Unrolling Cryptographic Circuits: A Simple Countermeasure Against Side-Channel Attacks

Shivam Bhasin, Sylvain Guilley, Laurent Sauvage, Jean-Luc Danger

Unrolling Cryptographic Circuits: A Simple Countermeasure Against Side-Channel Attacks

Shivam Bhasin[1], Sylvain Guilley[1] Laurent Sauvage[1] and Jean-Luc Danger[1].

Institut TELECOM / TELECOM ParisTech, CNRS LTCI (UMR 5141)
Département COMELEC, 46 rue Barrault, 75 634 PARIS Cedex 13, FRANCE.

Abstract. Cryptographic cores are used to protect various devices but their physical implementation can be compromised by observing dynamic circuit emanations in order to derive information about the secrets it conceals. Protection against these attacks, also called side channel attacks are major concern of the cryptographic community. Masking and dual-rail precharge logic are promoted as its countermeasures but each has its own vulnerabilities. In this article, we propose a simple countermeasure which comprises unrolling rounds of a cryptographic algorithm such that multiple rounds are executed per clock cycle. This will require a stronger hypothesis on multiple bits due to deeper diffusion of the key. Results show that it resist against correlation power analysis on Hamming distance and Hamming weight model if the datapath is cleared after each operation. We also evaluated mutual information metric on the design and results show that unrolled DES is less vulnerable.

Keywords: Data encryption standard, side-channel attack, architectural countermeasure, mutual information metric.

1 Introduction

With the generalization of open networks, information society regards security as a critical factor. Modern cryptographic algorithms which ensure security are robust and free from practical cryptanalysis. However, other methods which target the physical implementation of an algorithm can be deployed to break the security. These attacks can be mounted by merely observing or perturbing the targeted system. Observing the activity of the system and its correlation with potential guesses can yield sensible information. Such attacks are better known as Side Channel Attacks (SCAs) [1]. When a device is perturbed such that it yields a non-nominal output, this together with expected output can lead to the secret key. Such attacks are called as Differential Fault Analyses (DFAs) [2]. The passive attacks that consist in observing the chip are difficult to protect

since the chip is even not aware of the attack. Therefore these attacks are considered more critical.

SCAs try to recognize synchronous operations (rounds of cryptographic operations) in the leakage of a device. Then for a chosen round, the leakage is correlated with some guesses to reveal secret information. It is possible to guess some key bits because the value of key remains same for one or a set of synchronous operations. For example if we consider DES, cryptanalysis is impractical as we need a huge number of plaintext or ciphertext. Whereas with power attacks only the power consumption of a few hundreds of encryption are needed to break a non-protected implementation. For instance in DPA contest [3], the participants have demonstrated that DES could be broken in 141 traces in average. Therefore it is essential to protect implementations against SCA.

State of the art countermeasures can be widely classified into two categories *i.e.* information making and information hiding. Masking [4] countermeasures rely on confusing the attacker. A random generated mask is used while running the algorithm such as the mask affects the intermediate states without affecting the end result. Owing to this technique, the attacker observes leakage corresponding to mask and not the actual key bits. Although a nicely masked circuit can resist first order SCA but higher order SCA can still compromise the security of the design

Information hiding as the name suggests hides the information from attacker. The algorithm is implemented in such a way that leakage remains constant irrespective of the computations performed. Dual-rail precharge logic (DPL) [5] is a countermeasure based on information hiding. The principle of this countermeasure is to generate a design equivalent and with opposite behaviour of the target design such that every part of the circuit is perfectly balanced. This way the activity of the doubled design remains constant. There are some countermeasures which combine hiding and masking techniques in order to achieve higher level of security. The major problem of these countermeasures is that it is hard to design a perfectly balanced circuit. Even minor imbalance in space (unbalanced dual nets) or time (early evaluation) can be exploited by sophisticated attacking techniques to reveal sensitive information.

In [6], the effect of pipelining on security is studied. In this article, we investigate the other trend, namely pipelining less; this way, all registers become unpredictable depending on the key (*i.e.* a hypothesis test involves too many key hypotheses). The idea is to implement the design in such a way that the key changes more than once during a synchronous operation. In other words, more than one round of a cryptographic algo-

rithm are executed in one synchronous operation. The rest of the paper is organized as follows. Section 2 explains the theory of the proposed countermeasure. It also details the implementation details of a fully unrolled DES. Section 3 evaluates fully unrolled DES against the iterative DES using correlation power analysis (CPA [7]). Finally, section 4 concludes the paper.

2 Proposed Countermeasure

2.1 Rationale of the Countermeasure

In a cryptographic block product algorithm, data is ciphered by repeating a set of operations with a different key value each time generated from the previous key. These set of operations are called as rounds. The number of rounds are chosen such that linear and differential cryptanalysis are more difficult than an exhaustive key search. Normally, cryptographic circuits are designed to perform either some operations of a round or the whole round in one clock cycle. Thus the value of the key remains the same for one or more clock cycles. The attacker can guess some of the key bits and correlate it with leakage acquired. A correct guess will give a much higher correlation as compared to wrong guesses.

Most of the traditional SCA attacks target the registers where the result of each round is stored. This is because the leakage from the register is high due to its load and the leakage is synchronised to the clock. In combinatorial logic, the leakage is low and spread over time. If the result of a round is stored in the register at the end of each clock cycle, attacker can easily retrieve the subkey by guessing and correlating. Now, if the key is changed more than once during one clock cycle *i.e.* multiple rounds are executed per clock cycle the key used for one round is further diffused deeper into the design and mixed with the second key and so on. Thus exploiting this property we propose to design the cryptographic coprocessors in such a way that it executes multiple rounds in one clock cycle. We call this as unrolling the rounds of the algorithm. Also we define unrolling factor as the number of rounds unrolled. An implementation unrolled twice means that two rounds are performed at every clock cycle. A didactic presentation of the loop unrolling technique is given by Kris Gaj and Pawel Chodowiec in the chapter 10 of [8], along with a discussion about its pros and cons from a performance point of view.

Figure 1(a) shows the architecture of one round of a normal iterative cryptographic algorithm while figure 1(b) shows the architecture of an unrolled cryptographic algorithm. An idea of the difficulty to mount a side

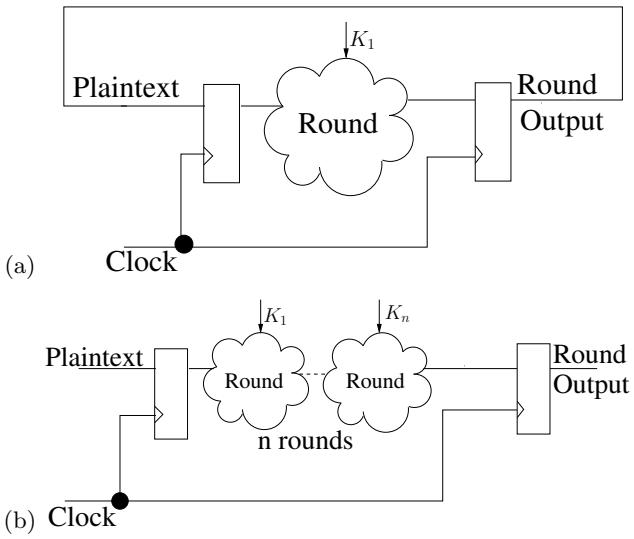


Fig. 1. (a) Architecture of a iterative cryptographic algorithm. (b) Architecture of a fully unrolled cryptographic algorithm.

channel attack on the unrolled version can be estimated from the following discussion. Suppose, we have two implementations of a cryptographic algorithm: one iterative and the other unrolled with an unrolling factor of 2 as shown in fig 1(a) and (b) respectively. Let us see the signal and the noise when the attack is mounted on 1-bit. In the iterative design, the signal will be the sum of the power activity of all the combinatorial gates and flip-flop involved in calculating that bit. The noise shall be sum of power activity of other gates and flip-flops. In the unrolled design, if we implement an attack on 1-bit in the first of the two rounds, the signal will be the power activity of the gates involved only as the result is not memorised. The noise shall be twice the previous value as components are doubled. As explained before the power activity of a combinatorial gates is lesser than the power activity of a register. This results in SNR reduction of more than twice.

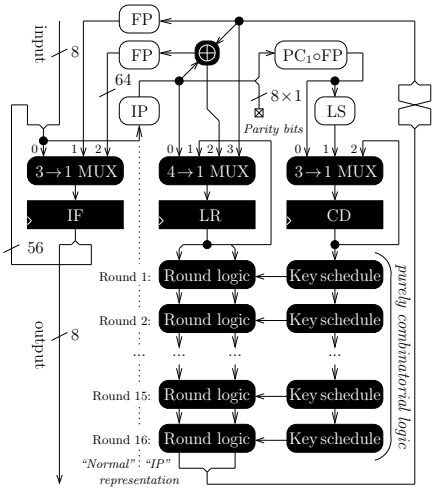
A rough evaluation of the theoretical complexity of this countermeasure in terms of area is given by the unrolling factor. Thus a design unrolled twice will have double the area of its original design as far as combinatorial part is concerned. In terms of performance, the trade-off is almost the same as original design. Unrolling factor of n will multiply the critical path by n times and thus maximum frequency is reduced $1/n$ times. Since n rounds are executed per clock cycle, N/n clock cycles

are needed to execute the whole algorithm where N is the total number of rounds. Thus the throughput is approximately the same for original and unrolled design. The practical results are better than the one described below as some of the unnecessary components like multiplexers are removed while unrolling. Thus the area is less than n times and the operating frequency is more than $1/n$ times. We also point out that the unrolling does not impact the possibility of the encrypting block to be used in any mode of operation (CBC, CFB, OFB, *etc.*).

Fully unrolled DES implementation: An iterative architecture can be made combinatorial, by removing its register transfers occurring during the rounds [9]. In the case of DES, the algorithm combinatorial depth is thus roughly increased by a factor of sixteen, but the registers LR and CD remain frozen during sixteen clock cycles, which makes up for the delay through the gates. The architecture, based on that described in [10], and the floorplan are depicted in Fig. 2(a) and (b). It is a special case of the so called *brutal countermeasure* mentioned in [11], where the "glued blocks" actually make up the entire datapath. The inputs 1 of the LR multiplexer and 2 of the CD multiplexer play the role of enable for the corresponding registers. The key schedule consists in a sequence of pre-computed circular shifts which can be implemented just by switching wires and requires no logic. Such a technique is only valid for certain algorithms like DES and the absence of logic in key schedule avoids leakage. Thus attacks like [12] cannot be mounted anymore.

The synthesizers, in default mode, attempt to fit a timing path into one clock cycle. To synthesize such a design there is need to relax the timing constraints. In the combinatorial DES specific case, the logic driven by LR and CD has time equivalent to sixteen clock cycles to execute. This piece of information cannot be easily inferred, thus user constraints must be set. They basically consist in specifying spare clock cycles for some timing arcs. The timing paths that are concerned thus start at registers LR and CD, plus the Boolean signal originating from the control that tells whether the current operation is a ciphering or a deciphering , where the shifts can be interpreted left or right-wise. The "multi-cycle" constraints listed in Fig. 3 express the fact that outputs of LR and CD are sixteen times slower that the clock and that the signal to decide between ciphering and deciphering is a false timing path. This last path is indeed never critical because the choice between encryption and decryption is not modified during one computation. The key schedule can be implemented by mere routing of wires, with no logic usage. Indeed, every round key in DES is obtained by simply selecting the adequate bits from the 56 bit

(a) (b)



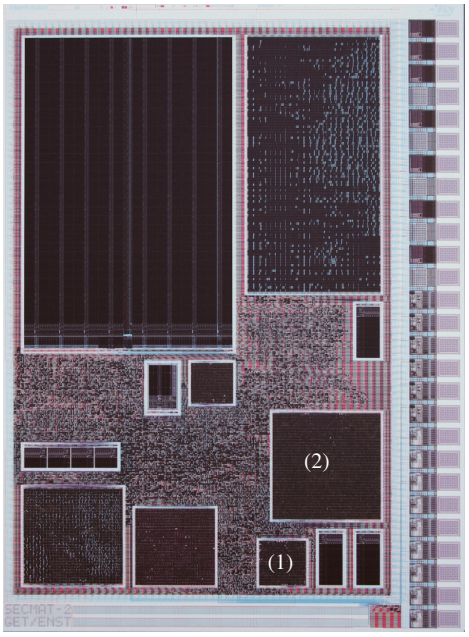


Fig. 2. (a) Unrolled DES Architecture. (b) Floorplan of the ASIC implementing DES iterative (1) and DES unrolled (2).

master key. However, this peculiar property applies to DES only and cannot be generalized for all the cryptographic algorithms.

```
set_current_module des_datapath_combi_wrapper; # Internal constraints
set_current_instance [find -hier -inst I_REG_LR];
# The following constraint (1+15 cycles allowed for the computation)
# concerns the whole bus:
set_cycle_addition -from [get_info [lindex [find -port q] 0] bus] 15;
set_current_instance [find -hier -inst I_REG_CD];
set_cycle_addition -from [get_info [lindex [find -port q] 0] bus] 15;
set_current_module des_datapath_combi; # External constraint
set_false_path -from [find -port sel_left_not_right]; # Encrypt/Decrypt
```

Fig. 3. TCL timing constraints crafted for the "multi-cycle" DES combinatorial datapath synthesis by Cadence `bgx_shell`.

3 Experimental Results

We implemented an iterative DES and a fully unrolled DES on SecMatV2: an academic ASIC for security evaluation of cryptoprocessors implemented in 130 nm technology from STMicroelectronics. The placement constraint used for both modules is that their placement density is 95%. Therefore we found that iterative DES consumes an area of 24787 μm^2 while the unrolled DES consumes an area of 139816 μm^2. The ratio in terms of surface is thus as low as 5.64 lower than expected *i.e.* 16, the unrolling factor which is due to removal of registers, removal of logic involved in the iteration management (multiplexers), round boundaries optimization. Also the key schedule is completely dissolved in mere routing which is a property specific to DES algorithm. In terms of performance for a nominal operating frequency, the iterative DES needs almost 5 times more time for single encryption. However, the operating frequency is not the maximal operating frequency in this case.

The average side-channel curves for one DES encryption are shown in Fig. 4(a) and 4(b) respectively for the iterative reference DES and the combinatorial instance. It clearly appears in Fig. 4 that the variations increase during the encryption.

Side-channel attacks can be roughly divided into two categories. On one hand correlation attacks make the assumption of a known leakage model; several models corresponding to different values of the secret are

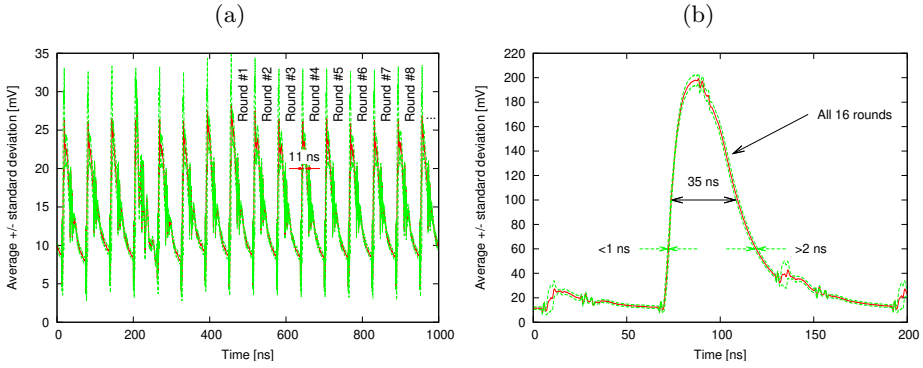


Fig. 4. (a) Sequential iterative DES encryption signature, with the average variation margin, for statistics collected on 10k measurements. (b) Average combinatorial DES encryption signature, with the average variation margin, for statistics collected on 100k measurements.

devised. The model that correlate the better with the concrete measurements discloses the secret. On the other hand, template attacks divide into two steps. The first step is done off-line; it consists in pre-characterizing the circuit in an almost blind fashion, for as many representative values of the message and key inputs. Stochastic attacks are a variant where the pre-characterization is made more simple by injecting some partial knowledge about the target's leakage. The second step is the on-line attack proper. The attacker attempts to recognize the secret by matching measurements obtained from a fixed albeit unknown secret key.

We show that correlation attacks are made very implausible on a fully combinatorial implementation, due to the signal's desynchronization, even in the early rounds (represented in Fig. 5). First of all, we apply the same attack that is successful on the iterative reference implementation. It consists in a correlation of the measurements with the consecutive values of the right datapath register R_0, that leaks $\mathcal{L}(initial : R_0, final : L_0 \oplus f(R_0, K_1)) = |R_0 \oplus L_0 \oplus f(R_0, K_1)|$. The attack results on DES iterative and unrolled are shown in Tab. 1 and 2 respectively . Without any surprise, this attack completely fails on the combinatorial instance of DES, since the targeted transition has disappeared in the unrolled implementation. We would like to emphasize that each time a encryption is done, the datapath should be cleared. This can be done like precharge in DPL or by propagating random values without interference from the key. This is because, if two consecutive computations are done then some correlation can be found on the basis of previous computation.

Table 1. Key recovery attack on the iterative reference DES using a CPA over 10K traces.

Sbox index	Key		Lock_t	SNR	Max CPA
	Actual	Guessed	$0 \leq \cdot \leq 10\,000$		[%]
1	56	56	4 314	4.38603	8.40
2	11	11	7 848	3.94818	5.68
3	59	59	1 247	5.29027	6.81
4	38	38	3 555	5.09747	5.94
5	0	0	2 272	7.25941	8.86
6	13	13	3 868	4.52662	8.10
7	25	25	4 399	4.69634	6.28
8	55	55	273	6.81590	14.68

Table 2. Key recovery attack on the unrolled DES using a CPA over 100K traces.

Sbox index	Key		Lock_t	SNR	Max CPA
	Actual	Guessed	$0 \leq \cdot \leq 100\,000$		[%]
1	56	58	87 976	1.83827	3.25
2	11	21	75 073	3.04394	1.52
3	59	17	97 462	2.07826	2.69
4	38	25	71 369	1.63005	4.85
5	0	53	70 590	3.45533	2.18
6	13	26	99 982	3.01725	1.18
7	25	22	70 433	2.07131	3.37
8	55	47	74 552	2.78395	3.26

3.1 Attack on the Unrolled DES

Now let us see a case when the previously described constraints are not respected i.e. two encryption are done without clearing the datapath. We explore two leakage models, namely the Hamming weight (HW) and the Hamming distance (HD), on two neuralgic positions of the algorithm, namely the Feistel function output (P1) and the round output right half (P2). We find that the HD on P1 completely discloses the key. The results are given in Tab. 3. We can see that for all the eight broken substitution boxes, the signal-to-noise ratio (SNR) is much smaller than for the case of the reference circuit. The results for the sbox 4 are printed in italics, because actually two keys are guessed simultaneously in a unrolled implementation, due to a mathematical property of this sbox. The fourth sbox S_4 of DES has the following property: $\forall x, y \in \{0,1\}^6$, $S_4(x) \oplus S_4(y)$

Table 3. Key recovery attack using the a CPA with a Hamming distance model (with respect to the previous encryption) over 100K traces.

Sbox index	Key		Lock_t	SNR	Max CPA
	Actual	Guessed	$0 \leq \cdot \leq 100\,000$		[%]
1	56	56	16 557	2.20267	2.17
2	11	11	44 092	2.15008	2.09
3	59	59	36 090	2.50697	2.22
4	*38*	*9*	*3 291*	*3.73242*	*5.01*
5	0	0	27 164	1.96649	2.28
6	13	13	20 138	2.13591	2.65
7	25	25	17 862	2.11245	2.86
8	55	55	37 317	2.77701	2.75

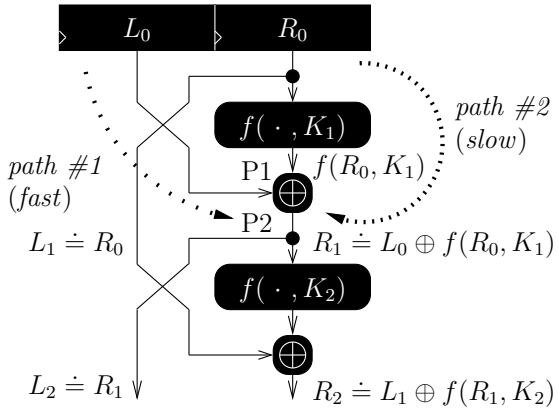


Fig. 5. Notations used to describe the combinatorial DES leakage functions.

and $\mathrm{S}_4(x \oplus \texttt{0x2f}) \oplus \mathrm{S}_4(y \oplus \texttt{0x2f})$ are palindromic. This fact can be shown by computing exhaustively the two expressions and comparing them.

Therefore, we have a remarkable Hamming distance conservation property: $\forall x, y \in \{0,1\}^6$, $|\mathrm{S}_4(x) \oplus \mathrm{S}_4(y)| = |\mathrm{S}_4(x \oplus \texttt{0x2f}) \oplus \mathrm{S}_4(y \oplus \texttt{0x2f})|$. As a conclusion, in a Hamming distance model, two keys are retrieved in pairs: the correct one and one another (false), equal to the correct key translated by `0x2f`.

To show that the correlations of the sboxes output (locus P1) are very disrupted due to their combinatorial nature, we have computed the DPA peaks, shown in Fig. 6. We favor DPA [13] over CPA [7], because, as explained in the technical article [14], the covariance used by DPA

extracts the activity of some nets in the netlist, which is interesting for leakage characterization. As for the CPA, it is more suitable for attacks, because the normalization by the trace standard deviation corrects the fact that the leakage is not necessarily maximum at the times where the side-channel is [15]. The DPA covariance $|f(R_r^{-1}, K_{r+1}) \oplus f(R_r, K_{r+1})|$ for all $r \in [0, 6]$ are plotted in Fig. 6. We have also added the transition in R_0 between two consecutive messages, because it indicates the computation beginning and its end. The beginning consists of the R_0 register sampling at the rising edge of the clock. The end corresponds to the other transition (final $\rightarrow$ initial), in the R_0 register input latches, that are transparent, and that dissipate even in the absence of a clock event. We observe that the DPA covariances do not especially show peaks ordered in time. This indicates the link between the data and the side-channel measurement is destroyed as early as the first couple of rounds.

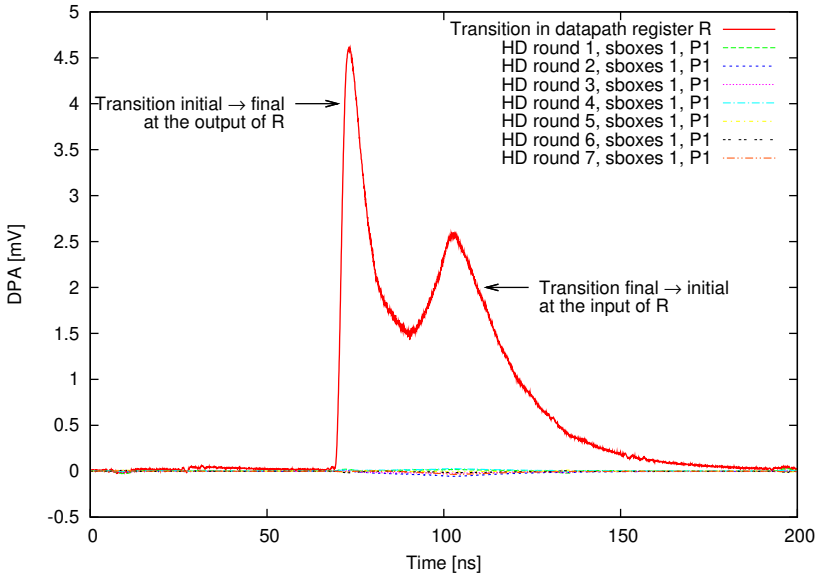


Fig. 6. DPA covariance for the register transfer R_0, and round correlations for the first sbox outputs.

To conclude with the security analysis, we discuss briefly on the unsuitability of other SCAs. Template attacks are expected to become less a concern as technology typical feature sizes shrink and characteristics dispersion increases [16]. Preliminary works on 130 nm technologies [17]

suggest that the intra-die technological mismatches are the preponderant source of variation, surpassing the imperfections of the logic style.

3.2 Evaluation Based on Mutual Information Metric

Mutual information analysis (MIA) has been introduced in [18] and further discussed in [19]. This analysis captures whatsoever dependence between measurements and a leakage model. It is thus a tool suited for an information leakage evaluation, as pointed out in [20]. The default leakage model does not assume any device-specific knowledge. Therefore it considers plain dependency with one sensitive and predicable word within the device. The notions of sensitivity and predictability have been defined in [21]. Basically, a variable is sensitive if it depends on one secret, and predictable if testing all the hypotheses for this variable is computationally tractable. The leakage-agnostic approach is the one employed in template attacks [22].

We have computed the mutual information (MI) between the right half of the datapath for sbox #1 and each point of our experimental traces. The results are plotted in Fig. 7 for the 80k traces of the iterative DES module and the 100k traces of the unrolled one. In the iterative circuit, the MI is roughly the same for each round. However, it depends on the round index for the combinatorial circuit; therefore we represent a couple of them in Fig. 7. It appears clearly that the sequential circuit is leaking more information about the first round than the combinatorial. Hence the vulnerability is less significant for our proposed countermeasure.

4 Conclusion and Perspectives

Information masking and hiding are two protection techniques against side-channel attacks. We propose a new countermeasure which comprises unrolling of rounds of a cryptographic algorithm to execute during a single clock. Results show that unrolling is secure against power attacks with a constraint of clearing the datapath after each encryption. We also evaluated mutual information metric on the design and results show that unrolled DES is less vulnerable. Further work involves testing this countermeasure with other algorithms like AES, *etc.* Also it could be interesting to partially unroll the algorithm like the rounds which are soft targets for an attacker.

Finally, we mention the potential advantage of algorithms unrolling against some fault attacks; for instance, it is impossible to inject faults via

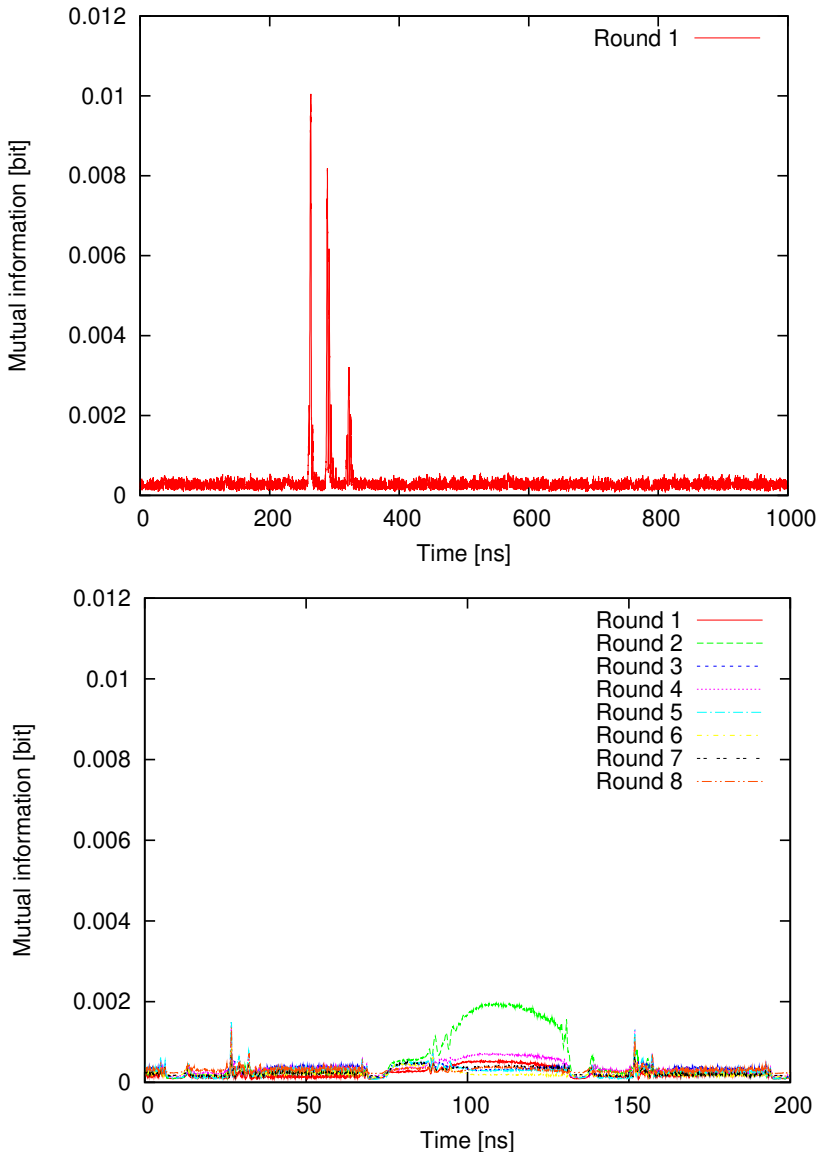


Fig. 7. Mutual information metric for sequential (*top*) and combinatorial (*bottom*) DES.

a setup time violation [23–25], produced by either under-powering or over-clocking the unrolled module. The resistance of partially or completely unrolled architectures against other DFAs is thus an interesting research direction.

Acknowledgments

This work has been partly financed by the french national research agency (ANR), through the `ANR-07-ARFU-010` grant "SeFPGA" (Secured Embedded FPGAs). We acknowledge interesting discussions and encouragements with Renaud Pacalet from the LabSoC laboratory of TELECOM ParisTech at Sophia-Antipolis.

References

1. Kocher, P.C., Jaffe, J., Jun, B.: Differential Power Analysis. In: CRYPTO. Volume 1666 of LNCS., Springer (1999) pp 388–397
2. Biham, E., Shamir, A.: Differential Fault Analysis of Secret Key Cryptosystems. In: CRYPTO. Volume 1294 of LNCS., Springer (August 1997) 513–525 Santa Barbara, California, USA. DOI: 10.1007/BFb0052259.
3. TELECOM ParisTech SEN research group: DPA Contest (2008–2009) `http://www.DPAcontest.org/`.
4. Akkar, M.L., Giraud, C.: An Implementation of DES and AES Secure against Some Attacks. In LNCS, ed.: Proceedings of CHES'01. Volume 2162 of LNCS., Springer (May 2001) 309–318 Paris, France.
5. Tiri, K., Verbauwhede, I.: A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation. In: DATE'04, IEEE Computer Society (February 2004) 246–251 Paris, France. DOI: 10.1109/DATE.2004.1268856.
6. Standaert, F.X., Örs, S.B., Preneel, B.: Power Analysis of an FPGA: Implementation of Rijndael: Is Pipelining a DPA Countermeasure? In: CHES. Volume 3156 of LNCS., Springer-Verlag (August 11-13 2004) 30–44 Cambridge (Boston), MA, USA.
7. Brier, É., Clavier, C., Olivier, F.: Correlation Power Analysis with a Leakage Model. In: CHES. Volume 3156 of LNCS., Springer (August 11–13 2004) 16–29 Cambridge, MA, USA.
8. Koç, K.Ç.: Cryptographic Engineering. Springer US (2009)
9. Guilley, S., Chaudhuri, S., Sauvage, L., Danger, J.L., Beyrouthy, T., Fesquet, L.: Updates on the Potential of Clock-Less Logics to Strengthen Cryptographic Circuits against Side-Channel Attacks. In: ICECS. IEEE (December 13–16 2009) 351–354 Medina, Yasmine Hammamet, Tunisia. DOI: 10.1109/ICECS.2009.5411008.
10. Guilley, S., Hoogvorst, P., Pacalet, R.: A Fast Pipelined Multi-Mode DES Architecture Operating in IP Representation. Integration, The VLSI Journal **40**(4) (July 2007) 479–489 DOI: 10.1016/j.vlsi.2006.06.004.
11. Roche, T., Tavernier, C.: Multi-Linear cryptanalysis in Power Analysis Attacks: MLPA. In: Western European Workshop on Research in Cryptology, WEWoRC 2009. (July 7-9 2009) Graz, Austria.

12. Aabid, M.A.E., Guilley, S., Hoogvorst, P.: Template Attacks with a Power Model. Cryptology ePrint Archive, Report 2007/443 (December 2007) http://eprint.iacr.org/2007/443/.
13. Kocher, P.C., Jaffe, J., Jun, B.: Differential Power Analysis. In: Proceedings of CRYPTO'99. Volume 1666 of LNCS., Springer-Verlag (1999) 388–397
14. Guilley, S., Hoogvorst, P., Pacalet, R., Schmidt, J.: Improving Side-Channel Attacks by Exploiting Substitution Boxes Properties. In Presse Universitaire de Rouen et du Havre, ed.: BFCA. (2007) 1–25 May 02–04, Paris, France, http://www.liafa.jussieu.fr/bfca/books/BFCA07.pdf.
15. Guilley, S., Sauvage, L., Danger, J.L., Selmane, N., Pacalet, R.: Silicon-level solutions to counteract passive and active attacks. In: FDTC, 5th Workshop on Fault Detection and Tolerance in Cryptography, IEEE-CS, Washington DC, USA (aug 2008) 3–17
16. Quisquater, J.J., Standaert, F.X.: Physically Secure Cryptographic Computations: From Micro to Nano Electronic Devices. In: DSN, Workshop on Dependable and Secure Nanocomputing (WDSN), IEEE Computer Society (June 28 2007) Invited Talk, 2 pages, Edinburgh, UK.
17. Guilley, S., Flament, F., Pacalet, R., Hoogvorst, P., Mathieu, Y.: Security Evaluation of a Balanced Quasi-Delay Insensitive Library. In: DCIS, Grenoble, France, IEEE (nov 2008) 6 pages, Session 5D – Reliable and Secure Architectures, ISBN: 978-2-84813-124-5, full text in HAL: http://hal.archives-ouvertes.fr/hal-00283405/en/.
18. Gierlichs, B., Batina, L., Tuyls, P., Preneel, B.: Mutual information analysis. In: CHES, 10th International Workshop. Volume 5154 of Lecture Notes in Computer Science., Springer (August 10-13 2008) 426–442 Washington, D.C., USA.
19. Prouff, E., Rivain, M.: Theoretical and Practical Aspects of Mutual Information Based Side Channel Analysis. In Springer, ed.: ACNS. Volume 5536 of LNCS. (June 2-5 2009) 499–518 Paris-Rocquencourt, France.
20. Veyrat-Charvillon, N., Standaert, F.X.: Mutual Information Analysis: How, When and Why? In: CHES. Volume 5747 of LNCS., Springer (September 6-9 2009) 429–443 Lausanne, Switzerland.
21. Standaert, F.X., Peeters, É., Rouvroy, G., Quisquater, J.J.: An Overview of Power Analysis Attacks Against Field Programmable Gate Arrays. Proceedings of the IEEE **94**(2) (February 2006) 383–394 (Invited Paper).
22. Chari, S., Rao, J.R., Rohatgi, P.: Template Attacks. In: CHES. Volume 2523 of LNCS., Springer (August 2002) 13–28 San Francisco Bay (Redwood City), USA.
23. Faurax, O., Tria, A., Freund, L., Bancel, F.: Robustness of circuits under delay-induced faults: test of AES with the PAFI tool. In: IOLTS, IEEE Computer Society (8-11 July 2007) 185–186 Heraklion, Crete, Greece.
24. Selmane, N., Guilley, S., Danger, J.L.: Setup Time Violation Attacks on AES. In: EDCC, The seventh European Dependable Computing Conference, Kaunas, Lithuania (May 7-9 2008) 91–96 ISBN: 978-0-7695-3138-0, DOI: 10.1109/EDCC-7.2008.11.
25. Khelil, F., Hamdi, M., Guilley, S., Danger, J.L., Selmane, N.: Fault Analysis Attack on an FPGA AES Implementation. In: NTMS, Tangier, Morocco, IEEE (nov 2008) 1–5 DOI: 10.1109/NTMS.2008.ECP.45.