 \rightarrow 00$, the XOR gate does not consume, since CMOS is a static technology. In any case $00 \rightarrow \{01, 10\}$, the gate definitely consumes. Finally, the most backend-dependant transition is $00 \rightarrow 11$, where the logical output value does not change, but the gate still consumes. In summary, the leakage model is not as simple as a Hamming distance between the inputs values, however it remains understandable. On the contrary, the situation of a deep net is illustrated in the bottom half of Fig. 4. The glitching activity of the output $S(x)$ is so complex that predicting the leakage becomes very difficult.

For this reason, the shallow XOR gates that implement the key mixing are good targets for correlation power analysis. Moreover, the relative isochronism in successive executions and intra-bit synchronization can make attacks more powerful.

C. Breaking the USM Implementation

The *Shallow Attack* previously described seems typically appropriate to extract sensible information in the case of an USM masked implementation of a cryptographic algorithm.

As experimental results, we have been able to break both the unprotected DES implementation as well as the USM masked DES implementation with the *Shallow Attack*. As we could have predicted, the classical Differential Power Analysis proves on some registers of the unprotected DES implementation is more efficient than the *Shallow Attack*. This is due to the good properties of the registers power consumption (synchronization of data and efficient power consumption model).

Moreover, the key used on the USM DES implementation can be recovered by a classical CPA. The explanation of that phenomenon follows: Let x be the plain message known by the attacker, k_1 and k_2 be respectively the round keys used for the first and second DES round, f be the round function computed during each DES round. We then focus on the activity of the right-hand side 32 bits X register at the first DES round and assume that register's power consumption is proportional to the amount of transitions occurring in it.

Considering an unprotected DES implementation, the activity A at the X register output can be expressed by the Hamming distance of the two consecutive values when switching to the first round:

$$A = HW [x \oplus f(x \oplus k_1)], \quad (4)$$

where HW represents the Hamming Weight function. It is exploited by a Correlation Power Analysis to retrieve the key after making 64 key hypotheses for each S-Box.

On the USM masked implementation of the DES algorithm, the data is always masked when stored into the X register as shown in Fig. 3. The activity A at the first round is then:

$$A = HW [(x \oplus m_1) \oplus (f(x \oplus k_1) \oplus m_2)]. \quad (5)$$

As m_1 and m_2 are random data, one cannot guess the transitions of the register. However, the particularity of the USM implementation makes that the data is unmasked ($x \oplus k$) just before entering into the S-Box. The activity A of this particular net at the first round transition is:

$$\begin{aligned} A &= HW [(x \oplus k_1) \oplus (f(x \oplus k_1) \oplus k_2)] \\ &= HW [(x \oplus f(x \oplus k_1)) \oplus (k_1 \oplus k_2)]. \end{aligned} \quad (6)$$

This activity is very close to that of a non-protected implementation expressed in Eq. (4). The added term $(k_1 \oplus k_2)$ is a constant which implies more calculation in the shallow attack as it is necessary to consider 6 bits of k_1 and one additional bit of k_2 for every bit of x .

The attack results in term of number of traces to disclose the key for each S-Box are summarized in Tab. I. They prove the robustness improvement obtained with USM but also its limit because both the shallow attack and the DPA attack succeed. The attacks on ROM implementation have failed even with 100,000 traces and are consequently the most robust. The goal of the next section IV is to evaluate this ROM implementation against Higher-Order attacks.

TABLE I
DPA vs SHALLOW ATTACK RESULTS.

(a) DPA on unprotected DES								
S-Box	S1	S2	S3	S4	S5	S6	S7	S8
MTD ¹	2974	2635	997	3317	965	2034	1803	1133
(b) Shallow attack on unprotected DES								
S-Box	S1	S2	S3	S4	S5	S6	S7	S8
MTD ¹	4981	7772	1290	3565	4859	5578	1870	4302
(c) DPA on USM implementation								
S-Box	S1	S2	S3	S4	S5	S6	S7	S8
MTD ¹	20657	43513	11347	11779	16012	23517	94944	23998
(d) Shallow attack on USM implementation								
S-Box	S1	S2	S3	S4	S5	S6	S7	S8
MTD ¹	14009	82392	10137	73913	99975	5287	76725	30802

¹MTD: Measurements To Disclose.

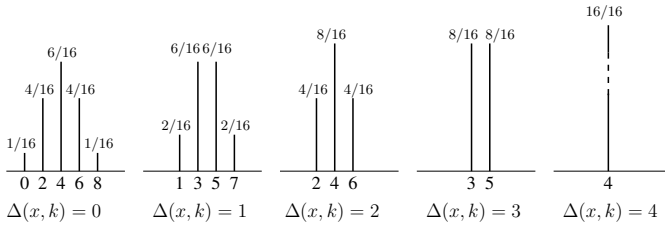


Fig. 5. pdfs corresponding to the five possible $\Delta(x)$.

IV. EVALUATION OF THE ROM IMPLEMENTATION AGAINST HO-DPA

A. ROM Implementation

In the ROM implementation illustrated in Fig 2, the S' ROM is directly implemented into the FPGAs RAM, which is configured in read-only mode. On the one hand the size overhead mentioned in chapter II shows the drawbacks of such implementation for large S-Boxes as in AES. On the other hand this implementation does not reveal any sensible data all along the encryption path. The classical Correlation Power Analysis as well as the Shallow Attack described above, did not allow us to extract a single S-Box subkey used by the cryptoprocessor using up to 100,000 traces.

B. Second-Order DPA

The zero-offset attack on the two paths (Fig. 2) masked implementation proposed in [14] and tested in [9] is a second-order DPA based on analysis of power consumption distributions.

More precisely, the considered probability density function (pdf) of the activity corresponds to those of the combined X and M registers of Fig. 2. The activity of these two registers is expressed by:

$$A = HW[\Delta(x) \oplus \Delta(m)] + HW[\Delta(m)], \quad (7)$$

where Δ is the Hamming distance of a register output:

$$\Delta(x) = x \oplus S(x \oplus k), \quad \Delta(m) = m \oplus m'.$$

Considering 4-bit registers, there are five possible distributions depending on the $\Delta(x)$ values as shown in Fig. 5.

In a real application, the noise coming from other computing blocks and the environment shapes the pdf as a sum

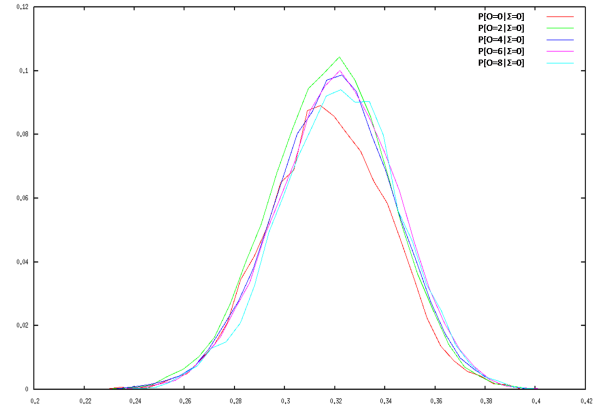


Fig. 6. Real world pdf for $\Delta(x) = 0$.

of Gaussian distributions. We reproduce the attack described in [9] on a fully-fledged masked DES implementation using a ROM in an Altera Stratix II FPGA on the SASEBO-B evaluation board provided by the RCIS [10]. The attack platform is partly described in [11] by performing electromagnetic field acquisition of the decoupling capacitors of the FPGA. This allows to measure the power consumption in a non intrusive way.

The attack algorithm is the following:

- 1) Apply n plaintext messages ($x_i, i \in [1, n]$) and collect n observations of power consumption (traces A_i).
- 2) For each, S-Box make assumptions about the key k_j with $j \in [0, 63]$ and sort the traces according to $\Delta(x)$:

$$\begin{cases} \Delta(k_0) = \Delta(x_0, k_0), \Delta(x_1, k_0), \dots, \Delta(x_n, k_0), \\ \Delta(k_1) = \Delta(x_0, k_1), \Delta(x_1, k_1), \dots, \Delta(x_n, k_1), \\ \dots \\ \Delta(k_{63}) = \Delta(x_0, k_{63}), \Delta(x_1, k_{63}), \dots, \Delta(x_n, k_{63}). \end{cases}$$

- 3) For each, $\Delta(k_j)$ compute the probability:

$$P[A|\Delta(k_j)] = \prod_{i=0}^{n-1} P[A = A_i|\Delta(k_j, x_i)].$$

- 4) Apply the maximum likelihood approach: the correct key corresponds to the maximum probability $P[A|\Delta(k_i)]$.

This HO-DPA attack implementation succeeded on noisy simulated traces, but failed when applied to our real world DES implementation using 200,000 power consumption traces.

Fig. 6 represents the five different pdfs obtained on the FPGA platform with $\Delta(x) = 0$. Compared to the leftmost figure of Fig. 5, it shows that it is hardly possible to discriminate them in a real environment. This explains why the attack proposed in [9] fails as there is a need to perfectly know the mean an variance to calculate $P[A|\Delta(k_i)]$.

C. Proposed Variance-based Power Analysis (VPA)

By choosing a fixed and appropriate (key, message) couple in regard to a specific S-Box, the distribution of power consumption has the same mean, but a different variance as shown in Fig. 5. For instance the variance difference between the pdf for $\Delta(x) = 0$ and $\Delta(x) = 4$ should be enough discriminating even without the knowledge of the exact variance. This leads to the *Variance-based Power Analysis* or VPA which is a kind of partition distinguisher as proposed in [12].

TABLE II
COMPARISON BETWEEN REAL SUBKEYS USED DURING A DES
ENCRYPTION AND SUBKEYS GUESSED BY A VPA.

S-Box	S1	S2	S3	S4
Real Key	0x38	0x0b	0x3b	0x26
Gussed key	0x38	0x0a	0x3b	0x26
S-Box	S5	S6	S7	S8
Real Key	0x00	0x0d	0x19	0x37
Gussed key	0x00	0x0d	0x19	0x37

It is based on the variance computation of the power consumption traces during a time window corresponding to the first DES round, while ciphering random messages. The VPA algorithm is the following:

- 1) Apply n plaintext messages ($x_i, i \in [1, n]$) and collect n observations of power consumption (traces A_i).
- 2) For each S-Box, make assumptions about the key k_j with $j \in [0, 63]$:
 - Sort the traces A_i to get five activity sets $set_l, l \in [0, 4]$, corresponding to the five $\Delta(x_l, k_j)$ possible values.
 - Compute the variance v_l for each set set_l .
 - Compute a VPA indicator F_{k_j} being a linear combination of the variances with weights $w_l: F_{k_j} = \sum_{l=0}^4 w_l \bullet v_l$.
- 3) The correct guess of the key k_j corresponds to $\arg\max_{k_j} F_{k_j}$.

D. Experimental Results

The VPA is carried out on a ROM masked DES implementation. It is tested on 200,000 traces of a masked DES implementation with different weights (w_0, w_1, w_2, w_3, w_4) values. The weights of the F function producing the best results were (0, 1, 0, -1, 0).

Fig. 7 shows the 10 keys having the higher VPA Indicator values for each DES S-Box. These indicators have been normalized in order to make the best key candidate having an indicator value equal to 1. Then for each S-Box, the round subkey guessed by our VPA algorithm is the key corresponding to the highest indicator value (the most left one on the figures).

Amongst the eight DES S-Boxes subkeys used during the first round of our DES implementation, seven of these subkeys have been guessed by the VPA. The table II compares the per S-Box keys guessed by the VPA to the real keys used during the DES encryption.

Note that the indicator illustrated by these results does not use the sets producing maximum and minimum variance observations ($\Delta(x) = 0$ and $\Delta(x) = 4$). The use of such sets decreased the overall performance of the attack. The fact there are four times less traces for $\Delta(x) = 0$ and $\Delta(x) = 4$ than for $\Delta(x) = 1$ and $\Delta(x) = 3$ could explain this behaviour. Attacks with more traces or taking advantage of already cracked subkeys should improve the efficiency.

V. CONCLUSION AND PERSPECTIVES

Even if the masking countermeasures is one of the most efficient manner to thwart SCA, this study shows its vulnerabilities against first-order and second-order DPA. The USM and ROM implementations have been compared in terms of complexity and robustness. The ROM solution is by far the more robust

against first order DPA but leads to higher memory sizes. The second-order DPA as the one already presented in literature is hardly possible on a fully-fledged DES cryptoprocessor. We presented a second-order attack based on variance analysis which is powerful enough to attack a DES implemented in an FPGA. This attack is quite efficient on ROM implementation (7 S-Boxes cracked out of 8) and requires a reasonable number of traces (200K). It could certainly be improved by taking advantage of reliable S-Boxes to refine the weight coefficients in an incremental manner. Another perspective is to link this attack with other partition distinguishers like the Mutual Information Analysis (MIA).

REFERENCES

- [1] 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.
- [2] S. Chari, C. Jutla, J. Rao, and P. Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In *CRYPTO*, volume 1666 of LNCS, August 1999. ISBN: 3-540-66347-9.
- [3] J. D. Golić and C. Tymen. Multiplicative Masking and Power Analysis of AES. In *CHES*, volume 2523, pages 198–212. Springer, 2002. San Francisco, USA.
- [4] L. Goubin and J. Patarin. DES and differential power analysis. In *CHES*, LNCS, pages 158–172. Springer, Aug 1999.
- [5] Johannes Blömer and Jorge Guajardo and Volker Krummel. Provably Secure Masking of AES. In LNCS, editor, *Proceedings of SAC'04*, volume 3357, pages 69–83. Springer, August 2004. Waterloo, Canada.
- [6] T. Messerges. Using second-Order Power Analysis to Attack DPA resistant Software. In *CHES*, volume 1965 of LNCS, pages 71–77. Springer, August 2000.
- [7] T. S. Messerges. Securing the AES Finalists Against Power Analysis Attacks. In *Fast Software Encryption'00*, pages 150–164. Springer-Verlag, April 2000. New York.
- [8] E. Oswald, S. Mangard, N. Pramstaller, and V. Rijmen. A Side-Channel Analysis Resistant Description of the AES S-box. In LNCS, editor, *Proceedings of FSE'05*, volume 3557 of LNCS, pages 413–423. Springer, February 2005. Paris, France.
- [9] É. 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.
- [10] SASEBO board from the Japanese RCIS-AIST: <http://www.rcis.aist.go.jp/special/SASEBO/index-en.html>.
- [11] L. Sauvage, S. Guilley, J.-L. Danger, Y. Mathieu, and M. Nassar. Successful Attack on an FPGA-based Automatically Placed and Routed WDDL+ Crypto Processor. In *DATE, track A4 (Secure embedded implementations)*, April 20–24 2009. Nice, France. Electronic version: <http://hal.archives-ouvertes.fr/hal-00325417/en/>.
- [12] F.-X. Standaert, B. Gierlichs, and I. Verbauwhede. Partition vs. comparison side-channel distinguishers: An empirical evaluation of statistical tests for univariate side-channel attacks against two unprotected cmos devices. In *ICISC*, pages 253–267, 2008.
- [13] 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*, August 2006. Madrid, Spain.
- [14] 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.

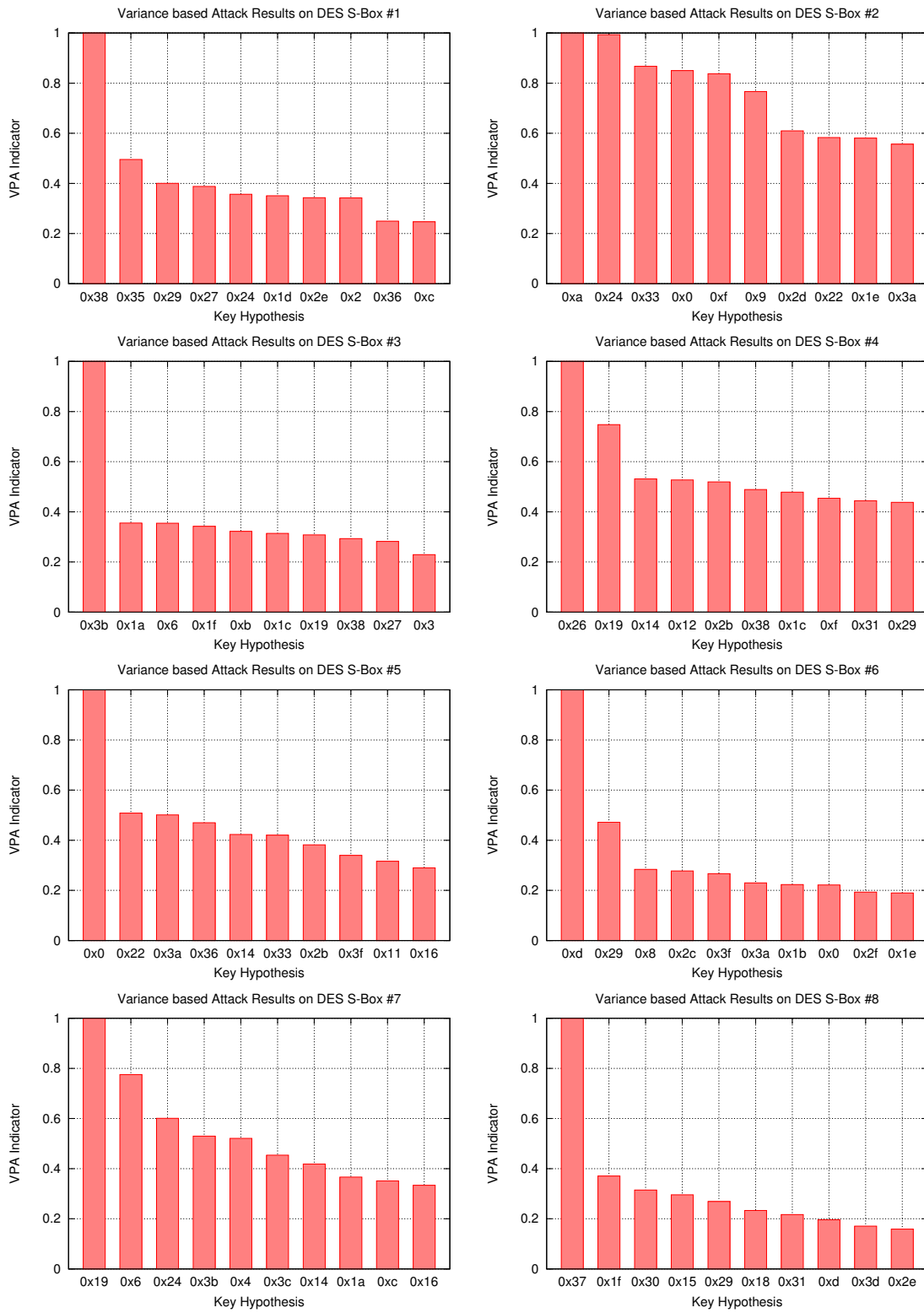


Fig. 7. Variance-based Power Analysis (VPA) results on 200,000 power consumption traces of a ROM masked DES implementation.