# Mixing Additive and Multiplicative Masking for Probing Secure Polynomial Evaluation Methods

Axel Mathieu-Mahias, Michael Quisquater

## ▶ To cite this version:

Axel Mathieu-Mahias, Michael Quisquater. Mixing Additive and Multiplicative Masking for Probing Secure Polynomial Evaluation Methods. IACR Transactions on Cryptographic Hardware and Embedded Systems, IACR, 2018, 2018 (1), pp.175-208. 10.13154/tches.v2018.i1.175-208 . hal-02177722

**HAL Id: hal-02177722**

**https://hal.archives-ouvertes.fr/hal-02177722**

Submitted on 9 Jul 2019

# Mixing Additive and Multiplicative Masking for Probing Secure Polynomial Evaluation Methods

Axel Mathieu-Mahias and Michaël Quisquater [*]

University of Versailles-St-Quentin-en-Yvelines
{axel.mathieu-mahias,michael.quisquater}@uvsq.fr

**Abstract.** Masking is a sound countermeasure to protect implementations of block-cipher algorithms against Side Channel Analysis (SCA). Currently, the most efficient masking schemes use Lagrange's Interpolation Theorem in order to represent any S-box by a polynomial function over a binary finite field. Masking the processing of an S-box is then achieved by masking every operation involved in the evaluation of its polynomial representation. While the common approach requires to use the well-known Ishai-Sahai-Wagner (ISW) scheme in order to secure this processing, there exist alternatives. In the particular case of power functions, Genelle, Prouff and Quisquater proposed an efficient masking scheme (GPQ). However, no generalization has been suggested for polynomial functions so far. In this paper, we solve the open problem of extending GPQ for polynomials, and we also solve the open problem of proving that both the original scheme and its variants for polynomials satisfy the $t$-SNI security definition. Our approach to extend GPQ is based on the cyclotomic method and results in an alternate cyclotomic method which is three times faster in practice than the original proposal in almost all scenarios we address. The best-known method for polynomial evaluation is currently CRV which requires to use the cyclotomic method for one of its step. We also show how to plug our alternate cyclotomic approach into CRV and again provide an alternate approach that outperforms the original in almost all scenarios. We consider the masking of $n$-bit S-boxes for $n \in [4; 8]$ and we get in practice 35% improvement of efficiency for S-boxes with dimension $n \in \{5, 7, 8\}$ and 25% for 6-bit S-boxes.

**Keywords:** Side-channel countermeasure · Masking · Polynomial evaluation · Probing security · Block cipher · Authenticated encryption.

## 1 Introduction

Side channel attacks exploit physical leakages of a device during the computation. This leakage may unveil sensitive information on the data manipulated by an implementation. Since their introduction in the late nineties [Koc96,KJJ99], numerous side-channel attacks have been successfully mounted on cryptosystems, motivating the design of provably secure countermeasures against such realistic threats.

The most common strategy is based on masking. Such a countermeasure randomly splits every sensitive variable into several shares such that all of them are required to retrieve any information about the original data. Internal computations no longer operate directly on complete data but rather on their corresponding shares. The number of random shares used to split (or mask) a sensitive variable is referred to as the masking order. Typically, a masking scheme of order greater or equal to $d$ resists to an attack of order $d$ (that exploits

[*] This research has been partially funded by ANR project ANR-14-CE28-0015 BRUTUS.

IACR Transactions on Cryptographic Hardware and Embedded Systems ISSN 2569-2925,
Vol. 2018, No. 1, pp. 175–208
DOI:10.13154/tches.v2018.i1.175-208

physical information from $d$ leakage points of a circuit). Indeed, without all the (random) shares and the associated masked sensitive variable, an attacker gets information he can only relate to random values.

Another advantage is that higher-order masking schemes, for which sensitive data are split into $d$ shares (with $d > 2$), are sound countermeasures in realistic leakage models. One of them is the noisy leakage model for which it has been shown in [CJRR99, PR13] that the complexity of recovering sensitive data grows exponentially with the number of shares. There also exists the more theoretical but simpler probing model introduced by Ishai, Sahai and Wagner in [ISW03] for which an attacker exploit the information carried through circuit wires during computations. Ishai et al. were interested in securing circuits against an adversary who can probe a limited number of wires. More precisely, they showed how to transform any boolean circuit of size $|C|$ into a larger construction of size $\mathcal{O}(|C| \cdot t^2)$ that is secure against an attacker that is able to probe $t$ wires at a time. Their security proof implies the simulation of transformed `AND` gates processing variables splitted into $d \geq 2t + 1$ shares. A secure AND gate has size $\mathcal{O}(t^2)$ and shall be referred to as the `ISW` gadget in the following. The probing model has been extensively used to prove the security of numerous constructions. More recently, the work of Duc, Dziembowski and Faust [DDF14] showed a security reduction from the noisy leakage model to the model of probing adversaries. This result renders the use of the probing security more legitimate than before. Yet, proving the security of cryptographic algorithms in the probing model remains a challenge. Namely, a construction is not necessarily secure even if each of its basic blocks has been proven to be secure. The composition of basic blocks may induces flaws in a scheme. This issue has been recently addressed in [BBD$^+$15] in which was introduced a stronger security definition for the probing model, referred to as $t$-`SNI`, and used to guarantee that the composition of masked blocks remains secure.

A cryptographic algorithm is a sequence of affine and nonlinear functions. Linear/affine functions are simply masked by applying any of such functions to every share separately. However, the processing of masked nonlinear functions is less straightforward. The input shares of a nonlinear transformation have to be handled carefully in order to guarantee the security of the masking countermeasure.

In [RP10], Rivain and Prouff proposed the first efficient and provably secure masking scheme in the probing model for AES whose S-box consists in computing inversions in the finite field $\mathbb{F}_{2^8}$. Their idea was to express the corresponding inverse function $x \mapsto x^{254}$ as a sequence of squares and nonlinear multiplications over $\mathbb{F}_{2^8}$. While squares are linear functions and are therefore easy to mask, they adapted the `ISW` multiplication gadget over $\mathbb{F}_2$ to the desired extension field $\mathbb{F}_{2^8}$ in order to mask the nonlinear multiplications. The proposed scheme was originally supposed to achieve $d^{\text{th}}$ order security. However, the composition of their mask refreshing procedure with the `ISW` multiplication gadget induced a security flaw in the overall scheme [CPRR13]. A solution proposed in the same article was to avoid the use of the mask refreshing gadget by adapting the `ISW` scheme. The resulting secure multiplications are referred to as bilinear multiplications in the literature. It was only recently that the original scheme (without the bilinear multiplications) has been fixed in [BBD$^+$15]. Namely, they proved that the composition of the multiplication gadget with a different mask refreshing procedure results in a safe construction with $d \geq t+1$ shares by showing that both previous gadgets satisfy the $t$-`SNI` security definition.

The approach followed by Rivain and Prouff was extended to any $n$-bit S-box by Carlet, Goubin, Prouff, Quisquater and Rivain (`CGPQR`) in [CGP$^+$12]. They showed that any $n$-bit S-box can be expressed as a sequence of linear transformations and nonlinear multiplications over $\mathbb{F}_{2^n}$, that is represented by a polynomial $S(x) = \sum_i a_i x^i$ over $\mathbb{F}_{2^n}$ using Lagrange's interpolation theorem. Thus, the CGPQR masking scheme consists in evaluating securely such polynomial over $\mathbb{F}_{2^n}$ by masking with ISW every nonlinear multiplication involved in the corresponding sequence. However, as the masking order grows, the secure processing of nonlinear multiplications quickly becomes expensive. Therefore, they also described two efficient heuristics called cyclotomic and parity-split methods that optimize the number of nonlinear multiplications required to evaluate the polynomial representation of generic S-boxes. Several methods have also improved CGPQR by further optimizing this number of nonlinear multiplications. Roy and Vivek [RV13] further reduced the complexity of several well known S-boxes and the currently best-known method for fast polynomial evaluation in $\mathbb{F}_{2^n}$ has been proposed by Coron Roy and Vivek in [CRV14] and is referred to as the CRV method in the rest of the paper. Recently, other constructions of multiplication circuits in finite fields than ISW have been proposed [BBP$^+$17]. However, ISW remains the most efficient $t$-SNI scheme for orders of practical interest (i.e. orders 1, 2 and 3).

Different approaches can be used as alternatives to the higher-order CGPQR masking scheme [ISW03, GM11, PR11, GPQ11b, Cor14, BFG15, CPRR15]. Among them, the higher-order masking scheme introduced by Genelle, Prouff and Quisquater (GPQ) in [GPQ11b] is a more efficient alternative for the AES than [RP10] (see [GSF14]). The GPQ scheme is particularly efficient to mask S-boxes which are power functions but no generalization to mask generic S-boxes has been proposed so far.

## 1.1 Our contributions

In this paper, we begin to prove the security of the GPQ masking scheme in the probing model under the stronger $t$-SNI security definition. Then, we show how to solve the open problem of extending GPQ to mask generic S-boxes (not only power functions). Specifically, our approach is based on the generic cyclotomic method proposed in [CGP$^+$12], whose security so far relied on the ISW scheme. We show how to refine the use of GPQ when combined with the cyclotomic method so that it results in an alternate cyclotomic approach for polynomial evaluation over $\mathbb{F}_{2^n}$ that no longer requires ISW. We provide a description of our construction and prove that it satisfies the $t$-SNI requirements. We also provide an alternate approach for CRV. The latter requires the cyclotomic method in order to build a set of monomials in one of its steps. We show how to plug our alternate cyclotomic method into CRV, in order to efficiently compute those power functions with our previous construction. Moreover, our approach allows us to derive new parameters for CRV considered as irrelevant with the original proposal, but which are well-suited in our case. We then show that our alternate CRV construction is $t$-SNI. In practice, we consider the same scenarios for both our alternate approaches. We report the cost of polynomial evaluations with our approaches compared to the original ones where S-boxes are of dimension $n \in [4; 8]$. We improve by a factor 3 the efficiency of the original cyclotomic method in almost all scenarios (for $n \in \{5, 6, 7, 8\}$) and we improve by 35% the efficiency of the original CRV for S-boxes of dimension $n \in \{5, 7, 8\}$ and 25% for 6-bit S-boxes.

## 1.2 Road map

The paper is organized as follows. Section 2 provides background notions on masking and surrounding the probing model. We present GPQ in Section 3 along with our $t$-SNI security proof. In Section 4 we recall aspects of the cyclotomic method, we address the extension of GPQ to the masking of generic S-boxes and we give security proofs regarding our alternate cyclotomic construction. In Section 5, we describe the CRV method before showing how to derive an alternate approach that also enables to consider new parameters, and we also provide security proofs. Section 6 reports implementation results using our alternate approaches compared to the originals for S-boxes of dimension $n \in [4; 8]$. Eventually Section 7 concludes the paper.

# 2 Basics and Definitions

In this paper, $n$ denotes the bit-length of processed data. By default, variables in this paper are assumed to be defined in the field $\mathbb{F}_{2^n} \cong (\mathbb{F}_2[x]/(p(x)), \oplus, \otimes)$, where $p(x)$ is an irreducible polynomial of $\mathbb{F}_2[x]$ of degree $n$, $\oplus$ is the bitwise XOR operation and $\otimes$ denotes the polynomial multiplication modulo $p(x)$. These variables can also sometimes be viewed as elements of the vector space $\mathbb{F}_2^n$ defined over the field $(\mathbb{F}_2, \oplus, \odot)$, where $\odot$ is the AND operation. Some transformations may involve $n$-bit operations $\mathtt{XOR}, \mathtt{AND}$ which shall be referred to by $\oplus^n, \odot^n$. The inverse of an element $x \in \mathbb{F}_{2^n}^*$ for the law $\otimes$ is $x^{-1}$ where $\mathbb{F}_{2^n}^*$ denotes the set of invertible elements of $\mathbb{F}_{2^n}$.

## 2.1 Basics on masking

As explained in the introduction, the masking countermeasure splits every sensitive variable $x$ into $d = t+1$ shares $x_0, \ldots, x_d$ in such a way that the following relation is satisfied for a group operation $\perp$. Namely,

$$x_0 \perp x_1^{-1} \perp \ldots \perp x_d^{-1} = x \,. \tag{1}$$

where $x_i^{-1}$ denotes the inverse of $x_i$ w.r.t $\perp$. Usually, the $d$ shares $x_1 \ldots, x_d$ are randomly generated and $x_0$ is processed so that (1) is satisfied. In this paper, $\perp$ either denotes the field addition $\oplus$ or the field multiplication $\otimes$. When $\perp = \oplus$ (resp. $\perp = \otimes$), the relation (1) induces an additive masking (resp. a multiplicative masking) of $x$. A $(d+1)$-tuple $(x_0, \ldots, x_d)$ satisfying (1) for $\perp = \oplus$ (resp. for $\perp = \otimes$) is called a $d^{\text{th}}$ order additive (resp. $d^{\text{th}}$ order multiplicative) sharing of $x$.

## 2.2 Security definitions

In the probing model, proofs are based on simulation. Namely, if any adversary observation set (*i.e.* set of probed wires) can be simulated without the knowledge of any input variable then the $t$ probes are of no use to an attacker. We remind several security definitions introduced in [BBD+15] that are useful to prove the security of a construction in the probing model under the stronger $t$-SNI security definition.

An adversary can probe input wires, internal wires or output wires. An adversary observation set is denoted by $\Omega$ and divided into two sets $\mathcal{I}$ and $\mathcal{O}$ such that $\mathcal{I}$ is the set of input or internal probed wires while $\mathcal{O}$ is the set of output probed wires. For any set $\Omega = (\mathcal{I}, \mathcal{O})$ of at most $t$ probed wires, it is obvious that $|\mathcal{I}| + |\mathcal{O}| \leq t$.

The following security definitions rely on whether or not it is possible to simulate $\Omega$. Namely, if $\Omega$ can be perfectly simulated without knowledge of any input variable then the $t$ probes used by the attacker to build $\Omega$ are not dependent on any secret. Indeed, an input variable is a $(d+1)$-sharing generated such that the knowledge of $d$ of its shares does not reveal the original data. Thus, as long as the simulation of $\Omega$ only requires strictly less than $d+1$ shares of each input variable, then $\Omega$ can be simulated without knowing any secret. Consequently the $t$ probes reveal nothing to the attacker.

The set of input shares required simulating an adversary set of probed wires is denoted by $\mathcal{S}$. The latter also indicates which specific shares (*i.e.* which wires) are considered for each input. The upper bounds on the cardinality of $\mathcal{S}$ lead to more or less strong security definitions of [BBD+15] which are reminded hereafter. For simplicity we consider a gadget taking as input a single $(d+1)$-sharing $x$ and that outputs a single $(d+1)$-sharing $y$.

**$t$-NI security.** Let $G$ be a gadget which takes as input a $(d+1)$-sharing $(x_0, \ldots, x_d)$ of $x$ and outputs a $(d+1)$-sharing $(y_0, \ldots, y_d)$ of $y$. The gadget $G$ is said to be $t$-NI secure if for every adversary set of $t$ probed wires $\Omega = (\mathcal{I}, \mathcal{O})$ with $t \leq d$, there exists a set $\mathcal{S}$ of input shares such that $|\mathcal{S}| \leq t$ and $\mathcal{S}$ is sufficient to simulate the adversary observation set $\Omega$ on G.

**affine-NI security.** Let $G$ be a gadget which takes as input a $(d+1)$-sharing $(x_0, \ldots, x_d)$ of $x$ and outputs a $(d+1)$-sharing $(y_0, \ldots, y_d)$ of $y$. The gadget $G$ is said to be affine-NI secure if for every adversary set of $t$ probed wires $\Omega = (\mathcal{I}, \mathcal{O})$ with $t \leq d$, there exists a set $\mathcal{S}$ of input shares such that $|\mathcal{S}| \leq |\mathcal{I}| + |\mathcal{O}|$ and $\mathcal{S}$ is sufficient to simulate the adversary observation set $\Omega$ on G.

**$t$-SNI security.** Let $G$ be a gadget which takes as input a $(d+1)$-sharing $(x_0, \ldots, x_d)$ of $x$ and outputs a $(d+1)$-sharing $(y_0, \ldots, y_d)$ of $y$. The gadget $G$ is $t$-SNI secure if for every adversary set of $t$ probed wires $\Omega = (\mathcal{I}, \mathcal{O})$ with $t \leq d$, there exists a set $\mathcal{S}$ of input shares such that $|\mathcal{S}| \leq |\mathcal{I}|$ and $\mathcal{S}$ is sufficient to simulate the adversary observation set $\Omega$ on G.

**$t$-SNI security (binary gadgets).** Let $G$ be a gadget which takes as inputs a $(d+1)$-sharing $(x_0, \ldots, x_d)$ of $x$, a $(d+1)$-sharing $(y_0, \ldots, y_d)$ of $y$, and outputs a $(d+1)$-sharing $(z_0, \ldots, z_d)$ of $z$. The gadget $G$ is said to be $t$-SNI secure if for every adversary set of $t$ probed wires $\Omega = (\mathcal{I}, \mathcal{O})$ with $t \leq d$, there exist sets $\mathcal{S}^1$ of input shares of $x$ and $\mathcal{S}^2$ of input shares of $y$ such that $|\mathcal{S}^1| \leq |\mathcal{I}|, |\mathcal{S}^2| \leq |\mathcal{I}|$ and $\mathcal{S}^1 \cup \mathcal{S}^2$ is sufficient to simulate the adversary observation set $\Omega$ on G.

## 2.3 Useful t-SNI gadgets

Several constructions in this article may involve gadgets whose security has already been analyzed in the literature. We hereafter recall the secure multiplication algorithm as described in [RP10] and the mask refreshing procedure introduced by Duc, Dziembowski and Faust in [DDF14]. Furthermore, it has been shown in [BBD+15] that both gadgets are $t$-SNI.

**Algorithm 1** SecMult [RP10]

```
Require: An order $d$, a $(d+1)$-sharing of $x$ and a $(d+1)$-sharing of $y$.
Ensure: A $(d+1)$-sharing $(z_0, \ldots, z_d)$ of $(x \otimes y)$.
for $i = 0$ to $d$ do
    $z_i \leftarrow x_i \otimes y_i$
end for
for $i = 0$ to $d$ do
    for $j = i + 1$ to $d$ do
        $r \overset{\$}{\leftarrow} \mathbb{F}_{2^n}$
        $z_i \leftarrow z_i \oplus r$
        $r \leftarrow x_i \otimes y_j \oplus r \oplus x_j \otimes y_i$
        $z_j \leftarrow z_j \oplus r$
    end for
end for
return $(z_0, \ldots, z_d)$
```

Alg. 2 presents the multiplication-based refreshing algorithm of [DDF14].

**Algorithm 2** Multiplication-Based Mask Refreshing Algorithm

```
Require: An order $d$ and a $(d+1)$-sharing $(x_0, \ldots, x_d)$ of $x$.
Ensure: A $(d+1)$-sharing $(z_0, \ldots, z_d)$ of $x$.
for $i = 0$ to $d$ do
    $z_i \leftarrow x_i$
end for
for $i = 0$ to $d$ do
    for $j = i+1$ to $d$ do
        $r \overset{\$}{\leftarrow} \mathbb{F}_{2^n}$
        $z_i \leftarrow z_i \oplus r$
        $z_j \leftarrow z_j \oplus r$
    end for
end for
return $(z_0, \ldots, z_d)$
```

# 3 The GPQ scheme

Introduced by Genelle, Prouff and Quisquater in [GPQ10, GPQ11a, GPQ11b], the GPQ scheme securely evaluates power functions by mixing additive and multiplicative masking. Namely, (1) holds alternatively for $\perp = \oplus$ and $\perp = \otimes$. The additive masking is used to secure affine functions while multiplicative masking efficiently masks power functions as illustrated in Fig. 1. Thus, special transformations are necessary to convert an additive sharing into a multiplicative one and conversely. This strategy was initially addressed by Akkar and Giraud [AG01] but turned out to be not secure when a multiplicatively masked variable equals zero [GT02]. Genelle, Prouff and Quisquater solved this issue by proposing a secure implementation of the Dirac function that enables to multiplicatively mask the value zero [GPQ11a,GPQ10]. For the sake of self-completeness, we recall some algorithms of [GPQ10,GPQ11a,GPQ11b] that constitute GPQ and we also conduct a security analysis throughout this section to prove that the scheme actually satisfies the $t$-SNI property and not only the $t$-NI definition (as proven in the original paper).

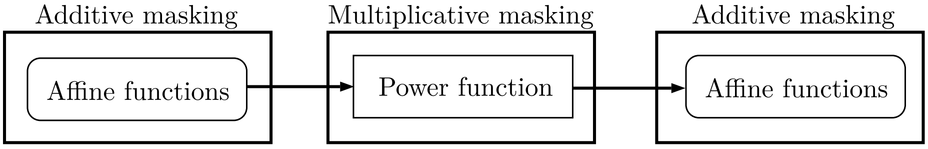

**Figure 1:** GPQ mixes additive and multiplicative masking.

From the above discussion, an additively masked element of $\mathbb{F}_{2^n}$ is mapped into $\mathbb{F}_{2^n}^*$ by adding it to its Dirac value so that the resulting non-zero element can be multiplicatively masked. Further details are given below.

## 3.1 Dirac

The Dirac function $\delta$ is defined over $\mathbb{F}_{2^n}$ by $\delta(x) = 1$ if $x = 0$ and $\delta(x) = 0$ otherwise. Hence for any $x \in \mathbb{F}_{2^n}$, it results $(x \oplus \delta(x)) \in \mathbb{F}_{2^n}^*$. The computation of the Dirac function of $x \in \mathbb{F}_{2^n}$ may be performed as follows.

Let $\overline{x} = (\overline{x_0}, \ldots, \overline{x_{n-1}})$ denote the bitwise complement of $x = (x_0, \ldots, x_{n-1})$, we have

$$\delta(x) = \overline{x_0} \odot \overline{x_1} \odot \cdots \odot \overline{x_{n-1}}\,,$$

where $\odot$ denotes the `AND` operation.

Computing one Dirac function at a time for several field elements may not be interesting in terms of efficiency (due to `AND` operations that have to be secured with `ISW`). However, bit-slicing enables to compute several Dirac functions simultaneously at a reasonable cost [GPQ11a]. The latter approach is therefore preferred. In a nutshell, it computes the Dirac function of $n$ elements of $\mathbb{F}_{2^n}$ viewed as a $(n \times n)$-matrix whose lines are actually treated as elements of $\mathbb{F}_2^n$. In the following, the field elements involved in the `Secure-Dirac` procedure are referred to as $x^{(k)}$ with $k \in \{0, 1, \ldots, n-1\}$. We hereafter recall the resulting algorithm that we also used in our implementations.

**Algorithm 3** `Secure-Dirac`

```
Require: An order $d$, a length $n$ and a $(d+1)$-sharing $(M_0, \cdots, M_d)$ of a binary $(n \times n)$-matrix $M$ whose lines are the $x^{(k)}$'s.
Ensure: A $(d+1)$-sharing $(\Delta_0, \cdots, \Delta_d)$ of the $n$-bit vector $\Delta = (\delta(x^{(0)}), \cdots, \delta(x^{(n-1)}))$.
    ** Compute the bitwise complement $\overline{M_0}$ of the $(n \times n)$-matrix $M_0$.
 1: $M_0 \longleftarrow \overline{M_0}$
    ** Transpose the $(n \times n)$ matrices $M_i$ for every $i \leq d$.
 2: for $i = 0$ to $d$ do
 3:     $t_i \longleftarrow (M_i)^\intercal$
 4: end for
    ** Refresh the shares.
 5: $(t_0^{(0)}, \ldots, t_d^{(0)}) \longleftarrow \text{Refresh}(t_0^{(0)}, \ldots, t_d^{(0)})$
 6: $(\Delta_0, \ldots, \Delta_d) \longleftarrow (t_0^{(0)}, \ldots, t_d^{(0)})$
    ** Process the Dirac computations.
 7: for $i = 1$ to $n-1$ do
 8:     $(\Delta_0, \cdots, \Delta_d) \longleftarrow (\Delta_0, \cdots, \Delta_d) \odot^n (t_0^{(i)}, \ldots, t_d^{(i)})$
 9: end for
10: return $(\Delta_0, \cdots, \Delta_d)$
```

The $\odot^n$ operation (Step 8 of Alg. 3) performs $n$ secure multiplications over $\mathbb{F}_2$.

*Remark* 1. In order to prove the following Lemma, we had to add a refreshing procedure (step 5) that was not originally required. In particular, this step requires the use of Alg. 2.

*Lemma* 1. `Secure-Dirac`$(\cdot)$ *is* $t$-`SNI`. *Let* $(M_i)_{0 \leq i \leq d}$ *be the input and let* $(\Delta_i)_{0 \leq i \leq d}$ *be the output of Alg. 3. For any adversary set of at most* $t$ *probed wires* $\Omega = (\mathcal{I}, \mathcal{O})$*, with* $t \leq d$*, there exists a set of input shares* $\mathcal{S}$ *such that* $|\mathcal{S}| \leq |\mathcal{I}|$ *and* $\mathcal{S}$ *is sufficient to simulate the adversary observation set* $\Omega$*.*

*Proof.* See Appendix A.1. □

For a given set of $n$ additively masked field elements, their Dirac values can be computed with Alg. 3 and have to be added to their corresponding elements before converting them into multiplicative maskings. The complexity of the `Secure-Dirac` procedure is given at the end of this section. We now address the conversion transformations that enable to switch encodings for a non-zero masked element between its additive and multiplicative sharing.

## 3.2 Conversions

The general strategy consists in replacing sequentially each additive (resp. multiplicative) mask of the $(d+1)$-additive (resp. multiplicative) sharing of an element $x \in \mathbb{F}_{2^n}^*$ by a multiplicative (resp. additive) one. This strategy results in the following two algorithms. Alg. 4 describes the steps for an additive to multiplicative masking conversion and Alg. 5 describes the multiplicative to additive masking conversion. As in [GPQ11b], these transformations are respectively called AMtoMM and MMtoAM.

**Algorithm 4** AMtoMM

```
Require: A $(d+1)$-additive sharing $(x_0, \ldots, x_d)$ of $x \in \mathbb{F}_{2^n}^*$
Ensure: A $(d+1)$-multiplicative sharing $(z_0, \ldots, z_d)$ of $x \in \mathbb{F}_{2^n}^*$
$z_0 \leftarrow x_0$
for $i = 1$ to $d$ do
    $z_i \overset{\$}{\leftarrow} \mathbb{F}_{2^n}^*$
    $z_0 \leftarrow z_0 \dot{\otimes} z_i$
    for $j = 1$ to $d - i$ do
        $U \overset{\$}{\leftarrow} \mathbb{F}_{2^n}$
        $x_j \leftarrow z_i \dot{\otimes} x_j$
        $**$ Refreshing of the additive share
        $x_j \leftarrow x_j \oplus U$
        $z_0 \leftarrow z_0 \oplus x_j$
        $x_j \leftarrow U$
    end for
    $x_{d-i+1} \leftarrow z_i \dot{\otimes} x_{d-i+1}$
    $z_0 \leftarrow z_0 \oplus x_{d-i+1}$
end for
return $(z_0, z_1, \ldots, z_d)$
```

This conversion has been proven in [GPQ11b] to satisfy the $t$-NI definition. We now prove the following theorem that states that AMtoMM$(\cdot)$ actually satisfies the $t$-SNI requirements.

*Theorem* 1. AMtoMM$(\cdot)$ *conversion is* $t$-SNI. *Let* $(x_i)_{0\leq i\leq d}$ *be the input and let* $(z_i)_{0\leq i\leq d}$ *be the output of Alg. 4. For any adversary set of at most* $t$ *probed wires* $\Omega = (\mathcal{I}, \mathcal{O})$, *with* $t \leq d$, *there exists a set of input shares* $\mathcal{S}$ *such that* $|\mathcal{S}| \leq |\mathcal{I}|$ *and* $\mathcal{S}$ *is sufficient to simulate the adversary observation set* $\Omega$.

*Proof.* See Appendix A.2. □

The other conversion that deals with getting an additive masking from a multiplicative one is described by Alg. 5. This conversion has only been proven to satisfy the $t$-NI property. We prove similarly to the previous conversion that it satisfies the stronger security definition.

*Theorem* 2. MMtoAM$(\cdot)$ *conversion is* $t$-SNI. *Let* $(x_i)_{0\leq i\leq d}$ *be the input and let* $(z_i)_{0\leq i\leq d}$ *be the output of Alg. 5. For any adversary set of at most* $t$ *probed wires* $\Omega = (\mathcal{I}, \mathcal{O})$, *with* $t \leq d$, *there exists a set of input shares* $\mathcal{S}$ *such that* $|\mathcal{S}| \leq |\mathcal{I}|$ *and* $\mathcal{S}$ *is sufficient to simulate the adversary observation set* $\Omega$.

*Proof.* See Appendix A.3. □

The GPQ scheme involves Alg. 3, 4 and 5. We now give further details about the evaluation of power functions with GPQ.

**Algorithm 5** MMtoAM

```
Require: A $(d+1)$-multiplicative sharing $(z_0, \dots, z_d)$ of $x \in \mathbb{F}_{2^n}^*$
Ensure: A $(d+1)$-additive sharing $(x_0, \dots, x_d)$ of $x \in \mathbb{F}_{2^n}^*$
x_0 ← z_0
for $i = 1$ to $d$ do
    $x_i \overset{\$}{\leftarrow} \mathbb{F}_{2^n}$
    $x_0 \leftarrow x_0 \oplus x_i$
    $x_0 \leftarrow x_0 \dot{\otimes} z_i^{-1}$
    for $j = 1$ to $i$ do
        $x_j \leftarrow x_j \dot{\otimes} z_i^{-1}$
        $U \overset{\$}{\leftarrow} \mathbb{F}_{2^n}$
        ** Refreshing of the additive share
        $x_j \leftarrow x_j \oplus U$
        $z_0 \leftarrow z_0 \oplus x_j$
        $x_j \leftarrow U$
    end for
end for
return $(x_0, x_1, \dots, x_d)$
```

## 3.3 Power function processing

We outline the processing of a power function as follows. Consider a power $\alpha \in [0, 2^n - 1]$ and an element $x \in \mathbb{F}_{2^n}$ that is initially additively masked. First, the GPQ processing requires to compute the Dirac function of $x$ and add the result to it in order to map the field element into $\mathbb{F}_{2^n}^*$. Then, $x$ is converted into a multiplicative sharing in order to process the power function $x \mapsto x^\alpha$. Afterwards, $x^\alpha$ is converted back into an additive sharing and the resulting element is mapped from $\mathbb{F}_{2^n}^*$ back into $\mathbb{F}_{2^n}$ to be further processed by linear operations only. This processing is illustrated Fig. 2.

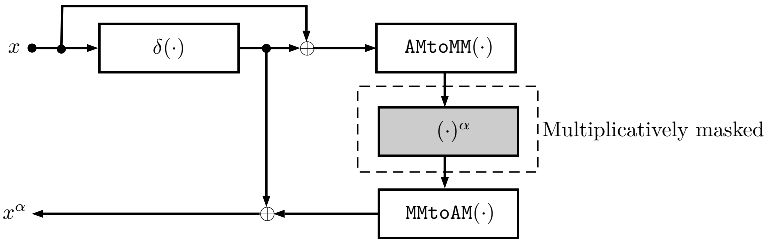

**Figure 2:** GPQ power function processing : $x \mapsto x^\alpha$.

The classical approach to securely process a power function $x \mapsto x^\alpha$ consists in expressing it in terms of squares and multiplications over $\mathbb{F}_{2^n}$, the latter being secured with the ISW multiplication gadget. This approach was first proposed by Rivain and Prouff in [RP10] for AES whose S-box can be represented as a single monomial over $\mathbb{F}_{2^n}$ (*i.e.* $x \mapsto x^{254}$). The study of masking power functions has been generalized by the work of Carlet, Goubin, Prouff, Quisquater and Rivain in [CGP+12]. They defined the notion of masking complexity for a $n$-bit S-box as the minimal number of nonlinear multiplications required to evaluate its polynomial representation over $\mathbb{F}_{2^n}$, and they computed the masking complexity of all power functions over $\mathbb{F}_{2^n}$ for $n \leq 11$. Their approach involves the notion of cyclotomic class and addition chain which are recalled hereafter.

**Cyclotomic class.** The cyclotomic class of $\alpha$ denoted by $C_\alpha$, $\alpha \in [0; 2^n - 2]$ is defined by

$$C_\alpha = \{\alpha \cdot 2^i \bmod 2^n - 1 \; ; \; i \in [0; n-1]\} .$$

As the Frobenius map $x \mapsto x^2$ over $\mathbb{F}_{2^n}$ is linear, any $\alpha_i \in C_\alpha$ can be computed from any $\alpha_j \in C_\alpha$ with $\alpha_j \neq \alpha_i$ only using linear transformations. Hence, powers whose exponents lie in the same cyclotomic class have the same masking complexity. The authors of [CGP+12] have related the problem of computing the masking complexity for an element $\alpha$ whose cyclotomic class is $C_\alpha$ to finding the shortest addition chain for $\alpha$, $C_{\alpha_0} \to C_{\alpha_1} \to \ldots \to \mathsf{C}_{\alpha_k}$, such that $C_{\alpha_0} = C_1, C_{\alpha_k} = C_\alpha$, and for every $i \in [1; k]$, there exist $j, l \leq i$ such that $\alpha_i = \alpha_j + \alpha_l$ where $\alpha_j \in C_{\alpha_j}$ and $\alpha_l \in C_{\alpha_l}$. The resulting chain decomposes any power $x^\alpha$ in terms of linear operations (*i.e.* squares) and nonlinear multiplications between powers whose exponents belong to different cyclotomic classes. On the contrary to the classical approach, `GPQ` does not require `ISW` to secure the sequence that decomposes a power function. More precisely, multiplications which are nonlinear when an additive masking is involved may be performed by element-wise field multiplications between the shares of the multiplicatively masked values and hence `ISW` is no longer required. In fact, a power function $x \mapsto x^\alpha$ can even be tabulated with `GPQ` leading to great efficiency gains. Such an implementation choice costs $2^n$ bytes of memory to store the table, which is reasonable for a power function over $\mathbb{F}_{2^n}$ with $n < 10$.

In the following, when power functions cannot be tabulated, we use the procedure `Eval-Chain`$(\cdot)$ that takes as inputs a multiplicatively masked element $x$ and an addition chain for $\alpha$ and that outputs the desired power $x^\alpha$ multiplicatively masked. Note that the cost of `Eval-Chain`$(\cdot)$ is negligible with `GPQ`. However, in order to minimize the complexity of an evaluation, it is always better to find the shortest possible addition chains.

**Algorithm 6** Secure Power Function Evaluation

```
Require: An order $d$, an addition chain $\mathcal{A}$ for $\alpha$, and a $(d+1)$-additive sharing of $x$
Ensure: A $(d+1)$-additive sharing $(y_0, \ldots, y_d)$ of $x^\alpha$
  ** Mapping from $\mathbb{F}_{2^n}$ to $\mathbb{F}^*_{2^n}$.
  $(\Delta_0, \ldots, \Delta_d) \leftarrow$ Secure-Dirac $(x_0, \ldots, x_d)$
  $(x_0, \ldots, x_d) \leftarrow (x_0, \ldots, x_d) \oplus (\Delta_0, \ldots, \Delta_d)$
  ** Convert into multiplicative masking
  $(z_0, \ldots, z_d) \leftarrow$ AMtoMM $(x_0, \ldots, x_d)$
  ** Evaluate the chain
  $(z_0^\alpha, \ldots, z_d^\alpha) \leftarrow$ Eval-Chain $((z_0, \ldots, z_d), \mathcal{A})$
  ** Convert back into additive masking
  $(y_0, \ldots, y_d) \leftarrow$ MMtoAM $(z_0^\alpha, \ldots, z_d^\alpha)$
  ** Mapping from $\mathbb{F}^*_{2^n}$ to $\mathbb{F}_{2^n}$.
  $(y_0, \ldots, y_d) \leftarrow (y_0, \ldots, y_d) \oplus (\Delta_0, \ldots, \Delta_d)$
  return $(y_0, \ldots, y_d)$
```

### 3.3.1 Complexity

Let us denote by $\mathtt{C}_\delta, \mathtt{C}_{\mathtt{AMtoMM}}$ and $\mathtt{C}_{\mathtt{MMtoAM}}$ respectively the costs of Alg. 3, 4 and 5 and by $\mathtt{C}_{\mathtt{GPQ}}$ the overall cost of a power function processing with `GPQ` (Alg. 6). For each algorithm, we express their cost in terms of the costs of their elementary operations. To that end, let us also denote by $\mathtt{C}_{\mathtt{T}}, \mathtt{C}_\oplus, \mathtt{C}_\odot, \mathtt{C}_\otimes$ respectively the costs of $(n \times n)$-matrix transpositions,

$\oplus$,$\odot$ and $\otimes$ operations. At last, $\mathtt{C}_{\oplus^n}$ and $\mathtt{C}_{\odot^n}$ denote the cost of $n$-bit operations $\oplus, \odot$. We have,

$$\mathtt{C}_\delta = \frac{(d+1)}{n} \times \mathtt{C}_\mathtt{T} + \frac{((2d\,(n-1)+n)(d+1)}{n} \times \mathtt{C}_{\oplus^n} + \frac{(n-1)(d+1)^2}{n} \times \mathtt{C}_{\odot^n}\,,$$

$$\mathtt{C}_\mathtt{AMtoMM} = d^2 \times \mathtt{C}_\oplus + \frac{d\,(3+d)}{2} \times \mathtt{C}_\otimes\,,$$

$$\mathtt{C}_\mathtt{MMtoAM} = d\,(2+d) \times \mathtt{C}_\oplus + \frac{d\,(3+d)}{2} \times \mathtt{C}_\otimes\,,$$

which gives,

$$\mathtt{C}_\mathtt{GPQ} = \mathtt{C}_\delta + \mathtt{C}_\mathtt{AMtoMM} + \mathtt{C}_\mathtt{MMtoAM}\,.$$

### 3.3.2 Security

We now prove that GPQ is $t$-SNI. This is made accurate in the following theorem.

*Theorem* 3. GPQ *is* $t$-SNI. *Let* $(x_i)_{0\le i\le d}$ *be the input and let* $(y_i)_{0\le i\le d}$ *be the output of Alg. 6 (or equivalently of Fig. 3). For any adversary set of* $t$ *probed wires* $\Omega = (\mathcal{I}, \mathcal{O})$*, with* $t \le d$*, there exists a set* $\mathcal{S}$ *of input shares such that* $|\mathcal{S}| \le |\mathcal{I}|$ *and* $\mathcal{S}$ *is sufficient to simulate the adversary observation set* $\Omega$*.*

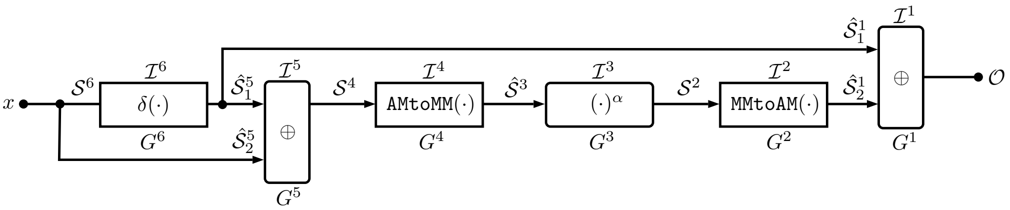

**Figure 3:** GPQ secure Gadget $(\cdot)^\alpha$.

*Proof.* As in [BBD+15], the proof is constructed by composition. Namely, we construct the simulator for the circuit of Fig. 3 by simulating sequentially each inner gadget from right to left.

Let $\Omega = (\mathcal{I}, \mathcal{O})$ be an observation set that has to be simulated, made on the whole circuit of Fig. 3 such that $\mathcal{I} = \bigcup_{1\le i\le 6} \mathcal{I}^i$ and such that the global constraint $\sum_{i=1}^6 |\mathcal{I}^i| + |\mathcal{O}| \le t$ is satisfied.

**Gadget 1** - Let $\Omega^1 = (\mathcal{I}^1, \mathcal{O})$ be an observation set made on Gadget 1. Since $G^1$ is affine-NI, we know that for every observation set $\Omega^1$, there exists a set of input shares $\hat{\mathcal{S}}^1 = (\hat{\mathcal{S}}^1_1, \hat{\mathcal{S}}^1_2)$ such that $|\hat{\mathcal{S}}^1| \le |\mathcal{I}^1 \cup \mathcal{O}|$ and the set $\hat{\mathcal{S}}^1$ is sufficient to simulate $\Omega^1$.
**Gadget 2** - Let $\Omega^2 = (\mathcal{I}^2, \hat{\mathcal{S}}^1_2)$ be an observation set made on Gadget 2. Since MMtoAM$(\cdot)$ is t-SNI and $|\mathcal{I}^2 \cup \hat{\mathcal{S}}^1_2| \le |\mathcal{I}^2 \cup \mathcal{I}^1 \cup \mathcal{O}| \le t$ (by simulation of Gadget 1 and the global constraint), we know that for every observation set $\Omega^2$, there exists a set of input shares $\mathcal{S}^2$ such that $|\mathcal{S}^2| \le |\mathcal{I}^2|$ and the set $\mathcal{S}^2$ is sufficient to simulate $\Omega^2$.
**Gadget 3** - Let $\Omega^3 = (\mathcal{I}^3, \mathcal{S}^2)$ be an observation set made on Gadget 3. Since $G^3$ is affine-NI, we know that for every observation set $\Omega^3$, there exists an observation set $\hat{\mathcal{S}}^3$ such that $|\hat{\mathcal{S}}^3| \le |\mathcal{I}^3 \cup \mathcal{S}^2| \le |\mathcal{I}^3| + |\mathcal{I}^2|$ and the set $\hat{\mathcal{S}}^3$ is sufficient to simulate $\Omega^3$.

**Gadget 4** - Let $\Omega^4 = (\mathcal{I}^4, \hat{\mathcal{S}}^3)$ be an observation set made on Gadget 4. Since AMtoMM$(\cdot)$ is $t$-SNI and $|\mathcal{I}^4 \cup \hat{\mathcal{S}}^3| \leq t$ (by simulation of Gadget 3 and the global constraint) , we know that for every observation set $\Omega^4$, there exists an observation set $\mathcal{S}^4$ such that $|\mathcal{S}^4| \leq |\mathcal{I}^4|$ the set $\mathcal{S}^4$ is sufficient to simulate $\Omega^4$.
**Gadget 5** - Let $\Omega^5 = (\mathcal{I}^5, \mathcal{S}^4)$ be an observation set made on Gadget 5. Since $G^5$ is affine-NI, we know that for every observation set $\Omega^5$ there exists an observation set $\hat{\mathcal{S}}^5$ such that $|\hat{\mathcal{S}}^5| \leq |\mathcal{I}^5 \cup \mathcal{S}^4| \leq |\mathcal{I}^5| + |\mathcal{I}^4|$ and the set $\hat{\mathcal{S}}^5$ is sufficient to simulate $\Omega^5$.
**Gadget 6** - Let $\Omega^6 = (\mathcal{I}^6, (\hat{\mathcal{S}}^5 \cup \hat{\mathcal{S}}_1^1))$ be an observation set made on Gadget 6. Since $\delta(\cdot)$ is t-SNI and $|\mathcal{I}^6 \cup \hat{\mathcal{S}}^5 \cup \hat{\mathcal{S}}_1^1| \leq |\mathcal{I}^6 \cup \mathcal{I}^5 \cup \mathcal{I}^4 \cup \mathcal{I}^1 \cup \mathcal{O}| \leq t$ (by simulation of gadgets 5 and 1 and by the global constraint), we know that for every observation set $\Omega^6$, there exists an observation set $\mathcal{S}^6$ such that $|\mathcal{S}^6| \leq |\mathcal{I}^6|$ and the set $\mathcal{S}^6$ is sufficient to simulate $\Omega^6$.

To simulate the whole circuit, that is the observation set $\Omega = (\bigcup_{1\leq i \leq 6} \mathcal{I}^i, \mathcal{O})$, the simulator requires $|\mathcal{S}^6 \cup \hat{\mathcal{S}}^5|$ shares of $x$. Since $|\mathcal{S}^6| \leq |\mathcal{I}^6|$, and $|\hat{\mathcal{S}}^5| \leq |\mathcal{I}^5| + |\mathcal{I}^4|$, we have that $|\mathcal{S}^6 \cup \hat{\mathcal{S}}^5| \leq \sum_{i=1}^{6} |\mathcal{I}^i| \leq t$ and therefore GPQ satisfies the $t$-SNI property. □

# 4 Polynomial GPQ : The Alternate Cyclotomic Method

In this section, we describe how to extend the GPQ scheme to the masking of generic S-boxes. The main idea is outlined as follows. Since any $n$-bit S-box can be represented by a polynomial $S(x) = \sum a_i x^i$ over $\mathbb{F}_{2^n}$, a secure evaluation of $S(x)$ thus requires to securely process the corresponding sequence of linear operations and power functions. As mentioned in the previous section, the common approach, referred to as the CGPQR method, would in turn decompose each power function in terms of squares and nonlinear multiplications over $\mathbb{F}_{2^n}$. Thereby, this approach involves ISW in order to secure these nonlinear multiplications. We propose to use GPQ to process the power functions in such manner that ISW is no longer required. However, a naive evaluation of the above writing of $S$ that processes each power $x^i$ with GPQ is not recommended in terms of efficiency. Indeed, such an evaluation would require the computation of a Dirac function along with conversions from an additive masking to a multiplicative masking and conversely for each monomial involved in the polynomial representation. Note that those transformations are costly to process (asymptotically they have the same complexity $\mathcal{O}(d^2)$ as ISW multiplications), thus we seek to minimize their number during a polynomial evaluation. A solution is provided by the cyclotomic method of [CGP$^+$12] which we briefly present hereafter. Our approach is then detailed along with security proofs for the new proposed constructions.

## 4.1 Original Cyclotomic Method

Since the family of cyclotomic classes $C_\alpha$ is a partition of $[0, 2^n - 1]$, hence the polynomial representation of any S-box can be written

$$S(x) = a_0 + \left( \sum_{i=1}^{q} L_i(x^{\alpha_i}) \right) + a_{2^n-1} x^{2^n-1} , \tag{2}$$

where $L_i(x)$ denotes the linearized polynomial $\sum_j a_{i,j} x^{2^j}$ and $q$ is the number of distinct cyclotomic classes of $[0; 2^n - 2]$. The cyclotomic method simply consists in deriving the powers $x^{\alpha_i}$ for each cyclotomic class as well as $x^{2^n-1}$ if $a_{2^n-1} \neq 0$ and in evaluating $S(x)$. Following the CGPQR approach, it is required to find an addition chain for the $x^{\alpha_i}$'s, $C_{\alpha_0} \to C_{\alpha_1} \to \ldots \to C_{\alpha_k}$ such that $C_{\alpha_0} = C_1$ and for every $x^{\alpha_i}$, there exists $j \in [1, k]$ such that $C_{\alpha_i} = C_{\alpha_j}$. The addition chain decomposes the $x^{\alpha_i}$'s as a sequence of squares and nonlinear multiplications over $\mathbb{F}_{2^n}$. The rest of the powers can be derived with Frobenius maps. Using CGPQR, ISW is involved to derive at least one power of each distinct cyclotomic classes of $[0; 2^n - 2]$. Therefore, the shorter the chain is, the better.

#### 4.1.1 Complexity

Let us denote by $\mathtt{C}_{\mathtt{Cyclo}}$ the cost of evaluating polynomials with the CGPQR scheme and the cyclotomic method. Let us also denote by $C_{\mathtt{SecMult}}$ the cost of a finite field multiplication which is secured with ISW and let $q$ be the number of distinct cyclotomic classes of $[0; 2^n - 2]$. Then the cost of masking generic S-boxes is

$$\mathtt{C}_{\mathtt{Cyclo}} = (q-1) \times \mathtt{C}_{\mathtt{SecMult}},$$

or $\mathtt{C}_{\mathtt{Cyclo}} = (q-2) \times \mathtt{C}_{\mathtt{SecMult}}$ if the S-box which is considered is balanced (see [CGP+12]).

We now propose a different writing of (2) adapted for an evaluation with GPQ.

### 4.2 Our alternate proposal

We have that $x^{\alpha_i} = (x + \delta(x))^{\alpha_i} + \delta(x)$ which gives $L_i(x^{\alpha_i}) = L_i((x + \delta(x))^{\alpha_i}) + L_i(\delta(x))$ by linearity of $L_i$. Thus, (2) can be written as

$$S(x) = a_0 + L_1(x^{\alpha_1}) + \sum_{i=2}^{q} (L_i((x + \delta(x))^{\alpha_i}) + L_i(\delta(x))) + a_{2^n-1} x^{2^n-1},$$

where $L_i(\delta(x)) = \sum_j a_{i,j} \delta(x)^{2^j} = \sum_j a_{i,j} (1)^{2^j} \delta(x) = L_i(1) \cdot \delta(x)$, which gives

$$S(x) = a_0 + L_1(x^{\alpha_1}) + \sum_{i=2}^{q} (L_i((x + \delta(x))^{\alpha_i}) + L_i(1) \cdot \delta(x)) + a_{2^n-1} x^{2^n-1}.$$

According to the field equation, $x^{2^n-1} = 0$ if $x = 0$ and $x^{2^n-1} = 1$ otherwise. It follows that since $\delta(x) = 1$ if $x = 1$ and $\delta(x) = 0$ otherwise, we have $x^{2^n-1} = \delta(x) + 1$. Finally,

$$S(x) = a_0 + a_{2^n-1} + L_1(x^{\alpha_1}) + \sum_{i=2}^{q} L_i((x + \delta(x))^{\alpha_i}) + \left( \sum_{i=2}^{q} L_i(1) + a_{2^n-1} \right) \cdot \delta(x). \quad (3)$$

The above writing of $S(x)$ yields to a novel version of the cyclotomic method which shall be referred to as the alternate cyclotomic method in the following and which also extends GPQ to the evaluation of polynomials over $\mathbb{F}_{2^n}$.

We outline the steps of such an evaluation in Alg. 7. Similarly to the processing of a single power function (see Section 3.3), the procedure Eval-Chain$(\cdot)$ (Step 4 of Alg. 7) takes as inputs an element $(x + \delta(x)) \in \mathbb{F}_{2^n}^*$ along with an addition chain for all the $x^{\alpha_i}$'s, evaluates the latter without ISW and outputs the desired powers $(x + \delta(x))^{\alpha_i}$ still multiplicatively masked. Note that the sequence of operations provided by the chain may lead to computing powers which are not one of the $(x + \delta(x))^{\alpha_i}$'s. However, only the $(x + \delta(x))^{\alpha_i}$'s are converted back into additive maskings at the end of the evaluation. Moreover, since Frobenius maps are less costly than conversions, the linearized polynomial $L_1(x)$ of (3), whose monomials are only powers of two, is always computed in additive masking.

**Algorithm 7** Alternate Cyclotomic

```
Require: An order $d$, an addition chain $\mathcal{A}$, and a $(d+1)$-additive sharing of $x$
Ensure: A $(d+1)$-additive sharing $(S_0, \dots, S_d)$ of $S(x)$
    NOTE : The (d+1)-additive sharing $(x_0, \dots, x_d)$ of $x$ is stored in memory

    ** Mapping from $\mathbb{F}_{2^n}$ to $\mathbb{F}^*_{2^n}$.
    $(\Delta_0, \dots, \Delta_d) \leftarrow$ Secure-Dirac$(x_0, \dots, x_d)$
    $(x_0, \dots, x_d) \leftarrow (\Delta_0, \dots, \Delta_d) \oplus (x_0, \dots, x_d)$

    ** Convert into multiplicative masking and evaluate the addition chain.
    $(z_0, \dots, z_d) \leftarrow$ AMtoMM$(x_0, \dots, x_d)$
    $(z^{\alpha_2}, \dots, z^{\alpha_k}) \leftarrow$ Eval-Chain$((\mathtt{z_0}, \dots, \mathtt{z_d}), \mathcal{A})$

    ** Compute the linearized polynomials.
    $(L_0, \dots, L_d) \leftarrow$ Linearize-Poly$(x_0, \dots, x_d)$
    for $i = 2$ to $q$ do
        $(x_0^{(i)}, \dots, x_d^{(i)}) \leftarrow$ MMtoAM$(z_0^{\alpha_i}, \dots, z_d^{\alpha_i})$
        $(l_0^{(i)}, \dots, l_d^{(i)}) \leftarrow$ Linearize-Poly$(x_0^{(i)}, \dots, x_d^{(i)})$
        $(L_0, \dots, L_d) \leftarrow (L_0, \dots, L_d) \oplus (l_0^{(i)}, \dots, l_d^{(i)})$
    end for

    ** Mapping from $\mathbb{F}^*_{2^n}$ to $\mathbb{F}_{2^n}$.
    $a \leftarrow a_0$
    for $i = 1$ to $2^n - 1$ do
        $a \leftarrow a \oplus (a_i \cdot (\Delta_0, \dots, \Delta_d))$
        $(S_0, \dots, S_d) \leftarrow a_0 \oplus a_{2^n-1} \oplus (L_0, \dots, L_d)$
    end for

    return $(S_0, \dots, S_d)$
```

### 4.2.1 Complexity

Let us denote by $\mathtt{C}_{\mathtt{Alt-cyclo}}$ the cost of evaluating polynomials using the alternate cyclotomic method (Alg. 7). We do not take into account the costs of Eval-Chain$(\cdot)$ and Linearize-Poly$(\cdot)$ procedures as they can be computed with linear transformations. The cost of our alternate cyclotomic method for the evaluation of polynomials is therefore

$$\mathtt{C}_{\mathtt{Alt-cyclo}} = \mathtt{C}_{\delta} + \mathtt{C}_{\mathtt{AMtoMM}} + (q-2) \times \mathtt{C}_{\mathtt{MMtoAM}} \,,$$

where $q$ is the number of distinct cylotomic classes of $[0; 2^n - 2]$.

For the sake of clarity, Table 1 lists the complexities of our proposal and the original method in terms of elementary operations as a function of the order $d$. Also, as operations $\oplus, \oplus^n, \odot, \odot^n$ have the same complexity in practice (see Section 6), we list them together. Operation $\mathtt{M}^{\mathtt{T}}$ denotes $(n \times n)$-matrix transpositions.

In order to proceed to a fair comparison it should be noted that field multiplications with our proposal do not have the same weight as the ones that are implemented following the original method. Our approach allows to implement field multiplications more efficiently (see Section 6).

**Table 1:** Complexities of our proposal and the original method in terms of elementary operations.

| Operations | Our proposal | [CGP+12] |
|---|---|---|
| $\otimes$ | $(q-1)\,d^2+(3q-3)\,d$ | $(q-2)\,d^2+(2q-4)\,d+(q-2)$ |
| $\oplus,\oplus^n,\odot,\odot^n$ | $\left(q+2-\frac{3}{n}\right)d^2+\left(2q+1-\frac{4}{n}\right)d+\left(2-\frac{1}{n}\right)$ | $(2q-4)\,d^2+(2q-4)\,d$ |
| $\mathtt{M}^{\mathsf{T}}$ | $\frac{d}{n}+\frac{1}{n}$ | – |

From Table 1 it is obvious that the complexities of both our proposal and the original method mainly depend on the number of cyclotomic classes.

The end of the section is dedicated to prove the security of the resulting method under the $t$-SNI security definition.

### 4.2.2 Security

In order to analyze the security of our alternate cyclotomic method and for the sake of clarity, we divide the processing of the corresponding Gadget Alt-Cy$(\cdot)$ into two parts as illustrated Fig. 4. The security of Gadgets Alt-Cy$^1(\cdot)$ and Alt-Cy$^2(\cdot)$ is analyzed separately. Then the security of Alt-Cy$(\cdot)$ is induced by the secure composition of Gadgets Alt-Cy$^1(\cdot)$ and Alt-Cy$^2(\cdot)$. Note that Gadget $R$ is a refreshing Gadget (Alg. 2).

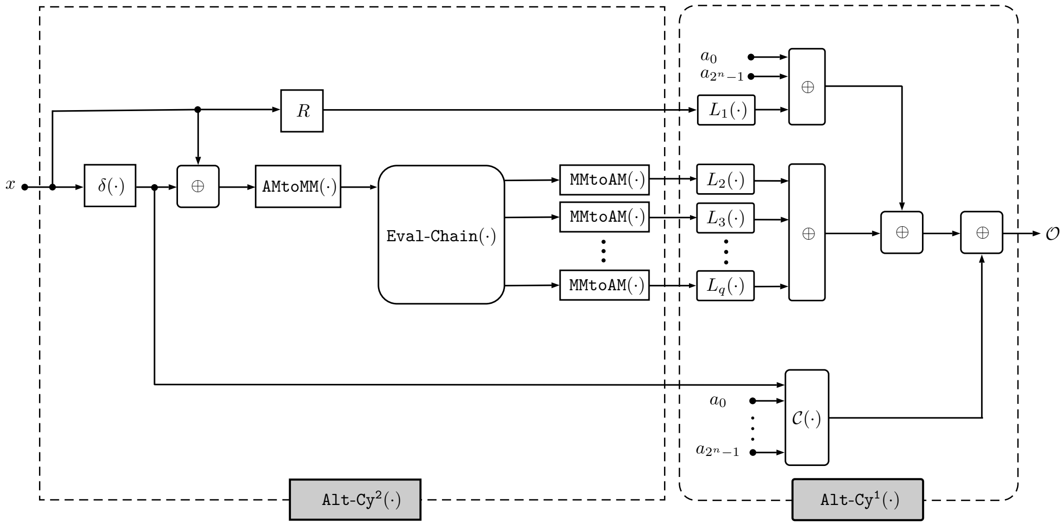

**Figure 4:** Gadget Alt-Cy$(\cdot)$ : circuit of the alternate cyclotomic processing.

Gadget Alt-Cy$^1(\cdot)$ only involves affine gadgets. Indeed, Gadget $L_i(\cdot)$ corresponds to the procedure Linearize-Poly$(\cdot)$ (step 9) of Alg. 7 which only involves linear operations (*i.e.* squares and additions) and Gadget $\mathcal{C}(\cdot)$ corresponds to step 8 which involves scalar multiplications and additions. Since the composition of affine gadgets is affine, hence Alt-Cy$^1(\cdot)$ is affine-NI. Regarding Gadget Alt-Cy$^2(\cdot)$ we prove the following Lemma.

*Lemma* 2. Alt-Cy$^2(\cdot)$ *is* $t$-SNI. *Let* $(x_i)_{0\le i\le d}$ *be the input and let* $(x'_i)_{0\le i\le d}$, $((x+\delta(x))_i^{\alpha_j})_{0\le i\le d}$ *with* $j\in[2;q]$ *and* $(\delta(x)_i)_{0\le i\le d}$ *be the outputs of Gadget* Alt-Cy$^2(\cdot)$. *For any adversary* *set of $t$ probed wires $\Omega = (\mathcal{I}, \mathcal{O})$, with $t \leq d$, there exists a set $\mathcal{S}$ of input shares such that $|\mathcal{S}| \leq |\mathcal{I}|$ and $\mathcal{S}$ is sufficient to simulate the adversary observation set $\Omega$.*

*Proof.* See Appendix B.1. □

*Theorem 4. Alternate cyclotomic is $t$-*`SNI`*. Let $(x_i)_{0 \leq i \leq d}$ be the input and let $(S_i)_{0 \leq i \leq d}$ be the output of Alg. 7 or equivalently of Gadget* `Alt-Cy`*$(\cdot)$ (see Fig. 4). For any adversary set of $t$ probed wires $\Omega = (\mathcal{I}, \mathcal{O})$, with $t \leq d$, there exists a set $\mathcal{S}$ of input shares such that $|\mathcal{S}| \leq |\mathcal{I}|$ and $\mathcal{S}$ is sufficient to simulate the adversary observation set $\Omega$.*

We illustrate Fig. 5 the circuit corresponding to our alternate cyclotomic method. We already analyzed Gadgets `Alt-Cy`$^1(\cdot)$ and `Alt-Cy`$^2(\cdot)$ and now we prove the security of the full construction by composition.

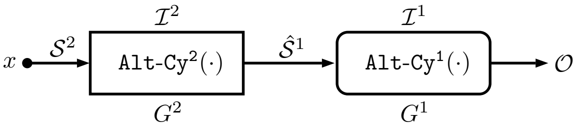

**Figure 5:** Gadget `Alt-Cy`$(\cdot)$.

*Proof.* Let $\Omega = ((\mathcal{I}^1 \cup \mathcal{I}^2), \mathcal{O})$ be an observation set to simulate for the circuit represented Fig. 5, such that the global constraint $|\mathcal{I}^1| + |\mathcal{I}^2| + |\mathcal{O}| \leq t$ is satisfied.

**Gadget 1.** Let $\Omega^1 = (\mathcal{I}^1, \mathcal{O})$. Since $G^1$ is `affine-NI`, we know that there exists an observation set $\hat{\mathcal{S}}^1$ such that $|\hat{\mathcal{S}}^1| \leq |\mathcal{I}^1| + |\mathcal{O}|$ and the set of input shares $\hat{\mathcal{S}}^1$ is sufficient to simulate the adversary observation set $\Omega^1$ made on Gadget 1.
**Gadget 2.** Let $\Omega^2 = (\mathcal{I}^2, \hat{\mathcal{S}}^1)$. Since $G^2$ is $t$-`SNI` (by Lemma 2) and $|\mathcal{I}^2 \cup \hat{\mathcal{S}}^1| \leq t$ (by simulation of Gadget 1 and the global constraint), we know that there exists an observation set $\mathcal{S}^2$ such that $|\mathcal{S}^2| \leq |\mathcal{I}^2|$ and $\mathcal{S}^2$ is sufficient to simulate $\Omega^2$.

In order to simulate Gadget `Alt-Cy`$(\cdot)$, the corresponding simulator requires the shares $\mathcal{S}^2$ of its input $x$ and $|\mathcal{S}^2| \leq |\mathcal{I}^2| \leq \sum_{i=1}^{2} |\mathcal{I}^i| \leq t$. Therefore, Gadget `Alt-Cy`$(\cdot)$ is $t$-`SNI`. □

Gadgets `Alt-Cy`$^1(\cdot)$ and `Alt-Cy`$^2(\cdot)$ involved in our alternate cyclotomic method are also used in the next section in which we describe how to combine them to propose an alternate `CRV` method.

# 5 The Alternate CRV Method

In this section we describe an alternate approach for the `CRV` method proposed by Coron, Roy and Vivek in [CRV14] which is currently the best known method for polynomial evaluation over $\mathbb{F}_{2^n}$. The idea is to plug our polynomial evaluation method with `GPQ` (*i.e* our alternate cyclotomic method) into the `CRV` construction. First, we recall the original method, then we describe our alternate approach and we also show how it enables to derive new parameters more adapted to our case. Finally, we prove that the resulting construction is $t$-`SNI`.

## 5.1 Original `CRV` method

The `CRV` method first consists in choosing a collection $S$ of $l$ cyclotomic classes among which $C_0$ and $C_1$ are always counted. Then it defines the union set $L$ of all integers in those cyclotomic classes. The original approach states that the set $S$ has to be carefully chosen so that the monomials $x^L$ can be computed with only $l-2$ nonlinear multiplications. It is moreover required that every monomial of $[0, 2^n - 1]$ can be written as a product of some two monomials generated from $L$.

Denoting by $\mathcal{P}(x^L)$ the set of all polynomials in $\mathbb{F}_{2^n}$ whose monomials belong to the set $x^L$, `CRV` generates randomly $k-1$ polynomials $q_i(x) \in \mathcal{P}$ and tries to find $k$ polynomials $p_i(x) \in \mathcal{P}$ such that

$$S(x) = \sum_{i=1}^{k-1} p_i(x) \cdot q_i(x) + p_k(x)\,. \tag{4}$$

From (4), `CRV` tries to solve a system of $2^n$ linear equations with $k \times |L|$ unknowns which are the coefficients of the $p_i$'s. Such a system admits a solution for every choice of $S$ if it has rank $2^n$. To be of full rank, the necessary condition $k \cdot |L| \geq 2^n$ has to be satisfied.

### 5.1.1 Complexity

Let us denote by $\mathtt{C}_{\mathtt{CRV}}$ the overall cost of `CRV`. As mentioned in the above description, the set of monomials $x^L$ requires $l-2$ nonlinear multiplications to be built, and $k-1$ additional nonlinear multiplications are necessary to compute (4). Following the `CGPQR` method, those nonlinear multiplications are secured with `ISW`, which cost is denoted by $\mathtt{C}_{\mathtt{SecMult}}$. Thus,

$$\mathtt{C}_{\mathtt{CRV}} = (l + k - 3) \times \mathtt{C}_{\mathtt{SecMult}}\,.$$

## 5.2 Our Alternate Proposal

The original approach imposes a constraint on the choice of cyclotomic classes that form the set $S$. Underlying this constraint is in fact the cyclotomic method. The latter enables to evaluate polynomials composed of $l$ cyclotomic classes with $l-2$ nonlinear multiplications, as long as each nonlinear multiplication allows to reach a different cyclotomic class. Also, monomials that belong to $C_0$ or $C_1$ do not require nonlinear multiplications to be derived (see Section 4.1 of [CGP+12]).

On the other hand, our alternate cyclotomic approach does not imply to secure these $l-2$ nonlinear multiplications with `ISW` and thus makes the previous constraint obsolete. It evaluates polynomials with `GPQ` instead. Therefore, we propose to plug our alternate cyclotomic approach into the `CRV` construction only to build the precomputed set $x^L$. We emphasize that computing (4) still requires $k-1$ `ISW` multiplications.

### 5.2.1 New parameters

Our approach allows more freedom degree on the choice of cyclotomic classes to build $x^L$. Also, we can consider larger sets $S$. As an example, let us consider the secure evaluation of 8-bit S-boxes. It has been shown in [CRV14] that choosing $l = 7$ and the set of cyclotomic classes $L = C_0 \cup C_1 \cup C_3 \cup C_7 \cup C_{29} \cup C_{87} \cup C_{251}$ gives a full rank system for some random choice of the polynomials $q_i(x)$. The precomputed set from which the monomials of the $q_i$'s are picked up can thus be built with 5 nonlinear multiplications. Moreover, in order to satisfy the necessary condition $k \cdot |L| \geq 2^n$, such a choice for $L$ ($|L| = 49$) implies that $k = 6$.

In our approach, we increase the size of $S$ only to decrease the parameter $k$. To that end, we chose $l = 10$ and $L = C_0 \cup C_1 \cup C_{15} \cup C_{31} \cup C_{39} \cup C_{43} \cup C_{53} \cup C_{61} \cup C_{111} \cup C_{119}$ ($|L| = 69$)

which implies that $k = 4$ and we have checked that the corresponding system is of full rank. Such settings would require a total of 11 nonlinear multiplications following the original approach. However, they are better suited for our alternate cyclotomic approach than those proposed in [CRV14]. We also determined new sets of parameters for the cases $n \in \{5, 7\}$, which are given along with implementation results Section 6.

### 5.2.2 Complexity

Let us recall that $C_{\texttt{SecMult}}$ denotes the cost of a finite field multiplication which is secured with ISW. The cost to build the precomputed set of monomials is denoted by $C_{\texttt{Set}}$ and we denote by $C_{\texttt{Alt-CRV}}$ the overall cost of our alternate CRV proposal. Note that $C_{\texttt{Set}}$ represents the cost of our alternate cyclotomic proposal for polynomials that can be generated from $l$ distinct cyclotomic classes. Thus,

$$\mathtt{C}_{\texttt{Set}} = \mathtt{C}_{\delta} + \mathtt{C}_{\texttt{AMtoMM}} + (l-2) \times \mathtt{C}_{\texttt{MMtoAM}}\,,$$

and

$$\mathtt{C}_{\texttt{Alt-CRV}} = \mathtt{C}_{\texttt{Set}} + (k-1) \times \mathtt{C}_{\texttt{SecMult}}\,.$$

For the sake of clarity, Table 2 lists the complexities of our proposal and the original method in terms of elementary operations as a function of the order $d$. Also, as operations $\oplus, \oplus^n, \odot, \odot^n$ have the same complexity in practice (see Section 6), they are listed together. Operation $\mathtt{M}^\intercal$ denotes $(n \times n)$-matrix transpositions.

**Table 2:** Complexities of our proposal and the original method in terms of elementary operations.

| Operation | Our proposal |
|---|---|
| $\otimes$ | $(l+k-2)\,d^2 + (3l+2k-5)\,d + (k-1)$ |
| $\oplus, \oplus^n, \odot, \odot^n$ | $\left(l+2k-\frac{3}{n}\right) d^2 + \left(2l+2k-\frac{4}{n}-1\right) d + \left(2-\frac{1}{n}\right)$ |
| $\mathtt{M}^\intercal$ | $\frac{d}{n} + \frac{1}{n}$ |
| | [CRV14] |
| $\otimes$ | $(l+k-1)\,d^2 + (2l+2k-2)\,d + (l+k-1)$ |
| $\oplus$ | $(2l+2k-2)\,d^2 + (2l+2k-2)\,d + (l+k-1)$ |

As previously mentioned, in order to proceed to a fair comparison it should be noted that field multiplications with our proposal do not have the same weight as the ones that are implemented following the original method. Our approach allows to implement field multiplications more efficiently (see Section 6).

### 5.2.3 Security

We now describe the resulting alternate CRV construction that incorporates our alternate cyclotomic approach to build the precomputed set of monomials. Gadgets $\texttt{Alt-Cy}^1(\cdot)$ and $\texttt{Alt-Cy}^2(\cdot)$ of our alternate cyclotomic approach which have been analyzed in the previous section are thus involved in the full construction as illustrated Fig. 6. Note that $\texttt{Alt-Cy}^2(\cdot)$ enables to derive powers of the precomputed set and each gadget $\texttt{Alt-Cy}^1(\cdot)$ enables to generate distinct linearized polynomials by affecting different coefficients to powers belonging to the precomputed set. Therefore, each composition of a Gadget $\texttt{Alt-Cy}^2(\cdot)$ with a Gadget $\texttt{Alt-Cy}^1(\cdot)$ generates a new polynomial. Note also that Gadgets $\texttt{CRV}^1_\texttt{i}(\cdot)$ correspond to the products of the $p_i(x)$'s with the $q_i(x)$'s of (4) for which we prove the following Lemma.

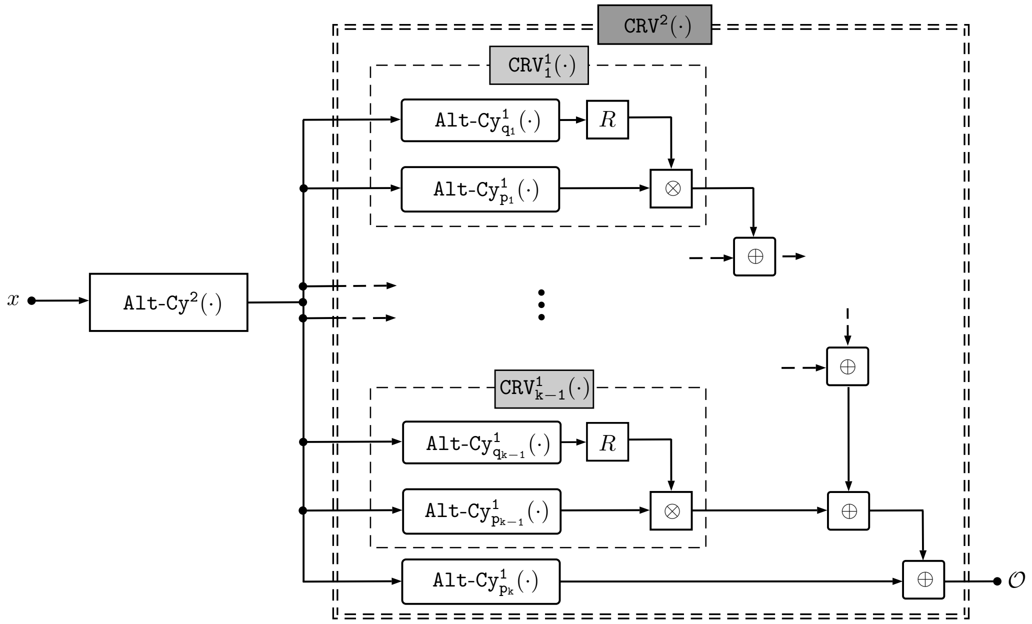
$\texttt{CRV}^2(\cdot)$
$\texttt{CRV}^1_\texttt{1}(\cdot)$
$\texttt{Alt-Cy}^1_{\texttt{q}_1}(\cdot)$
$R$
$\texttt{Alt-Cy}^1_{\texttt{p}_1}(\cdot)$
$\otimes$
$\oplus$
$x$
$\texttt{Alt-Cy}^2(\cdot)$
$\oplus$
$\texttt{CRV}^1_{\texttt{k-1}}(\cdot)$
$\texttt{Alt-Cy}^1_{\texttt{q}_{\texttt{k-1}}}(\cdot)$
$R$
$\texttt{Alt-Cy}^1_{\texttt{p}_{\texttt{k-1}}}(\cdot)$
$\otimes$
$\oplus$
$\texttt{Alt-Cy}^1_{\texttt{p}_\texttt{k}}(\cdot)$
$\oplus$
$\mathcal{O}$

**Figure 6:** Gadget $\texttt{Alt-CRV}(\cdot)$ : circuit of the alternate $\texttt{CRV}$ method.

*Lemma* 3. *Gadget* $\texttt{CRV}^1_\texttt{i}(\cdot)$ *is* $t$-$\texttt{SNI}$. *Let* $(x_i)_{0\le i\le d}$, $(\delta(x)_i)_{0\le i\le d}$ *and* $\left((x+\delta(x))_i^{\alpha_j}\right)_{0\le i\le d}$ *with* $j\in[2;q]$ *be the inputs and let* $(y_i)_{0\le i\le d}$ *be the output of Gadget* $\texttt{CRV}^1_\texttt{i}(\cdot)$ *represented Fig. 6. For any adversary set of $t$ probed wires $\Omega=(\mathcal{I},\mathcal{O})$, with $t\le d$, there exists a set $\mathcal{S}$ of input shares such that $|\mathcal{S}|\le|\mathcal{I}|$ from which $\Omega$ can be perfectly simulated.*

*Proof.* See Appendix C.1. □

Regarding Gadget $\texttt{CRV}^2(\cdot)$ we prove the following Lemma.

*Lemma* 4. $\texttt{CRV}^2(\cdot)$ *is* $t$-$\texttt{NI}$. *Let* $(x_i)_{0\le i\le d}$, $(\delta(x)_i)_{0\le i\le d}$ *and* $\left((x+\delta(x))_i^{\alpha_j}\right)_{0\le i\le d}$, $j\in[2,q]$ *be the inputs and let* $(y_i)_{0\le i\le d}$ *be the output of Gadget* $\texttt{CRV}^2(\cdot)$ *represented Fig. 14. For any adversary set of $t$ probed wires $\Omega=(\mathcal{I},\mathcal{O})$, with $t\le d$, there exists a set $\mathcal{S}$ of input shares such that $|\mathcal{S}|\le t$ from which $\Omega$ can be perfectly simulated.*

*Proof.* See Appendix C.2. □

We now prove the following theorem regarding the full construction of our alternate $\texttt{CRV}$ approach.

*Theorem* 5. *Alternate* $\texttt{CRV}$ *is* $t$-$\texttt{SNI}$. *Let* $(x_i)_{0\le i\le d}$ *be the input and let* $(y_i)_{0\le i\le d}$ *be the output of Gadget* $\texttt{Alt-CRV}(\cdot)$ *represented Fig. 6 and 7. For any adversary set of $t$ probed wires $\Omega=(\mathcal{I},\mathcal{O})$, with $t\le d$, there exists a set $\mathcal{S}$ of input shares such that $|\mathcal{S}|\le|\mathcal{I}|$ from which $\Omega$ can be perfectly simulated.*

*Proof.* Let $\Omega=((\mathcal{I}^1\cup\mathcal{I}^2),\mathcal{O})$ be an observation set we want to simulate, made on the whole circuit represented Fig. 7 (or equivalently Fig. 6) such that $|\mathcal{I}^2|+|\mathcal{I}^1|+|\mathcal{O}|\le t$.

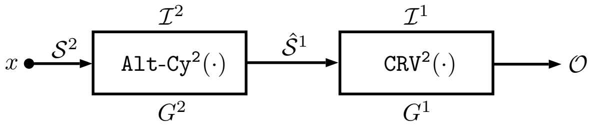

**Figure 7:** Gadget Alt-CRV$(\cdot)$.

**Gadget 1.** Let $\Omega^1 = (\mathcal{I}^1, \mathcal{O})$. Since $G^1$ is affine-NI (by Lemma 4) and $|\mathcal{I}^1 \cup \mathcal{O}| \leq t$ (by the global constraint), we know that there exists an observation set $\mathcal{S}^1$ such that $|\mathcal{S}^1| \leq |\mathcal{I}^1| + |\mathcal{O}|$ and the set of input shares $\mathcal{S}^1$ is sufficient to simulate the adversary observation set $\Omega^1$ made on Gadget 1.
**Gadget 2.** Let $\Omega^2 = (\mathcal{I}^2, \hat{\mathcal{S}}^1)$. Since $G^2$ is $t$-SNI (by Lemma 2) and $|\mathcal{I}^2 \cup \hat{\mathcal{S}}^1| \leq |\mathcal{I}^2| + |\mathcal{I}^1| + |\mathcal{O}| \leq t$ (by simulation of Gadget 1 and the global constraint), we know that there exists an observation set $\mathcal{S}^2$ such that $|\mathcal{S}^2| \leq |\mathcal{I}^2|$ and $\mathcal{S}^2$ is sufficient to simulate $\Omega^2$.

To simulate Gadget Alt-CRV$(\cdot)$, the simulator needs the shares $\mathcal{S}^2$ of $x$ and $|\mathcal{S}^2| \leq |\mathcal{I}^2| \leq |\mathcal{I}^1| + |\mathcal{I}^2| \leq t$. Therefore Alt-CRV$(\cdot)$ is $t$-SNI. □

# 6 Implementation Results

In this section we compare the efficiency of our alternate approach for the cyclotomic and the CRV methods with that of the original approach (CGPQR) for orders $d = 1, 2, 3$. We wrote the codes in assembly language for an 8051 based 8-bit architecture with bit-addressable memory and we provide implementation results for different settings related to S-boxes of size 4 to 8. For the sake of clarity, we begin to explicit our implementation choices and provide timings (in cycles) for several elementary operations. For elementary operations $\oplus$ and $\odot$, we experienced $\mathtt{C}_\oplus = \mathtt{C}_\odot = 1$ cycle.

**Finite field multiplications.** We tabulated them for S-boxes of dimension $n = 4$ at the cost of $2^8$ bytes of memory and we experienced $\mathtt{C}_\otimes = 10$ cycles. For larger dimensions, the memory required to store such tables becomes prohibitive. In cases $n \in [5; 8]$, we implemented finite fields multiplications using exp/log tables. This approach still requires to store two tables with $2^n$ entries each, but offers a good trade-off between execution time and memory cost. The most tricky part of the exp/log multiplication is to manage the case where the inputs equal 0 while avoiding any conditional branch. In our GPQ based alternate approaches, there are always one non-zero input involved in field multiplications which yields to slightly more efficient field multiplications than in the classical CGPQR approach. A time constant field multiplication is executed in $\mathtt{C}_\otimes = 38$ cycles in the context of CGPQR while it only takes $\mathtt{C}_\otimes = 25$ cycles in our alternate proposals.

**(8 × $n$)-matrix transposition.** We recalled in Section 3.1 a bit-sliced approach that computes $n$ Dirac functions simultaneously over $\mathbb{F}_{2^n}$. The procedure (Alg. 3) involves $(n \times n)$-matrix transpositions. However, on 8-bit architectures we are able to simultaneously compute 8 Dirac functions at a time for any S-box dimension lower or equal to 8 which consequently requires $(8 \times n)$-matrix transpositions. We experienced $\mathtt{C}_\mathtt{T} = 150$ cycles to transpose $(8 \times n)$-matrices for $5 \leq n \leq 8$. Note that 8-bit architectures allow us to fill each register with two elements of $\mathbb{F}_{2^4}$ in order to faster the transformation in that particular case leading to a cost $\mathtt{C}_\mathtt{T} = 75$ cycles.

We give costs of the transformations involve in GPQ along with the cost of a secure multiplication using ISW (Alg. 1) in Table 3.

**Table 3:** Costs of Secure-Dirac (Alg. 3), AMtoMM (Alg. 4), MMtoAM (Alg. 5) and SecMult (Alg. 1).

| | Costs (in cycles) | | | |
|---|---|---|---|---|
| Order ($d$) | $\mathtt{C}_\delta$ | $\mathtt{C}_{\mathtt{AMtoMM}}$ | $\mathtt{C}_{\mathtt{MMtoAM}}$ | $\mathtt{C}_{\mathtt{SecMult}}$ |
| 1 | 43 | 51 | 53 | 156 |
| 2 | 72 | 129 | 133 | 354 |
| 3 | 105 | 234 | 240 | 632 |

## 6.1 Cyclotomic method

The cyclotomic method only consists in evaluating a polynomial whose monomials may belong to any of the $q$ distinct cyclotomic classes of $[0, 2^n - 2]$. The classical CGPQR scheme requires to secure each of the $q-1$ nonlinear multiplications with ISW ($q-2$ if the S-box is balanced, see [CGP+12]). Considering our proposal, a secure polynomial evaluation implies to process 1 Secure-Dirac($\cdot$), 1 AMtoMM($\cdot$) and $q-1$ MMtoAM($\cdot$) ($q-2$ if the S-box is balanced). Table 4 lists the costs (in cycles) to evaluate polynomials over $\mathbb{F}_{2^n}$ with $n \in [4; 8]$.

**Table 4:** Costs of evaluating S-boxes of size $4 \leq n \leq 8$ with the cyclotomic method and our alternate proposal.

| | | $n$ | | | | |
|---|---|---|---|---|---|---|
| Method | Order ($d$) | 4 | 5 | 6 | 7 | 8 |
| Our proposal | 1 | **83** | **246** | **553** | **860** | **1677** |
| [CGP+12] | | 132 | 780 | 1716 | 2652 | 5148 |
| Our proposal | 2 | 276 | **585** | **1362** | **2138** | **4205** |
| [CGP+12] | | **174** | 1770 | 3894 | 6018 | 11682 |
| Our proposal | 3 | 477 | **1036** | **2445** | **3854** | **7603** |
| [CGP+12] | | **293** | 3160 | 6952 | 10744 | 20856 |

When finite field multiplications can be tabulated (when $n = 4$), our proposal does not lead to improvement of efficiency. In this case, the original approach is preferred. In all other scenarios, our proposal is approximatively 3 times faster at orders $d = 1, 2, 3$. Those results illustrates the efficiency of our extended version of GPQ for polynomials.

## 6.2 CRV method

Regarding the CRV method, its processing can be divided into two main stages. First it requires to generate polynomials whose monomials are derived from a set of $l$ distinct cyclotomic classes. This stage requires $l-2$ nonlinear multiplications with the classical approach or 1 Secure-Dirac($\cdot$), 1 AMtoMM($\cdot$) and $l-2$ MMtoAM($\cdot$) with ours. Then the evaluation is completed with $k-1$ additional nonlinear multiplications secured with ISW for both approaches.

**New parameters.** As mentioned in Section 5.2, our proposal enables to consider new settings for parameters $l$,$k$ and $L$. We list in Table 5 the settings that led to better performances in practice. We were able to derive more efficient parameters for S-boxes of dimension n with $n \in \{5,7,8\}$.

**Table 5:** New settings for parameters $k$ and $l$ of the CRV method.

| $n$ | $l$ | $k$ | $\|L\|$ | $L$ |
|---|---|---|---|---|
| 5 | 5 | 2 | 21 | $C_0 \cup C_1 \cup C_5 \cup C_7 \cup C_{15}$ |
| 7 | 8 | 3 | 50 | $C_0 \cup C_1 \cup C_3 \cup C_9 \cup C_{11} \cup C_{15} \cup C_{21} \cup C_{43}$ |
| 8 | 10 | 4 | 69 | $C_0 \cup C_1 \cup C_{15} \cup C_{31} \cup C_{39} \cup C_{43} \cup C_{53} \cup C_{61} \cup C_{111} \cup C_{119}$ |

We report in Table 6 the cost (in cycles) of the CRV method with the original approach compared to our proposal. Parameters $l$ and $k$ have been chosen accordingly to [CRV14] for the original approach, while our alternate proposal uses our new settings for S-boxes of dimension $n \in \{5,7,8\}$.

**Table 6:** Costs of evaluating S-boxes of size $4 \leq n \leq 8$ with the CRV method and our alternate proposal.

| | | $n$ | | | | |
|---|---|---|---|---|---|---|
| Method | Order ($d$) | 4 | 5 | 6 | 7 | 8 |
| Our proposal | 1 | 127 | **402** | **559** | **713** | **972** |
| [CRV14] | | **88** | 624 | 780 | 1092 | 1560 |
| Our proposal | 2 | 276 | **939** | **1296** | **1685** | **2300** |
| [CRV14] | | **204** | 1416 | 1770 | 2478 | 3540 |
| Our proposal | 3 | 477 | **1668** | **2305** | **3012** | **4117** |
| [CRV14] | | **368** | 2528 | 3160 | 4424 | 6320 |

Again in the particular case $n = 4$, the original approach is preferred since finite field multiplications can be tabulated. However, our alternate proposal outperforms the original in every other scenario.

## 7 Conclusion

In this paper, we have proven the security of the power function masking scheme GPQ under the $t$-SNI definition. We have extended the GPQ scheme to the evaluation of polynomials over $\mathbb{F}_{2^n}$ and we have proven the security of the resulting construction under the $t$-SNI definition. Our extension results in an alternate cyclotomic method which we have plugged into the CRV construction in order to speed up polynomial evaluations. We have analyzed our alternate CRV construction and we have proven that it is $t$-SNI. Moreover, we have provided new sets of parameters that improve even more the efficiency of our alternate approach for CRV. We have given implementation results in several realistic scenarios where S-boxes are of dimension $n \in \{4,5,6,7,8\}$. Given those results, we argue that our $t$-SNI proposal for polynomial evaluation over $\mathbb{F}_{2^n}$ is a better alternative than the original approach in all scenarios where finite field multiplications are not tabulated.

## References

[AG01] Mehdi-Laurent Akkar and Christophe Giraud. An implementation of DES and aes, secure against some attacks. In *Cryptographic Hardware and Embedded Systems - CHES 2001, Third International Workshop, Paris, France, May 14-16, 2001, Proceedings*, number Generators, pages 309–318, 2001.

[BBD$^+$15] Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, and Benjamin Grégoire. Compositional verification of higher-order masking: Application to a verifying masking compiler. *IACR Cryptology ePrint Archive*, 2015:506, 2015.

[BBP$^+$17] Sonia Belaïd, Fabrice Benhamouda, Alain Passelègue, Emmanuel Prouff, Adrian Thillard, and Damien Vergnaud. Private multiplication over finite fields. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, pages 397–426, 2017.

[BFG15] Josep Balasch, Sebastian Faust, and Benedikt Gierlichs. Inner product masking revisited. In *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, pages 486–510, 2015.

[CGP$^+$12] Claude Carlet, Louis Goubin, Emmanuel Prouff, Michaël Quisquater, and Matthieu Rivain. Higher-order masking schemes for s-boxes. In *Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers*, pages 366–384, 2012.

[CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, pages 398–412, 1999.

[Cor14] Jean-Sébastien Coron. Higher order masking of look-up tables. In *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, pages 441–458, 2014.

[CPRR13] Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, and Thomas Roche. Higher-order side channel security and mask refreshing. In *Fast Software Encryption - 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers*, pages 410–424, 2013.

[CPRR15] Claude Carlet, Emmanuel Prouff, Matthieu Rivain, and Thomas Roche. Algebraic decomposition for probing security. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, pages 742–763, 2015.

[CRV14] Jean-Sébastien Coron, Arnab Roy, and Srinivas Vivek. Fast evaluation of polynomials over binary finite fields and application to side-channel countermeasures. In *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, pages 170–187, 2014.

[DDF14] Alexandre Duc, Stefan Dziembowski, and Sebastian Faust. Unifying leakage models: From probing attacks to noisy leakage. In *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, pages 423–440, 2014.

[GM11] Louis Goubin and Ange Martinelli. Protecting AES with shamir's secret sharing scheme. In *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, pages 79–94, 2011.

[GPQ10] Laurie Genelle, Emmanuel Prouff, and Michaël Quisquater. Secure multiplicative masking of power functions. In *Applied Cryptography and Network Security, 8th International Conference, ACNS 2010, Beijing, China, June 22-25, 2010. Proceedings*, pages 200–217, 2010.

[GPQ11a] Laurie Genelle, Emmanuel Prouff, and Michaël Quisquater. Montgomery's trick and fast implementation of masked AES. In *Progress in Cryptology - AFRICACRYPT 2011 - 4th International Conference on Cryptology in Africa, Dakar, Senegal, July 5-7, 2011. Proceedings*, pages 153–169, 2011.

[GPQ11b] Laurie Genelle, Emmanuel Prouff, and Michaël Quisquater. Thwarting higher-order side channel analysis with additive and multiplicative maskings. In *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, pages 240–255, 2011.

[GSF14] Vincent Grosso, François-Xavier Standaert, and Sebastian Faust. Masking vs. multiparty computation: how large is the gap for aes? *J. Cryptographic Engineering*, 4(1):47–57, 2014.

[GT02] Jovan Dj. Golic and Christophe Tymen. Multiplicative masking and power analysis of AES. In *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, pages 198–212, 2002.

[ISW03] Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In D. Boneh, editor, *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481. Springer, 2003.

[KJJ99] P. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In M.J. Wiener, editor, *Advances in Cryptology – CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.

[Koc96] 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, 1996.

[PR11] Emmanuel Prouff and Thomas Roche. Higher-order glitches free implementation of the AES using secure multi-party computation protocols. In *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, pages 63–78, 2011.

[PR13] Emmanuel Prouff and Matthieu Rivain. Masking against side-channel attacks: A formal security proof. In *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, pages 142–159, 2013.

[RP10] Matthieu Rivain and Emmanuel Prouff. Provably secure higher-order masking of aes. In Stefan Mangard and François-Xavier Standaert, editors, *CHES*, volume 6225 of *Lecture Notes in Computer Science*, pages 413–427. Springer, 2010.

[RV13] Arnab Roy and Srinivas Vivek. Analysis and improvement of the generic higher-order masking scheme of FSE 2012. In *Cryptographic Hardware and Embedded Systems - CHES 2013 - 15th International Workshop, Santa Barbara, CA, USA, August 20-23, 2013. Proceedings*, pages 417–434, 2013.

# A Security analysis of GPQ

## A.1 Proof of Lemma 1.

For the sake of clarity, we divide the bit-sliced Secure-Dirac procedure into two stages as illustrated Fig. 8 and we give further details about the transformation before proving its security.

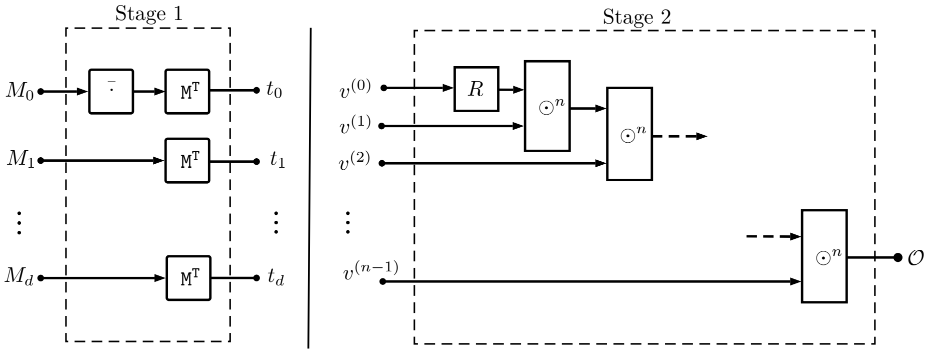

**Figure 8:** Gadget $\delta(\cdot)$.

Stage 1 is composed of the steps 1 to 4 of Alg. 3 and stage 2 illustrates the steps 5 to 9. As mentionned previously, the Secure-Dirac procedure uses bit-slicing and therefore processes simultaneously several elements of $\mathbb{F}_2^n$. We consider the case of $n$-bit architectures for which the transformation processes $n$ elements of $\mathbb{F}_2^n$ at a time. These $n$ elements are represented by a matrix $M \in \mathbb{F}_2^{n\times n}$ in such a way that each line of $M$ is one element of $\mathbb{F}_2^n$. Since the procedure manipulates masked data, stage 1 therefore takes as input a $(d+1)$-sharing $(M_0, \ldots, M_d)$ of $M$, with $M_j \in \mathbb{F}_2^{n\times n}$ for every $j \in [0; d]$. We hereafter exhibit the matrices involve in the computation. Namely, we have

$$M_j = \begin{pmatrix} \left(\pi_0\left(x^{(0)}\right)\right)_j & \left(\pi_1\left(x^{(0)}\right)\right)_j & \cdots & \left(\pi_{n-1}\left(x^{(0)}\right)\right)_j \\ \left(\pi_0\left(x^{(1)}\right)\right)_j & \left(\pi_1\left(x^{(1)}\right)\right)_j & \cdots & \left(\pi_{n-1}\left(x^{(1)}\right)\right)_j \\ \vdots & \vdots & \ddots & \vdots \\ \left(\pi_0\left(x^{(n-1)}\right)\right)_j & \left(\pi_1\left(x^{(n-1)}\right)\right)_j & \cdots & \left(\pi_{n-1}\left(x^{(n-1)}\right)\right)_j \end{pmatrix}$$

where $\left(\pi_k\left(x^{(i)}\right)\right)_j$ is the projection of the $k^{\text{th}}$ bit of the $j^{\text{th}}$ share of the element $x^{(i)}$ with $i$ in $[0; n-1]$, $j$ in $[0; d]$ and $k$ in $[0; n-1]$.

Stage 1 transposes the matrices $M_j$ for every $j \in [0; d]$. Note that the bit-wise complement (step 1 of Alg. 3) is only performed over the elements of $M_0$. The transposed matrices are denoted by $t_j$ and we therefore have $t_0 = \left(\overline{M_0}\right)^{\intercal}$ and $t_j = (M_j)^{\intercal}$ for $j \in [1; d]$. Then, stage 1 outputs the $d+1$ binary $(n \times n)$-matrices $t_0, \ldots, t_d$ such that

$$t_j = \begin{pmatrix} \left(\pi_0\left(x^{(0)}\right)\right)_j & \left(\pi_0\left(x^{(1)}\right)\right)_j & \cdots & \left(\pi_0\left(x^{(n-1)}\right)\right)_j \\ \left(\pi_1\left(x^{(0)}\right)\right)_j & \left(\pi_1\left(x^{(1)}\right)\right)_j & \cdots & \left(\pi_1\left(x^{(n-1)}\right)\right)_j \\ \vdots & \vdots & \ddots & \vdots \\ \left(\pi_{n-1}\left(x^{(0)}\right)\right)_j & \left(\pi_{n-1}\left(x^{(1)}\right)\right)_j & \cdots & \left(\pi_{n-1}\left(x^{(n-1)}\right)\right)_j \end{pmatrix}.$$

Finally, stage 2 takes as inputs $n$ distinct vectors $v^{(k)} = \left(t_0^{(k)}, \ldots, t_d^{(k)}\right)$ where $t_j^{(k)}$ is the $k^{\text{th}}$ line of the matrix $t_j$. In other words, $t_j^{(k)}$ is a $n$-tuple composed of the $k^{\text{th}}$ bits of the $j^{\text{th}}$ shares of all input elements and $v^{(k)}$ is therefore composed of the $k^{\text{th}}$ bit of all the shares of every input elements. In the following proof, we assume that if a single internal bit has to be simulated then the whole word corresponding to this single bit is required.

As in [BBD+15], the proof is constructed by composition. Namely, we construct the simulator for the whole circuit by simulating sequentially each inner gadget from right to left. We begin our security analysis by stage 2 which we also divide into two parts (see Fig. 9).

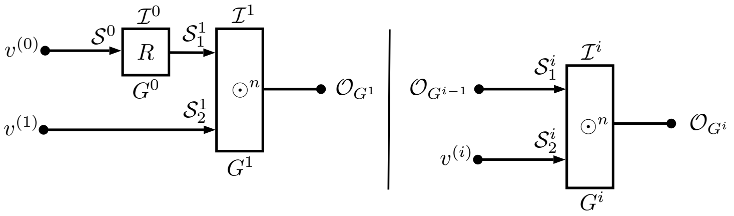

**Figure 9:** Stage 2 of Gadget $\delta(\cdot)$.

*Proof.* Let $\Omega = (\mathcal{I}, \mathcal{O})$ be an observation set made on stage 2 such that $\mathcal{I} = \bigcup_{0 \leq i \leq n-1} \mathcal{I}^i$ and such that the global constraint $\sum_{i=0}^{n-1} |\mathcal{I}^i| + |\mathcal{O}| \leq t$ is satisfied.

Let us first consider the right side of Fig. 9. For every $i \in [2, n-1]$, we want to simulate the observation set $\Omega^i = (\mathcal{I}^i, \mathcal{O}_{G^i})$ made on Gadget $i$. Since Gadget $i$ is $t$-SNI and $|\mathcal{I}^{n-1} \cup \mathcal{O}| \leq t$ (by global constraint) and $|\mathcal{I}^i \cup \mathcal{O}_{G^i}| \leq t$ for $i \in [2, n-2]$ (by global constraint and simulation of Gadget $i+1$), we know that there exists an observation set $\mathcal{S}^i = (\mathcal{S}_1^i, \mathcal{S}_2^i)$ such that $|\mathcal{S}_1^i| \leq |\mathcal{I}^i|$, $|\mathcal{S}_2^i| \leq |\mathcal{I}^i|$ and $\mathcal{S}_1^i \cup \mathcal{S}_2^i$ is sufficient to simulate Gadget $i$ (*i.e.* simulate $\Omega^i$) for every $i \in [2, n-1]$. As illustrated Fig. 9 and by the $t$-SNI property of Gadget $i$ for every $i \in [2, n-1]$, the simulation of these Gadgets therefore requires at most $|\mathcal{I}^i|$ shares of $v^{(i)}$, and at most $|\mathcal{I}^i|$ shares of the output of Gadget $i-1$ for every $i \in [2, n-1]$.

Let us now also take into account the left side of Fig. 9. It has been shown in [BBD+15] that such a composition of $t$-SNI gadgets is $t$-SNI thanks to the additional $t$-SNI refreshing Gadget (Alg. 2). Thus, in order to simulate the observation set $\Omega^{0,1} = ((\mathcal{I}^0 \cup \mathcal{I}^1), \mathcal{O}_{G^1})$

and since $|\mathcal{I}^0 \cup \mathcal{I}^1 \cup \mathcal{O}_{G^1}| \leq t$ (by global constraint and simulation of Gadgets $i$ for every $i \in [2, n-1]$), the corresponding simulator requires at most $|\mathcal{I}^0|$ shares of $v^{(0)}$ and at most $|\mathcal{I}^1|$ shares of $v^{(1)}$.

Altogether, the simulation of stage 2 requires at most $|\mathcal{I}^i|$ shares of $v^{(i)}$ for every $i \in [0, n-1]$.

Let us now take into account stage 1. As mentioned previously, $v^{(i)} = \left(t_0^{(i)}, \ldots, t_d^{(i)}\right)$ for every $i \in [0, n-1]$, which means that the $v^{(i)}$'s are composed of the $i^{\text{th}}$ bit of all the shares of every input elements. Let us also remind that we assume that if a single internal bit has to be simulated then the whole word corresponding to this single bit is required. For the sake of clarity, we illustrate in Fig. 10 the propagation of a share $v^{(i)}$ throughout stage 1. As an example, we consider the case where the simulation requires the first share $t_0^{(i)}$ of $v^{(i)}$ and show how it is actually related to the first shares of each input elements.

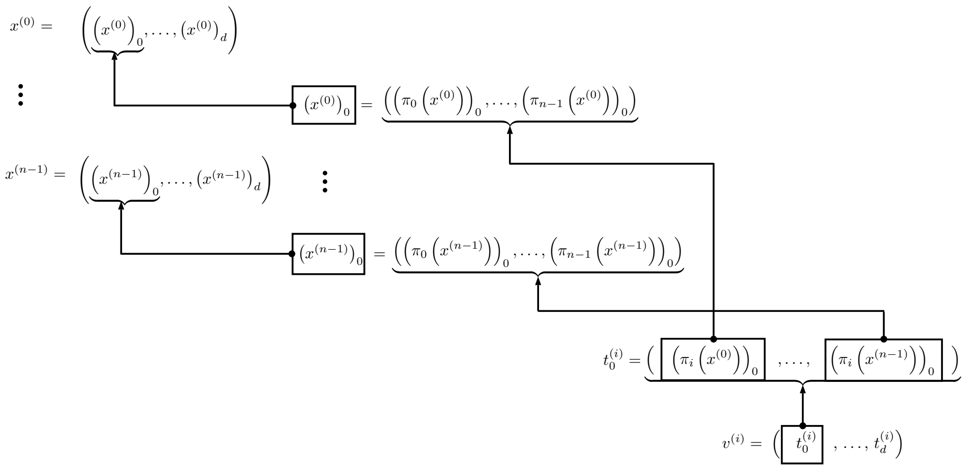
$x^{(0)} = \left(\left(x^{(0)}\right)_0, \ldots, \left(x^{(0)}\right)_d\right)$
$\left(x^{(0)}\right)_0 = \left(\left(\pi_0\left(x^{(0)}\right)\right)_0, \ldots, \left(\pi_{n-1}\left(x^{(0)}\right)\right)_0\right)$
$x^{(n-1)} = \left(\left(x^{(n-1)}\right)_0, \ldots, \left(x^{(n-1)}\right)_d\right)$
$\left(x^{(n-1)}\right)_0 = \left(\left(\pi_0\left(x^{(n-1)}\right)\right)_0, \ldots, \left(\pi_{n-1}\left(x^{(n-1)}\right)\right)_0\right)$
$t_0^{(i)} = \left(\left(\pi_i\left(x^{(0)}\right)\right)_0, \ldots, \left(\pi_i\left(x^{(n-1)}\right)\right)_0\right)$
$v^{(i)} = \left(t_0^{(i)}, \ldots, t_d^{(i)}\right)$

**Figure 10:** Linking the shares of the $v^i$'s to the shares of the input elements.

The above figure shows that if the simulation requires the $j^{\text{th}}$ share of $v^{(i)}$, then the simulation actually requires the $j^{\text{th}}$ shares of all input elements. Moreover and as discussed above, the simulation of the procedure requires $|\mathcal{I}^i|$ shares of $v^{(i)}$ for every $i \in [0, n-1]$. Oberserve that the simulation may involve $|\mathcal{I}^i|$ distinct shares of $v^{(i)}$ for every $i = [0, n-1]$. Therefore, at most $\sum_{i=0}^{n-1} |\mathcal{I}^i|$ distinct shares of each input element of the `Secure-Dirac` procedure are actually required. Also, by the global constraint $\sum_{i=0}^{n-1} |\mathcal{I}^i| \leq t$ and consequently the $t$-`SNI` property is satisfied for the whole circuit of the `Secure-Dirac` procedure. □

*Remark* 2. In order to satisfy the $t$-`SNI` property, the shares of $v^{(0)}$ have to be refreshed in stage 2 thanks to Algorithm 2. This mask refreshing was not required in the original approach that only proves the security of `Secure-Dirac` under the less stronger $t$-`NI` security definition.

## A.2 Proof of Theorem 1

The `AMtoMM`$(\cdot)$ transformation converts an additively masked element $x \in \mathbb{F}_{2^n}^*$ into a multiplicative masking. Initially $x$ is represented by a $(d+1)$-additive sharing $(X_0, \ldots, X_d)$

involving $d$ additive masks $(X_i)_{1\leq i\leq d}$ such that $\sum_{i=0}^{d} X_i = x$. A sequence of transformations is carried out over the successive intermediate maskings of $x$ to finally produce a $(d+1)$-multiplicative sharing $(Z_0, \dots, Z_d)$ of $x$ such that $\prod_{i=0}^{d} Z_i = x$.

More precisely, Alg. 4 randomly generates multiplicative masks $(Z_i)_{0\leq i\leq d-1}$ and computes the sequence $X^{(i+1)} = \pi_i(X^{(i)}, Z_i)$ for $i$ in $[0; d-1]$ where $X^{(i)} = (X_0^{(i)}, X_1^{(i)}, \dots, X_{d-i}^{(i)})$. Thus, $X^{(0)}$ is the input of Alg. 4, $X^{(d)}$ is the output and the other $X^{(i)}$'s are intermediate maskings of $x$. We illustrate the above discussion in Fig. 11.

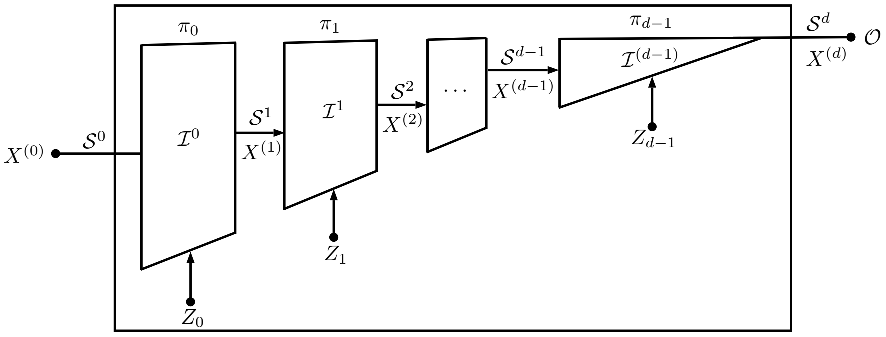

**Figure 11:** Internal functioning of Gadget AMtoMM$(\cdot)$.

The $\mathcal{I}^i$'s are random vectors whose components are some intermediate variables that appear during the corresponding transformation $\pi_i$ and may also be composed of some of $X^{(i)}$'s shares. The $\mathcal{S}^i$'s specify which components of $X^{(i)}$ are considered. In the following we denote by $\widetilde{X}^{(i)}$ vectors only composed of the shares of $X^{(i)}$ and specified by $\mathcal{S}^i$. Also, $\mathcal{O} = \widetilde{X}^{(d)} \cup Z_{|_J}$ where $Z_{|_J} = (Z_i)_{i\in J}$.

Our goal is to prove that Gadget AMtoMM$(\cdot)$ is t-SNI.

*Proof.* Let $\Omega = (\mathcal{I}, \mathcal{O})$ be an adversary observation set constructed over Gadget AMtoMM$(\cdot)$ with $\mathcal{I} = \cup_{i=0}^{d-1}\mathcal{I}^i$ such that the global constraint $|\mathcal{I} \cup \mathcal{O}| \leq t$ is satisfied.
In order to prove that Alg. 4 is t-SNI, we prove that $\Omega = (\mathcal{I}, \mathcal{O})$ may be simulated from a set of its input shares $\mathcal{S}^0$ with $|\mathcal{S}^0| \leq |\mathcal{I}|$. More precisely, we prove that any adversary view $\Omega$ satisfying the global constraint may be expressed as a function $\rho$ of $\widetilde{X}^{(0)}$ and a uniform random vector $\mathcal{U}$ such that $\Omega = \rho(\widetilde{X}^{(0)}, \mathcal{U})$, where $\widetilde{X}^{(0)} \in \mathbb{F}_{2^n}^{|\mathcal{S}^0|}$ and $|\mathcal{S}^0| \leq |\mathcal{I}|$. This is achieved in two steps. We first prove that the adversary view $\Omega$ may be expressed in terms of a vector $(\widetilde{X}^{(0)}, \widetilde{X}^{(1)}, \dots, \widetilde{X}^{(d)}, z_{|_{J\cup L}})$ and a uniform random vector $(\mathcal{U}_{d-2}, \cdots, \mathcal{U}_0)$. We then prove that $(\widetilde{X}^{(0)}, \widetilde{X}^{(1)}, \dots, \widetilde{X}^{(d)}, z_{|_{J\cup L}})$ may be expressed as $h(\widetilde{X}^{(0)}, \mathcal{U}')$ with $|\mathcal{S}^0| \leq |\mathcal{I}|$ and $\mathcal{U}'$ is a uniform random vector.

Let us first build the sequence of $\mathcal{S}^i$'s.

If $X^{(d)}$ is a component of $\mathcal{O}$, then $\mathcal{S}^d = 0$ and thus $|\mathcal{S}^d| = 1$, otherwise $\mathcal{S}^d = \emptyset$ and $|\mathcal{S}^d| = 0$. Then the other $\mathcal{S}^i$'s are defined according to the following discussion.

We start from the end of the evaluation of AMtoMM$(\cdot)$ and we therefore first consider the transformation

$$\pi_{d-1}(X^{(d-1)}, Z_{d-1}) \longrightarrow (X^{(d)}),$$

where $X^{(d-1)} = (X_0^{(d-1)}, X_1^{(d-1)})$ and $X^{(d)} = X_0^{(d)}$.

We list hereafter the variables computed during this transformation :

- $W_0^{(d)} = X_0^{(d-1)} \cdot Z_{d-1}$
- $W_1^{(d)} = X_1^{(d-1)} \cdot Z_{d-1}$
- $X_0^{(d)} = W_0^{(d)} + W_1^{(d)}$

Let $\mathcal{I}^{d-1}$ be a subset of $\{X_0^{(d-1)}, X_1^{(d-1)}, W_0^{(d)}, W_1^{(d)}\}$ and $\overset{\circ}{\mathcal{I}}{}^{d-1} = \mathcal{I}^{d-1} \backslash \{X_0^{(d-1)}, X_1^{(d-1)}\}$. On the one hand, if $\overset{\circ}{\mathcal{I}}{}^{d-1} = \emptyset$ then $\mathcal{I}^{d-1}$ is only composed of input variables of $\pi_{d-1}$, thus all variables of $\mathcal{I}_{d-1}$ may be expressed from $\widetilde{X}^{(d-1)} \in \mathbb{F}_{2^n}^{|S^{d-1}|}$ with $|S^{d-1}| = |\mathcal{I}^{d-1}|$. On the other hand, if $\overset{\circ}{\mathcal{I}}{}^{d-1} \neq \emptyset$, noting that $W_0^{(d)}$ (resp. $W_1^{(d)}$) can be expressed from $X_0^{(d-1)}$ (resp. $X_1^{(d-1)}$) and $Z_{d-1}$, $\mathcal{I}^{d-1}$ may be expressed in terms of $\widetilde{X}^{(d-1)}$ with $|S^{d-1}| \leq |\mathcal{I}^{d-1}|$. Consequently in any cases, all variables of $\mathcal{I}^{d-1}$ may be expressed as a function of $Z_{d-1}$ and $\widetilde{X}^{(d-1)} \in \mathbb{F}_{2^n}^{|S^{d-1}|}$, i.e.

$$\mathcal{I}^{d-1} = \rho_{d-1}(\widetilde{X}^{(d-1)}, Z_{d-1}) \text{ with } |S^{d-1}| \leq |\mathcal{I}^{d-1}|\,. \tag{5}$$

Let us define the sets $S_i = \{0, \ldots, d-i\}$ and $T_i = \{1, \ldots, d-i-1\}$ for $i = (d-2) \cdots 0$. Our goal is now to prove that for any $i = (d-2) \cdots 0$, for any $\mathcal{I}^i$ with $\overset{\circ}{\mathcal{I}}{}^i \neq \emptyset$ and for any subset $\mathcal{S}^{i+1} \subseteq S_{i+1}$ there exists a subset $\mathcal{S}^i \subseteq S_i$, a uniform vector $\mathcal{U}_i$ stochastically independent of all other random vectors and an application $\rho_i$ such that

$$\mathcal{I}^i = \rho_i(\widetilde{X}^{(i)}, \widetilde{X}^{(i+1)}, Z_i, \mathcal{U}_i) \text{ with } |\mathcal{S}^i| \leq |\mathcal{I}^i| + |\mathcal{S}^{i+1}|\,. \tag{6}$$

We list hereafter the variables involved in the $\pi_i$ transformation, with $i \in \{0, \ldots, d-2\}$ :

- $X_j^{(i+1)} \sim \mathcal{U}(\mathbb{F}_{2^n})$ with $j \in T_i$
- $W_j^{(i+1)} = X_j^{(i)} \cdot Z_i$ with $j \in S_i$
- $Y_j^{(i+1)} = W_j^{(i+1)} + X_j^{(i+1)}$ with $j \in T_i$
- $H_j^{(i+1)} = W_0^{(i+1)} + \sum_{t=1}^{j} Y_t^{(i+1)} = H_{j-1}^{(i+1)} + Y_j^{(i+1)}$ with $j \in T_i$. Note that $H_0^{(i+1)} = W_0^{(i+1)}$.
- $X_0^{(i+1)} = H_{d-i-1}^{(i+1)} + W_{d-i}^{(i+1)}$.

We have

$$\mathcal{I}^i = (X^{(i)}_{|_{I_i}}, W^{(i+1)}_{|_{K_i}}, Y^{(i+1)}_{|_{L_i}}, H^{(i+1)}_{|_{Q_i}})\,,$$

with $I_i \subseteq S_i$, $K_i \subseteq S_i$, $L_i \subseteq T_i$, $Q_i \subseteq T_i$.

First note that $X^{(i)}_{|_{I_i}}$ and $W^{(i)}_{|_{K_i}}$ may be expressed as a function of $X^{(i)}_{|_{I_i \cup K_i}}$ and $Z_i$. Therefore, $\mathcal{S}^i \supseteq I_i \cup K_i$. For any $j \in \mathcal{S}^{i+1} \cap L_i$, $\widetilde{X}^{(i+1)}$ and $X_j^{(i)}$ are necessary to simulate $Y_j^{(i+1)}$. Therefore, $\mathcal{S}^i \supseteq \mathcal{S}^{i+1} \cap L_i$. For any $j \in L_i \backslash \mathcal{S}^{i+1}$, $Y_j^{(i+1)}$ may be simulated from a uniform random variable stochastically independent from any other random variables.

If $Q_i = \emptyset$, $\mathcal{I}^i$ may be expressed as a function of $\widetilde{X}^{(i+1)}$, $\widetilde{X}^{(i)}$ (with $\mathcal{S}^i = I_i \cup K_i \cup (\mathcal{S}^{i+1} \cap L_i)$), $Z_i$ and a uniform random vector $\mathcal{U}_i$ stochastically independent from any other random variables. It follows that

$$|\mathcal{S}^i| \leq |I_i| + |K_i| + |\mathcal{S}^{i+1}| \leq |\mathcal{I}^i| + |\mathcal{S}^{i+1}|\,.$$

The condition (6) is therefore satisfied in this case.

If $Q_i \neq \emptyset$, i.e. $Q_i = \{s_1, \cdots, s_t\}$. Applying a well determined invertible linear application to $H^{(i+1)}_{|Q_i}$, we observe that the simulation of $H^{(i+1)}_{|Q_i}$ is equivalent to the simulation of the vector

$$\left( X^{(i)}_0 \cdot Z_i + \sum_{j=1}^{s_1} Y^{(i+1)}_j, \sum_{j=s_1+1}^{s_2} Y^{(i+1)}_j, \ldots, \sum_{j=s_{t-1}+1}^{s_t} Y^{(i+1)}_j \right) .$$

If the interval $[1; s_1] \subseteq \mathcal{S}^{i+1}$ then the first component of the above vector may be expressed as a function of $\widetilde{X}^{(i+1)}$, $Z_i$ and the random variables $X^{(i)}_j$ for $j \in [1; s_1] \cup \{0\}$. Otherwise, the first component may be simulated from a uniform random variable stochastically independent from any other random variables. If the interval $[s_i; s_{i+1}] \subseteq \mathcal{S}^{i+1}$ then the corresponding component of the vector may be expressed as a function of $\widetilde{X}^{(i+1)}$, $Z_i$ and the random variables $X^{(i)}_j$ for $j \in [s_i; s_{i+1}]$. Otherwise, this component may be simulated from a uniform random variable stochastically independent from any other random variables. It follows that $\mathcal{I}^i$ may be expressed as a function of $\widetilde{X}^{(i)}$ (with $\mathcal{S}^i = I_i \cup K_i \cup (\mathcal{S}^{i+1} \cap L_i) \cup \mathcal{S}^{i+1} \cup \{0\}$), $Z_i$ and a uniform random vector $\mathcal{U}_i$ stochastically independent from any other random variables. Since $Q_i \neq \emptyset$ by assumption, it follows that

$$|\mathcal{S}^i| \leq |I_i| + |K_i| + |L_i| + |\mathcal{S}^{i+1}| + 1 \leq |\mathcal{I}^i| + |\mathcal{S}^{i+1}| .$$

The condition (6) is therefore satisfied in this case.

Observe that for any $\mathcal{I}^i$ with $\overset{\circ}{\mathcal{I}^i} = \emptyset$ we have $\mathcal{I}^i = \widetilde{X}^{(i)}$ with $\mathcal{S}^i = \mathcal{I}^i$. In this case, we have therefore $|\mathcal{S}^i| = |\mathcal{I}^i|$.

Define $L = \{i \mid \overset{\circ}{\mathcal{I}^i} \neq \emptyset\}$ with $\overset{\circ}{\mathcal{I}^i} = \mathcal{I}^i \setminus \widetilde{X}^{(i)}$. Let us prove that $(\widetilde{X}^{(0)}, \widetilde{X}^{(1)}, \ldots, \widetilde{X}^{(d)}, Z_{|_{J \cup L}})$ may be expressed as $h(\widetilde{X}^{(0)}, \mathcal{U}')$ with $|\mathcal{S}^0| \leq |\mathcal{I}|$, $\mathcal{U}'$ is a uniform random variable stochastically independent of $\widetilde{X}^{(0)}$ and $\widetilde{X}^{(i)}$ are vectors only composed of the shares of $X^{(i)}$ which are specified by $\mathcal{S}^i$.

Suppose that there does not exist an indice $k$ such that $|\mathcal{S}^k| = d - k + 1$, then none of the sets $\mathcal{S}^i$ are $S_i = \{0, \ldots, d - i\}$ and thus all the $\widetilde{X}^{(i)}$'s and $Z_{|_{J \cup L}}$ are uniform stochastically independent random vectors. It follows that $(\widetilde{X}^{(0)}, \widetilde{X}^{(1)}, \ldots, \widetilde{X}^{(d)}, Z_{|_{J \cup L}})$ may clearly be expressed as $h(\widetilde{X}^{(0)}, \mathcal{U}')$.

Define now $k$ as the smallest index such that $|S^k| = d - k + 1$. Note that $k \geq 1$ by the global constraint $|\mathcal{I} \cup \mathcal{O}| \leq t$.

For the case $k = d$, all the $\widetilde{X}^{(i)}$ with $i$ in $[0; d-1]$ are uniform stochastically independent random vectors of $\mathbb{F}_{2^n}^{|\mathcal{S}^i|}$ respectively. Also, $(\widetilde{X}^{(d)}, Z_{|_{J \cup L}})$ is a uniform random vector of $(\mathbb{F}^*_{2^n})^{1+|J \cup L|}$ with $|J \cup L| \leq t - 1$ according to the global constraint $|\mathcal{I} \cup \mathcal{O}| \leq t$. It follows that if $k = d$ then $(\widetilde{X}^{(0)}, \widetilde{X}^{(1)}, \ldots, \widetilde{X}^{(d)}, Z_{|_{J \cup L}})$ may be expressed as $h(\widetilde{X}^{(0)}, \mathcal{U}')$.

Let us now assume that $1 \leq k \leq d - 1$. Gathering conditions (5) and (6), we have $|\mathcal{S}^k| \leq \sum_{i=k}^{d-1} |\mathcal{I}^i| \leq |\mathcal{I}|$. Also, $|\mathcal{I}| + |\mathcal{O}| \leq |\mathcal{I} \cup \mathcal{O}| \leq t$ by the global constraint. It follows that $|\mathcal{O}| \leq t - |\mathcal{S}^k|$. Note that

$$X^{(k)} = \underbrace{(x \cdot \prod_{i=0}^{k-1} Z_i + \sum_{j=1}^{d-i} X_j^{(k)}}_{X_0^{(k)}}, X_1^{(k)}, \ldots, X_{d-k}^{(k)}).$$

Remembering that $t \leq d$ and that $|S^k| = d - k + 1$, we have $|\mathcal{O}| \leq k - 1$. It follows that at most $k - 1$ $Z_i$'s have to be simulated among the $k$ $Z_i$'s in the expression of $X_0^{(k)}$, *i.e.* $p^* = x \cdot \prod_{i=0}^{k-1} Z_i$ is therefore a uniform random variable of $\mathbb{F}_{2^n}^*$ stochastically independent of the random variables in the set $\mathcal{O}$. All other random variables $\widetilde{X}^{(i)}$ with $i > k$ and $Z_{|_{(J \cup L) \cap \{k,\ldots,d-1\}}}$ may be build from $p^*$, random vectors of $\mathbb{F}_{2^n}^{|\mathcal{S}^i|}$ and $(\mathbb{F}_{2^n}^*)^{|(J \cup L) \cap \{k,\ldots,d-1\}|}$. From the above discussion, it follows that $(\widetilde{X}^{(0)}, \widetilde{X}^{(1)}, \ldots, \widetilde{X}^{(d)}, Z_{|_{J \cup L}})$ may be expressed as $h(\widetilde{X}^{(0)}, \mathcal{U}')$ with $|\mathcal{S}^0| \leq |\mathcal{I}|$ and $\mathcal{U}'$ is a uniform random vector.
Finally, from conditions (5) and (6) it follows that $\Omega = (\mathcal{I}, \mathcal{O})$ may be expressed as $\rho(\widetilde{X}^{(0)}, \mathcal{U})$ which means that `AMtoMM`$(\cdot)$ is `t-SNI`.

□

### A.3 Proof of Theorem 2

The proof concerning `MMtoAM`$(\cdot)$ is very similar to the one of `AMtoMM`$(\cdot)$. It consists essentially in interchanging the role of the additive masks with the multiplicative ones in the previous proof of `AMtoMM`$(\cdot)$.

## B Security analysis of alternate cyclotomic.

### B.1 Proof of Lemma 2

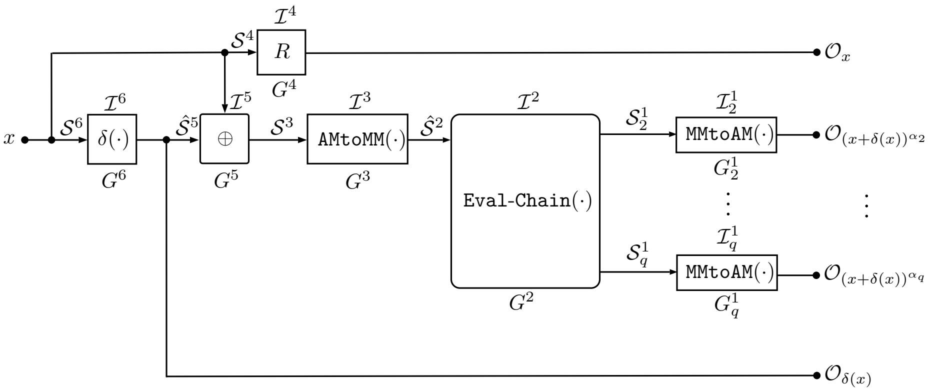

**Figure 12:** Gadget `Alt-Cy`$^2(\cdot)$.

As previously, we build the simulator for the circuit Fig. 12 by simulating sequentially each inner gadget from right to left.

*Proof.* Let $\Omega = (\mathcal{I}, \mathcal{O})$ be the adversary observation set that we want to simulate and which is made on the whole circuit of Fig. 12, with $\mathcal{O} = (\mathcal{O}_x, \mathcal{O}_{(x+\delta(x))^{\alpha_2}}, \ldots, \mathcal{O}_{(x+\delta(x))^{\alpha_q}}, \mathcal{O}_{\delta(x)})$ such that $\mathcal{I} = \left(\bigcup_{j=2}^{q} \mathcal{I}_j^1\right) \cup \left(\bigcup_{i=2}^{6} \mathcal{I}^i\right)$ and such that $\left(\sum_{j=2}^{q} |\mathcal{I}_j^1| + \sum_{i=2}^{6} |\mathcal{I}^i|\right) + |\mathcal{O}| \leq t$.

**Gadgets 1.** For every $j \in [2;q]$, let $\Omega_j^1 = (\mathcal{I}_j^1, \mathcal{O}_{(x+\delta(x))^{\alpha_j}})$ be an observation set made on $G_j^1$. Since $G_j^1$ is $t$-SNI and $|\mathcal{I}_j^1 \cup \mathcal{O}_{(x+\delta(x))^{\alpha_j}}| \leq t$ (by global constraint), we know that for every observation set $\Omega_j^1$ there exists an observation set $\mathcal{S}_j^1$ for $G_j^1$ such that $|\mathcal{S}_j^1| \leq |\mathcal{I}_j^1|$ and $\mathcal{S}_j^1$ is sufficient to simulate $\Omega_j^1$ for every $j \in [2,q]$.

**Gadget 2.** Let $\Omega^2 = (\mathcal{I}^2, (S_2^1, \ldots, S_q^1))$. Since $G^2$ is `affine-NI`, we know that for every observation set $\Omega^2$ there exists an observation set $\hat{\mathcal{S}}^2$ such that $|\hat{\mathcal{S}}^2| \leq |\mathcal{I}^2 \cup \left(\bigcup_{2\leq j\leq q} S_j^1\right)| \leq |\mathcal{I}^2| + \sum_{j=2}^{q} |\mathcal{I}_j^1|$ and the set of input shares $\hat{\mathcal{S}}^2$ is sufficient to simulate the adversary observation set $\Omega^2$ made on Gadget 2.

**Gadget 3.** Let $\Omega^3 = (\mathcal{I}^3, \hat{\mathcal{S}}^2)$. Since $G^3$ is $t$-SNI and $|\mathcal{I}^3 \cup \hat{\mathcal{S}}^2| \leq t$ (by simulation of Gadget 2 and the global constraint), we know that for every observation set $\Omega^3$ there exists an observation set $\mathcal{S}^3$ such that $|\mathcal{S}^3| \leq |\mathcal{I}^3|$ and $\mathcal{S}^3$ is sufficient to simulate $\Omega^3$.

**Gadget 4.** Let $\Omega^4 = (\mathcal{I}^4, \mathcal{O}_x)$. Since $G^4$ is $t$-SNI and $|\mathcal{I}^4 \cup \mathcal{O}_x| \leq t$ (by the global constraint), we know that for every observation set $\Omega^4$ there exists an observation set $\mathcal{S}^4$ such that $|\mathcal{S}^4| \leq |\mathcal{I}^4|$ and $\mathcal{S}^4$ is sufficient to simulate $\Omega^4$.

**Gadget 5.** Let $\Omega^5 = (\mathcal{I}^5, \mathcal{S}^3)$. Since $G^5$ is `affine-NI`, we know that for every observation set $\Omega^5$ there exists an observation set $\hat{\mathcal{S}}^5$ such that $|\hat{\mathcal{S}}^5| \leq |\mathcal{I}^5 \cup \mathcal{S}^3| \leq |\mathcal{I}^5| + |\mathcal{I}^3|$ and the set of input shares $\hat{\mathcal{S}}^5$ is sufficient to simulate the adversary observation set $\Omega^5$ made on Gadget 5.

**Gadget 6.** Let $\Omega^6 = (\mathcal{I}^6, \hat{\mathcal{S}}^5 \cup \mathcal{O}_{\delta(x)})$. Since $G^6$ is $t$-SNI and $|\mathcal{I}^6 \cup \hat{\mathcal{S}}^5 \cup \mathcal{O}_{\delta(x)}| \leq |\mathcal{I}^6| + |\mathcal{I}^5| + |\mathcal{I}^3| + |\mathcal{O}_{\delta(x)}| \leq t$ (by simulation of Gadget 5 and the global constraint), we know that for every observation set $\Omega^6$ there exists an observation set $\mathcal{S}^6$ such that $|\mathcal{S}^6| \leq |\mathcal{I}^6|$ and $\mathcal{S}^6$ is sufficient to simulate $\Omega^6$.

In order to simulate Gadget `Alt`-$Cy^2(\cdot)$, the corresponding simulator requires the shares $S^6 \cup \hat{\mathcal{S}}^5 \cup \mathcal{S}^4$ and $|S^6 \cup \hat{\mathcal{S}}^5 \cup \mathcal{S}^4| \leq |\mathcal{I}^6| + |\mathcal{I}^5| + |\mathcal{I}^4| + |\mathcal{I}^3| \leq \sum_{j=2}^{q} |\mathcal{I}_j^1| + \sum_{i=2}^{6} |\mathcal{I}^i| \leq t$. Therefore, Gadget $\texttt{Alt-Cy}^2(\cdot)$ is $t$-SNI. □

# C Security analysis of alternate CRV

## C.1 Proof of Lemma 3.

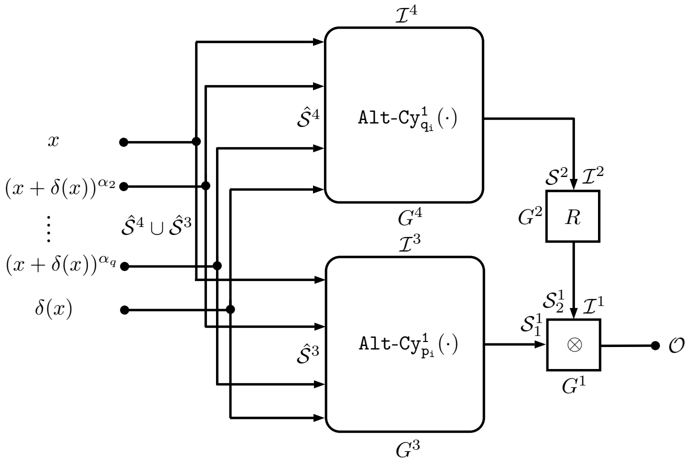

**Figure 13:** Gadget $\texttt{CRV}^1_\texttt{i}(\cdot)$.

*Proof.* Let $\Omega = (\mathcal{I}, \mathcal{O})$ be an observation set that we want to simulate and which is made on the whole circuit of Fig. 13, such that $\mathcal{I} = \bigcup_{1 \le i \le 4} \mathcal{I}^i$ and such that the global constraint $\sum_{i=1}^{4} |\mathcal{I}^i| + |\mathcal{O}| \le t$ is satisfied.

**Gadget 1.** Let $\Omega^1 = (\mathcal{I}^1, \mathcal{O})$. Since $G^1$ is $t$-`SNI` and $|\mathcal{I}^1 \cup \mathcal{O}| \le t$ (by global constraint), we know that there exists an observation set $\mathcal{S}^1 = (\mathcal{S}_1^1, \mathcal{S}_2^1)$ such that $|\mathcal{S}_1^1| \le |\mathcal{I}^1|$, $|\mathcal{S}_2^1| \le |\mathcal{I}^1|$ and $\mathcal{S}^1$ is sufficient to simulate $\Omega^1$.
**Gadget 2.** Let $\Omega^2 = (\mathcal{I}^2, \mathcal{S}_2^1)$. Since $G^2$ is $t$-`SNI` and $|\mathcal{I}^2 \cup \mathcal{S}_2^1| \le t$ (by simulation of Gadget 1 and the global constraint), we know that there exists an observation set $\mathcal{S}^2$ such that $|\mathcal{S}^2| \le |\mathcal{I}^2|$ and $\mathcal{S}^2$ is sufficient to simulate $\Omega^2$.
**Gadget 3.** Let $\Omega^3 = (\mathcal{I}^3, \mathcal{S}_1^1)$. Since $G^3$ is `affine-NI`, we know that there exists an observation set $\hat{\mathcal{S}}^3$ such that $|\hat{\mathcal{S}}^3| \le |\mathcal{I}^3 \cup \mathcal{S}_1^1| \le |\mathcal{I}^3| + |\mathcal{I}^1| \le t$ and the set of input shares $\hat{\mathcal{S}}^3$ is sufficient to simulate the adversary observation set $\Omega^3$ made on Gadget 3.
**Gadget 4.** Let $\Omega^4 = (\mathcal{I}^4, \mathcal{S}^2)$. Since $G^4$ is `affine-NI`, we know that there exists an observation set $\hat{\mathcal{S}}^4$ such that $|\hat{\mathcal{S}}^4| \le |\mathcal{I}^4 \cup \mathcal{S}^2| \le |\mathcal{I}^4| + |\mathcal{I}^2|$ and the set of input shares $\hat{\mathcal{S}}^4$ is sufficient to simulate the adversary observation set $\Omega^4$ made on Gadget 4.

In order to simulate Gadget $\texttt{CRV}^1_{\texttt{i}}(\cdot)$, the corresponding simulator requires the shares $\hat{\mathcal{S}}^4 \cup \hat{\mathcal{S}}^3$ of each of its inputs and $|\hat{\mathcal{S}}^4 \cup \hat{\mathcal{S}}^3| \le |\mathcal{I}^4| + |\mathcal{I}^3| + |\mathcal{I}^2| + |\mathcal{I}^1| \le \sum_{i=1}^{4} |\mathcal{I}^i| \le t$. Therefore, Gadget $\texttt{CRV}^1_{\texttt{i}}(\cdot)$ is $t$-`SNI`. □

## C.2 Proof of Lemma 4.

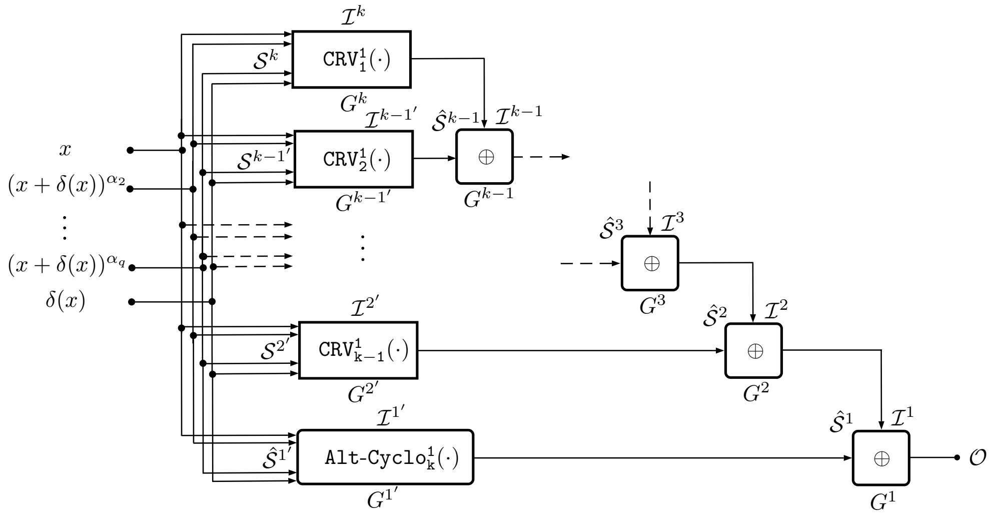

**Figure 14:** Gadget $\texttt{CRV}^2(\cdot)$.

*Proof.* Let $\Omega = (\mathcal{I}, \mathcal{O})$ be an observation set to simulate, made on the whole circuit of Fig. 14, such that $\mathcal{I} = \left(\bigcup_{1 \le i \le k} \mathcal{I}^i\right) \cup \left(\bigcup_{1 \le i \le k-1} \mathcal{I}^{i'}\right)$ and such that the global constraint $\left(\sum_{i=1}^{k} |\mathcal{I}^i| + \sum_{i=1}^{k-1} |\mathcal{I}^{i'}|\right) + |\mathcal{O}| \le t$ is satisfied.

**Gadget 1.** Let $\Omega^1 = (\mathcal{I}^1, \mathcal{O})$. Since $G^1$ is `affine-NI`, we know that there exists an observation set $\hat{\mathcal{S}}^1$ such that $|\hat{\mathcal{S}}^1| \le |\mathcal{I}^1 \cup \mathcal{O}| \le |\mathcal{I}^1| + |\mathcal{O}|$ and the set of input shares $\hat{\mathcal{S}}^1$ is sufficient to simulate $\Omega^1$.

**Gadget 1'.** Let $\Omega^{1'} = (\mathcal{I}^{1'}, \hat{\mathcal{S}}^1)$. Since $\texttt{Alt-Cy}^1_{\texttt{t}}(\cdot)$ is $\texttt{affine-NI}$, we know that there exists an observation set $\hat{\mathcal{S}}^{1'}$ such that $|\hat{\mathcal{S}}^{1'}| \leq |\mathcal{I}^{1'} \cup \hat{\mathcal{S}}^1| \leq |\mathcal{I}^{1'}| + |\mathcal{I}^1| + |\mathcal{O}|$ and the set of input shares $\hat{\mathcal{S}}^{1'}$ is sufficient to simulate $\Omega^{1'}$.
**Gadget i**, for every $i \in [2; k-1]$. Let $\Omega^i = (\mathcal{I}^i, \hat{\mathcal{S}}^{i-1})$. Since $\oplus$ is $\texttt{affine-NI}$, we know that there exists an observation set $\hat{\mathcal{S}}^i$ for $G^i$ such that $|\hat{\mathcal{S}}^i| \leq |\mathcal{I}^i \cup \hat{\mathcal{S}}^{i-1}| \leq |\bigcup_{1\leq j\leq i} \mathcal{I}^j \cup \mathcal{O}| \leq \sum_{j=1}^{i} |\mathcal{I}^j| + |\mathcal{O}|$. Moreover, the set of input shares $\hat{\mathcal{S}}^i$ is sufficient to simulate the adversary observation set $\Omega^i$ made on Gadget $i$.
**Gadget i'**, for every $i \in [2; k-1]$. Let $\Omega^{i'} = (\mathcal{I}^{i'}, \hat{\mathcal{S}}^i)$. Since $\texttt{CRV}^1_{\texttt{i}}(\cdot)$ is $t$-$\texttt{SNI}$ and $|\mathcal{I}^{i'} \cup \hat{\mathcal{S}}^i| \leq t$ (by simulation of Gadgets j for every $j \in [2, i]$ and the global constraint), we know that there exists an observation set $\mathcal{S}^{i'}$ for $G^{i'}$ such that $|\mathcal{S}^{i'}| \leq |\mathcal{I}^{i'}|$ and the set of input shares $\mathcal{S}^{i'}$ is sufficient to simulate the adversary observation set $\Omega^{i'}$ made on Gadget $i'$.
**Gadget k.** Let $\Omega^k = (\mathcal{I}^k, \hat{\mathcal{S}}^{k-1})$. Since $\texttt{CRV}^1_1(\cdot)$ is $t$-$\texttt{SNI}$ and $|\mathcal{I}^k \cup \hat{\mathcal{S}}^{k-1}| \leq \sum_{i=1}^{k} |\mathcal{I}^i| \cup |\mathcal{O}| \leq t$ (by simulation of Gadget i for $i \in [1, k-1]$ and the global constraint), we know that there exists an observation set $\mathcal{S}^k$ such that $|\mathcal{S}^k| \leq |\mathcal{I}^k|$ and $\mathcal{S}^k$ is sufficient to simulate $\Omega^k$.

In order to simulate Gadget $\texttt{CRV}^2(\cdot)$, the corresponding simulator requires the shares $\bigcup_{1\leq i\leq k-1} \mathcal{S}^{i'} \cup \mathcal{S}^k$ and $|\bigcup_{1\leq i\leq k-1} \mathcal{S}^{i'} \cup \mathcal{S}^k| \leq |\mathcal{I}^k| + \sum_{i=1}^{k-1} |\mathcal{I}^{i'}| + |\mathcal{I}^1| + |\mathcal{O}| \leq t$. Therefore Gadget $\texttt{CRV}^2(\cdot)$ is $\texttt{affine-NI}$. □