# An Algebraic Masking Method to Protect AES Against Power Attacks*

Nicolas T. Courtois[1] and Louis Goubin[1,2]

[1] Axalto Crypto Research & Advanced Security, 36-38 rue de la Princesse, BP 45, F-78430 Louveciennes Cedex, France, courtois@minrank.org
[2] PRiSM Versailles University, 45 avenue des Etats-Unis, F-78035 Versailles Cedex, France, Louis.Goubin@prism.uvsq.fr

**Abstract.** The central question in constructing a secure and efficient masking method for AES is to address the interaction between additive masking and the inverse S-box of Rijndael. All recently proposed methods to protect AES against power attacks try to avoid this problem and work by decomposing the inverse in terms of simpler operations that are more easily protected against DPA by generic methods.
In this paper, for the first time, we look at the problem in the face, and show that this interaction is not as intricate as it seems. In fact, any operation, even complex, can be directly protected against DPA of any given order, if it can be embedded in a group that has a compact representation. We show that a secure computation of a whole masked inverse can be done directly in this way, using the group of homographic transformations over the projective space (but not exactly, with some non-trivial technicalities). This is used to propose a general high-level algebraic method to protect AES against power attacks of any given order.
**Key Words:** Rijndael, AES, inverse S-box, homographic transformations, linear fractional transformations, Möbius transformations, the zero-masking problem, Differential Power analysis, higher-order DPA.

## 1 Introduction

A secure implementation of (even a very secure) cryptographic algorithm, is by no means easy to achieve in portable cryptographic devices such as a smart cards. Indeed, it is hard to protect a secret that is entirely in the hands of a potential attacker. The nature of cryptography makes that all kinds of additional information, even very remotely correlated with the secret quantities manipulated in the cryptographic algorithm, are very likely to either directly leak information on the secret quantities, or indirectly, will help to improve some cryptographic attack. Moreover, in cryptographic devices such as smart cards, the performance and cost considerations make that there is little or no margin between the required

* This work was partially supported by the French Ministry of Research RNRT X-CRYPT project and by the European Commission via ECRYPT network of excellence IST-2002-507932.

level of security and the best known attack. Thus all kind of side channels are potentially fatal to the security.

The idea of side-channel attacks is very old and well known, for example it is possible to break many implementations of one-time pad by studying the output not in terms of zeros and ones, but with an oscilloscope... In 1998 practical side channel attacks for smart cards have been demonstrated by Kocher, Jaffe and Jun. They introduce a class of attacks called Power Analysis, using the power consumption traces to recover the key of an encryption scheme implemented in a smart card. The Power Analysis of first and higher order is already described by the authors in the original paper [22], and also in [23]. These attacks have been later applied to attack (and protect against such attacks) many secret key schemes, see for example [22, 28] and in particular to AES in [10, 5, 15, 1, 19, 34]. Before 2000, only first order attacks were demonstrated in practice.

In order to protect the secret key schemes against first order power attacks, two generic methods have been proposed: the transformed masking method [26, 1] and the duplication method [20, 21, 10, 11]. Moreover, in the particular case of AES, specific methods have been proposed [1, 38, 19, 6, 37, 35, 36, 30, 31]. Although few of these methods can be extended to higher order attacks, it was long considered that it was not a problem, since these elaborated attacks were believed hard to implement.

In 2000, T. Messerges [27] showed that second order attacks are perfectly doable in practice. In 2004 Waddle and Wagner improved this attack [39] and show that by using the Fast Fourier Transform (FFT) the attacker suffers only a logarithmic overhead in terms of runtime, compared to "ordinary" DPA. In particular, with the progressive replacement of DES by AES as a universal encryption standard, and with slow advances in hardware protections, efficient software protections of AES against higher order side-channel attacks have become a hot topic.

Up till now, the only proposed method to build high-order DPA-resistant implementations is due to Akkar and Goubin [2], see also [3]. It can be applied to the case of AES, however it uses many precomputed tables, which is memory-consuming in software and not really acceptable in hardware. In this paper we propose a new algebraic AES-specific masking method that may appear complex, yet it allows to achieve the same (very ambitious) goal without tables, and thus is suitable for both hardware and software implementations. The main contribution of this paper can be summarised as follows: a fully masked non-linear S-box of Rijndael can still be "embedded" (in a special way) in an algebraic group that has a

compact representation. This allows to construct efficient and somewhat mathematically elegant protections against power analysis of any given order. The paper is organized as follows:

- In the next section we recall the principles of first and higher order DPA attacks, summarise the existing counter-measures for AES, and define our objectives.
- In the third section we introduce the mathematical background necessary.
- In Sections 4-5 we describe the actual new solution. We present a security proof for the basic version and sketch a general solution for DPA of any given order.

## 2 AES and Side-channel Attacks

### 2.1 Known Side-channel Attacks

In this paper we limit to passive non-intrusive attacks against implementations of cryptographic algorithms, such as DPA, or any other passive side-channel attack.

**The Differential Power Analysis** (first order) attack works as follows: one records the power consumption of several runs of a cryptographic algorithm implemented on a smart card. Then one guesses a few bits of the secret or derived key, which allows to compute for each curve, some intermediate bit(s) that appear inside the cryptographic algorithm. For example the first bit of the entry of the first S-box. Then one separates the curves in two classes, these for which this bit is expected to be 0, and those for which this bit is expected to be 1. If the guess on the key is correct, one will be able to distinguish the two classes by various statistical techniques, if the guess was incorrect, both classes should look the same.

An algorithm is susceptible to be attacked by DPA if there is an intermediate value that depends on the plaintext and on key bits or derived bits.

**Higher Order DPA** (of order $k$) just generalizes the above attack: we will use up to $k$ points on the curves, or equivalently, up to $k$ intermediate variables. The information obtained on each of these values (e.g. it's power trace) should be efficiently combined in order to produce an output, that is correlated to some internal value/combination of values that again depends on the plaintext and on key or derived bits.

### 2.2 Adversarial Model for General Side-channel Attacks

In order to prevent all possible side-channel attacks, whatever is the leakage model, it is possible to assume that the (passive) adversary has full

access to some of the intermediate results of the computation. This cannot hold for all values, and Blömer, Merchan and Krummel do define on Fig. 1. in [6] a minimal set of two assumptions that are judged necessary to be able to protect the implementation of a cryptographic algorithm against side-channel attacks. We adapt and comment on these assumptions.

1. First of all, a random number generator must be available (that unlike what we read in [6] does not have to be really random and can be pseudo-random as long as it is not resettable, i.e. at least some real entropy is gathered and used by the device). This random generator must be protected against manipulation/perturbation (not necessarily against reading).
2. The secret key (and part of it) $K$ can be securely combined by a group operation with this random number $R$. This operation that manipulates $K$ should be secure both for reading and manipulation (!). Then both $R$ and $R \oplus K$ can go further unprotected.
   We observe that this assumption of [6] is not sufficient to protect against DPA of higher order. Our corrected assumption is follows. One can chose random $R_1, \ldots, R_k$, then securely compute $K' = K \oplus R_1 \oplus \ldots \oplus R_k$ and then the values $K', R_1, \ldots, R_k$ can go further unprotected.

   These assumptions seem to be acceptable and realistic, because:

0. smart cards may indeed protect some basic operations very carefully in hardware at the gate level,
1. it is difficult to produce really precise perturbations to random numbers that will not be removed by more randomisation,
2. we assumed that we manipulate the key only once with a random mask, and in a way that is independent of the plaintext. This makes DPA impossible and SPA can again probably be prevented by classical noise/randomisation techniques (without DPA the noise should cover the signal to recover),
3. finally even if the attacker were able to read $R$ (which is permitted by the model), we can have again DPA attacks on $R \oplus K$, the only place the key is exposed. This is linear and as such, it is known to be fundamentally much more resistant to DPA than non-linear operations, see [33].

We will assume that the power of the attacker is restricted to observing at most $k$ intermediate values that appear during the computation. Usually and in [6] it is also implicitly assumed that some "atomic" computations are secure and the only "observable" data are intermediate data at some level of abstraction. This assumption seems strong but we claim

that as far as protecting against power attacks of any given order can be achieved, it is not necessary and can be relaxed. If we assume that the code of the algorithm is public, each part of computation is some deterministic function of a small number of intermediate values, and will be secure if $k$ is high enough.

In addition doing this assumption at some level allows reasoning and security proofs much simpler. Then, refining the attack to an access to more internal states in a more precise decomposition may force us to use a protection with a higher $k$, but the overall method should work and remain the same.

### 2.3 A Representation of Rijndael

In this paper we will view AES (and other versions of Rijndael) in an abstract way, as a functional composition of three kind of operations denoted by X, I and L:

- X In Rijndael we simply XOR a byte of an expanded key with a byte of the current state. We assume that the expanded key is also computed by a composition of basic operations X,I and L.
- I Special substitution S-box that is called $Inv$. Many authors identify this function with the inverse $X \mapsto 1/X$ in the finite field. We will see that $Inv$ is not exactly the same thing as $1/X$ in $GF(256)$.
- L Linear operations (and more precisely linear or affine transformations over $GF(2)$, that are fixed and do not depend on the key or on the data).

Typically a DPA counter-measure for Rijndael is designed to protect an implementation of any cipher that is a composition of these operations.

### 2.4 Previously Proposed Masking Methods for AES

A natural method to protect (against side-channel attacks) a cipher using operations of type X and L is the transformed masking method [26, 1] in which any value inside the computation of the enciphering algorithm is masked with XOR by some random value. This masking method can also resist to higher-order DPA if the mask is split into several sub-masks as follows $R = R_1 \oplus \ldots \oplus R_k$, $R$ is never computed but all sub-masks are used one after another.

The main question that arises in all proposed masking methods for AES is how to protect the AES S-box, and more precisely how to protect the $Inv$ operation. In the past, the question has been answered with more or less success as follows:

1. Transformed Multiplicative Masking (TMM) of Akkar and Giraud [1]. Switching from additive (XOR) to multiplicative masking and back is possible and easy due to the ring structure of $GF(256)$. Unluckily, a multiplicative masking with respect to the multiplication in $GF(256)$ cannot be secure, because there is a special value $0 \in GF(256)$ that for any mask remains the same (i.e. is never masked).
2. In [38] Trichina *et al* proposed a simplified version of this method, which is not secure for exactly the same reason, see [19, 30].
3. Embedded Multiplicative Masking (EMM) of Golić and Tymen.
   One solution to the "zero-masking problem" is proposed in [19]. The method consists of a randomised embedding of $GF(256)$ in a larger ring in which zero can be represented by several possible elements (on two bytes). Then each byte taken separately has a uniform distribution.
4. A method based on univariate polynomials of Blömer, Merchan and Krummel [6]. This can be seen as a perfectly general method that can be applied to any S-box, as any function over a finite field can be seen as a univariate polynomial. Luckily, the polynomial representation $Inv(x) = x^{254}$ for any $x$ (including 0) has only one monomial. This makes the method more efficient than in the general case and suitable for Rijndael.
5. In [37] Trichina and Korkishko propose a software-oriented masking method based on log tables.
6. In [32] Rostovtsev and Shemyakina propose to use isomorphisms of the underlying finite field.
7. Tower Fields Methods by Oswald, Trichina, *et al* [35, 36, 30, 31] are designed for hardware implementations. In these methods, the computing of $Inv$ in $GF(2^{2k})$ is reduced to a secure computation with masked values of multiplications and inverses in $GF(2^k)$, by representing $GF(2^{2k})$ as a quadratic extension of $GF(2^k)$. Multiplications can be computed with additive masking and we are left with the problem of a secure computation of $Inv$ at the lower level. Two versions have been proposed:
   - In [35, 36] Trichina, Korkishko and Hee Lee go down to the field $GF(16)$. At this level the problem is solved by a completely general method in which it done as a masked computation of a combinatorial circuit with XOR and AND gates.
   - In [30, 31] Oswald, Mangard, Pramstaller and Rijmen go down to $GF(4)$ on which $Inv$ is multivariate linear (linear over $GF(2)$, not over $GF(4)$) and easy to protect.

### 2.5 New Method - Defining the Target

In this paper we present a novel algebraic masking method for $Inv$. Our method is strictly more powerful and somewhat mathematically more elegant that all the other known methods. After the initial (insecure) proposal of multiplicative masking all the methods subsequently proposed has become more and more generic in a sense that had to **decompose** the Rijndael S-box in simpler operations and protect each of them separately. On the contrary in our new method we operate at higher level, and protect directly more complex operations by representing them as elements of some group. This is made possible by exhibiting a new algebraic structure that though a bit tricky to use, allows to protect Rijndael against side-channel analysis.

In our method, we will be able to make masked computations of $Inv$ in one single step without "mask switching". We wish to go directly from a masked input to a masked output and thus the operation that we wish to protect a fully masked Rijndael inversion defined as:

$$F_{(R,R')} : X \mapsto Inv(X \oplus R) \oplus R'$$

Though complex, this operation can be directly protected. The basic idea is as follows: if we can embed this operation in some group, we can protect it against DPA of any fixed order: we represent it as a composition of several transformations, for example $k$, out of which any subset of $k-1$ transformations are just a set of random uniformly distributed operations.

We will exploit the group of homographic transformations with some non-trivial mathematical and implementation technicalities: strictly speaking $Inv$ does **not** belong to this group (while the real inverse in $GF(256)$ does). In our solution we will represent $F_{(R,R')}$ as a combination of a homographic transformation and of mapping that exchanges two points. This representation has a very interesting feature: these two type of mappings do "almost" commute (except that the two points to exchange are different). Thus we will also be able to mask completely the exchange of two points by additional homographic mappings.

## 3 Homographic Functions

In this paper we will work in $GF(2^n)$, for Rijndael S-boxes we have $n = 8$. The function $Inv$ can be defined over $GF(2^n)$ in the same way as in Rijndael with the usual $0 \mapsto 0$:

$$Inv(X) = \begin{cases} X^{-1} & \text{in } GF(2^n) \text{ if } X \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

We also have the real inverse function that can be either defined as

- a function $GF(256)^* \to GF(256)^*$ or

– as a projective function $\overline{GF(256)} \rightarrow \overline{GF(256)}$ with $\overline{GF(256)} = GF(256) \cup \{\infty\}$. This is our preferred version that will be used in this paper.

In mathematics the functions of the form $X \mapsto \frac{aX+b}{cX+d}$ are called *homographic functions* (a.k.a. linear fractional transformations or Möbius transformations, see [40]). It is well known that they can be represented by $2 \times 2$ matrices $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$. The composition of these functions is equivalent to multiplying their matrices. A cross-ratio of 4 pairwise different points $R(t,u,v,w) = \frac{t-u}{t-w} / \frac{v-u}{v-w}$ is known to be an invariant for such transformations, see [12, 4].

### 3.1 What is the Difference Between $Inv$ and $1/X$ ?

Unfortunately the function $Inv$ of Rijndael is **not** strictly speaking a homographic function. It is equal to a function of the form $X \mapsto \frac{aX+b}{cX+d}$ except in one point, when 0 is mapped to 0. This "completion" with $0 \mapsto 0$ has many important and non-trivial properties, see [12]. There are three ways of defining a practical "inverse" function for the finite field $GF(256)$:

1. We can have a bijection on 255 elements $GF(256)^* \rightarrow GF(256)^*$.
2. We can have a bijection on 256 elements $Inv : GF(256) \rightarrow GF(256)$ that is used in Rijndael.
3. We can have a bijection on 257 elements $\overline{GF(256)} \rightarrow \overline{GF(256)}$ with $\overline{GF(256)} = GF(256) \cup \{\infty\}$. The "real" inverse is defined as:

$$\overline{Inv}(X) = \begin{cases} X^{-1} & \text{if } X \notin \{0, \infty\} \\ 0 \mapsto \infty & \\ \infty \mapsto 0 & \end{cases}.$$

This is an eminently interesting version that is of central interest to us. We can compose this function with other homographic permutations defined as follows:

$$H_{a,b,c,d}(X) \stackrel{def}{=} \begin{cases} \frac{aX+b}{cX+d} & \text{if } X \notin \{-\frac{d}{c}, \infty\} \\ & \\ -\frac{d}{c} \mapsto \infty & \text{with } det\begin{pmatrix} a & b \\ c & d \end{pmatrix} \neq 0. \\ \infty \mapsto \frac{a}{c} & \end{cases}$$

The set of such invertible homographic mappings forms a group $\mathcal{H}$ under the usual composition law $\circ$. The matrix representation $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ in $GL_2(GF(2^n))$ is redundant and our group $\mathcal{H}$ is in fact isomorphic to a subgroup $SL_2(GF(2^n))$ of $GL_2(GF(2^n))$ in which all matrices are of determinant 1. Each element of $\mathcal{H}$ can be represented by "essentially" three elements of $GF(2^n)$.

### 3.2 Composition / Group Properties and Subtleties

We call an Almost-Invariant, any property that is invariant with a probability close to one. We have the following theorem:

**Theorem 3.3 (Jakobsen-Knudsen-Courtois, cf. [12]).**
For any function $Y = F(X)$ that composes in any order
(a) $N$ applications of $Inv$ in $GF(2^n)$,
(b) any number of XORs with different subkeys or constants,
there exist $(a, b, c, d) \in GF(2^n)^4$ such that:

$$\mathbb{P}_{X \in GF(2^n)} \left[ Y = \frac{aX + b}{cX + d} \mid Y = F(X) \right] \geq \left(1 - \frac{1}{2^n}\right)^N \geq \left(1 - \frac{N}{2^n}\right)$$

**Remark:** When $N$ is small, the probability is very close to 1 and the difference can be neglected, see [12, 24, 25]. Several related Theorems are given in [12]. One rather surprising fact is that, in spite of the fact the "homographic approximation" given by the Theorem 3.3 has rather excessively high probability, the probability will decrease with composition and can eventually become negligible. We have the following result due to Courtois [12]:

**Theorem 3.4 (The Group Generated by $Inv$ and XORs).**
The group generated by composing $Inv$ and constant/key additions is exactly the group of all permutations of $GF(2^n)$.

**Note:** The only difference between Theorem 3.3 and Theorem 3.4 is replacing $Inv$ by the real inverse. The homographic function $1/X$ over $\overline{GF(256)}$ composes well with constant/key additions and with constant/key multiplications to form a quite small group. When we replace $1/X$ by $Inv$, "nearly" the same thing, the group generated changes dramatically (becomes the group of all permutations). This fact is closely related to the famous question whether DES is a group or not and has some interesting consequences for block cipher cryptanalysis, see [12].

## 4 The New Masking Method

Let $\tau_{ab}$, be a function that that swaps two points $a \neq b$:

$$\tau_{ab}(X) = \begin{cases} X & \text{if } X \notin \{a, b\} \\ a \mapsto b & \\ b \mapsto a & \end{cases} .$$

We observe that $Inv$ is a restriction to $GF(256)$ of $\overline{Inv} \circ \tau_{O\infty}$. We have

$$Inv = \overline{Inv} \circ \tau_{O\infty} = \tau_{O\infty} \circ \overline{Inv}$$

Now, our target is to protect the whole operation as follows:

$$F_{(R,R')} : X \mapsto Inv(X \oplus R) \oplus R'$$

Though it is defined over $GF(2^n)$, nothing prevents us from extending it by deciding that $\overline{F_{(R,R')}}(\infty) = \infty$. We will be using operations in $\overline{GF(2^n)}$ to implement operations in $GF(2^n)$ and if we are able to securely implement $\overline{F_{(R,R')}}$ we obtain also a secure implementation of $F_{(R,R')}$.

Our new target is $\overline{F_{(R,R')}}$ and it can also be seen as a composition of an invertible homographic transformation and another mapping that swaps two points:

$$\overline{F_{(R,R')}} = H_{a,b,c,d} \circ \tau_{\infty R} = \tau_{\infty R'} \circ H_{a,b,c,d}$$

With $a, b, c, d$ that can bo computed explicitly as:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} R' & RR'+1 \\ 1 & R \end{pmatrix} \qquad \text{with } det \begin{pmatrix} R' & RR'+1 \\ 1 & R \end{pmatrix} = 1.$$

We will use the second decomposition (the first might be used as well). We have:

$$\overline{F_{(R,R')}} = \tau_{\infty R'} \circ H_{a,b,c,d}$$

It is important to see that it is not sufficient to find a secure implementation of $H_{a,b,c,d}$ and a secure implementation of $\tau_{\infty R'}$. This is because when the (real, not masked) value of the input of the Rijndael S-box is equal to 0, the output value of $H_{a,b,c,d}$ is always $\infty$ ($\infty$ is not masked by $R'$). This is a projective version of the "zero-masking problem". In a secure implementation both operations have to be "jointly" protected. In particular the implementation of $\tau_{\infty R'} \circ H_{a,b,c,d}$ must hide both points that are exchanged $\infty$ and $R'$, not only the point $R'$.

### 4.1 Joint Secure Implementation

First we describe a method for usual DPA (1st order). We assume that in the implementation of Rijndael each entry and each output of the $Inv$ function are protected by a couple of masks $R$ and $R'$ that vary from one S-box to another. We omit the description of how to protect the linear parts of the algorithm.

1. We pick a random $\alpha \in \mathcal{H}$.
2. We compute $\alpha^{-1} \in \mathcal{H}$.
3. By evaluating $\alpha^{-1}$ on $R'$ and $\infty$, we compute two points $g, h \in \overline{GF(2^n)}$ such that:
$$\tau_{\infty R'} = \alpha \circ \tau_{gh} \circ \alpha^{-1}$$
It is possible to show that they are a random and uniformly distributed couple of distinct points in $\overline{GF(2^n)}$. A proof of this fact is given in Appendix A.

4. We compute $H' = \alpha^{-1} \circ H_{a,b,c,d}$.
5. We store the sequence $\alpha, \tau_{gh}, H'$ in RAM as matrices of $SL_2(GF(2^n))$, except $\tau_{gh}$ that is stored as a couple of points in $\overline{GF(2^n)}$.
6. Our a secure implementation of $F_{(R,R')}$ is defined as the following sequence of transformations to be applied to $X$ in order from right to left:

$$\alpha, \quad \tau_{gh}, \quad H'$$

**Security of our countermeasure against DPA**. Observing each of the three transformations specified above is just a random transformation of some type that gives no information to the attacker.

It is also the case for $H'$ and $\tau_{gh} \circ H'$. This allows to see that all the intermediate values $H'(X)$ and $\tau_{gh}(H'(X))$ before the final masked output are fully de-correlated from $X$, from the real values inside the AES, and from the key.

In Appendix A we give a complete proof that this method is fully secure for (usual) DPA of order 1. This holds even while in our model the adversary is powerful enough to observe the whole values such as $H'$ (3 bytes).

**An attack of order 2:** Our method may a look a bit over-kill but isn't. We will show (which is not obvious to see at all) that it does not protect against DPA of order 2. We limit to show this in the case when the attacker is quite powerful, and can indeed see two full intermediate values appearing in our (high) level of abstraction. In particular, if the attacker "looks" at $X$ and $H_{a,b,c,d}$ that appear in 4., he can compute $H_{a,b,c,d}(X)$. This value is nearly random, except it will always be $\infty$ when a 0 appears inside the unprotected Rijndael. Then a version of zero-masking attack [19, 30] is possible.

### 4.2 Joint Secure Implementation - General Case

Again each Rijndael S-box is protected by a different couple of masks $R$ and $R'$. Moreover, for higher order DPA we assume that these masks are never computed and never stored during the computation but each of them is a sum of $k$ sub-masks $R_i$ such that

$$\begin{cases} R = R_1 \oplus \ldots \oplus R_k \\ R' = R'_1 \oplus \ldots \oplus R'_k. \end{cases}$$

Each time we wish to manipulate $R$ to XOR or multiply it with some value, we do not compute $R$, but we use the group/ring structure to obtain the result by successively adding the parts corresponding to each $R_i$. (In addition the order of the $R_i$ used can be also randomised).

Here is how to compute $F_{(R,R')}$ securely with respect to the DPA of order $k$. We present the simplest (but presumably not the fastest) solution.

1. We pick $k$ random $\alpha_i \in \mathcal{H}$ and $k$ random $\beta_i \in \mathcal{H}$.
2. Let $\alpha = \bigcirc_{i=1..k}\alpha_i$ and $\alpha^{-1} = \bigcirc_{i=k..1}\alpha_i^{-1}$. Let $\beta = \bigcirc_{i=1..k}\beta_i$ and $\beta^{-1} = \bigcirc_{i=k..1}\beta_i^{-1}$. We will **never** compute any of these four values.
3. We put
$$G_i = \begin{pmatrix} 1 & R_i \\ 0 & 1 \end{pmatrix} \qquad G'_i = \begin{pmatrix} 1 & R'_i \\ 0 & 1 \end{pmatrix}$$
and we will **never** compute $H_{a,b,c,d}$ (cf. the attack above) but observe that:
$$H_{a,b,c,d} = G'_k \circ \cdots \circ G'_1 \circ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \circ G_1 \circ \cdots \circ G_k$$
4. Instead, we will compute iteratively $H'' = \alpha^{-1} \circ H_{a,b,c,d} \circ \beta^{-1}$ in a secure manner.
For this we first compute $\alpha_1^{-1} \circ G'_k$ then $\alpha_2^{-1} \circ (\alpha_1^{-1} \circ G'_k) \circ G'_{k-1}$,
Up to: $A = \alpha^{-1} \circ G'_k \circ \cdots \circ G'_1$.
Then we compute successively $G_k \circ \beta_k^{-1}$ then $G_{k-1} \circ (G_k \circ \beta_k^{-1}) \circ \beta_{k-1}^{-1}$ up to $B = G_1 \circ \cdots \circ G_k \circ \beta^{-1}$. Finally we compute
$$H'' = A \circ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \circ B.$$
5. We need to securely evaluate $\alpha^{-1}$ on $R'$ and $\infty$, to compute two points $g$, $h \in \overline{GF(2^n)}$ such that:
$$\tau_{\infty R'} = \alpha \circ \tau_{gh} \circ \alpha^{-1}$$
For $\infty$ we simply successively compute $\alpha_1^{-1}(\infty), \alpha_2^{-1}(\alpha_1^{-1}(\infty)), \ldots$.
For $R'$, that is $\neq \infty$ we embed $R'$ into $\mathcal{H}$ as already defined simple translations $G'_i$. Then we successively compute $\alpha_1^{-1} \circ A_1$, then $\alpha_2^{-1} \circ (\alpha_1^{-1} \circ A_1) \circ A_2$ etc..
At the end we get some $G' \in \mathcal{H}$ and compute $h = G'(0)$.
6. The following sequence of transformations to be stored in RAM, is a secure implementation of $F_{(R,R')}$ to be applied to $X$ in order from right to left:
$$\alpha_1, \ldots, \alpha_k, \quad \tau_{gh}, \quad H'', \quad \beta_1, \ldots, \beta_k$$

## 5 Implementation Issues

**Representation of $\overline{GF(256)}$.** The representation can still be on 8-bits, and the case $\infty$ can be hard-coded into algorithm code. Instead of writing some variable $a = \infty$ and the evaluating some homographic function at this point, we will directly compute the result of the next operation at $\infty$. We will use conditions to check which case we are in, yet the

implementation will be secure because **any** implementation of our way of rewriting Rijndael should be secure.

**Representation of $\mathcal{H}$.** Each element of $\mathcal{H}$ can be represented by an element of $SL_2(GF(2^n))$ which amounts to store "essentially" three elements of $GF(2^n)$.

## 6 Conclusion

In this paper we studied the interaction between additive masking and the inverse S-box of Rijndael the happens to be less complex than it seems. It allows to construct a method to protect in one single step the whole masked inverse against all passive side-channel attacks (e.g. DPA, DEMA, etc.). This is achieved by embedding (with additional non-trivial technicalities) this operation in a group of homographic transformations over the projective space that happens to have a compact representation (essentially 3 bytes).

**Further Research:** This paper is a proof of concept for a new non-trivial algebraic masking method. Due to the space requirements we do not present yet an optimised ready-to-use solution for DPA of order 1 or 2. With our methodology: embedding in a group, it is clear that we can construct a provably secure masking solution in the spirit of [6] yet secure against attacks of any given order, not only first-order DPA. At present we give a security proof for order 1 and sketch a general solution. It is also clear that our solution is reasonably fast and practical for both software and hardware implementations (there are no large tables to generate). A specific optimised solution with precise comparison to other known masking schemes will be given in a separate paper.

## References

1. Mehdi-Laurent Akkar, C. Giraud: *An Implementation of DES and AES secure against Some Attacks*. In CHES'2001, LNCS 2162, pp. 309-318, Springer, 2001.
2. Mehdi-Laurent Akkar, Louis Goubin, *A Generic Protection against High-Order Differential Power Analysis*. In FSE'2003, LNCS 2887, Springer, pp. 192-205, 2003.
3. Mehdi-Laurent Akkar, Régis Bevan, Louis Goubin, *Two Power Analysis Attacks against One-Mask Methods*. In FSE'2004, LNCS 3017, Springer, pp. 332-347, 2004.
4. Kazuaro Aoki, Serge Vaudenay: *On the Use of GF-Inversion as a Cryptographic Primitive*, SAC 2003, LNCS 3006, pp. 234-247, Springer 2004.
5. Eli Biham, Adi Shamir: *Power Analysis of the Key Scheduling of the AES Candidates*. the second Advanced Encryption Standard (AES) Candidate Conference, March 1999. Available from `http://csrc.nist.gov/encryption/aes/round1/Conf2/aes2conf.htm`

6. Johannes Blömer, Jorge Guajardo Merchan, Volker Krummel: *Provably Secure Masking of AES,* In SAC'2004, LNCS 3357, pp. 69-83, Springer, 2005. Available on `eprint.iacr.org/2004/101`.
7. Johannes Blömer, Jean-Pierre Seifert: *Fault based cryptanalysis of the Advanced Encryption Standard,* In Proceedings of Financial Cryptography 2004, LNCS 2742, pp 162-181, Springer 2004. Available on `eprint.iacr.org/2002/075`.
8. Eric Brier, Christophe Clavier, Francis Olivier: *Optimal Statistical Power Analysis,* Available on `eprint.iacr.org/2003/152`.
9. Vincent Carlier, Hervé Chabanne, Emmanuelle Dottax, Hervé Pelletier: *Electromagnetic Side Channels of an FPGA Implementation of AES,* `eprint.iacr.org/2004/145`.
10. Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, Pankaj Rohatgi: *A Cautionary Note Regarding Evaluation of AES Candidates on Smart-Cards*. the second Advanced Encryption Standard (AES) Candidate Conference, March 1999. Available from `http://csrc.nist.gov/encryption/aes/round1/Conf2/aes2conf.htm`
11. Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, Pankaj Rohatgi: *Towards Sound Approaches to Counteract Power-Analysis Attacks*. In Crypto 99, LNCS 1666, pp. 398-412, Springer, 1999.
12. Nicolas Courtois: *The Inverse S-box, Non-linear Polynomial Relations and Cryptanalysis of Block Ciphers,* in AES 4 Conference, Bonn May 10-12 2004, LNCS 3373, pp. 170-188, Springer, 2005.
13. Nicolas Courtois: *The Inverse S-box and Two Paradoxes of Whitening,* Long extended version of the Crypto 2004 rump session presentation, *Whitening the AES S-box,* Available at `http://www.minrank.org/invglc_rump_c04.zip`. Also explained in Appendix B of the extended version of [12].
14. Joan Daemen, Vincent Rijmen: *AES proposal: Rijndael,* The latest revised version of the proposal is available on the internet, `http://csrc.nist.gov/encryption/aes/rijndael/Rijndael.pdf`
15. Joan Daemen, Vincent Rijmen: *Resistance Against Implementation Attacks: A Comparative Study of the AES Proposals,* In 2nd AES Candidate Conference, March 1999. Available from `http://csrc.nist.gov/encryption/aes/round1/Conf2/aes2conf.htm`
16. Pierre Dusart, Gilles Letourneux, Olivier Vivolo: *Differential Fault Analysis on A.E.S.,* Rapport interne n2003-01, Laco, Université de Limoges, France. Available on `eprint.iacr.org/2003/010`.
17. Christophe Giraud: *DFA on AES,* In Proceedings of AES4 Conference, LNCS 3373, Springer, 2005. Available on `eprint.iacr.org/2003/008`.
18. Jovan Dj. Golić: *Multiplicative Masking and Power Analysis of AES,* claimed as being presented at an (internal) Gemplus Quarterly meeting, La Ciotat, France, October 30-31, 2001, In CHES 2002, LNCS 2523, pp. 198-212, Springer, 2003. Available at `http://eprint.iacr.org/2002/091/`.
19. Jovan Dj. Golić, Christophe Tymen: *Multiplicative Masking and Power Analysis of AES,* In CHES 2002, LNCS 2523, pp. 198-212, Springer, 2003. Available at `http://eprint.iacr.org/2002/091/`.
20. Louis Goubin, Jacques Patarin: *Procédé de sécurisation d'un ensemble électronique de cryptographie à clé secrète contre les attaques par analyse physique*. European Patent, Axalto, (previously SchlumbergerSema), February 4th, 1999, Publication Number: 2789535.
21. Louis Goubin, Jacques Patarin: *DES and Differential Power Analysis – The Duplication Method*. In CHES'99, LNCS 1717, pp. 158-172, Springer, 1999.

22. Paul Kocher, Joshua Jaffe, Benjamin Jun: *Introduction to Differential Power Analysis and Related Attacks*. Technical Report, Cryptography Research Inc., 1998. Available from `http://www.cryptography.com/dpa/technical/index.html`
23. Paul Kocher, Joshua Jaffe, Benjamin Jun: *Differential Power Analysis*. In Crypto 99, LNCS 1666, pp. 388-397, Springer, 1999.
24. Thomas Jakobsen, Lars Knudsen: *Attacks on Block Ciphers of Low Algebraic Degree,* Journal of Cryptology 14(3): 197-210 (2001).
25. Thomas Jakobsen, Lars R. Knudsen: *The Interpolation Attack on Block Ciphers,* FSE 97, LNCS 1267, Springer, pp.28-40, 1997.
26. Thomas S. Messerges: *Securing the AES Finalists Against Power Analysis Attacks,* FSE'00, LNCS 1978, pp. 150-164, Springer, 2001.
27. Thomas S. Messerges: *Using second-Order Power Analysis to Attack DPA Resistant software,* . In CHES'2000, LNCS 1965, pp. 238-251, Springer, 2000.
28. Thomas S. Messerges, Ezzat A. Dabbish, Robert H. Sloan: *Investigations of Power Analysis Attacks on Smartcards*. the USENIX Workshop on Smartcard Technology, pp. 151-161, May 1999. Available from `http://www.eecs.uic.edu/ tmesserg/papers.html`
29. Thomas S. Messerges, Ezzat A. Dabbish, Robert H. Sloan: *Power Analysis Attacks of Modular Exponentiation in Smartcards*. In CHES'99, LNCS 1717, pp. 144-157, Springer, 1999.
30. Elisabeth Oswald, Stefan Mangard, Norbert Pramstaller: *Secure and Efficient Masking of AES - A Mission Impossible ?,* `eprint.iacr.org/2004/134`.
31. Elisabeth Oswald, Stefan Mangard, Norbert Pramstaller, Vincent Rijmen: *A Side-Channel Analysis Resistant Description of the AES S-box,* In FSE 2005, to appear in LNCS Springer, 2005.
32. A.G. Rostovtsev and O.V. Shemyakina; *AES side channel attack protection using random isomorphisms,* Available on `http://eprint.iacr.org/2005/087.pdf`
33. Emmanuel Prouff: *DPA attacks and S-boxes*, In FSE'05, to appear in LNCS, Springer, 2005.
34. François-Xavier Standaert, Siddika Berna Ors, Bart Preneel: *Power Analysis of an FPGA Implementation of Rijndael: Is Pipelining a DPA Countermeasure ?* In CHES'2004, LNCS 3156, pp. 30-44, Springer, 2004.
35. Elena Trichina: *Combinational Logic Design for AES SubByte Transformation on Masked Data,* . Available on `eprint.iacr.org/2003/236`.
36. Elena Trichina, Tymur Korkishko, Kyung Hee Lee: *Small Size, Low Power, Side Channel-Immune AES Coprocessor, Design and Synthesis Results,* 4th AES Conference, LNCS 3373, to appear in Springer, 2005.
37. Elena Trichina, Lesya Korkishko: *Secure and Efficient AES Software Implementation for Smart Cards,* In Proceedings of WISA 2004, LNCS 3325, Pages 425-439, Springer, 2005. Available on `eprint.iacr.org/2004/149`.
38. Elena Trichina, Domenico De Seta, Lucia Germani: *Simplifed Adaptive Multiplicative Masking for AES,* In CHES 2002,LNCS 2535, pp. 187-197, Springer, 2003.
39. Jason Waddle, David Wagner: *Towards Efficient Second-Order Power Analysis,* In CHES'2004, LNCS 3156, pp. 1-15, Springer, 2004.
40. The Wikipedia entry "Möbius transformation", freely available at `http://en.wikipedia.org/wiki/Mobius_group`

## A Appendix: Proof of Security for First-Order DPA

In all this section, we will denote $q = 2^n$ (for AES, we have $n = 8$). We will make use of the following fundamental lemmas:

**Lemma 1.** *Let $(u_1, u_2, u_3)$ and $(v_1, v_2, v_3)$ two triplets of elements in $\overline{GF(q)}$, such that $u_j \neq u_k$ and $v_j \neq v_k$ whenever $j \neq k$. Then there is exists a unique $\beta \in \mathcal{H}$ such that $\beta(u_i) = v_i$ ($i = 1, 2, 3$).*

*Proof.* See [40]. □

**Lemma 2.** *Let $u \in \overline{GF(q)}$ and $v \in \overline{GF(q)}$. There are exactly $q(q-1)$ homographic transformations $\beta \in \mathcal{H}$ such that $\beta(u) = v$.*

*Proof.* Let us define $u_1 = u$ and $v_1 = v$. Moreover, choose arbitrary values for $u_2$ and $u_3$, such that $u_1$, $u_2$, $u_3$ are pairwise distinct.

Let $A_v$ be the set of pairs $(v_2, v_3)$ such that $(v, v_2, v_3)$ are pairwise distinct. It is easy to see that $A_v$ contains $q(q-1)$ elements: there are $q$ choices for $v_2 \in \overline{\mathrm{GF}(q)} \setminus \{v\}$ and $q-1$ choices for $v_3 \in \overline{\mathrm{GF}(q)} \setminus \{v, v_2\}$.

Let $\mathcal{H}_{u,v}$ the set of homographic transformations which map $u$ onto $v$. From Lemma 1, there is a one-to-one correspondance between $A_v$ and $\mathcal{H}_{u,v}$ (for a given $(v_2, v_3)$, consider the transformation which maps $(u_1, u_2, u_3)$ onto $(v_1, v_2, v_3)$). Therefore $\mathcal{H}_{u,v}$ also contains $q(q-1)$ elements. □

**Security proof (first-order DPA).**
Now, let us recall the first-order method (see section 4). Let

$$F_{(R,R')} : X \mapsto \mathrm{Inv}(X \oplus R) \oplus R'.$$

By extending the function $F$ to $\overline{\mathrm{GF}(q)} = \mathrm{GF}(q) \cup \{\infty\}$, we obtain a function $\overline{F_{(R,R')}}$ such that

$$\overline{F_{(R,R')}} = \tau_{\infty,R'} \circ H_{a,b,c,d},$$

the homographic transformation $H_{a,b,c,d}$ corresponding to the following matrix:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} R' & RR'+1 \\ 1 & R \end{pmatrix} \in \mathrm{SL}_2(\mathrm{GF}(q)).$$

To study the resistance of the implementation against first-order DPA attacks, we prove that the distribution of each intermediate value occurring during the computation of $Y = F(X)$ neither depends on $X \oplus R$ nor on $Y \oplus R'$ (*i.e.* the input/output values without mask).

**Precomputations (*i.e.* computations which do not depend on $X$).**
For given $R$ and $R'$, the following steps are performed:

1. Define $\alpha \in \mathcal{H}$ by randomly and uniformly choosing a matrix $\begin{pmatrix} p & q \\ r & s \end{pmatrix} \in \mathrm{SL}_2(\mathrm{GF}(q))$.
2. Define $H' = \alpha^{-1} \circ H_{a,b,c,d} \in \mathcal{H}$, by computing $\begin{pmatrix} a' & b' \\ c' & d' \end{pmatrix} \in \mathrm{SL}_2(\mathrm{GF}(q))$ with
$$\begin{cases} a' = -sR' + q \\ b' = -s(RR' + 1) + qR \\ c' = rR' - p \\ d' = r(RR' + 1) - pR \end{cases}$$
Since $\mathcal{H}$ is a group, the distribution of $H' \in \mathcal{H}$ is also uniform.
3. Compute
$$g = \frac{-sR' + q}{rR' - p} = \alpha^{-1}(R') \qquad \text{and} \qquad h = \frac{-s}{r} = \alpha^{-1}(\infty).$$
For a given value $g \in \overline{\mathrm{GF}(q)}$, from Lemma 2, there are exactly $q(q-1)$ transformations $\alpha \in \mathcal{H}$ such that $g = \alpha^{-1}(R')$. Therefore $g$ is uniformly distributed in $\overline{\mathrm{GF}(q)}$. The same is true for $h = \alpha^{-1}(\infty)$.

**Note 1:** We have $g = \alpha^{-1}(R')$ and $h = \alpha^{-1}(\infty)$, so that $\alpha \circ \tau_{g,h} = \tau_{\infty,R'} \circ \alpha$.

**Note 2:** By using lemma 1, we can also prove that the *pairs* $(g, h)$, $g \neq h$, are uniformly ditributed. Let $(u, v)$ two distinct elements of $\overline{\mathrm{GF}(q)}$. Considering $(u_1, u_2) = (u, v)$, $(v_1, v_2) = (g, h)$ and an arbitrary $u_3 \notin \{u_1, u_2\}$, Lemma 1 proves that exactly $q-1$ homographic transformations exist in $\mathcal{H}$ sending $(u, v)$ onto $(g, h)$ (one for each possible choice of $v_3 \notin \{v_1, v_2\}$).

**Computations depending on $X$.**
To compute $Y = F(X)$, the following steps are performed:

1. Compute $Z = H'(X)$.
2. Compute $T = \tau_{g,h}(Z)$.
3. Compute $Y = \alpha(T)$.

Let us detail the intermediate values $Z$ and $T$.

– The first step is:

$$Z = H'(X) = \frac{a'X + b'}{c'X + d'}.$$

For each fixed pair $(X, Z)$ with $X \in \mathrm{GF}(q)$ and $Z \in \overline{\mathrm{GF}(q)}$, Lemma 2 indicates that exactly $q(q-1)$ homographic functions $H'$ exist (in $\mathcal{H}$) such that $H'(X) = Z$. Therefore, the distribution of $Z$ is uniform on $\overline{\mathrm{GF}(q)}$ (the probability being taken on the possible choices of $\alpha$).
**Note.** We also have:

$$Z = \frac{(-sR' + q)X + (-s(RR' + 1) + qR)}{(rR' - p)X + (r(RR' + 1) - pR)} = \frac{(-sR' + q)(X + R) - s}{(rR' - p)(X + R) + r}.$$

– In the second step

$$T = \tau_{g,h}(Z)$$

the transposition $\tau_{g,h}$ acts as the identity map, except in one special case: $Z = h = \frac{-s}{R}$ (corresponding to $X = R$). For each fixed pair $(X, T)$ with $X \in \mathrm{GF}(q)$ and $T \in \overline{\mathrm{GF}(q)}$, there are two cases:

- First case: if $X \neq R$, we must have $Z = T$ (because $Z \neq \frac{-s}{R}$ whatever we choose for $\alpha$). Therefore, from Lemma 2, we have exactly $q(q-1)$ transformations $\alpha$ such that $\tau_{g,h} \circ H'(X) = T$.
- Second case: if $X = R$, then $Z = \frac{-s}{R} = h$, thus $T = g = \alpha^{-1}(R')$. Therefore, from Lemma 2, we have exactly $q(q-1)$ transformations $\alpha$ such that $\tau_{g,h} \circ H'(X) = T$.

In both cases, the distribution of $T$ is uniform on $\overline{\mathrm{GF}(q)}$ (the probability being taken on the possible choices of $\alpha$).
**Note.** It should be noted that the other special case $Z = g = \frac{-sR'+q}{rR'-p}$ never happens (this would correspond to $X = \infty$, which is impossible in our scenario).