A New Approach to Counteract DPA Attacks on Block Ciphers

Christophe Giraud and Emmanuel Prouff

Oberthur Card Systems,
25 rue Auguste Blanche,
92 800 Puteaux, France.
{c.giraud,e.prouff}@oberthurcs.com

Abstract. Since the publication of Differential Power Analysis (DPA) in 1998, many countermeasures have been published to counteract this very efficient kind of attacks. All these countermeasures follow the same approach : they try to make sensitive operations uncorrelated with the input. Such a method is very costly in terms of both timing and memory space. In this paper, we suggest a new approach where block ciphers are designed to inherently thwart DPA attacks. The idea we develop in this paper is based on a theoretical analysis of DPA attacks and it essentially consists in embedding existing iterated block ciphers in a secure layer. We analyse the security of our proposal and we show that it induces very small overheads.

Keywords: Countermeasure, Power Analysis, Block ciphers, Smart cards, S-boxes.

1 Introduction

When a new block cipher is designed, many criteria to counteract theoretical attacks such as linear and differential cryptanalysis must be satisfied [19,4]. However, if this algorithm is straightforwardly implemented on embedded devices such as smart cards, many other specific attacks can then be applied to recover secret parameters. One family of these attacks is called *side-channel attacks* because it uses the characteristics of embedded environments such as timing, power consumption or electromagnetic radiations. Since their publication in 1996 [16], they have been successfully put into practice and numerous papers have been published on the subject (see for instance [17,1,21]).

Despite the efficiency of side channel attacks on block ciphers, the criterion of being side-channel resistant has never been required for the design of new cryptosystems[1]. Thus, in order to protect cryptographic

[1] However in the case of AES, the feasibility of adding countermeasures against power analysis has been taken into account.

implementations from side-channel attacks, developers must implement specific countermeasures [25,20]. The latter are always very costly in terms of both memory space and timings.

In this paper we describe a way to obtain Differential Power Analysis (DPA) resistant block ciphers. The idea we develop consists in adding a layer before and after traditional block ciphers such as DES or AES. The family of layers we propose do not decrease the theoretical security of the underlying algorithm and induce a very small efficiency penalty compared to the traditional countermeasures.

This paper is organized as follows. Firstly we recall some properties about vectorial functions and we describe iterated block cipher cryptosystems in a formal way. Then we study in detail Differential Power Analysis on iterated block ciphers. In Section 4, this analysis is used to describe a way of designing efficient DPA-resistant iterated block ciphers. In Section 5, the security of our proposal is discussed. Finally, we conclude in Section 6.

2 Notations and Preliminaries

2.1 Preliminaries About Vectorial Functions in Cryptography

We call (n, m)-function any mapping F from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$ and we denote by $\mathcal{B}_{n,m}$ the set of (n, m)-functions. If m equals 1, then the function is called Boolean.
If $F \in \mathcal{B}_{n,m}$ is *affine*, then we call *direction of* F the linear function $L \in \mathcal{B}_{n,m}$ such that it exists a vector $B \in \mathbb{F}_2^m$ for which $F(X) = L(X)+B$, $X \in \mathbb{F}_2^n$.
A function $F \in \mathcal{B}_{n,m}$ is said to be *balanced* if every element $Y \in \mathbb{F}_2^m$ admits the same number 2^{n-m} of pre-images by F.

To every function $F \in \mathcal{B}_{n,m}$, we associate the m-tuple $(f_1, \cdots, f_m)$ of Boolean functions on $\mathbb{F}_2^n$ such that we have $F(X) = (f_1(X), \cdots, f_m(X))$.

The *Walsh transform* of a (n, m)-function F is defined on $\mathbb{F}_2^n \times \mathbb{F}_2^m$ by the formula:

$$W_F(u, v) = \sum_{X \in \mathbb{F}_2^n} (-1)^{v \cdot F(X) + u \cdot X} \quad , \tag{1}$$

where we recall that $v \cdot F$ equals the Boolean function $\bigoplus_{i=1}^m v_i f_i$.

Remark 1. Notice that $W_F(0, v)$ equals $\pm 2^n$ iff $v \cdot F$ is constant and equals 0 iff $v \cdot F$ is balanced.

As we recall in the following proposition, the balancedness of a function can be characterized through its Walsh transform's coefficients.

Proposition 1 *Let n and m be two positive integers and let F be a (n,m)-function, F is balanced iff $W_F(0,v)$ equals zero for every vector $v \in \mathbb{F}_2^{m*}$.*

A fundamental principle introduced by Shannon [28] for the design of conventional cryptographic systems is *confusion*, which aims at concealing any algebraic structure. The main characteristic quantifying the confusion induced into the system is the *non-linearity*. The non-linearity N_F of a (n,m)-function F can be defined through its Walsh transform's coefficients [8,23] by:

$$\mathrm{N}_F = 2^{n-1} - \frac{1}{2} \max_{v \in \mathbb{F}_2^{m*},\, u \in \mathbb{F}_2^n} |\sum_{X \in \mathbb{F}_2^n} (-1)^{v \cdot F(X) + u \cdot X}|. \tag{2}$$

Let n be a positive integer and let f and g be two Boolean functions defined on $\mathbb{F}_2^n$, then the *correlation coefficient* of f and g is defined by $\mathrm{Cor}(f,g) = \frac{1}{2^n} \sum_{X \in \mathbb{F}_2^n} (-1)^{f(X)+g(X)}$. The notion of correlation coefficient can be generalized to functions $F, G \in \mathcal{B}_{n,m}$: let (u,v) be an element of $\mathbb{F}_2^{m^2}$, then *the correlation coefficient $Cor_{u,v}(F,G)$ of F and G with respect to* (u,v) is defined by:

$$\mathrm{Cor}_{u,v}(F,G) = \frac{1}{2^n} \sum_{X \in \mathbb{F}_2^n} (-1)^{u \cdot F(X) + v \cdot G(X)} \quad . \tag{3}$$

Functions F and G are *uncorrelated* iff $\mathrm{Cor}_{u,v}(f,g)$ equals zero for every pair of non-zero elements $u, v \in \mathbb{F}_2^m$. If outputs of F and G are statistically independent, then F and G are *uncorrelated*.

A useful tool for quantifying the cryptographic resistance of functions is the notion of *derivative*. The derivative of F with respect to a vector $a \in \mathbb{F}_2^n$ is the (n,m)-function $D_aF : X \mapsto F(X) + F(X+a)$. The notion of derivative is related to *differential* and *higher-order differential* attacks [4,15,18]. An element a such that D_aF is constant is called a *linear structure* of F. As argued by Evertse in [10], functions F used in block ciphers must only admit the null-vector for linear structure.

2.2 Iterated Block Ciphers

Let n be a positive integer. To define an iterated block cipher in a formal way, we consider a family $(F_K)_{K \in \mathcal{K}}$ of (n,n)-functions, indexed by a value $K \in \mathcal{K}$ where $\mathcal{K}$ is called the *round key space*. The *encryption function*

Enc_k of an iterated block cipher with block size n, with R rounds and with round functions F_K is defined by:

$$X^{(i)} = F_{K_i}\left(X^{(i-1)}\right) \text{ for } 1 \leq i \leq R, \tag{4}$$

where $X^{(0)}$ is the *plaintext* and $X^{(R)}$ is the *ciphertext*.

The vector $k = (K_1, \ldots, K_R)$ is called the *key* and its coordinates are the *round keys*. The latter may be derived from a unique master key which is shorter than the concatenation of all the round keys.

Round functions F_K of iterated block ciphers are designed to ensure the *diffusion* and the *confusion* of information. The *confusion* part is usually obtained by composing two affine functions A and A' with a non-linear function S as follows:

$$F_K(X) = A' \circ S \circ A(X + K),\ X \in \mathbb{F}_2^n\ . \tag{5}$$

In such a system, the round key K is said to be introduced *by addition*. The main role of the function S is to ensure the confusion of information brought in the system, whereas the functions A and A' only ensure the diffusion of information. To allow an efficient computation of S, outputs of this function are usually defined as the concatenation of independent small vectors (usually of 8-bit length), each of them depending on a small number of bits of the inputs. We call *decomposition order of a* (p, m)*-function* S the smallest integer t such that it exists a family $(S_1, \cdots, S_t)$ of t functions from $\mathbb{F}_2^{p/t}$ into $\mathbb{F}_2^{m/t}$ satisfying the following relation for every $Y = (y_1, \cdots, y_p) \in \mathbb{F}_2^p$:

$$S(Y) = S_1(Y_{1(\frac{p}{t})})||S_2(Y_{2(\frac{p}{t})})|| \cdots ||S_t(Y_{t(\frac{p}{t})})\ , \tag{6}$$

where $Y_{i(j)}$ denotes the vector $(y_{1+(i-1)j}, \cdots, y_{j+(i-1)j})$ and $||$ denotes the *concatenation operation*.

Let $F = (f_1, \cdots, f_m)$ be an element of $\mathcal{B}_{n,m}$. For every index $j \leq m$, let us denote by $\mathcal{I}_j$ one of the smallest subsets of $\{1, \cdots, n\}$ such that for every $X = (x_1, \cdots, x_n)$, the value $f_j(x_1, \cdots, x_n)$ only depends on the input-bits x_i, with i belonging to $\mathcal{I}_j$. We call here *diffusion order* of F the value $\min_{j=1,\cdots,m}(\#\mathcal{I}_j)$.

In the rest of the paper, we focus our analysis on block ciphers where round functions F_K satisfy (5) and where the function S is defined as the concatenation of smaller functions S_i, usually called *S-boxes*. Moreover, we assume that S is a (p, m)-function admitting t for decomposition order and that A and A' are two affine surjective functions from $\mathbb{F}_2^n$ into $\mathbb{F}_2^p$ and

from $\mathbb{F}_2^m$ into $\mathbb{F}_2^n$ respectively. To make appear the main parameters, we call (A, S, A', R)-*iterated block cipher* the R-rounds iterated block cipher defined by the functions A, S and A'.

3 Multi-bit DPA Attacks on Iterated Block Ciphers

The aim of this section is to describe DPA attacks on (A, S, A', R)-iterated block ciphers. In particular, we point out the relationship between the efficiency of these attacks and the decomposition order t of S.

3.1 Introduction

In [17], Kocher introduced a new kind of attacks called Differential Power Analysis, especially efficient for cryptanalyzing algorithms embedded in smart cards (*cf.* [20,25]). Kocher's method is based on the fact that computers and microchips leak information about the operations they process. Initial attack of Kocher, called *single-bit DPA attack*, was generalized in *multi-bit DPA attack* in [21]. Since an algorithm which thwarts multi-bit DPA attacks is also resistant to single-bit DPA attacks, we only pay attention in the rest of this paper for the multi-bit case.

The goal of a cryptanalyst performing a DPA attack on the first round of a block cipher is to obtain information about the secret round key $\dot{K}$. This information is deduced by analyzing the values taken by a particular function, usually called *power consumption function* and denoted by $C_{\dot{K}}$, on a set of well-chosen plaintexts $\{X^{(0)}\}$.

Let $(A_i)_{i \leq t}$ be the family of the t affine $(n, p/t)$-functions such that $A = A_1||A_2||\cdots||A_t$. When the block cipher is an (A, S, A', R)-iterated block cipher, a multi-bit DPA attack on its first round is a *divide-and-conquer* attack. Indeed, several multi-bit DPA attacks are simultaneously applied to each function $S_i \circ A_i$, $i \leq t$, and the first round key is recovered by putting together all the information obtained from these parallel attacks.

In what follows, we describe without lost of generality the multi-bit DPA attack on $S_1 \circ A_1$.

3.2 Power Consumption Function

In a multi-bit DPA attack on $S_1 \circ A_1$, each value $C_{\dot{K}}(X)$, $X \in \mathbb{F}_2^n$, can be viewed as the energy required to flip bits from a previous state to state $S_1 \circ A_1(X + \dot{K})$. To have a formal definition of $C_{\dot{K}}$, one has to

introduce a theoretical model for the power consumption of devices. In this paper, we use the *Hamming distance model* introduced in [5] as a generalization of the *Hamming weight model* (*cf.* [2]). In the Hamming distance model, it is assumed that switching a bit from 0 to 1 requires the same amount of energy as switching it from 1 to 0. We denote by λ the average power consumption to switch a bit from 0 to 1 and we denote by $\alpha(X, \dot{K})$ the value of the data which is replaced by $S_1 \circ A_1(X + \dot{K})$. We call *state function* the function $\alpha : (X, K) \in \mathbb{F}_2^{n^2} \mapsto \alpha(X, K)$. For every pair (X, K), we assume throughout this paper that the power consumption $C_K(X)$ satisfies the following relation:

$$C_K(X) = \lambda \times \mathrm{H}\left(\alpha(X, K) + S_1 \circ A_{1,K}(X)\right) + \mu \ , \tag{7}$$

where $A_{1,K}$ denotes the function $X \mapsto A_1(X + K)$, H denotes the *Hamming weight function* and we assume here that μ denotes a random noise.

Remark 2. Equality $v \cdot F = \frac{1}{2} - \frac{1}{2}(-1)^{v \cdot F}$ is satisfied for every function $F \in \mathcal{B}_{n,m}$ and for every vector $v \in \mathbb{F}_2^m$. By applying it to Relation (7), we have $C_K(X) = \frac{n\lambda}{2} - \frac{\lambda}{2} \times \sum_{\substack{u \in \mathbb{F}_2^n \\ \mathrm{H}(u)=1}} (-1)^{u \cdot (\alpha(X,K) + S_1 \circ A_{1,K}(X))} + \mu$.

In the rest of the paper, we consider the restriction $\alpha(\cdot, \dot{K})$ of the state function α to the set $\mathbb{F}_2^n \times \{\dot{K}\}$, where $\dot{K}$ denotes the actual round key. To simplify notations, we denote by α the function $X \mapsto \alpha(X, \dot{K})$.

3.3 Multi-bit DPA Attacks

Assume that a cryptanalyst measured all the values $C_{\dot{K}}(X^{(0)})$, $X^{(0)}$ ranges over $\mathbb{F}_2^n$. It has been shown in [26] that a multi-bit DPA attack on $S_1 \circ A_1$ is done by searching for round keys $K \in \mathbb{F}_2^n$ which maximize the value:

$$\delta_{\dot{K}}(K) = \left| \sum_{v \in \mathbb{F}_2^{m/t},\ \mathrm{H}(v)=1} \Delta_{K,\dot{K}}(v) \right| \ , \tag{8}$$

where $\Delta_{K,\dot{K}}(v)$ is defined by:

$$\Delta_{K,\dot{K}}(v) = \frac{-1}{2^{n-1}} \sum_{X^{(0)} \in \mathbb{F}_2^n} (-1)^{v \cdot (S_1 \circ A_{1,K})(X^{(0)})} C_{\dot{K}}(X^{(0)}) \ . \tag{9}$$

Remark 3. In a cryptographic context, Relation (9) can be rewritten (*cf.* [26]) as:

$$\Delta_{K,\dot{K}}(v) = \lambda \sum_{\substack{u \in \mathbb{F}_2^{m/t} \\ \mathrm{H}(u)=1}} \mathrm{Cor}_{v,u}\left(S_1 \circ A_{1,K}, S_1 \circ A_{1,\dot{K}} + \alpha\right) \quad .$$

In practice, the computation of $\Delta_{K,\dot{K}}(v)$ (and thus the one of $\delta_{\dot{K}}(K)$) can be made much more efficiently than summing 2^n elements. To explain how the particular structure of the function $S_1 \circ A_{1,K}$ can be used to simplify the computations, we make the following two assumptions whose relevances in the context of symmetric cryptography are argued in [5,12,26]:

Assumption 1 [2] *Let u and v be two distinct elements of $\mathbb{F}_2^{m/t}$ such that $\mathrm{H}(u) = \mathrm{H}(v) = 1$. For every pair $(K, \dot{K})$ of round keys, functions $v \cdot (S_1 \circ A_{1,K})$ and $u \cdot (S_1 \circ A_{1,\dot{K}})$ are uncorrelated.*

Assumption 2 *The state function α is constant.*

Moreover, to simplify notations we assume that the state function α is the null function. The generalization of the study to the case of a constant function α different from zero is straightforward.

When α is the null function, then Assumption 1 and Remark 3 imply $\Delta_{K,\dot{K}}(v) = \frac{\lambda}{2^n} W_{D_{K+\dot{K}}(S_1 \circ A_1)}(0, v)$. Moreover, if A_1 is surjective (which is the case in practice), then we have:

$$\Delta_{K,\dot{K}}(v) = \frac{\lambda}{2^{p/t}} W_{D_{L_1(K+\dot{K})} S_1}(0, v) \quad , \tag{10}$$

where L_1 denotes the direction of A_1. Thus, one can deduce the following relation from (8):

$$\delta_{\dot{K}}(K) = 2^{1-p/t} \left| \sum_{\substack{v \in \mathbb{F}_2^{m/t} \\ \mathrm{H}(v)=1}} \sum_{X \in E_1} (-1)^{v \cdot (S_1 \circ A_1[X+K])} C_{\dot{K}}(X) \right| \quad , \tag{11}$$

where E_1 denotes one pre-image set of $\mathrm{Im}(A_1)$.

[2] Even if this assumption is not always true (for DES for instance), it is usually highly recommended that an S-box has the property described in this assumption.

Remark 4. From (11), we deduce that the computation of $\delta_{\dot{K}}(K)$ is reduced to a summation of $m/t \times 2^{p/t}$ terms and to the computation of a set E_1.

From (8) and (10), one deduces that $\delta_{\dot{K}}(K)$ satisfies relation $0 \leq \delta_{\dot{K}}(K) \leq \frac{\lambda m}{t}$ for every $(K, \dot{K})$. Moreover, for every $\dot{K}$, the upper bound $\frac{\lambda m}{t}$ is achieved iff $W_{D_{L_1(K+\dot{K})}S_1}(0, v)$ equals $\pm 2^{p/t}$ for every $v \in \mathbb{F}_2^{m/t}$ of Hamming weight equal to 1. Thus, one deduces from Remark 1 that $\delta_{\dot{K}}(K)$ is maximal iff $D_{L_1(K+\dot{K})}S_1$ is constant, *i.e.* iff $L_1(K+\dot{K})$ is a linear structure of S_1. Since the null-vector $0_{\frac{p}{t}}$ on $\mathbb{F}_2^{p/t}$ is a linear structure of S_1, then for every $\dot{K}$ the value $\delta_{\dot{K}}(K)$ is maximal when K satisfies $L_1(K + \dot{K}) = 0_{\frac{p}{t}}$, *i.e.* when K belongs to $\dot{K} + \mathrm{Ker}(L_1)$. Since E_1 contains exactly one element of $\dot{K} + \mathrm{Ker}(L_1)$, one deduces the following procedure for a multi-bit DPA attack on $S_1 \circ A_1$:

Procedure 3.1 Multi-bit DPA attack on $S_1 \circ A_1$

INPUTS: two functions A_1 and S_1 and an expected round key $\dot{K}$
OUTPUT: a subset $\mathcal{D}_1 \subseteq \mathbb{F}_2^n$ of keys K such that $\delta_{\dot{K}}(K)$ is maximal

1. $E_1 = \emptyset$
2. **for all** $Y \in \mathrm{Im}(A_1)$
   compute one element $X^{(0)}$ such that $X^{(0)} \in A_1^{-1}(Y)$
   add $X^{(0)}$ to E_1
3. **for all** $X^{(0)} \in E_1$
   measure $C_{\dot{K}}(X^{(0)})$
4. **for all** $K \in E_1$
   compute $\delta_{\dot{K}}(K) = 2^{-p/t+1} \left| \sum_{\substack{v \in \mathbb{F}_2^{m/t} \\ H(v)=1}} \sum_{X^{(0)} \in E_1} (-1)^{v \cdot (S_1 \circ A_{i,K})(X^{(0)})} C_{\dot{K}}(X^{(0)}) \right|$
   if $\delta_{\dot{K}}(K)$ is maximal **then** store K in $\mathcal{D}_1$

Fact 1 *The complexity of the construction of E_1 is about $\mathrm{O}\left(n^3\right)$. As the function A_1 is surjective from $\mathbb{F}_2^n$ into $\mathbb{F}_2^{p/t}$, then $\#E_1$ equals $2^{p/t}$. Thus, Procedure 3.1 requires $2^{p/t}$ measurements and the complexity of the main loop (Step 4) is $\mathrm{O}(\frac{m}{t} 2^{\frac{p^2}{t^2}})$.*

The set $\mathcal{D}_1$ contains one element of $\dot{K} + \mathrm{Ker}(L_1)$. Finally, the key $\dot{K}$ is retrieved by applying Procedure 3.1 to all the functions $S_i \circ A_i$, $i = 1, \cdots, t$, and by obtaining one element of each coset $\dot{K} + \mathrm{Ker}(L_i)$ (which implies the computation of the sets $E_1, \ldots, E_t$).

3.4 Examples of the Construction of the E_i Sets

DES The non-linear function S used in DES is a $(48, 32)$-function defined as the concatenation of 8 S-boxes S_1, ..., S_8 from $\mathbb{F}_2^6$ into $\mathbb{F}_2^4$. The input of the round function is a 32-bit length vector which is transformed by a public expansion function Exp in a 48-bit length vector. The computation of $\mathrm{Exp}(X)$ and the computation of $\mathrm{Exp}^{-1}[\mathrm{Exp}(X)]$, $X \in \mathbb{F}_2^{32}$, being immediate one can assume that inputs of the round functions of DES are 48-bit length. For DES, if one chooses A_i the function $(x_1, \cdots, x_{48}) \mapsto (x_{6i-5}, \cdots, x_{6i})$, the computation of E_i is immediate: $E_i = \{0_{6i-6}\} \times \mathbb{F}_2^6 \times \{0_{48-6i}\}$.

AES The S-box used is a permutation on $\mathbb{F}_2^{128}$ defined as the concatenation of 16 permutations S_1, ..., S_{16} on $\mathbb{F}_2^8$. All the functions S_i are equal to the same function $X \in \mathbb{F}_{2^8} \mapsto (1 + \delta_0(X))X^{-1}$ (where δ_0 denotes the Dirac function). One can assume that mappings A_i in AES are the functions $(x_1, \cdots, x_{128}) \mapsto (x_{8i-7}, \cdots, x_{8i})$. The computations of E_i are immediate : $E_i = \{0_{8i-8}\} \times \mathbb{F}_2^8 \times \{0_{64-8i}\}$.

4 A New Way to Counteract DPA Attacks

4.1 Introduction

In Section 3.3, we showed that a DPA attack on an (A, S, A', R)-iterated block cipher can be efficiently mounted when the computation of sets E_i is feasible and when the number $t \times 2^{p/t}$ of required measurements is small. As n and t are lower than 128 in practice, the computation of sets E_i is very fast (indeed the complexity of such a computation is $\mathrm{O}\left(t \times n^3\right)$). Moreover, for many block ciphers (such as DES or AES) the value $t \times 2^{p/t}$ is small (see Section 3.4). Thus, DPA attacks are usually very efficient and must be thwarted by adding specific countermeasures when implementing on embedded devices (see for example [11,3,9,25,20]). In this section, we develop another way to counteract DPA attacks. Our approach consists in designing DPA-resistant block ciphers which do not require additional DPA-countermeasures. To achieve this aim, we add a layer before traditional iterated block ciphers to increase the complexity of the computation of the sets E_i. Of course, such a layer is also added after the block cipher to counteract DPA attacks on the last round.

In this section, we describe a way to design the layers $P_{k'}^0$ and $P_{k'}^1$ such that DPA attacks on the system depicted in Fig. 1 are unfeasible. To reach this goal, layers $P_{k'}^0$ and $P_{k'}^1$, parameterized by a secret key $k' \in \mathcal{K}'$, must

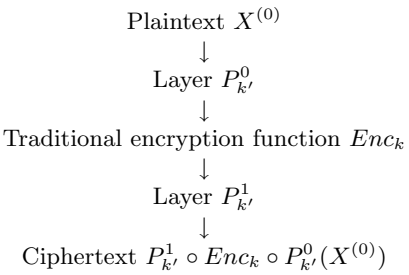

Fig. 1. Schematic representation of our DPA-resistant block cipher

fulfil several security requirements. Moreover, these layers must be easy to implement and effective from both timing and memory points of view.

4.2 Our Proposal

For simplicity reasons, we denote the layer $P^0_{k'}$ by $P_{k'}$. Let us assume that $\mathcal{K}'$ equals $\mathbb{F}_2^{n'}$. For every $k' \in \mathbb{F}_2^{n'}$, we define the layer $P_{k'}$ by :

$$\begin{array}{rl} P_{k'} : \mathbb{F}_2^n \to \mathbb{F}_2^n & \\ X \mapsto L \circ \pi_{k'} \circ L(X) & , \end{array} \tag{12}$$

where L is an involutive linear function whose diffusion order is greater than or equal to $n/2$ and where $(\pi_{k'})_{k' \in \mathbb{F}_2^{n'}}$ is a family of bit-permutations indexed by the elements k'. Let $\pi : k' \mapsto \pi_{k'}$ be the function which associates to every k' the corresponding bit-permutation $\pi_{k'}$. The family $(\pi_{k'})_{k' \in \mathbb{F}_2^{n'}}$ is defined such that π is an injective *highly non-linear* function from $\mathbb{F}_2^{n'}$ into the set of bit-permutations on $\mathbb{F}_2^n$ (which implies that n' must be chosen lower than or equal to $\log_2(n!)$).

For security reasons, we use the function $\pi_{k'}$ as the secret parameter of the layer $P_{k'}$. This implies that the key of the layer is not the vector k' but the function $\pi_{k'}$ itself. Thus the derivation of $\pi_{k'}$ from k' has to be done as part of the set up of the new DPA-resistant block cipher described in Fig. 1[3].

To cipher and decipher, most of block ciphers such as DES use the same core and only differ in the key scheduling. In order to keep this involutive property, we define the layer $P^1_{k'}$ as the inverse of the layer $P_{k'}$. The layer $P^1_{k'}$ is thus the function $X \in \mathbb{F}_2^n \mapsto L \circ \pi_{k'}^{-1} \circ L(X) \in \mathbb{F}_2^n$

[3] For instance, $\pi_{k'}$ can be pre-computed on a computer and stored into the device during the personalization step.

(recall that $L = L^{-1}$ since L is involutive). As $\pi_{k'}$ is a bit-permutation, the function $\pi_{k'}^{-1}$ is also a bit-permutation.

Remark 5. Let Dec_k denotes the decryption function corresponding to the function Enc_k. Choosing $P_{k'}^1 = P_{k'}^{-1}$ implies a DPA-resistant decryption function which is built by using exactly the same layers used to build our DPA-resistant encryption function. Indeed, one can easily check that $P_{k'}^{-1} \circ Dec_k \circ P_{k'}$ inverts the function $P_{k'}^{-1} \circ Enc_k \circ P_{k'}$.

4.3 Discussion about $P_{k'}$

The Diffusion Part. Since L is a linear function having a diffusion order greater than or equal to $n/2$, it can be represented by a binary matrix whose row vectors have an Hamming weight greater than or equal to $n/2$.

Example 1. On can choose for function L the one which is represented by the complementary to the Identity matrix, *i.e.*

$$L = \begin{pmatrix} 0 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 0 \end{pmatrix} .$$

This matrix, which has a diffusion order equal to $n-1$, is also used in the linear layer of `ICEBERG` [29]. As L is involutive, L^{-1} also admits $n-1$ for diffusion order.

The Confusion Part. In the layer $P_{k'}$, the function $(x, k') \mapsto \pi_{k'}(x)$ plays the role of an S-box. For reasons which are discussed in Section 5.2, this function is defined to be linear in x and highly non-linear in k'. For every $k' \in \mathbb{F}_2^{n'}$, let us represent a permutation $\pi_{k'}$ on $\mathbb{F}_2^n$ by the n-tuple $(i_1, \cdots, i_n)$ such that $\pi_{k'}$ maps 1 to i_1, 2 to i_2, ..., n to i_n. The function $\pi : k' \in \mathbb{F}_2^{n'} \mapsto \pi_{k'}$ used in (12) can thus be defined as an application which injectively associates to a vector k' the binary representation of the n-tuple $(i_1, \cdots, i_n)$ associated to $\pi_{k'}$. If one denotes by ℓ the value $\lceil n \log_2(n) \rceil$, then π can be viewed as a function from $\mathbb{F}_2^{n'}$ into $\mathcal{I} = \{y \in \mathbb{F}_2^{\ell}$; y is the binary representation of a bit-permutation on $\mathbb{F}_2^n\}$. To design an injective highly non-linear (n', ℓ)-function π, one can for instance easily adapt the constructions proposed in [7,23,14]. As the average non-linearity of (n', ℓ)-functions is high [24,27], another solution consists in choosing the (n', ℓ)-function π at random among the injective functions from $\mathbb{F}_2^{n'}$ into $\mathcal{I}$.

4.4 Implementation Efficiency

From a developer's point of view, the operation L is a binary matrix multiplication which can be efficiently implemented by using shifts and logical operations AND and XOR. For example, if one chooses the function L suggested in Example 1, the multiplication with the i^{th} row can be done by XORing every bit of the input except the i^{th} one. Moreover, it means that the matrix representing this function L does not need to be stored in the card.

Regarding the implementation of π, once the parameter k' is chosen, the representation of the bit-permutation $\pi_{k'}$ as index table is precomputed and is stored in the card. In practice, this operation can be done in the factory during the personalization of the card.

The RAM consumption of the layer $P_{k'}$ is negligible. Indeed, $P_{k'}$ can be performed by using the same n-bit temporary buffer which is used during the execution of Enc_k to store the temporary results.

To conclude, the implementation of $P_{k'}$ only requires basic assembly instructions and does not require any supplemental RAM memory compared to the embedded function Enc_k. The layer $P_{k'}$ can thus be efficiently implemented in a smart card environment.

5 Security Analysis of our Solution

In this section, we assume that a computation requiring more than 2^{64} elementary operations is impossible to perform in practice. Moreover, we assume that n and n' are greater than or equal to 64.

The encryption function depicted in Fig. 1 is DPA-resistant if the layer $P_{k'}$ satisfies three requirements: render impossible a DPA attack on the underlying block cipher, being resistant to DPA attacks and thwart the classical attacks (such as the linear or the differential attack).

5.1 DPA Attacks on the Underlying Encryption Function

In the solution proposed in (12), we adapt the classical principles applied to design round functions of iterated block ciphers by mixing diffusion operations with confusion operations. Thus, as it is classical in symmetric cryptography, one can assume the following property for $P_{k'}$ and $P_{k'}^{-1}$.

Property 1. For every $y \in \mathbb{F}_2^n$, the probability of finding a vector $x \in \mathbb{F}_2^n$ such that $y = P_{k'}(x)$ (resp. $y = P_{k'}^{-1}(x)$) equals 2^{-n} when k' is unknown. Moreover, knowing a pair (x, y) such that $y = P_{k'}(x)$ (resp. $y = P_{k'}^{-1}(x)$),

there does not exist an attack more efficient than the exhaustive search on $\mathbb{F}_2^{n'}$ to retrieve k'.

As argued in Section 3, a DPA on the first round of an (A, S, A', R)-iterated block cipher is a divide and conquer attack: the round key is recovered by mounting several DPA-attacks separately on all the S-boxes $S_i \circ A_i$. Moreover, we relate the efficiency of a DPA-attack on any function $S_i \circ A_i$ to the number of plaintexts needed to design a set E_i such as defined in Procedure 3.1. Let us assume that the (A, S, A', R)-iterated block cipher is embedded as described in Fig. 1. By definition, the set E_i is a set of $2^{p/t}$ plaintexts satisfying $A_i \circ P_{k'}(E_i) = \mathbb{F}_2^{p/t}$. Thus, designing E_i is equivalent to inverse the function $P_{k'}$ on a set $\mathcal{I}$ such that $A_i(\mathcal{I}) = \mathbb{F}_2^{p/t}$. Let us assume that we can find out one of the sets E_i in less than 2^n elementary operations and with a probability of success greater than 2^{-n}. This implies that we are able to find $2^{p/t}$ pre-images of $P_{k'}$ in less than 2^n elementary operations and with a probability of success greater than 2^{-n}. As this contradicts Property 1, one deduces that the design of a set E_i requires at least 2^n elementary operations: this makes the DPA attack inefficient when n is greater than or equal to 64.

5.2 DPA Attacks on the Layer $P_{k'}$

Since the diffusion order of L is greater than $n/2$, then every output-bit of $y = \pi_{k'} \circ L(X^{(0)})$ depends on at least $n/2$ bits of the plaintext $X^{(0)}$. This makes the computation of the coefficients $\Delta_{\dot{k'},k'}$ and $\delta_{\dot{k'}}(k')$ by using Relations (8) and (9) impossible when $n \times n'$ is greater than 64. As a consequence, a single-bit (or a multi-bit) DPA attack cannot be mounted directly on the function $\pi_{k'}$.

As the parameter k' is not introduced by addition, the manipulation of the coordinates of x depends on the value of k'. This makes a decomposition of $(x, k') \mapsto \pi_{k'}(x)$ such as performed for S in (6) impossible. Consequently, the decomposition order t of this function is 1 and a multi-bit DPA attack such as described in Section 3 is inefficient. Let Z denotes the value $L(X^{(0)})$. As done in Section 3, a cryptanalyst can try to obtain information on k' by mounting one or several DPA-attack(s) on some parts of the computation of the values $\pi_{k'}(Z)$ when Z ranges over $\mathbb{F}_2^n$. However, as the decomposition order of $(x, k') \mapsto \pi_{k'}(x)$ equals 1, then it is impossible for an attacker to re-build the bit-permutation $\pi_{k'}$ step by step as the concatenation of smaller bit-permutations (acting for example on 8-bit words).

Let us assume that $\pi_{k'}$ is given by its index representation $(i_1, \cdots, i_n)$. To counteract any other kind of Power Analysis on the computation of $\pi_{k'}(Z)$ from Z, one can for instance operate on the bits of Z (that is to access to the index representation) in a random order.

Remark 6. Usually, computing the output-bits of a cryptographic function in a random order is a very costly operation. For instance, in the case of AES each output-bit of the inverse function $Z \in \mathbb{F}_{2^8} \mapsto Y = (1 + \delta_0(Z))Z^{-1}$ depends on 8 bits of Z. Thus, the computation of every bit-coordinate of Y from the ones of Z requires a large number of logical operations. In our case, since $\pi_{k'}$ is itself a bit-permutation, randomizing the execution of $\pi_{k'}$ is for free.

5.3 Resistance of $P_{k'}$ to Classical Attacks

To prevent statistical attacks, functions involved as cryptographic primitives must be balanced. The following proposition proves that the confusion part of $P_{k'}$ is balanced.

Proposition 1. *The $(n \times n', n)$-function $(x, k') \mapsto \pi_{k'}(x)$ is balanced.*

Proof. Functions in $(\pi_{k'})_{k' \in \mathbb{F}_2^{n'}}$ being bit-permutations on $\mathbb{F}_2^n$, they are balanced on $\mathbb{F}_2^n$. This implies that for every $y \in \mathbb{F}_2^n$ and every $k' \in \mathbb{F}_2^{n'}$, there exists exactly one vector $x \in \mathbb{F}_2^n$ such that $\pi_{k'}(x) = y$. Function π being injective, for every $y \in \mathbb{F}_2^n$ the number of pairs (x, k') such that $y = \pi_{k'}(x)$ is $2^{n'}$. One deduces that π is balanced from $\mathbb{F}_2^n \times \mathbb{F}_2^{n'}$ into $\mathbb{F}_2^n$. □

Remark 7. Functions $\pi_{k'}$ being linear, the $(n \times n', n)$-function $(x, k') \mapsto \pi_{k'}(x)$ belongs to the class of vectorial Maiorana-MacFarland's functions (see [7,22,13] for more details about the cryptographic properties of these functions).

To prevent a differential analysis, two parameters k'_1 and k'_2 differing in a small number of bits must imply two functions $\pi_{k'_1}$ and $\pi_{k'_2}$ which are as different as possible. To ensure this property, the function π has been chosen to be highly non-linear.

In many attacks (such as the linear, the differential or the higher-order cryptanalysis), statistical properties of the round functions of block ciphers are used to make appear a relationship (a statistical bias) which must be satisfied after the penultimate round. These so-called *last round*

attacks use this statistical bias as a *distinguisher* to retrieve one round key after another in ascending order (see for instance [6]).

The function $P_{k'}^{-1}$ plays the role of the last round in the functions $P_{k'}^{-1} \circ Enc_k \circ P_{k'}$ and $P_{k'}^{-1} \circ Dec_k \circ P_{k'}$. Thus, if there exists a last round attack on these functions then there exists a family of plaintexts $(X_i)_i$ whose images after the penultimate round satisfy a particular relationship for almost all the pairs (k, k'). The images of the plaintexts X_i after the penultimate round correspond to the outputs of either $Enc_k \circ P_{k'}$ or $Dec_k \circ P_{k'}$. So, if the functions Enc_k and Dec_k are immunized against last round attacks, then one can assume that for every $k \in \mathbb{F}_2^n$ the functions Enc_k and Dec_k act as random permutations on $\mathbb{F}_2^n$. In this case, for any family $(X_i)_i$, the images of the plaintexts X_i through $Enc_k \circ P_{k'}$ or $Dec_k \circ P_{k'}$ are statistically independent since the distribution of the outputs of $P_{k'}$ (which is composition of balanced functions) is uniform. One deduces that if the functions Enc_k and Dec_k are immunized against last round attacks, then the functions $P_{k'}^{-1} \circ Enc_k \circ P_{k'}$ and $P_{k'} \circ Dec_k \circ P_{k'}^{-1}$ are also resistant to this kind of attacks.

6 Conclusion

In this paper we study in detail DPA attacks on iterated block ciphers. Based on this theoretical analysis, we suggest a way to design DPA-resistant iterated block ciphers by using a diffusion layer which is non-linearly parameterized by a secret parameter. The solution we suggest in this paper is designed to be very efficient in practice and easy to apply to any kind of existing iterated block cipher. In the area of embedded cryptography, this new approach allows a cryptologist to design specific block ciphers which are much more efficient in practice than usual block ciphers on which traditional DPA countermeasures must be added.

References

1. D. Agrawal, B. Archambeault, J. Rao, and P. Rohatgi. The EM Side-Channel(s). In B. Kaliski Jr., Ç. Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 29–45. Springer-Verlag, 2002.
2. M.-L. Akkar, R. Bévan, P. Dischamp, and D. Moyart. Power Analysis, What is Now Possible. In T. Okamoto, editor, *Advances in Cryptology – ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 489–502. Springer-Verlag, 2000.

3. M.-L. Akkar and C. Giraud. An implementation of DES and AES, secure against some attacks. In Ç. Koç, D. Naccache, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2001*, volume 2162 of *Lecture Notes in Computer Science*, pages 309–318. Springer-Verlag, 2001.
4. E. Biham and A. Shamir. Differential cryptanalysis of DES-like cryptosystems. *Journal of Cryptology*, 4(1):3–72, 1991.
5. E. Brier, C. Clavier, and F. Olivier. Correlation Power Analysis with a Leakage Model. In M. Joye and J.-J. Quisquater, editors, *Cryptographic Hardware and Embedded Systems – CHES 2004*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer-Verlag, 2004.
6. A. Canteaut and M. Videau. Degree of composition of Highly Nonlinear Functions and Applications to Higher Order Differential Cryptanalysis. In R. Rueppel, editor, *Advances in Cryptology – EUROCRYPT '92*, volume 658 of *Lecture Notes in Computer Science*, pages 518–533. Springer-Verlag, 1992.
7. C. Carlet and E. Prouff. Vectorial Functions and Covering Sequences. In *Finite Fields and Applications, Fq7*, volume 2948 of *Lecture Notes in Computer Science*, pages 215–248. Springer, 2004. To appear.
8. F. Chabaud and S. Vaudenay. Links between differential and linear cryptanalysis. In A. D. Santis, editor, *Advances in Cryptology – EUROCRYPT '94*, volume 950 of *Lecture Notes in Computer Science*, pages 356–365. Springer-Verlag, 1994.
9. J.-S. Coron. Resistance Against Differential Power Analysis for Elliptic Curve Cryptosystems. In Ç. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES '99*, volume 1717 of *Lecture Notes in Computer Science*, pages 292–302. Springer-Verlag, 1999.
10. J. Evertse. Linear structures in blockciphers. In D. Chaum and W. Price, editors, *Advances in Cryptology – EUROCRYPT '87*, volume 304 of *Lecture Notes in Computer Science*, pages 249–266. Springer-Verlag, 1987.
11. L. Goubin and J. Patarin. DES and Differential Power Analysis – The Duplication Method. In Ç. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES '99*, volume 1717 of *Lecture Notes in Computer Science*, pages 158–172. Springer-Verlag, 1999.
12. S. Guilley, P. Hoogvorst, and R. Pacalet. Differential Power Analysis Model and Some Results. In J.-J. Quisquater, P. Paradinas, Y. Deswarte, and A. E. Kalam, editors, *Smart Card Research and Advanced Applications VI – CARDIS 2004*, pages 127–142. Kluwer Academic Publishers, 2004.
13. K. Gupta and P. Sarkar. Improved Construction of Nonlinear Resilient S-Boxes. In L. Knudsen, editor, *Advances in Cryptology – ASIACRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 466–483. Springer-Verlag, 2002.
14. T. Johansson and E. Pasalic. A construction of resilient functions with high nonlinearity. In *Proceedings of the IEEE International Symposium on Information Theory*, 2000.
15. L. Knudsen. Truncated and Higher Order Differentials. In B. Preneel, editor, *Fast Software Encryption – FSE '94*, volume 1008 of *Lecture Notes in Computer Science*, pages 196–211. Springer-Verlag, 1994.
16. P. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In N. Koblitz, editor, *Advances in Cryptology – CRYPTO '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer-Verlag, 1996.
17. P. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In M. Wiener, editor, *Advances in Cryptology – CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer-Verlag, 1999.

18. X. Lai. Higher order derivatives and differential cryptanalysis. In *Symposium on Communication, Coding and Cryptography*, 1994.
19. M. Matsui. Linear cryptanalysis method for DES cipher. In T. Helleseth, editor, *Advances in Cryptology – EUROCRYPT '93*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer-Verlag, 1993.
20. T. Messerges. *Power Analysis Attacks and Countermeasures for Cryptographic Algorithms*. PhD thesis, University of Illinois, 2000.
21. T. Messerges, E. Dabbish, and R. Sloan. Investigations of Power Analysis Attacks on Smartcards. In *the USENIX Workshop on Smartcard Technology (Smartcard '99)*, pages 151–161, 1999.
22. K. Nyberg. Perfect nonlinear S-boxes. In J. Feigenbaum, editor, *Advances in Cryptology – EUROCRYPT '91*, volume 547 of *Lecture Notes in Computer Science*, pages 378–386. Springer-Verlag, 1991.
23. K. Nyberg. On the construction of highly nonlinear permutations. In R. Rueppel, editor, *Advances in Cryptology – EUROCRYPT '92*, volume 658 of *Lecture Notes in Computer Science*, pages 92–98. Springer-Verlag, 1992.
24. D. Olejar and M. Stanek. On cryptographic properties of random boolean functions. *J. UCS*, 4(8):705–717, 1998.
25. E. Oswald. *On Side-Channel Attacks and the Application of Algorithmic Countermeasures*. PhD thesis, Institute for Applied Information Processing and Communications - Graz University of Technology, May 2003.
26. E. Prouff. DPA attacks and S-Boxes. In H. Handschuh and H. Gilbert, editors, *Fast Software Encryption – FSE 2005*, volume 3557 of *Lecture Notes in Computer Science*, pages 424–442. Springer-Verlag, 2005.
27. F. Rodier. On the nonlinearity of Boolean functions. In *Proceedings of 2003 International Workshop on Coding and Cryptography (WCC 2003)*, pages 397–406, 2003.
28. C. E. Shannon. Communication theory of secrecy systems. *Bell System Tech. J.*, 28:656–715, 1949.
29. F.-X. Standaert, G. Piret, G. Rouvroy, J.-J. Quisquater, and J.-D. Legat. `ICEBERG` : An Involutive Cipher Efficient for Block Encryption in Reconfigurable Hardware. In M. Joye and J.-J. Quisquater, editors, *Cryptographic Hardware and Embedded Systems – CHES 2004*, volume 3156 of *Lecture Notes in Computer Science*, pages 279–298. Springer-Verlag, 2004.