Dynamic Random Probing Expansion with Quasi Linear Asymptotic Complexity

Sonia Belaïd[1], Matthieu Rivain[1], Abdul Rahman Taleb[1,2], and Damien Vergnaud[2,3]

[1] CryptoExperts, France
[2] Sorbonne Université, CNRS, LIP6, F-75005 Paris, France
[3] Institut Universitaire de France, France

{sonia.belaid,matthieu.rivain,abdul.taleb}@cryptoexperts.com
[4] damien.vergnaud@lip6.fr

Abstract. The masking countermeasure is widely used to protect cryptographic implementations against side-channel attacks. While many masking schemes are shown to be secure in the widely deployed probing model, the latter raised a number of concerns regarding its relevance in practice. Offering the adversary the knowledge of a fixed number of intermediate variables, it does not capture the so-called horizontal attacks which exploit the repeated manipulation of sensitive variables. Therefore, recent works have focused on the *random probing model* in which each computed variable leaks with some given probability p. This model benefits from fitting better the reality of the embedded devices. In particular, Belaïd, Coron, Prouff, Rivain, and Taleb (CRYPTO 2020) introduced a framework to generate random probing circuits. Their compiler somehow extends base gadgets as soon as they satisfy a notion called *random probing expandability* (RPE). A subsequent work from Belaïd, Rivain, and Taleb (EUROCRYPT 2021) went a step forward with tighter properties and improved complexities. In particular, their construction reaches a complexity of $\mathcal{O}(\kappa^{3.9})$, for a κ-bit security, while tolerating a leakage probability of $p = 2^{-7.5}$.

In this paper, we generalize the random probing expansion approach by considering a dynamic choice of the base gadgets at each step in the expansion. This approach makes it possible to use gadgets with high number of shares –which enjoy better asymptotic complexity in the expansion framework– while still tolerating the best leakage rate usually obtained for small gadgets. We investigate strategies for the choice of the sequence of compilers and show that it can reduce the complexity of an AES implementation by a factor 10. We also significantly improve the asymptotic complexity of the expanding compiler by exhibiting new asymptotic gadget constructions. Specifically, we introduce RPE gadgets for linear operations featuring a quasi-linear complexity as well as an RPE multiplication gadget with linear number of multiplications. These new gadgets drop the complexity of the expanding compiler from quadratic to quasi-linear.

Keywords: Random probing model, masking, side-channel security, RPE

1 Introduction

Implementations of cryptographic algorithms may be vulnerable to the powerful *side-channel attacks.* The latter exploit the power consumption, the electromagnetic radiations or the temperature variations of the underlying device which may carry information on the manipulated data. Entire secrets can be recovered within a short time interval using cheap equipment.

Among the several approaches investigated by the community to counteract side-channel attacks, *masking* is one of the most deployed in practice. Simultaneously introduced by Chari, Jutla, Rao, and Rohatgi [12] and by Goubin and Patarin [16] in 1999, it consists in splitting the sensitive variables into n random shares, among which any combination of $n-1$ shares does not reveal any secret information. When the shares are combined by bitwise addition, the masking is said to be *Boolean.* In this setting, the linear operations can be very easily implemented by applying on each share individually. Nevertheless, non-linear operations require additional randomness to ensure that any set of less than n intermediate variables is still independent from the original secret.

To reason on the security of masked implementations, the community has introduced so-called *leakage models*. They aim to define the capabilities of the attacker to formally counteract the subsequent side-channel attacks. Among them, the *probing model* introduced in 2003 by Ishai, Sahai, and Wagner [18] is probably the most widely used. In a nutshell, it assumes that an adversary is able to get the exact values of up to a certain number of intermediate variables. The idea is to capture the difficulty of learning information from the combination of noisy variables. Despite its wide use by the community [21, 20, 13, 7, 14], the probing model raised a number of concerns regarding its relevance in practice [5, 17]. It actually fails to capture the huge amount of information resulting from the leakage of all manipulated data. As an example, it typically ignores the repeated manipulation of identical values which would average the noise and remove uncertainty on secret variables (see horizontal attacks [5]). Another model, the *noisy leakage model* introduced by Prouff and Rivain and inspired from [12], offers an opposite trade-off. Although it captures well the reality of embedded devices by assuming that all the data leaks with some noise, it is not convenient to build security proofs. To get the best from both worlds, Duc, Dziembowski, and Faust proved in 2014 that a scheme secure in the probing model is also secure in the noisy leakage model [15]. Nevertheless, the reduction is not very tight in the standard probing model (considering a constant number of probes) since the security level decreases as the size of the circuit increases (*i.e.* a secure circuit C in the probing model is also secure in the noisy model but loses at least a factor $|C|$, where $|C|$ is the number of operations in the circuit).

The reduction from [15] relies on an intermediate leakage model, referred to as *random probing model*. The latter benefits from a tight reduction with the noisy leakage model which becomes independent of the size of the circuit. In a nutshell, it assumes that every wire in the circuit leaks with some constant leakage probability. This leakage probability is somehow related to the amount of side-channel noise in practice. A masked circuit is secure in the random probing model whenever its random probing leakage can be simulated without knowledge of the underlying secret data with a negligible simulation failure. In addition to the attacks already captured by the probing model, the random probing model further encompasses the powerful *horizontal attacks* which exploit the repeated manipulations of variables in an implementation.

To the best of our knowledge, five constructions tolerate a constant leakage probability so far [1, 4, 3, 9, 10]. The two former ones [1, 4] use expander graphs and do not make their tolerated probability explicit. In the third construction [3], Ananth, Ishai, and Sahai develop an expansion strategy on top of multi-party computation protocols. According to the authors of [9], their construction tolerates a leakage probability of around 2^{-26} for a complexity of $\mathcal{O}(\kappa^{8.2})$ with respect to the security parameter κ. Finally, the two more recent constructions [9, 10] follow an expansion strategy on top of masking gadgets achieving the so-called *random probing expandability* (RPE) notion. In a nutshell, every gate in the original circuit is replaced by a corresponding gadget for some chosen number of shares. The operation is repeated until the desired security level is achieved. The improved gadgets of [10] make it possible to tolerate of leakage probability of $2^{-7.5}$ for a complexity of $\mathcal{O}(\kappa^{3.9})$.

Our contributions. In this paper, we push the random probing expansion strategy one step further by analyzing a dynamic choice of the base gadgets. While the expanding compiler considered in [9, 10] consists in applying a compiler CC composed of base RPE gadgets a given number of times, say k, to the input circuit: $\widehat{C} = \mathsf{CC}^{(k)}(C)$, we consider a dynamic approach in which a new compiler is selected at each step of the expansion from a family of base compilers $\{\mathsf{CC}_i\}_i$. This approach is motivated by the generic gadget constructions introduced in [10] which achieve the RPE property for any number of shares n. While the asymptotic complexity of the expanding compiler decreases with n, the tolerated leakage probability p also gets smaller with n, which makes those constructions only practical for small values of n. We show that using our dynamic approach we can get the best of both worlds: our dynamic expanding compiler enjoys the best tolerated probability as well as the best asymptotic complexity from the underlying family of RPE compilers $\{\mathsf{CC}_i\}_i$. We further illustrate how this approach can reduce the complexity of a random probing secure AES implementation by a factor 10 using a dynamic choice of the gadgets from [10].

This first contribution further motivates the design of asymptotic RPE gadgets achieving better complexity. While the asymptotic constructions introduced in [10] achieve a quadratic complexity, we introduce new constructions achieving quasi-linear complexity. We obtain this result by showing that the quasi-linear refresh gadget from Battistello, Coron, Prouff, and Zeitoun [6] achieves a *strong random probing expandability* (SRPE) which makes it a good building block for linear RPE gadgets (addition, copy, multiplication by constant). We thus solve a first issue left open in [10]. With such linear gadgets, the complexity bottleneck of the expanding compiler becomes the number of multiplications in the multiplication gadget, which is quadratic in known RPE constructions. We then provide a new generic construction of RPE multiplication gadget featuring a linear number of multiplications. We obtain this construction by tweaking the probing-secure multiplication gadget from Belaïd, Benhamouda, Passelègue, Prouff, Thillard, and Vergnaud [8]. As in the original construction, our RPE gadget imposes some constraint on the underlying finite field. We demonstrate that for any number of shares there exist a (possibly large) finite field on which our construction can be instantiated and we provide some concrete instantiations for some (small) number of shares.

Using our new asymptotic gadget constructions with the dynamic expansion approach we obtain random probing security for a leakage probability of $2^{-7.5}$ with asymptotic complexity of $\mathcal{O}(\kappa^2)$. Moreover, assuming that the constraint on the finite field from our multiplication gadget is satisfied, we can make this asymptotic complexity arbitrary close to $\mathcal{O}(\kappa)$ which is optimal. In practice, this means that securing circuits defined on large field against random probing leakage can be achieved at a sub-quadratic nearly-linear complexity.

2 Preliminaries

Along the paper, we shall use similar notations and formalism as [9]. In particular, $\mathbb{K}$ shall denote a finite field. For any $n \in \mathbb{N}$, we shall denote $[n]$ the integer set $[n] = [1, n] \cap \mathbb{Z}$. For any tuple $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{K}^n$ and any set $I \subseteq [n]$, we shall denote $\boldsymbol{x}|_I = (x_i)_{i \in I}$. Any two probability distributions D_1 and D_2 are said *ε-close*, denoted $D_1 \approx_\varepsilon D_2$, if their statistical distance is upper bounded by ε, that is

$$\mathsf{SD}(D_1; D_2) := \frac{1}{2} \sum_x |p_{D_1}(x) - p_{D_2}(x)| \leq \varepsilon \ ,$$

where $p_{D_1}(\cdot)$ and $p_{D_1}(\cdot)$ denote the probability mass functions of D_1 and D_2.

2.1 Linear Sharing, Circuits, and Gadgets

In the following, the *n-linear decoding* mapping, denoted LinDec, refers to the function $\mathbb{K}^n \to \mathbb{K}$ defined as

$$\mathsf{LinDec} : (x_1, \ldots, x_n) \mapsto x_1 + \cdots + x_n \ ,$$

for every $n \in \mathbb{N}$ and $(x_1, \ldots, x_n) \in \mathbb{K}^n$. We shall further consider that, for every $n, \ell \in \mathbb{N}$, on input $(\widehat{x}_1, \ldots, \widehat{x}_\ell) \in (\mathbb{K}^n)^\ell$ the n-linear decoding mapping acts as

$$\mathsf{LinDec} : (\widehat{x}_1, \ldots, \widehat{x}_\ell) \mapsto (\mathsf{LinDec}(\widehat{x}_1), \ldots, \mathsf{LinDec}(\widehat{x}_\ell)) \ .$$

Definition 1 (Linear Sharing). *Let $n, \ell \in \mathbb{N}$. For any $x \in \mathbb{K}$, an n-*linear sharing *of x is a random vector $\widehat{x} \in \mathbb{K}^n$ such that $\mathsf{LinDec}(\widehat{x}) = x$. It is said to be* uniform *if for any set $I \subseteq [n]$ with $|I| < n$ the tuple $\widehat{x}|_I$ is uniformly distributed over $\mathbb{K}^{|I|}$. A n-*linear encoding *is a probabilistic algorithm LinEnc which on input a tuple $\boldsymbol{x} = (x_1, \ldots, x_\ell) \in \mathbb{K}^\ell$ outputs a tuple $\widehat{\boldsymbol{x}} = (\widehat{x}_1, \ldots, \widehat{x}_\ell) \in (\mathbb{K}^n)^\ell$ such that $\widehat{x}_i$ is a uniform n-sharing of x_i for every $i \in [\ell]$.*

An *arithmetic circuit* on a field $\mathbb{K}$ is a labeled directed acyclic graph whose edges are *wires* and vertices are *arithmetic gates* processing operations on $\mathbb{K}$. We consider circuits composed of gates from some base $\mathbb{B} = \{g : \mathbb{K}^\ell \to \mathbb{K}^m\}$, *e.g.*, addition gates, $(x_1, x_2) \mapsto x_1 + x_2$, multiplication gates, $(x_1, x_2) \mapsto x_1 \cdot x_2$, and copy gates, $x \mapsto (x, x)$. A *randomized arithmetic circuit* is equipped with an additional random gate which outputs a fresh uniform random value of $\mathbb{K}$.

In the following, we shall call an *(n-share, ℓ-to-m) gadget*, a randomized arithmetic circuit that maps an input $\widehat{\boldsymbol{x}} \in (\mathbb{K}^n)^\ell$ to an output $\widehat{\boldsymbol{y}} \in (\mathbb{K}^n)^m$ such that $\boldsymbol{x} = \mathsf{LinDec}(\widehat{\boldsymbol{x}}) \in \mathbb{K}^\ell$ and $\boldsymbol{y} = \mathsf{LinDec}(\widehat{\boldsymbol{y}}) \in \mathbb{K}^m$ satisfy $\boldsymbol{y} = g(\boldsymbol{x})$ for some function g.

Definition 2 (Circuit Compiler). *A* circuit compiler *is a triplet of algorithms $(\mathsf{CC}, \mathsf{Enc}, \mathsf{Dec})$ defined as follows:*

- CC *(circuit compilation) is a deterministic algorithm that takes as input an arithmetic circuit C and outputs a randomized arithmetic circuit $\widehat{C}$,*
- Enc *(input encoding) is a probabilistic algorithm that maps an input $\boldsymbol{x} \in \mathbb{K}^\ell$ to an encoded input $\widehat{\boldsymbol{x}} \in \mathbb{K}^{\ell'}$,*
- Dec *(output decoding) is a deterministic algorithm that maps an encoded output $\widehat{\boldsymbol{y}} \in \mathbb{K}^{m'}$ to a plain output $\boldsymbol{y} \in \mathbb{K}^m$,*

which satisfy the following properties:

- ***Correctness:*** *For every arithmetic circuit C of input length ℓ, and for every $\boldsymbol{x} \in \mathbb{K}^\ell$, we have*

$$\Pr\left(\mathsf{Dec}\big(\widehat{C}(\widehat{\boldsymbol{x}})\big) = C(\boldsymbol{x}) \;\middle|\; \widehat{\boldsymbol{x}} \leftarrow \mathsf{Enc}(\boldsymbol{x})\right) = 1 \ , \ \textit{where } \widehat{C} = \mathsf{CC}(C).$$

- ***Efficiency:*** *For some security parameter $\kappa \in \mathbb{N}$, the running time of $\mathsf{CC}(C)$ is $\mathrm{poly}(\kappa, |C|)$, the running time of $\mathsf{Enc}(\boldsymbol{x})$ is $\mathrm{poly}(\kappa, |\boldsymbol{x}|)$ and the running time of $\mathsf{Dec}\big(\widehat{\boldsymbol{y}}\big)$ is $\mathrm{poly}(\kappa, |\widehat{\boldsymbol{y}}|)$, where $\mathrm{poly}(\kappa, \ell) = \mathcal{O}(\kappa^{e_1} \ell^{e_2})$ for some constants e_1, e_2.*

2.2 Random Probing Security

Let $p \in [0, 1]$ be some constant leakage probability parameter, a.k.a. the *leakage rate*. In the p-random probing model, an evaluation of a circuit C leaks the value carried by each wire with a probability p, all the wire leakage events being mutually independent.

As in [9], we formally define the random-probing leakage of a circuit from the two following probabilistic algorithms:

– The *leaking-wires sampler* takes as input a randomized arithmetic circuit C and a probability $p \in [0,1]$, and outputs a set W, denoted as

$$W \leftarrow \mathsf{LeakingWires}(C, p) \ ,$$

where W is constructed by including each wire label from the circuit C with probability p to W (where all the probabilities are mutually independent).
– The *assign-wires sampler* takes as input a randomized arithmetic circuit C, a set of wire labels W (subset of the wire labels of C), and an input $\boldsymbol{x}$, and it outputs a $|W|$-tuple $\boldsymbol{w} \in \mathbb{K}^{|W|}$, denoted as

$$\boldsymbol{w} \leftarrow \mathsf{AssignWires}(C, W, \boldsymbol{x}) \ ,$$

where $\boldsymbol{w}$ corresponds to the assignments of the wires of C with label in W for an evaluation on input $\boldsymbol{x}$.

Definition 3 (Random Probing Leakage). *The* p-random probing leakage *of a randomized arithmetic circuit C on input $\boldsymbol{x}$ is the distribution $\mathcal{L}_p(C, \boldsymbol{x})$ obtained by composing the leaking-wires and assign-wires samplers as*

$$\mathcal{L}_p(C, \boldsymbol{x}) \stackrel{id}{=} \mathsf{AssignWires}(C, \mathsf{LeakingWires}(C, p), \boldsymbol{x}) \ .$$

Definition 4 (Random Probing Security). *A randomized arithmetic circuit C with $\ell \cdot n \in \mathbb{N}$ input gates is* (p, ε)-random probing secure *with respect to encoding* Enc *if there exists a simulator* Sim *such that for every $\boldsymbol{x} \in \mathbb{K}^\ell$:*

$$\mathsf{Sim}(C) \approx_\varepsilon \mathcal{L}_p(C, \mathsf{Enc}(\boldsymbol{x})) \ . \quad (1)$$

2.3 Random Probing Expansion

In [3], Ananth, Ishai and Sahai proposed an *expansion* approach to build a random-probing-secure circuit compiler from a secure multi-party protocol. This approach was later revisited by Belaïd, Coron, Prouff, Rivain, and Taleb who formalize the notion of *expanding compiler* [9].

The principle of the expanding compiler is to recursively apply a base compiler, denoted CC and which simply consists in replacing each gate of $\mathbb{B}$ in the input circuit by the corresponding gadget. Assume we have n-share gadgets G_g for each gate g in $\mathbb{B}$. The base compiler CC simply consists in replacing each gate g in these gadgets by G_g and by replacing each wire by n wires carrying a sharing of the value. We thus obtain n^2-share gadgets by simply applying CC to each gadget: $G_g^{(2)} = \mathsf{CC}(G_g)$. This process can be iterated an arbitrary number of times, say k, to an input circuit C:

$$C \xrightarrow{\mathsf{CC}} \widehat{C}_1 \xrightarrow{\mathsf{CC}} \cdots \xrightarrow{\mathsf{CC}} \widehat{C}_k \ .$$

The first output circuit $\widehat{C}_1$ is the original circuit in which each gate g is replaced by a base gadget G_g. The second output circuit $\widehat{C}_2$ is the original circuit C in which each gate is replaced by an n^2-share gadget $G_g^{(2)}$. Equivalently, $\widehat{C}_2$ is the circuit $\widehat{C}_1$ in which each gate is replaced by a base gadget. In the end, the output circuit $\widehat{C}_k$ is hence the original circuit C in which each gate has been replaced by a k-expanded gadget and each wire has been replaced by n^k wires carrying an (n^k)-linear sharing of the original wire.

The expanding compiler achieves random probing security if the base gadgets verify a property called *random probing expandability* [9]. We recall hereafter the original definition of the random probing expandability (RPE) property for 2-to-1 gadgets.

Definition 5 (Random Probing Expandability [9]). *Let $f : \mathbb{R} \to \mathbb{R}$. An n-share 2-to-1 gadget $G : \mathbb{K}^n \times \mathbb{K}^n \to \mathbb{K}^n$ is (t, f)-*random probing expandable (RPE) *if there exists a deterministic algorithm Sim_1^G and a probabilistic algorithm Sim_2^G such that for every input $(\widehat{x}, \widehat{y}) \in \mathbb{K}^n \times \mathbb{K}^n$, for every set $J \subseteq [n]$ and for every $p \in [0, 1]$, the random experiment*

$$\begin{aligned} &W \leftarrow \mathsf{LeakingWires}(G, p) \\ &(I_1, I_2, J') \leftarrow \mathsf{Sim}_1^G(W, J) \\ &out \leftarrow \mathsf{Sim}_2^G(W, J', \widehat{x}|_{I_1}, \widehat{y}|_{I_2}) \end{aligned}$$

ensures that

1. *the failure events $\mathcal{F}_1 \equiv \big(|I_1| > t\big)$ and $\mathcal{F}_2 \equiv \big(|I_2| > t\big)$ verify*
$$\Pr(\mathcal{F}_1) = \Pr(\mathcal{F}_2) = \varepsilon \quad \textit{and} \quad \Pr(\mathcal{F}_1 \wedge \mathcal{F}_2) = \varepsilon^2 \tag{2}$$
with $\varepsilon = f(p)$ (in particular $\mathcal{F}_1$ and $\mathcal{F}_2$ are mutually independent),
2. *J' is such that $J' = J$ if $|J| \leq t$ and $J' \subseteq [n]$ with $|J'| = n - 1$ otherwise,*
3. *the output distribution satisfies*
$$out \stackrel{id}{=} \big(\mathsf{AssignWires}(G, W, (\widehat{x}, \widehat{y}))\ ,\ \widehat{z}|_{J'}\big) \tag{3}$$

where $\widehat{z} = G(\widehat{x}, \widehat{y})$.

The RPE notion can be simply extended to gadgets with 2 outputs: the Sim_1^G simulator takes two sets $J_1 \subseteq [n]$ and $J_2 \subseteq [n]$ as input and produces two sets J_1' and J_2' satisfying the same property as J' in the above definition (w.r.t. J_1 and J_2). The Sim_2^G simulator must then produce an output including $\widehat{z}_1|_{J_1'}$ and $\widehat{z}_2|_{J_1'}$ where $\widehat{z}_1$ and $\widehat{z}_2$ are the output sharings. The RPE notion can also be simply extended to gadgets with a single input: the Sim_1^G simulator produces a single set I so that the failure event $(|I| > t)$ occurs with probability ε (and the Sim_2^G simulator is then simply given $\widehat{x}|_I$ where $\widehat{x}$ is the single input sharing). We refer the reader to [9] for the formal definitions of these variants.

Although the requirement of mutual independence for the failure events might seem strong, it can be relaxed which leads to the notion of *weak random probing expandability*. It is shown in [9] that this weaker notion actually implies the RPE notion for some ε which is derivable from the (joint) probability of the failure events.

The authors of [10] eventually introduced a tighter version the RPE security property, namely the tight random probing expandability (TRPE). In this setting, the failure events are re-define as $\mathcal{F}_j \equiv \big(|I_j| > \min(t, W)\big)$. Both RPE and TRPE notions can be split into two sub-notions (that are jointly equivalent to the original one) corresponding to the two possible properties of J' in Definition 5. Specifically, in (T)RPE1, the set J is constrained to satisfy $|J| \leq t$ and $J' = J$, while in (T)RPE2, J' is chosen by the simulator such that $J' \subseteq [n]$ and $|J'| = n - 1$.

2.4 Complexity of the Expanding Compiler

Consider circuits with base of gates $\mathbb{B} = \{g_1, \ldots, g_\beta\}$ for which we have n-share RPE gadgets $\{G_g\}_{g \in \mathbb{B}}$. Further denote G_{random} the n-share random gadget which generates n independent random values as a random n-sharing as well as CC the circuit compiler based from those gadgets. To each gadget a complexity vector is associated $N_G = (N_{g_1}, \dots, N_{g_\beta}, N_r)^\mathsf{T}$ where N_{g_i} stands for the number of gates g_i and N_r for the number of random gates in the gadget G. Then the *compiler complexity matrix* M_{CC} is the $(\beta+1)\times(\beta+1)$ matrix defined as

$$M_{\mathsf{CC}} = \big(N_{g_1} \mid \cdots \mid N_{g_\beta} \mid N_{G_{\text{random}}}\big) \quad \text{with} \quad N_{G_{\text{random}}} = (0, \dots, 0, n)^\mathsf{T} \ .$$

Given a circuit C with complexity vector N_C (which is defined as the gate-count vector as for gadgets), compiling it with the base gadgets gives a circuit $\widehat{C}$ of complexity vector $N_{\widehat{C}} = M_{\mathsf{CC}} \cdot N_C$. It follows that the kth power of the matrix M gives the gate counts for the level-k gadgets as:

$$M_{\mathsf{CC}}^k = \underbrace{M_{\mathsf{CC}} \cdots M_{\mathsf{CC}}}_{k \text{ times}} = \big(N_{g_1}^{(k)} \mid \cdots \mid N_{g_\beta}^{(k)} \mid N_{G_{\text{random}}}^{(k)}\big) \quad \text{with} \quad N_{G_{\text{random}}}^{(k)} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ n^k \end{pmatrix}$$

where $N_{g_i}^{(k)}$ are the gate-count vectors for the level-k gadgets $G_{g_i}^{(k)}$. Let us denote the eigen decomposition of M_{CC} as $M_{\mathsf{CC}} = Q \cdot \Lambda \cdot Q^{-1}$, we get

$$M_{\mathsf{CC}}^k = Q \cdot \Lambda^k \cdot Q^{-1} \quad \text{with} \quad \Lambda^k = \begin{pmatrix} \lambda_1^k & & \\ & \ddots & \\ & & \lambda_{\beta+1}^k \end{pmatrix}$$

where λ_i are the eigenvalues of M_{CC}. We then obtain an asymptotic complexity of

$$|\widehat{C}| = \mathcal{O}\big(|C| \cdot \sum_{i=1}^{\beta+1} |\lambda_i|^k\big) = \mathcal{O}\big(|C| \cdot \max(|\lambda_1|, \dots, |\lambda_{\beta+1}|)^k\big)$$

for a compiled circuit $\widehat{C} = CC^{(k)}(C)$.

The complexity of the expanding compiler can be further expressed in terms of the target random probing security level κ. This complexity is related to the notion of *amplification order* that we recall hereafter.

Definition 6 (Amplification Order).

– *Let* $f : \mathbb{R} \to \mathbb{R}$ *which satisfies*
$$f(p) = c_d\, p^d + \mathcal{O}(p^{d+\varepsilon})$$
as p *tends to* 0*, for some* $c_d > 0$ *and* $\varepsilon > 0$*. Then* d *is called the amplification order of* f*.*

– *Let* $t > 0$ *and* G *a gadget. Let* d *be the maximal integer such that* G *achieves* (t, f)*-RPE for* $f : \mathbb{R} \to \mathbb{R}$ *of amplification order* d*. Then* d *is called the amplification order of* G *(with respect to* t*). We will sometimes denote* f_G *as the function* f *corresponding to the gadget* G *for which* G *achieves* (t, f_G)*-RPE.*

We stress that the amplification order of a gadget G is defined with respect to the RPE threshold t. Namely, different RPE thresholds t are likely to yield different amplification orders d for G (or equivalently d can be thought of as a function of t).

As shown in [9], the complexity of the expanding compiler relates to the (minimum) amplification order of the gadgets composing the base compiler CC. If the latter achieve (t, f)-RPE with an amplification order d, the expanding compiler achieves $(p, 2^{-\kappa})$-random probing security with an expansion level k such that $f^{(k)}(p) \leq 2^{-\kappa}$, which yields a complexity blowup of

$$|\widehat{C}| = \mathcal{O}\big(|C| \cdot \kappa^e\big) \quad \text{with} \quad e = \frac{\log N_{\max}}{\log d} \tag{4}$$

where

$$N_{\max} = \max \ |\mathsf{eigenvalues}(M_{\mathsf{CC}})| \ , \tag{5}$$

where $\mathsf{eigenvalues}(\cdot)$ returns the tuple of eigenvalues (or modules of eigenvalues in case of complex numbers) of the input matrix.

Let us slightly explicit the complexity with the 3-gate base $\mathbb{B} = \{\text{add, mult, copy}\}$ as used in [9, 10]. Considering that multiplication gates are solely used in the multiplication gadget ($N_{G_{\text{add}},m} = N_{G_{\text{copy}},m} = 0$) which is the case in the constructions of [9, 10], it can be checked that (up to some permutation) the eigenvalues satisfy

$$(\lambda_1, \lambda_2) = \mathsf{eigenvalues}(M_{ac}) \ , \quad \lambda_3 = N_{G_{\text{mult}},m} \quad \text{and} \quad \lambda_4 = n$$

where M_{ac} is the top left 2×2 block matrix of M_{CC}

$$M_{ac} = \begin{pmatrix} N_{G_{\text{add}},a} & N_{G_{\text{copy}},a} \\ N_{G_{\text{add}},c} & N_{G_{\text{copy}},c} \end{pmatrix}$$

where $N_{x,y}$ denotes the number of gates x in a gadget y, with m for the multiplication, a for the addition, and c for the copy. We finally get

$$|\widehat{C}| = \mathcal{O}\big(|C| \cdot N_{\max}^k\big) \quad \text{with} \quad N_{\max} = \max(|\mathsf{eigenvalues}(M_{ac})|, N_{G_{\text{mult}},m}, n) \ . \tag{6}$$

As an illustration, the expanding compiler from [10] satisfies $N_{\max} = 3n^2 - 2n$ and $d = \frac{min(t+1,n-t)}{2}$ which yields an asymptotic complexity of $\mathcal{O}(\kappa^e)$ with

$$e = \frac{\log(3n^2 - 2n)}{\log(\lfloor (n+1)/4 \rfloor)}$$

which tends to 2 as n grows. In comparison, in this work, we shall achieve a quasi-linear complexity, *i.e.*, $N_{\max} = \mathcal{O}(n \log n)$.

2.5 Tolerated Leakage Rate

Finally, we recall the notion of *tolerated leakage rate* which corresponds to the maximum value p for which we have $f(p) < p$. This happens to be a necessary and sufficient condition for the expansion strategy to apply with (t, f)-RPE gadgets.

In practice, the tolerated leakage rate should be measured on concrete devices and fixed accordingly. Hence the motivation to exhibit gadgets which tolerate a high probability to cover any setting. So far, the asymptotic constructions provide a trade-off between tolerated leakage rate and complexity. However, we only know how to compute the former for small numbers of shares and the bounds for larger values are not tight.

As an illustration, the instantiation proposed in [9] tolerates a leakage probability up to $2^{-7.80}$, while the instantiation of [**?**] tolerates $2^{-7.50}$, both for 3-share base gadgets.

3 Dynamic Random Probing Expansion

As recalled in Section 2, the principle of the expanding compiler is to apply a base circuit compiler CC which is composed of base gadgets –one per gate type in the circuit– several times, say k, to the input circuit: $\widehat{C} = \mathsf{CC}^{(k)}(C)$. The level of expansion k is chosen in order to achieve a certain desired security level κ such that $f^{(k)}(p) \leq 2^{-\kappa}$.

In this section, we generalize this approach to choose the circuit compiler dynamically at the different steps of the expansion. Let $\{\mathsf{CC}_i\}_i$ be a family of circuit compilers, the *dynamic expanding compiler* for this family with respect to the expansion sequence $k_1, \ldots k_\mu$, is defined as

$$\widehat{C} = \mathsf{CC}_\mu^{k_\mu} \circ \mathsf{CC}_{\mu-1}^{k_{\mu-1}} \circ \ldots \cdots \circ \mathsf{CC}_1^{k_1}(C) \ . \tag{7}$$

The idea behind this generalization is to make the most from a family of RPE compilers $\{\mathsf{CC}_i\}_i$ which is defined with respect to the number of shares n_i in the base gadgets. If we assume that each compiler CC_i with n_i shares achieves the maximum amplification order $d_i = \frac{n_i+1}{2}$, then the benefit of using a compiler with higher number of shares is to increase the amplification order and thus reduce the number of steps necessary to achieve the desired security level κ. On the other hand, the tolerated leakage rate of existing constructions decreases with n_i. As we show hereafter, a dynamic increase of n_i can ensure both, the tolerated leakage rate of a small n_i and the better complexity of a high n_i.

3.1 Dynamic Expanding Compiler

We formally introduce the dynamic expanding compiler hereafter.

Definition 7 (RPE Compiler). *Let $\mathbb{B} = \{g : \mathbb{K}^\ell \to \mathbb{K}^m\}$ be an arithmetic circuit basis. Let $n, t \in \mathbb{N}$, and let $\{G_g\}_{g\in\mathbb{B}}$ be a family of (t, f_{G_g})-RPE n-share gadgets for the gate functionalities in $\mathbb{B}$. The* RPE compiler CC *associated to $\{G_g\}_{g\in\mathbb{B}}$ is the circuit compiler which consists in replacing each gate from a circuit over $\mathbb{B}$ by the corresponding gadget G_g. Moreover,*

- *the* expanding function *of CC is the function f defined as*

$$f : p \mapsto \max_g f_{G_g}(p)$$

- *the* amplification order *of CC is the integer d defined as*

$$d = \min_g d_{G_g}$$

  where d_{G_g} is the amplification order of f_{G_g},
- *the* gadget complexity *of CC is the integer s defined as*

$$s = \max_g |G_g|$$

  where $|G_g|$ denotes the number of wires in the gadget G_g,
- *the* tolerated leakage rate *of CC is the real number $q \in [0, 1)$ such that $f(p) < p$ for every $p < q$.*

In the following, we state the security and asymptotic complexity of the dynamic expanding compiler. We will consider a family of different RPE compilers where each compiler is indexed by an index i, *i.e.* a family of different RPE compilers is denoted as $\{\mathsf{CC}_i\}_i$ for different number of shares $\{n_i\}_i$. We start with a formal definition of the dynamic compiler:

Definition 8 (Dynamic Expanding Compiler). *Let $\{\mathsf{CC}_i\}_i$ be a family of RPE compilers with numbers of shares $\{n_i\}_i$. The* dynamic expanding compiler *for $\{\mathsf{CC}_i\}_i$ with* expansion levels k_1, *..., k_μ, is the circuit compiler $(\mathsf{CC}, \mathsf{Enc}, \mathsf{Dec})$ where*

1. *The input encoding Enc is a $\left(\prod_{i=1}^{\mu} n_i^{k_i}\right)$-linear encoding.*
2. *The output decoding Dec is the $\left(\prod_{i=1}^{\mu} n_i^{k_i}\right)$-linear decoding mapping.*
3. *The circuit compilation is defined as*

$$\mathsf{CC}(\cdot) = \mathsf{CC}_\mu^{k_\mu} \circ \mathsf{CC}_{\mu-1}^{k_{\mu-1}} \circ \ldots \cdots \circ \mathsf{CC}_1^{k_1}(\cdot) \ .$$

The following theorem states the random probing security of the dynamic expanding compiler. The proof of the theorem is very similar to the proof of RPE security (Theorem 2) from [9]. The main difference is that at each level of the expansion, we can use a different expanding compiler with different sharing orders. Besides that, the proof follows the same baselines as in [9]. The proof is provided in Appendix A.1.

Theorem 1 (Security). *Let $\{\mathsf{CC}_i\}_i$ be a family of RPE compilers with expanding functions $\{f_i\}_i$. The dynamic expanding compiler for $\{\mathsf{CC}_i\}_i$ with expansion levels k_1, ..., k_μ is (p, ε)-random probing secure with*

$$\varepsilon = f_\mu^{k_\mu} \circ \cdots \circ f_1^{k_1}(p) \ .$$

We now state the asymptotic complexity of the dynamic expanding compiler in the next theorem. The proof is given in Appendix A.2.

Theorem 2 (Asymptotic Complexity). *Let $\{\mathsf{CC}_i\}_i$ be a family of circuit compilers with complexity matrices $\{M_{\mathsf{CC}_i}\}_i$. For any input circuit C, the output circuit $\widehat{C} = \mathsf{CC}_\mu^{k_\mu} \circ \cdots\cdots \circ \mathsf{CC}_1^{k_1}(C)$ is of size*

$$|\widehat{C}| = |C| \cdot \mathcal{O}\Big(\prod_{i=1}^{\mu} |\lambda_i|^{k_i}\Big) \quad \text{with} \quad \lambda_i \quad \text{such that} \quad |\lambda_i| := \max \ |\mathsf{eigenvalues}(M_{\mathsf{CC}_i})| \ . \tag{8}$$

In the following, we shall call λ_i as defined above, the *eigen-complexity* of the compiler CC_i. We shall further call the product $\prod_{i=1}^{\mu} |\lambda_i|^{k_i}$ the *complexity blowup* of the dynamic expanding compiler. We note that minimizing the complexity blowup is equivalent to minimizing the log complexity blowup, which is

$$\sum_{i=1}^{\mu} k_i \cdot \log_2(|\lambda_i|) \ . \tag{9}$$

3.2 General Bounds for Asymptotic Constructions

The following theorem introduces general bounds on the tolerated leakage rate and the expanding function of an RPE compiler with respect to its amplification order and gadget complexity. The proof of the theorem is given in the supplementary material (Appendix A.3).

Theorem 3. *Let CC_i be an RPE circuit compiler of amplification order d_i and gadget complexity s_i. The tolerated leakage rate q_i of CC_i is lower bounded by*

$$q_i \geq \bar{q}_i := \frac{1}{\mathrm{e}} \left(\frac{1}{2\,\mathrm{e}}\right)^{\frac{1}{d_i-1}} \left(\frac{d_i}{s_i}\right)^{1+\frac{1}{d_i-1}} \tag{10}$$

For any $p < \bar{q}_i$, the expanding function f_i of CC_i is upper bounded by

$$f_i(p) \leq 2\binom{s_i}{d_i} p^{d_i} \leq 2\left(\frac{\mathrm{e}\cdot s_i}{d_i}\right)^{d_i} p^{d_i} \ . \tag{11}$$

The lower bound $\bar{q}_i$ on the tolerated leakage rate quickly converges to the ratio $\mathrm{e}^{-1}\cdot d_i/s_i$ as d_i grows. In other words, an RPE compiler family $\{\mathsf{CC}_i\}_i$ indexed by the number of shares n_i of its base gadgets tolerates a leakage probability which is linear in the ratio between its amplification order d_i and its complexity s_i. For known families of RPE compilers from [10] this ratio is in $\mathcal{O}(1/n_i)$.

From Theorem 3, we obtain the following bound for the composition $f_i^{(k)}$. The proof of the corollary is given in the supplementary material (Appendix A.4).

Corollary 1. *Let CC_i be an RPE compiler of expanding function f_i, amplification order d_i and gadget complexity s_i. For any $p < \bar{q}_i$ as defined in* (10)*, we have*

$$f_i^{(k)}(p) \ \leq \ \left[2\binom{s_i}{d_i}\right]^{\left(1+\frac{1}{d_i-1}\right)d_i^{k-1}} p^{d_i^k} \ \leq \ \left[\left(\frac{2^{\frac{1}{d_i}}\mathrm{e}s_i}{d_i}\right)^{\left(1+\frac{1}{d_i-1}\right)} p\right]^{d_i^k} \ .$$

The following lemma gives an explicit lower bound on the expansion level $\{k_i\}_i$ to reach some arbitrary target probability $p_{out} = 2^{-\kappa_{out}}$ from a given input probability $p_{in} = 2^{-\kappa_{in}}$ by applying $\mathsf{CC}_i^{(k_i)}$.

Lemma 1. *Let $p_{in} = 2^{-\kappa_{in}} < q_i$ and $p_{out} = 2^{-\kappa_{out}} \in (0, 1]$. For any integer k_i satisfying*

$$k_i \geq \log_{d_i}(\kappa_{out}) - \log_{d_i}(\kappa_{in} - \Delta_i)$$

with

$$\Delta_i := \left(1 + \frac{1}{d_i - 1}\right)\left(\frac{1}{d_i} + \log_2\left(\frac{\mathrm{e}s_i}{d_i}\right)\right)$$

we have

$$f_i^{(k_i)}(p_{in}) \leq p_{out} = 2^{-\kappa_{out}} \ .$$

In the above lemma, Δ_i represents a lower bound for κ_{in} which matches the upper bound $\bar{q}_i$ of $p_{in} = 2^{-\kappa_{in}}$. Assuming that s_i and d_i are both monotonically increasing with i, we get that the threshold Δ_i tends towards $\log_2\left(\frac{\mathrm{e}s_i}{d_i}\right)$.

From Lemma 1, we further get that the cost induced by the choice of the compiler CC_i to go from an input probability p_{in} to a target output probability p_{out} is

$$k_i \cdot \log_2(|\lambda_i|) \geq \frac{\log_2(|\lambda_i|)}{\log_2(d_i)}\big(\log_2(\kappa_{out}) - \log_2(\kappa_{in} - \Delta_i)\big) \tag{12}$$

(in terms of the log complexity blowup (9)). Note that this lower bound is tight: it could be replaced by an equality at the cost of ceiling the term between parentheses (*i.e.* the term corresponding to k_i). We further note that the above equation is consistent with the complexity analysis of the expanding compiler provided in [9]. Indeed going from a constant leakage probability $p_{in} = p$ to a target security level $p_{out} = 2^{-\kappa}$ by applying k_i times a single RPE compiler CC_i, we retrieve a complexity of $\mathcal{O}(\kappa^e)$ with $e = \frac{\log_2(|\lambda_i|)}{\log_2(d_i)}$.

Equation (12) shows that using CC_i to go from input probability p_{in} to output probability p_{out} induces a log complexity cost close to

$$\frac{\log_2(|\lambda_i|)}{\log_2(d_i)}\big(\log_2(\kappa_{out}) - \log_2(\kappa_{in})\big)$$

provided that κ_{in} is sufficiently greater than Δ_i. So given the latter informal condition, it appears that the parameter i minimizing the ratio $\frac{\log_2(|\lambda_i|)}{\log_2(d_i)}$ gives the best complexity.

Application. For the asymptotic construction introduced in [10], the RPE compiler CC_i features

- an amplification order $d_i = \mathcal{O}(n_i)$,
- a gadget complexity $s_i = \mathcal{O}(n_i^2)$,
- an eigen-complexity $|\lambda_i| = \mathcal{O}(n_i^2)$.

For such a construction, the ratio $\frac{\log_2(|\lambda_i|)}{\log_2(d_i)}$ is decreasing and converging towards 2 as n_i grows. On the other hand, Δ_i tends to $\log_2(n_i)$ which implies that CC_i should only be applied to an input probability lower than $\frac{1}{n_i}$.

3.3 Selection of the Expansion Levels

In this section, we investigate the impact of the choice of the expansion levels k_i on the complexity of the dynamic expanding compiler. We first assess the asymptotic complexity obtained from a simple approach and then provide some application results for some given gadgets.

In the following CC_0 shall denote an RPE compiler with constant parameters while $\{\mathsf{CC}_i\}_{i\geq 1}$ shall denote a family of RPE compilers indexed by a parameter i. We do this distinction since the goal of the CC_0 compiler shall be to tolerate the highest leakage rate and to transit from a (possibly high) leakage probability p to some lower failure probability p_i which is in turn tolerated by at least one compiler from $\{\mathsf{CC}_i\}_i$.

A Simple Approach. We consider a simple approach in which the compiler CC_0 is iterated k_0 times and then a single compiler CC_i is iterated k_i times. The complexity blowup of this compiler is $|\lambda_0|^{k_0}|\lambda_i|^{k_i}$. The first expansion level k_0 is chosen to ensure that the intermediate probability $p_i := f_0^{(k_0)}(p)$ is lower than $\bar{q}_i$ (the lower bound on the tolerated leakage rate of CC_i from Theorem 3). Then k_i is chosen so that $f_i^{(k_i)} \leq 2^{-\kappa}$.

Concretely, we set $\kappa_i := \Delta_i + 1$ which, by Lemma 1, gives

$$k_0 = \big\lceil \log_{d_0}(\Delta_i + 1) - \log_{d_0}(\log_2(p) - \Delta_0)\big\rceil \ , \tag{13}$$

and

$$k_i = \lceil \log_{d_i}(\kappa) \rceil = \mathcal{O}\big(\log_{d_i}(\kappa) \big) \ . \tag{14}$$

For some constant leakage probability p and some start compiler CC_0 with constant parameters, we get $k_0 = \mathcal{O}\big(\log_{d_0}(\Delta_i) \big)$ giving an asymptotic complexity blowup of

$$\mathcal{O}\big(|\lambda_0|^{k_0} |\lambda_i|^{k_i} \big) = \mathcal{O}\big(\Delta_i^{e_0} \kappa^{e_i} \big) \quad \text{with} \quad e_0 = \frac{\log_2(|\lambda_0|)}{\log_2(d_0)} \quad \text{and} \quad e_i = \frac{\log_2(|\lambda_i|)}{\log_2(d_i)} \ . \tag{15}$$

Then for any choice of i we get an asymptotic complexity blowup of $\mathcal{O}\big(\kappa^{e_i}\big)$ which is the same asymptotic complexity as the standard expanding compiler with base compiler CC_i. On the other hand, our simple dynamic compiler $\mathsf{CC}_i^{(k_i)} \circ \mathsf{CC}_0^{(k_0)}$ tolerates the same leakage rate as CC_0.

Using this simple approach we hence get the best of both worlds:

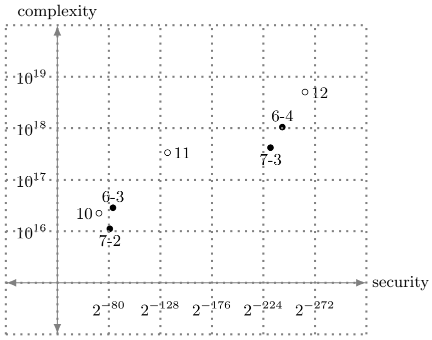
- a possibly inefficient RPE compiler CC_0 tolerating a high leakage rate q_0,
- a family of RPE compilers $\{\mathsf{CC}_i\}_i$ with complexity exponent $e_i = \frac{\log_2(|\lambda_i|)}{\log_2(d_i)}$ decreasing with i.

We stress that for monotonously increasing $|\lambda_i|$ and d_i, the asymptotic complexity of our simple approach is $\mathcal{O}(\kappa^e)$ where e can be made arbitrary close to $\lim_{i\to\infty} \frac{\log_2(|\lambda_i|)}{\log_2(d_i)}$.

Application. To illustrate the benefits of our dynamic approach, we simply get back to the experimentations on the AES implementation from [9]. The authors apply either a 3-share or 5-share compiler repeatedly until they reach their targeted security level. While using the 5-share compiler reduces the tolerated probability, we demonstrate that we can use both compilers to get the best tolerated probability as well as a better complexity.

Figure 1 illustrates the trade-offs in terms of achieved security level and complexity of the expansion strategy when using different compilers at each iteration of the expansion. Starting from a tolerated leakage probability p ($2^{-7.6}$ on the left and $2^{-9.5}$ on the right), the empty bullets ($\circ$) give this trade-off when only the 3-share compiler is iterated. In this case, the final security function ε from Theorem 1 is equal to $f_3^{(k_3)}(p)$ if we consider f_3 to be the failure function of the 3-share compiler, for a certain number of iterations k_3 which is written next to each empty bullet on the figure. On the other hand, the black bullets ($\bullet$) represent the trade-offs achieved in terms of complexity and security levels while combining both compilers with different numbers of iterations. In this case, we start the expansion with a certain number of iterations k_3 of the 3-share compiler, and then we continue with k_5 iterations of the 5-share compiler of failure function f_5, the final compiled circuit is then random probing secure with $\varepsilon = f_5^{(k_5)}(f_3^{(k_3)}(p))$ for $p \in \{2^{-7.6}, 2^{-9.5}\}$. The number of iterations of the compilers is written next to each black bullet in the format k_3-k_5.

For instance, starting from the best tolerated probability $2^{-7.6}$, the static compiler from [9, 10] requires 11 applications of the 3-share compiler to achieve a security level of at least 80 bits. This effort comes with an overall complexity of $10^{17.52}$. Using our dynamic approach, we can combine the 3-share and the 5-share to achieve this 80 bits security level for the same tolerated probability but with a complexity of $10^{16.04}$. That would require 7 iterations of the 3-share compiler and 2 iterations of the 5-share compiler. Starting from the same leakage probability, a security level of at least 128 bits is achieved also with 11 applications of the 3-share compiler with a complexity of $10^{17.52}$. In order to achieve at least the same security, we would need more iterations of both compilers in the dynamic approach. With 7 iterations of the 3-share compiler and 3 iterations of



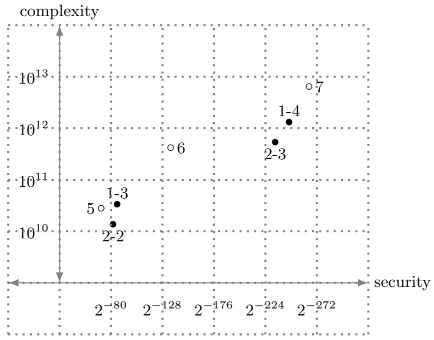


Fig. 1: Complexity of random probing AES for different security levels for a tolerated probability of $2^{-7.6}$ (left) or $2^{-9.5}$ (right).

the 5-share compiler, we get a complexity of $10^{17.62}$ which is very close to the complexity of the 3-share application alone, while achieving a security level of 231 bits. That is, we almost double the security level achieved using 11 iterations of the 3-share compiler with an almost equal complexity. For a tolerated probability of $2^{-7.6}$ and at least 128 bits of security, note that 11 applications of the 3-share compiler yield a security order of 2^{-135} while both other trade-offs directly yield security orders of 2^{-242} (6 iterations of 3-share and 4 iterations of 5-share) and 2^{-231} (7 iterations of 3-share and 3 iterations of 5-share), with one less iteration they would be below 128 bits, which explains their more important complexity. The same behavior can be observed with a starting tolerated leakage probability of $2^{-9.5}$ on the right.

The above results motivate the next contributions of this paper, namely finding RPE compilers which achieve the maximal amplification orders and which benefit from good asymptotic complexity (*i.e.* gadgets defined for any number of shares n with amplification order increasing with n) in order to optimize the security-efficiency trade-off and to tolerate the best possible leakage probability. We showed this far that the tolerated leakage probability decreases with an increasing number of shares n. So if we want to tolerate the best leakage probability, we would start with a few iterations of a compiler with a small number of shares and which tolerates a good leakage probability (which can be computed for instance with the verification tool VRAPS [9]), typically a 3-share construction. Meanwhile, after a few constant number of iterations, we can change to a different compiler which benefits from a better asymptotic complexity (as explained above with our simple approach). In the constructions from [10], the bottleneck in terms of asymptotic complexity was from the linear gadgets (addition and copy). Thanks to the quasilinear refresh gadget we introduce later in this paper, the bottleneck becomes the multiplication gadget (with n^2 multiplications), which we also improve in the following sections under some conditions on the base field.

4 Linear Gadgets with Quasi-Linear Complexity

In a first attempt, we aim to reduce the complexity of the linear gadgets that are to be used in our dynamic compiler.

In [10], the authors provide new constructions of generic addition and copy gadgets, using a refresh gadget G_{refresh} as a building block. The construction works for any number of shares and the authors prove the RPE security of the gadgets based on the security of G_{refresh}. In a nutshell, given a n-share refresh gadget G_{refresh}, the authors construct a copy gadget G_{copy} which on input sharing $(a_1, \ldots, a_n)$, outputs the sharings

$$\Big(G_{\text{refresh}}(a_1, \ldots, a_n), G_{\text{refresh}}(a_1, \ldots, a_n)\Big) \tag{16}$$

with two independent executions of G_{refresh}. The authors also construct an addition gadget G_{add} which, on input sharings $(a_1, \ldots, a_n)$ and $(b_1, \ldots, b_n)$, first refreshes the inputs separately, then outputs the sharewise sum of the results

$$\Big(G_{\text{refresh}}(a_1, \ldots, a_n) + G_{\text{refresh}}(b_1, \ldots, b_n)\Big). \tag{17}$$

If the refresh gadget G_{refresh} is TRPE of amplification order d, the authors show that G_{copy} is also TRPE of amplification order d, and G_{add} is TRPE of amplification order at least $\lfloor d/2 \rfloor$.

While the copy gadgets from [10] achieve an optimal amplification order, this is not the case yet for addition gadgets and we first aim to fill this gap. Precisely, we introduce a new property which, when satisfied by its inherent refresh gadget G_{refresh}, makes the addition gadget TRPE with the same amplification order as G_{refresh}. We then prove that this new property is actually satisfied by the refresh gadget from [6] which has quasi-linear complexity $\mathcal{O}(n \log n)$ in the sharing order n. Using this refresh gadget as a building block, we obtain linear gadgets G_{add} and G_{copy} with quasi-linear complexities.

Constructions of Linear Gadgets from a Stronger Building Block. We first define our new property (as a variant of properties defined in [9, 10]) which proves to be a useful requirement for refresh gadgets when used as a building block of linear gadgets.

Definition 9 (t-Strong TRPE2). *Let G be an n-share 1-input gadget. Then G is t-Strong TRPE2 (abbreviated t-STRPE2) if and only if for any set J' of output shares indices and any set W of internal wires of G such that $|W| + |J'| \leq t$, there exists a set J of output share indices such that $J' \subseteq J$ and $|J| = n - 1$ and such that the assignment of the wires indexed by W together with the output shares indexed by J can be perfectly simulated from the input shares indexed by a set I of cardinality satisfying $|I| \leq |W| + |J'|$.*

Remark 1. This new property directly implies the TRPE2 property with maximal amplification order introduced in [10]. Recall that G is t-TRPE2 with maximal amplification order if and only if for any set W of probed wires such that $|W| < \min(t + 1, n - t)$, there exists a set J of output shares indices such that $|J| = n - 1$ and such that an assignment of the wires indexed by W and the output shares indexed by J can be jointly perfectly simulated from input shares indexed in a set I such that $|I| \leq |W|$.

Having a refresh gadget which satisfies the property from Definition 9 results in tighter constructions for generic addition gadgets as stated in Lemma 2. Its proof is given in Appendix A.6.

Lemma 2. *Let G_{refresh} be an n-share refresh gadget and let G_{add} be the addition gadget described in Equation* (17)*. Then if G_{refresh} is (t, f)-TRPE for any $t \leq n - 1$ of amplification order $d \geq \min(t + 1, n - t)$ and G_{refresh} is $(n - 1)$-STRPE2, then G_{add} is (t, f')-RPE (resp. (t, f')-TRPE) for any $t \leq n - 1$ for some f' of amplification order* $\min(t + 1, n - t)$.

Instantiation of Linear Gadgets with Quasi-Linear Refresh Gadget. A refresh gadget with $\mathcal{O}(n \log n)$ complexity was introduced in [6]. In a nutshell, the idea is to add a linear number of random values on the shares at each step, to split the shares in two sets to apply the recursion, and then to add a linear number of random values again. For the sake of completeness, we provide the algorithmic description of this refresh gadget in Appendix A.7. It was proven to be $(n-1)$-SNI in [6]. In Lemma 3, we show that this gadget is also (t, f)-TRPE of amplification order $\min(t+1, n-t)$ and that it satisfies $(n-1)$-STRPE2. The proof is given in Appendix A.8.

Lemma 3. *Let $G_{refresh}$ be the n-share refresh gadget described above from [6]. Then $G_{refresh}$ is (t, f)-TRPE for some function $f : \mathbb{R} \to \mathbb{R}$ of amplification order $d \geq \min(t+1, n-t)$. $G_{refresh}$ is additionally $(n-1)$-STRPE2.*

Hence, we can instantiate the generic copy and addition gadgets described in (16) and (17) using the above refresh gadget as G_{refresh}. We thus obtain RPE gadgets G_{add} and G_{copy} enjoying optimal amplification order in quasi-linear complexity $\mathcal{O}(n \log n)$.

Regarding the asymptotic complexity of the expanding compiler, the eigenvalues λ_1, λ_2 from Section 2 are hence now both in $\mathcal{O}(n \log n)$. At this point, only the quadratic number of multiplications in the multiplication gadget still separates us from a compiler of quasi-linear complexity. We tackle this issue in the next section by constructing a generic multiplication gadget. We finally end up with a full expanding compiler with quasi-linear asymptotic complexity.

5 Towards Optimal Multiplication Gadgets

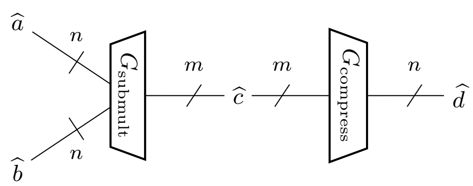


Fig. 2: n-share multiplication gadget G_{mult} from two subgadgets G_{submult} and G_{compress}

In what follows we should distinguish two types of multiplication gates: regular two-operand multiplications on $\mathbb{K}$, that we shall call bilinear multiplications, and multiplications by constant (or scalar multiplications) which have a single input operand and the constant scalar is considered as part of the gate description.

In previous works [9, 10], the number of bilinear multiplications is the prominent term of the expanding compiler's complexity. While the most deployed multiplication gadgets (*e.g.*, [18]) require a quadratic number of bilinear multiplications in the masking order, the authors of [8] exhibited a probing secure higher-order masking multiplication with only a linear number of bilinear multiplications. Their construction, which applies on larger fields, is built from the composition of two subgadgets G_{submult} and G_{compress}, as described in Figure 2. In a nutshell, on input sharings $\widehat{a}$ and $\widehat{b}$, the subgadget G_{submult} performs multiplications between the input shares of $\widehat{a}$ and $\widehat{b}$ as well as linear combinations of these products and it outputs a m-sharing $\widehat{c}$ of the product $a \cdot b$ where

$m \geq n$ [5]. Next, the compression gadget G_{compress} compresses the m-sharing $\widehat{c}$ back into an n-sharing $\widehat{d}$ of the product $a \cdot b$.

The authors of [8] instantiate this construction with a sub-multiplication gadget which performs only $\mathcal{O}(n)$ bilinear multiplications and with the compression gadget from [11]. In addition to bilinear multiplications, their sub-multiplication gadget additionally requires a quadratic number of linear operations (*i.e.*, addition, copy, multiplications by a constant) and random generation gates.

In the following, we rely on the construction [8] with its gadget G_{submult} which offers a linear number of bilinear multiplications to build a more efficient RPE multiplication gadget. In order to use it in our expanding compiler, we integrate an additional gate for the multiplication by a constant and discuss the resulting asymptotic complexity. We additionally demonstrate that the compression gadget of [8] is not $(n-1)$-SNI as claimed in the paper, and show that we can rely on other simple and more efficient compression gadgets which satisfy the expected properties.

5.1 Global Multiplication Gadget

We first define two new properties that G_{submult} and G_{compress} will be expected to satisfy to form a (t, f)-RPE multiplication gadget with the maximum amplification order from the construction [8].

Contrary to the usual simulation notions, the first *partial*-NI property distinguishes the number of probes on the gadget, and the number of input shares that must be used to simulate them. It additionally tolerates a *simulation failure* on at most one of the inputs (*i.e.*, no limitation on the number of shares for the simulation).

Definition 10 ((s,t)-partial NI). *Let G be a gadget with two input sharings $\widehat{a}$ and $\widehat{b}$. Then G is* (s,t)-partial NI *if and only any the assignment of any t wires of G can be perfectly simulated from shares $(a_i)_{i \in I_1}$ of $\widehat{a}$ and $(b_i)_{i \in I_2}$ of $\widehat{b}$ such that $|I_1| \leq s$* ***or*** *$|I_2| \leq s$.*

The second property is a variant of the classical TRPE property that we refer to as *comp-TRPE*.

Definition 11 ((t,f)-comp-TRPE). *Let G be a 1-to-1 gadget with m input shares and n output shares such that $m > n$. Let $t \leq n-1$ and $d = \min(t+1, n-t)$. Then G is* (t,f)-comp-TRPE *if and only if for all sets of internal wires W of G with $|W| \leq 2d-1$, we have:*

1. *$\forall\ J, |J| \leq t$ a set of output share indices of G, the assignment of the wires indexed by W and the output shares indexed by J can be jointly perfectly simulated from the input shares of G indexed by a set I, such that $|I| \leq |W|$.*
2. *$\exists\ J', |J'| = n-1$ a set of output share indices of G, such that the assignment of the wires indexed by W and the output shares indexed by J' can be jointly perfectly simulated from the input shares of G indexed by a set I, such that $|I| \leq |W|$.*

Similarly to what was done in [8] for the SNI property, we can prove that the composition of a gadget G_{submult} and G_{compress} which satisfy well chosen properties results in an overall multiplication gadget which is (t,f)-RPE specifically for any $t \leq n-1$ achieving the maximum amplification order $d = \min(t+1, n-t)$. This is formally stated in Lemma 4 which proof is given in Appendix A.9.

Lemma 4. *Consider the n-share multiplication gadget of Figure 2 formed by a 2-to-1 multiplication subgadget $G_{submult}$ of m output shares and a 1-to-1 compression gadget $G_{compress}$ of m input shares such that $m > n$. Let $t \leq n-1$ and $d = \min(t+1, n-t)$. If*

[5] In case of a sharewise multiplication for instance, we would have $m = n^2$.

– $G_{submult}$ *is* $(d-1)$*-NI and* $(d-1, 2d-1)$*-partial NI,*
– $G_{compress}$ *is* (t, f)*-comp-TRPE,*

then the multiplication gadget G_{mult} *is* (t, f)*-RPE of amplification order* d*.*

5.2 Construction of G_{compress}

In a first attempt, we analyze the compression function that was introduced in [11] and used to build a multiplication gadget in [8]. As it turns out not to be SNI or meet our requirements for the expanding compiler, we exhibit a new and also more efficient construction in a second attempt.

G_{compress} from [8, 11]. The authors of [8] use the $[m : n]$-compression gadget introduced in [11] for any input sharing m, using a $[2n : n]$-compression subgadget as a building block. In a nutshell, it first generates an ISW-refresh of the zero n-sharing $(w_1, \ldots, w_n)$. Then, these shares are added to the input ones $(c_1, \ldots, c_n)$ to produce the sequence of output shares $(c_1 + w_1, \ldots, c_n + w_n)$.

The compression gadget is claimed to be $(n-1)$-SNI in [8]. However, we demonstrate that it is not with the following counterexample. Let $n > 2$ and $i \in [n]$. We consider the set composed of a single output share of the compression procedure $J = \{(c_i + w_i) + c_{n+i}\}$ and the set of probes on the internal wires $W = \{w_i\}$. For the compression to be 2-SNI, we must be able to perfectly simulate both the wires in W and J with at most $|W| = 1$ share of the input $\widehat{c}$. However, we can easily observe that $(c_i + w_i) + c_{n+i} - w_i = c_i + c_{i+n}$ requires the two input shares c_i and c_{i+n} to be simulated, which does not satisfy the 2-SNI property. In conclusion, the above gadget is actually not SNI, and interestingly it is not sufficient either for our construction, *i.e.* it does not satisfy Definition 11. This observation motivates our need for a new compression gadget which satisfies the necessary property for our construction.

New Construction for G_{compress}. In Algorithm 1, we exhibit a new $[m : n]$-compression technique using an m-share refresh gadget G_{refresh} as a building block. We demonstrate in Lemma 5 that this new compression gadget satisfies the necessary properties for our construction as long as $m \geq 2n$. The proof is given in Appendix A.10.

Algorithm 1: $[m : n]$-compression gadget

Input : $(c_1, \ldots, c_m)$ such that $m \geq 2n$, m-share refresh gadget G_{refresh}
Output: $(d_1, \ldots, d_n)$ such that $\sum_{i=1}^{n} d_i = \sum_{i=1}^{m} c_i$
$K \leftarrow \lfloor m/n \rfloor$;
$(c'_1, \ldots, c'_m) \leftarrow G_{\text{refresh}}(c_1, \ldots, c_m)$;
$(d_1, \ldots, d_n) \leftarrow (c'_1, \ldots, c'_n)$;
for $i = 1$ **to** $K - 1$ **do**
  $(d_1, \ldots, d_n) \leftarrow (d_1 + c'_{1+i\cdot n}, \ldots, d_n + c'_{n+i\cdot n})$;
end
for $i = 1$ **to** $m - K \cdot n$ **do**
  $d_i \leftarrow d_i + c'_{i+K\cdot n}$;
end
return $(d_1, \ldots, d_n)$;

Lemma 5. *Let $G_{compress}$ be the $[m : n]$-compression gadget from Algorithm 1 such that $m \geq 2n$. If $G_{refresh}$ is $(m-1)$-SNI and $(m-1)$-STRPE2, then $G_{compress}$ is (t, f)-comp-TRPE (Definition 11).*

As shown in Section 4, the refresh gadget from [5] is actually $(m-1)$-SNI and $(m-1)$-STRPE2 for any sharing order m. This gadget can then be used as a building block for the $[m : n]$-compression gadget, giving it a complexity of $\mathcal{O}(m \log m)$ and satisfying the necessary properties. In addition, this further provides an improvement over the complexity of the proposed gadget in [8] which has a complexity of $\mathcal{O}(\lfloor \frac{m}{n} \rfloor n^2)$ (because it performs a n-share ISW-refreshing $\lfloor \frac{m}{n} \rfloor$ times, see [8] for more details on the algorithm).

5.3 Construction of $G_{\mathbf{submult}}$

To complete the construction of the overall multiplication gadget, we now exhibit relevant constructions for G_{submult}. We first rely on the construction from [8] which happens to achieve the desired goal in some settings. While all the cases are not covered by the state-of-the-art proposal, we then slightly modify the construction to meet all our requirements. Both constructions rely on linear multiplications that are not included yet on the expanding compiler. We thus start with a construction for this additional linear gadget that we further denote G_{cmult}.

Construction for $G_{\mathbf{cmult}}$. We give a natural construction for G_{cmult} in Algorithm 2 which simply multiplies each input share by the underlying constant value and then applies a (t, f)-RPE refresh gadget G_{refresh}. Basically, with a (T)RPE refresh gadget G_{refresh}, we obtain a (T)RPE linear multiplication gadget G_{cmult} as stated in Lemma 6. The proof is given in Appendix A.5.

Algorithm 2: n-share multiplication by a constant

Input : sharing $(a_1, \ldots, a_n)$, constant value $\widehat{c}$, n-share refresh gadget G_{refresh}
Output: sharing $(d_1, \ldots, d_n)$ such that $d_1 + \cdots + d_n = c.(a_1 + \ldots + a_n)$
$(b_1, \ldots, b_n) \leftarrow (c.a_1, \ldots, c.a_n)$;
$(d_1, \ldots, d_n) \leftarrow G_{\text{refresh}}((b_1, \ldots, b_n))$;
return $(d_1, \ldots, d_n)$;

Lemma 6. *Let $G_{refresh}$ be a (t, f)-(T)RPE n-share refresh gadget of amplification order d. Then G_{cmult} instantiated with $G_{refresh}$ is (t, f')-(T)RPE of amplification order d.*

Relying on an additional gate for the linear multiplication does not impact the security analysis and the application of the compilation, but it modifies the complexity analysis of the expanding compiler. From the analysis given in Section 2.4, a complexity vector is associated to each base gadget $N_G = (N_a, N_c, N_{cm}, N_m, N_r)^{\mathsf{T}}$ where N_a, N_c, N_{cm}, N_m, N_r stand for the number of addition gates, copy gates, constant multiplication gates, (bilinear) multiplication gates and random gates respectively in the corresponding gadget. The matrix M_{CC} is now a 5×5 square matrix defined as

$$M = \left(N_{G_{\text{add}}} \mid N_{G_{\text{copy}}} \mid N_{G_{\text{cmult}}} \mid N_{G_{\text{mult}}} \mid N_{G_{\text{random}}} \right)$$

including, for each vector, the number of linear multiplications. Five eigenvalues λ_1, λ_2, λ_3, λ_4, λ_5 are to be computed, *i.e.*, one more compared to the expanding compiler in the original setting.

We can consider as before that bilinear multiplication gates are solely used in G_{mult} ($N_{G_{\text{add}},m} = N_{G_{\text{copy}},m} = N_{G_{\text{cmult}},m} = 0$) and that constant multiplication gates are eventually solely used in G_{cmult} and G_{mult} ($N_{G_{\text{add}},cm} = N_{G_{\text{copy}},cm} = 0$) which is the case in the constructions we consider in this paper. It can be checked that (up to some permutation) the eigenvalues satisfy

$$(\lambda_1, \lambda_2) = \mathsf{eigenvalues}(M_{ac}) \ , \quad \lambda_3 = N_{G_{\text{cmult}},cm} \ , \quad \lambda_4 = N_{G_{\text{mult}},m} \quad \text{and} \quad \lambda_5 = n$$

where M_{ac} is the top left 2×2 block matrix of M_{CC}

$$M_{ac} = \begin{pmatrix} N_{G_{\text{add}},a} & N_{G_{\text{copy}},a} \\ N_{G_{\text{add}},c} & N_{G_{\text{copy}},c} \end{pmatrix} \ .$$

We get two complexity expressions for the expansion strategy

$$|\widehat{C}| = \mathcal{O}\big(|C| \cdot N_{\max}^k\big) \tag{18}$$

with $N_{\max} = \max(|\mathsf{eigenvalues}(M_{ac})|, N_{G_{\text{cmult}},cm}, N_{G_{\text{mult}},m}, n)$ and with the security parameter κ

$$|\widehat{C}| = \mathcal{O}\big(|C| \cdot \kappa^e\big) \quad \text{with} \quad e = \frac{\log N_{\max}}{\log d} \ .$$

Note that the exhibited construction for the linear multiplication gadget requires $N_{G_{\text{cmult}},cm} = n$ linear multiplications. Hence $\lambda_3 = N_{G_{\text{cmult}},cm} = \lambda_5 = N_{G_{\text{random}},r} = n$ and the global complexity (18) can be rewritten as

$$|\widehat{C}| = \mathcal{O}\big(|C| \cdot N_{\max}^k\big) \quad \text{with} \quad N_{\max} = \max(|\mathsf{eigenvalues}(M_{ac})|, N_{G_{\text{mult}},m})$$

if the number of multiplications is greater than n. The asymptotic complexity of the RPE compiler is thus not affected by our new base gadget G_{cmult}. We now describe our constructions of G_{submult}.

G_{submult} from [8]. The authors of [8] provide a $(n-1)$-NI construction for G_{submult} which outputs $2n-1$ shares while consuming only a linear number of bilinear multiplications in the masking order. We first recall their construction which relies on two square matrices of $(n-1)^2$ coefficients in the working field. As shown in [8], these matrices are expected to satisfy some condition for the compression gadget to be $(n-1)$-NI. Since we additionally want the compression gadget to be $(d-1, 2d-1)$-partial NI, we introduce a stronger condition and demonstrate the security of the gadget in our setting.

Let $\mathbb{F}_q$ be the finite field with q elements. Let $\boldsymbol{\gamma} = (\gamma_{i,j})_{1 \le i,j < n} \in \mathbb{F}_q^{(n-1)\times(n-1)}$ be a constant matrix, and let $\boldsymbol{\delta} = (\delta_{i,j})_{1 \le i,j < n} \in \mathbb{F}_q^{(n-1)\times(n-1)}$ be the matrix defined by $\delta_{i,j} = 1 - \gamma_{j,i}$ for all $1 \le i, j < n-1$. G_{submult} takes as input two n-sharings $\boldsymbol{a}$ and $\boldsymbol{b}$ and outputs a $(2n-1)$-sharing $\boldsymbol{c}$ such that:

- $c_1 = \Big(a_1 + \sum_{i=2}^{n}(r_i + a_i)\Big) \cdot \Big(b_1 + \sum_{i=2}^{n}(s_i + b_i)\Big)$
- $c_i = -r_i \cdot \Big(b_1 + \sum_{j=2}^{n}(\delta_{i-1,j-1} s_j + b_j)\Big)$ for $i = 2, \ldots, n$

- $c_{i+n-1} = -s_i \cdot \left(a_1 + \sum_{j=2}^{n} (\gamma_{i-1,j-1} r_j + a_j)\right)$ for $i = 2, \dots, n$

where r_i and s_i are randomly generated values for all $2 \leq i \leq n$. It can be easily checked that G_{submult} performs $2n-1$ bilinear multiplications, and that it is correct, *i.e.* $\sum_{i=1}^{2n-1} c_i = \sum_{i=1}^{n} a_i \cdot \sum_{i=1}^{n} b_i$.

In [8], the authors prove that a gadget is $(n-1)$-NI if one cannot compute a linear combination of any set of $n-1$ probes which can reveal all of the n secret shares of the inputs and which does not include any random value in its algebraic expression. We refer to [8] for more details on this result.

Based on this result, the authors demonstrate in [8], that G_{submult} is $(n-1)$-NI if the matrices $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$ satisfy Condition 1 that we recall below.

Condition 1 ***(from [8])*** *Let $\ell = 2 \cdot (n+1) \cdot (n-1) + 1$. Let $\boldsymbol{I}_{n-1} \in \mathbb{F}_q^{(n-1)\times(n-1)}$ be the identity matrix, $\mathbf{0}_{x \times y} \in \mathbb{F}_q^{x \times y}$ be a matrix of zeros (when $y = 1$, $\mathbf{0}_{x \times y}$ is also written $\mathbf{0}_x$), $\mathbf{1}_{x \times y} \in \mathbb{F}_q^{x \times y}$ be a matrix of ones, $\boldsymbol{D}_{\boldsymbol{\gamma},j} \in \mathbb{F}_q^{(n-1)\times(n-1)}$ be the diagonal matrix such that $D_{\boldsymbol{\gamma},j,i,i} = \gamma_{j,i}$, $\boldsymbol{T}_{n-1} \in \mathbb{F}_q^{(n-1)\times(n-1)}$ be the upper-triangular matrix with just ones, and $\boldsymbol{T}_{\boldsymbol{\gamma},j} \in \mathbb{F}_q^{(n-1)\times(n-1)}$ be the upper-triangular matrix for which $T_{\boldsymbol{\gamma},j,i,k} = \gamma_{j,i}$ for $i \leq k$:*

$$\boldsymbol{I}_{n-1} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & 1 \end{pmatrix} \qquad \boldsymbol{D}_{\boldsymbol{\gamma},j} = \begin{pmatrix} \gamma_{j,1} & 0 & \dots & 0 \\ 0 & \gamma_{j,2} & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & \gamma_{j,n-1} \end{pmatrix}$$

$$\boldsymbol{T}_{n-1} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 0 & 1 & & 1 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & 1 \end{pmatrix} \qquad \boldsymbol{T}_{\boldsymbol{\gamma},j} = \begin{pmatrix} \gamma_{j,1} & \gamma_{j,1} & \dots & \gamma_{j,1} \\ 0 & \gamma_{j,2} & & \gamma_{j,2} \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & \gamma_{j,n-1} \end{pmatrix}$$

We define the following matrices (with $n' = n-1$):

$$\boldsymbol{L} = \left(\begin{array}{c|c|c|c|c|c|c|c|c|c|c} 1 & \mathbf{0}_{1\times n'} & \mathbf{0}_{1\times n'} & \mathbf{0}_{1\times n'} & \mathbf{0}_{1\times n'} & \dots & \mathbf{0}_{1\times n'} & \mathbf{1}_{1\times n'} & \mathbf{1}_{1\times n'} & \dots & \mathbf{1}_{1\times n'} \\ \mathbf{0}_{n'} & \boldsymbol{I}_{n'} & \mathbf{0}_{n'\times n'} & \boldsymbol{I}_{n'} & \boldsymbol{I}_{n'} & \dots & \boldsymbol{I}_{n'} & \boldsymbol{T}_{n'} & \boldsymbol{T}_{n'} & \dots & \boldsymbol{T}_{n'} \end{array} \right)$$
$$\boldsymbol{M} = \left(\begin{array}{c|c|c|c|c|c|c|c|c|c|c} \mathbf{0}_{n'} & \mathbf{0}_{n'\times n'} & \boldsymbol{I}_{n'} & \boldsymbol{I}_{n'} & \boldsymbol{D}_{\boldsymbol{\gamma},1} & \dots & \boldsymbol{D}_{\boldsymbol{\gamma},n'} & \boldsymbol{T}_{n'} & \boldsymbol{T}_{\boldsymbol{\gamma},1} & \dots & \boldsymbol{T}_{\boldsymbol{\gamma},n'} \end{array} \right)$$

Condition 1 is satisfied for a matrix $\boldsymbol{\gamma}$ *if for any vector $\boldsymbol{v} \in \mathbb{F}_q^{\ell}$ of Hamming weight $\mathsf{hw}(\boldsymbol{v}) \leq n-1$ such that $\boldsymbol{L} \cdot \boldsymbol{v}$ contains no coefficient equal to 0 then $\boldsymbol{M} \cdot \boldsymbol{v} \neq \mathbf{0}_{n-1}$.*

In the above condition, the matrices $\mathbf{L}$ and $\mathbf{M}$ represent the vectors of dependencies for each possible probe. All the probes involving shares of $\widehat{a}$ for matrix $\boldsymbol{\gamma}$ (and symmetrically shares of $\widehat{b}$ for matrix $\boldsymbol{\delta}$) are covered in the columns of $\mathbf{L}$ and $\mathbf{M}$. Namely, the first column represents the probe a_1. As it does not involve any random, it results in a zero column in $\mathbf{M}$. The next columns represents the probes a_i, then the probes r_i. They are followed by columns for the probes $(a_i + r_i)$, then $(a_i + \gamma_{j-1,i-1} r_i)$ (for $2 \leq j \leq n$), then $a_1 + \sum_{i=2}^{k} (r_i + a_i)$ (for $2 \leq k \leq n$), and finally then $a_1 + \sum_{j=2}^{k} (\gamma_{i-1,j-1} r_j + a_j)$ (for $2 \leq i \leq n$ and $2 \leq k \leq n$). The above condition means that there is no linear combination of $(n-1)$ probes which can include the expression of all of the input shares, and no random variable.

From this result and by the equivalence between non-interference and tight non-interference developed in [8], we conclude that G_{submult} is $(d-1)$-NI for $d = \min(t+1, n-t)$ for any $t \leq n-1$. Lemma 4 also requires G_{submult} to be $(d-1, 2d-1)$-partial NI to get an overall RPE multiplication gadget. For G_{submult} to satisfy this second property, we need to rely on a stronger condition for matrices $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$ that we present in Condition 2.

Condition 2 *Let $z = 2 \cdot (n+1) \cdot (n-1) + 1$. Let $\boldsymbol{I}_{n-1} \in \mathbb{F}_q^{(n-1)\times(n-1)}$, $\mathbf{0}_{\ell \times n} \in \mathbb{F}_q^{\ell \times n}$, $\mathbf{1}_{\ell \times n} \in \mathbb{F}_q^{\ell \times n}$, $\boldsymbol{D}_{\gamma,j} \in \mathbb{F}_q^{(n-1)\times(n-1)}$, $\boldsymbol{T}_{n-1} \in \mathbb{F}_q^{(n-1)\times(n-1)}$, $\boldsymbol{T}_{\gamma,j} \in \mathbb{F}_q^{(n-1)\times(n-1)}$ and $\boldsymbol{L}$ and $\boldsymbol{M}$ the same matrices as defined in Condition 1.*

Condition 2 is satisfied for a matrix $\boldsymbol{\gamma}$ if and only if for any vector $\boldsymbol{v} \in \mathbb{F}_q^z$ of Hamming weight $\mathsf{hw}(\boldsymbol{v}) \leq n-1$, and for any $i_1, \ldots, i_K \in [z]$ such that $v_{i_1} \neq 0, \ldots, v_{i_K} \neq 0$ and the corresponding columns $i_1, \ldots, i_K$ in $\boldsymbol{L}$ and in $\boldsymbol{M}$ have no zero coefficient (i.e there are K probes of the form $a_1 + \sum_{i=2}^{n}(r_i + a_i)$ or $a_1 + \sum_{j=2}^{n}(\gamma_{i-1,j-1} r_j + a_j)$ for any $i \in \{2, \ldots, n\}$), if $\mathbf{M}.v = 0$, then we have $\mathsf{hw}(\boldsymbol{L} \cdot \boldsymbol{v}) \leq \mathsf{hw}(\boldsymbol{v}) - K$.

Based on this new condition, we can prove our second property G_{submult}, as stated in Lemma 7. The proof is given in Appendix A.11.

Lemma 7. *Let $t \leq n-1$ such that either n is even or $t \neq \lfloor \frac{n-1}{2} \rfloor$ and let $d = \min(t+1, n-t)$. Let $G_{submult}$ the multiplication subgadget introduced in [8]. If both matrices $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$ satisfy Condition 2, then $G_{submult}$ is $(d-1)$-NI and $(d-1, 2d-1)$-partial NI.*

The condition on t and n on Lemma 7 implies that the maximum amplification order for the multiplication gadget cannot be achieved for an odd number of shares (since the maximum order is reached when $t = \lfloor \frac{n-1}{2} \rfloor$). This is not a proof artifact but a limitation of the gadget G_{submult} with respect to the new $(d-1, 2d-1)$-partial NI property. We can easily show that under this extreme conditions on t and n, we have $2d-1 = n$. If we consider the instantiation of G_{submult} for $n = 3$ input shares, we obtain the following $2n-1 = 5$ output shares:

$$
\begin{aligned}
c_1 &= (a_1 + (r_2 + a_2) + (r_3 + a_3)) \cdot (b_1 + (s_2 + b_2) + (s_3 + b_3)) \\
c_2 &= -r_2 \cdot (b_1 + (\delta_{1,1} \cdot s_2 + b_2) + (\delta_{1,2} \cdot s_3 + b_3)) \\
c_3 &= -r_3 \cdot (b_1 + (\delta_{2,1} \cdot s_2 + b_2) + (\delta_{2,2} \cdot s_3 + b_3)) \\
c_4 &= -s_2 \cdot (a_1 + (\gamma_{1,1} \cdot r_2 + a_2) + (\gamma_{1,2} \cdot r_3 + a_3)) \\
c_5 &= -s_3 \cdot (a_1 + (\gamma_{2,1} \cdot r_2 + a_2) + (\gamma_{2,2} \cdot r_3 + a_3))
\end{aligned}
$$

To prove the $(d-1, 2d-1)$-partial NI property, we need to ensure that any set of at most $2d-1 = 3$ probes can be perfectly simulated from at most $d-1 = 1$ shares of one of the inputs and any number of shares from the other one. However, the three probes on c_1, c_3, c_4 reveal information on each of their sub-product. In particular, $(a_1 + (r_2 + a_1) + (r_3 + a_3))$ (from c_1), r_3 (from c_3) and $(a_1 + (\gamma_{1,1} \cdot r_2 + a_2) + (\gamma_{1,2} \cdot r_3 + a_3))$ (from c_4) would reveal $\widehat{a}$. Similarly, $(b_1 + (s_2 + b_2) + (s_3 + b_3))$ (from c_1), $(b_1 + (\delta_{2,1} \cdot s_2 + b_2) + (\delta_{2,2} \cdot s_3 + b_3))$ (from c_3) and s_2 (from c_4) would reveal $\widehat{b}$. Hence, the gadget is not $(d-1, 2d-1)$-partial NI. This counterexample with 3 shares can be directly extended to any odd number of shares.

This counterexample motivates a new construction for G_{submult} which would cover all values for n and t. In the following, we slightly modify the construction from [8] to achieve the maximum amplification order in any setting.

Remark 2. The current construction of G_{submult} outputs $m = 2n - 1$ shares, which does not satisfy the requirement $m \geq 2n$ shares for the compression gadget. Nevertheless, it is enough to add an artificial extra share c_{2n-1} equal to zero between both building blocks. In particular, the compression gadget (and subsequently the refresh gadget) does not expect the input sharing to be uniform to achieve the stated security properties.

New Construction for $\boldsymbol{G}_{\textbf{submult}}$. As stated earlier, Lemma 7 does not hold for G_{submult} in the case where n is odd and $t = (n-1)/2$. In order to cover this case, we propose a slightly modified version of G_{submult} with two extra random values r_1 and s_1. In this version, we let $\boldsymbol{\gamma} = (\gamma_{i,j})_{1 \leq i,j \leq n} \in \mathbb{F}_q^{n \times n}$ be a constant matrix, and let $\boldsymbol{\delta} \in \mathbb{F}_q^{n \times n}$ be the matrix defined by $\delta_{i,j} = 1 - \gamma_{i,j}$. The sub-gadget G_{submult} outputs $2n+1$ shares:

- $c_1 = \Big(\sum_{i=1}^{n}(r_i + a_i)\Big) \cdot \Big(\sum_{i=1}^{n}(s_i + b_i)\Big)$
- $c_{i+1} = -r_i \cdot \Big(\sum_{j=1}^{n}(\delta_{i,j}s_j + b_j)\Big)$ for $i = 1, \ldots, n$
- $c_{i+n+1} = -s_i \cdot \Big(\sum_{j=1}^{n}(\gamma_{i,j}r_j + a_j)\Big)$ for $i = 1, \ldots, n$

where r_i and s_i are randomly generated values. It can be easily checked that G_{submult} now performs $2n+1$ bilinear multiplications, and that it is correct, *i.e.* $\sum_{i=1}^{2n+1} c_i = \sum_{i=1}^{n} a_i \cdot \sum_{i=1}^{n} b_i$.

We now need the following slightly modified version of Condition 2 on $\boldsymbol{\gamma}$ and on $\boldsymbol{\delta}$, which instead of considering a linear combination of at most $n-1$ probes as in Condition 2, considers up to n probes:

Condition 3 *Let $z = (2n+4) \cdot n$. Let $\boldsymbol{I}_n \in \mathbb{F}_q^{n \times n}$ be the identity matrix, $\mathbf{0}_{\ell \times n} \in \mathbb{F}_q^{\ell \times n}$ be the matrix of zeros, $\mathbf{1}_{\ell \times n} \in \mathbb{F}_q^{\ell \times n}$ be the matrix of ones, $\boldsymbol{D}_{\gamma,j} \in \mathbb{F}_q^{n \times n}$ be the diagonal matrix such that $\boldsymbol{D}_{\gamma,j,i,i} = \gamma_{j,i}$, $\boldsymbol{T}_n \in \mathbb{F}_q^{n \times n}$ be the upper triangular matrix with just ones, $\boldsymbol{T}_{\gamma,j} \in \mathbb{F}_q^{n \times n}$ be the upper triangular matrix such that $\boldsymbol{T}_{\gamma,j,i,k} = \gamma_{j,i}$ for $i \leq k$. We define the following matrices:*

$$\boldsymbol{L} = \left[\begin{array}{c|c|c|c|c|c|c|c|c|c} \boldsymbol{I}_n & \mathbf{0}_{n\times n} & \boldsymbol{I}_n & \boldsymbol{I}_n & \ldots & \boldsymbol{I}_n & \boldsymbol{T}_n & \boldsymbol{T}_n & \ldots & \boldsymbol{T}_n \end{array}\right]$$
$$\boldsymbol{M} = \left[\begin{array}{c|c|c|c|c|c|c|c|c|c} \mathbf{0}_{n\times n} & \boldsymbol{I}_n & \boldsymbol{I}_n & \boldsymbol{D}_{\gamma,1} & \ldots & \boldsymbol{D}_{\gamma,n} & \boldsymbol{T}_n & \boldsymbol{T}_{\gamma,1} & \ldots & \boldsymbol{T}_{\gamma,n} \end{array}\right]$$

Then we say that $\boldsymbol{\gamma}$ satisfies Condition 3 if and only if

- *for any vector $\boldsymbol{v} \in \mathbb{F}_q^z$ of Hamming weight $\mathsf{hw}(\boldsymbol{v}) \leq n$,*
- *for any $i_1, \ldots, i_K \in [z]$ such that $v_{i_1} \neq 0, \ldots, v_{i_K} \neq 0$ and the corresponding columns $i_1, \ldots, i_K$ in $\boldsymbol{L}$ and in $\boldsymbol{M}$ have no zero coefficient (i.e there are K probes of the form $\sum_{i=1}^{n}(r_i + a_i)$ or $\sum_{j=1}^{n}(\gamma_{i,j}r_j + a_j)$ for any $i = 1, \ldots, n$),*

if $\boldsymbol{M} \cdot \boldsymbol{v} = 0$, then we have $\mathsf{hw}(\boldsymbol{L} \cdot \boldsymbol{v}) \leq \mathsf{hw}(\boldsymbol{v}) - K$.

Under this new condition, we obtain the following result.

Lemma 8. *Let $t \leq n-1$ and $d = \min(t+1, n-t)$. Let $G_{submult}$ as defined above with n-share inputs. If both matrices $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$ satisfy Condition 3, then $G_{submult}$ is $(d-1)$-NI and $(d-1, 2d-1)$-partial NI.*

Proof. The proof of the Lemma is in fact the same as the proof of Lemma 7. The only difference is that in this lemma, we also cover the special case of an odd value for the number of shares n and $t = \lfloor \frac{n-1}{2} \rfloor = \frac{n-1}{2}$. In the latter case, we consider in the proof up to n probes on the gadget G_{submult}, while in Lemma 7, we could only have up to $n-1$ probes on the gadget. Since Condition 3 covers the case of having up to n probes on G_{submult}, then we can follow the exact same procedure of the proof of Lemma 7 to prove the Lemma by considering the new condition. □

Remark 3. The number of output shares $m = 2n+1$ of G_{submult} satisfies the constraint required by G_{compress} in Algorithm 1 ($m \geq 2n$). We can thus use the compression gadget G_{compress} exactly as described in the algorithm on the input sharing $(c_1, \ldots, c_{2n+1})$, instantiated with the $\mathcal{O}(n \log n)$ refresh gadget from Section 4. Since the multiplication sub-gadget G_{submult} requires $\mathcal{O}(n)$ random values and G_{compress} requires $\mathcal{O}(n \log n)$ random values from the refresh gadget, the overall multiplication gadget G_{mult} also requires a quasi-linear number of random values $\mathcal{O}(n \log n)$.

5.4 Instantiations

We first state the existence of a matrix $\boldsymbol{\gamma}$ which satisfies Condition 3 over any finite field $\mathbb{F}_q$ for q large enough (with $\log(q) = \Omega(n \log n))$[6]. The proof technique follows closely the proof of [8, Theorem 4.5] and makes use of the non-constructive "probabilistic method". Specifically, it states that if one chooses $\boldsymbol{\gamma}$ uniformly at random in $\mathbb{F}_q^{n \log n}$, the probability that the matrix $\boldsymbol{\gamma}$ satisfies Condition 3 is strictly positive, when q is large enough. It is important to note that the proof relies on probability but the existence of a matrix $\boldsymbol{\gamma}$ which satisfies Condition 3 (for q large enough) is guaranteed without any possible error.

Theorem 4. *For any $n \geq 1$, for any prime power q, if $\boldsymbol{\gamma}$ is chosen uniformly in $\mathbb{F}_q^{n \times n}$, then*

$$\Pr[\boldsymbol{\gamma} \text{ satisfies Condition 3}] \geq 1 - 2 \cdot (12n)^n \cdot n \cdot q^{-1} \ .$$

In particular, for any $n \geq 1$, there exists an integer $Q = \mathcal{O}(n)^{n+1}$, such that for any prime power $q \geq Q$, there exists a matrix $\boldsymbol{\gamma} \in {\mathbb{F}_q}^{n \times n}$ satisfying Condition 3.

As when $\boldsymbol{\gamma}$ is uniformly random, so is $\boldsymbol{\delta}$, Theorem 4 immediately follows from the following proposition and the union bound.

Proposition 1. *For any $n \geq 1$, for any prime power q, if $\boldsymbol{\gamma}$ is chosen uniformly in $\mathbb{F}_q^{n \times n}$, then*

$$\Pr[\boldsymbol{\gamma} \text{ satisfies Condition 3}] \geq 1 - (12n)^n \cdot n \cdot q^{-1} \ .$$

In particular, for any $n \geq 1$, there exists an integer $Q = \mathcal{O}(n)^{n+1}$, such that for any prime power $q \geq Q$, there exists a matrix $\boldsymbol{\gamma} \in \mathbb{F}_q^{n \times n}$ satisfying Condition 3.

The proof of this proposition is very technical but follows essentially the proof of the analogous [8, Proposition 4.6]. It is provided in Appendix A.12.

In [8], Belaïd *et al.*. presented examples of matrices which satisfy their condition for 2 shares and 3 shares. Karpman and Roche [19] proposed afterwards new explicit instantiations up to order $n = 6$ over large finite fields and up to $n = 4$ over practically relevant fields such as $\mathbb{F}_{256}$. It is worth mentioning that the matrices proposed in [19] are actually incorrect (due to a sign error) but this can be easily fixed and we check that matrices obtained following [19] also achieve our Condition 3. These matrices for 3, 4 and 5 shares are provided in Appendix A.13.

[6] Such large finite fields may actually be useful to build efficient symmetric primitives (see for instance MiMC [2]).

6 Improved Asymptotic Complexity

In the previous sections, we exhibit the construction of a multiplication gadget G_{mult} which performs a linear number of multiplications between variables, and a quadratic number of multiplications by a constant operations. Using the results of Lemmas 5, 8 and 4, the constructed multiplication gadget is RPE and achieves the maximum amplification order $\lfloor \frac{n+1}{2} \rfloor$ for any number of shares n.

Using the three linear gadgets proposed in Section 4 (G_{add}, G_{copy}, G_{cmult}) with the $\mathcal{O}(n \log n)$ refresh gadgets, and the proposed construction of the multiplication gadget G_{mult}, we get an expanding compiler with a complexity matrix M_{CC} of eigenvalues:

$$(\lambda_1, \lambda_2) = (n, 6n\log(n) - 2n)\ , \quad \lambda_3 = n\ , \quad \lambda_4 = 2n + 1 \quad \text{and} \quad \lambda_5 = n.$$

Hence we have $N_{\max} = 6n\log(n) - 2n = \mathcal{O}(n \log n)$.

Figure 3 illustrates the evolution of the complexity exponent with respect to the number of shares n, for the best construction provided in [10] with quadratic complexity for an expanding compiler (orange curve), and our new construction with quasi-linear complexity (pink curve). While the best construction from [10] yields a complexity in $\mathcal{O}(|C| \cdot \kappa^e)$ for e close to 3 for reasonable numbers of shares, the new expanding compiler quickly achieves a sub-quadratic complexity in the same settings.

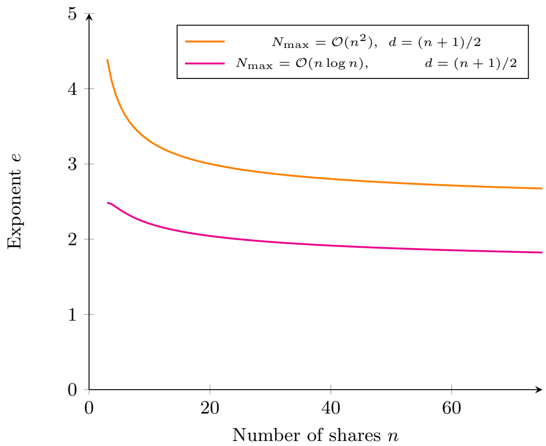


Fig. 3: Evolution of the complexity exponent $e = \log(N_{\max})/\log(d)$ with respect to the number of shares n. The orange curve matches the instantiation from[10] with quadratic asymptotic complexity ($N_{\max} = \mathcal{O}(n^2)$); the pink curve matches the new construction with quasi-linear asymptotic complexity ($N_{\max} = \mathcal{O}(n \log n)$).

7 Conclusion

In this paper we have put forward a dynamic expansion strategy for random probing security which can make the most of different RPE gadgets in terms of tolerated leakage probability and asymptotic complexity. We further introduce new generic constructions of gadgets achieving RPE for any number of shares n. When the base finite field of the circuit meets the requirement of our multiplication gadget, the asymptotic complexity of the obtained expanding compiler becomes arbitrary close to linear, which is optimal.

As for concrete instantiations, our small example on the AES demonstrates the benefits of our dynamic approach. Namely, it provides the best tolerated probability (from the best suited compiler) while optimizing the complexity using higher numbers of shares. Using two compilers with 3 and 5 shares instead of a single one already reduces the complexity by a factor 10.

To go further in the concrete use of our expanding compiler, future works could exhibit explicit constructions of matrices with (quasi)constant field size for our multiplication gadget. One could also investigate further designs of RPE multiplication gadgets with linear number of multiplications for arbitrary fields. Another interesting direction is to optimize the tolerated leakage probability for a set of (possibly inefficient) small gadgets to be used as starting point of the expansion in our dynamic approach before switching to more (asymptotically) efficient RPE gadgets.

References

1. Miklós Ajtai. Secure computation with information leaking to an adversary. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 715–724. ACM Press, June 2011.
2. Martin R. Albrecht, Lorenzo Grassi, Christian Rechberger, Arnab Roy, and Tyge Tiessen. MiMC: Efficient encryption and cryptographic hashing with minimal multiplicative complexity. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 191–219. Springer, Heidelberg, December 2016.
3. Prabhanjan Ananth, Yuval Ishai, and Amit Sahai. Private circuits: A modular approach. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 427–455. Springer, Heidelberg, August 2018.
4. Marcin Andrychowicz, Stefan Dziembowski, and Sebastian Faust. Circuit compilers with $O(1/\log(n))$ leakage rate. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 586–615. Springer, Heidelberg, May 2016.
5. Alberto Battistello, Jean-Sébastien Coron, Emmanuel Prouff, and Rina Zeitoun. Horizontal side-channel attacks and countermeasures on the ISW masking scheme. In Benedikt Gierlichs and Axel Y. Poschmann, editors, *CHES 2016*, volume 9813 of *LNCS*, pages 23–39. Springer, Heidelberg, August 2016.
6. Alberto Battistello, Jean-Sebastien Coron, Emmanuel Prouff, and Rina Zeitoun. Horizontal side-channel attacks and countermeasures on the ISW masking scheme. Cryptology ePrint Archive, Report 2016/540, 2016. `https://eprint.iacr.org/2016/540`.
7. Sonia Belaïd, Fabrice Benhamouda, Alain Passelègue, Emmanuel Prouff, Adrian Thillard, and Damien Vergnaud. Randomness complexity of private circuits for multiplication. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 616–648. Springer, Heidelberg, May 2016.
8. Sonia Belaïd, Fabrice Benhamouda, Alain Passelègue, Emmanuel Prouff, Adrian Thillard, and Damien Vergnaud. Private multiplication over finite fields. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 397–426. Springer, Heidelberg, August 2017.
9. Sonia Belaïd, Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, and Abdul Rahman Taleb. Random probing security: Verification, composition, expansion and new constructions. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 339–368. Springer, Heidelberg, August 2020.
10. Sonia Belaïd, Matthieu Rivain, and Abdul Rahman Taleb. On the power of expansion: More efficient constructions in the random probing model. *IACR Cryptol. ePrint Arch.*, 2021:434, 2021.
11. Claude Carlet, Emmanuel Prouff, Matthieu Rivain, and Thomas Roche. Algebraic decomposition for probing security. Cryptology ePrint Archive, Report 2016/321, 2016. `https://eprint.iacr.org/2016/321`.
12. Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 398–412. Springer, Heidelberg, August 1999.
13. Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, and Thomas Roche. Higher-order side channel security and mask refreshing. In Shiho Moriai, editor, *FSE 2013*, volume 8424 of *LNCS*, pages 410–424. Springer, Heidelberg, March 2014.

14. Jean-Sebastien Coron, Franck Rondepierre, and Rina Zeitoun. High order masking of look-up tables with common shares. Cryptology ePrint Archive, Report 2017/271, 2017. `https://eprint.iacr.org/2017/271`.
15. Alexandre Duc, Stefan Dziembowski, and Sebastian Faust. Unifying leakage models: From probing attacks to noisy leakage. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 423–440. Springer, Heidelberg, May 2014.
16. Louis Goubin and Jacques Patarin. DES and differential power analysis (the "duplication" method). In Çetin Kaya Koç and Christof Paar, editors, *CHES'99*, volume 1717 of *LNCS*, pages 158–172. Springer, Heidelberg, August 1999.
17. Hannes Groß, Ko Stoffelen, Lauren De Meyer, Martin Krenn, and Stefan Mangard. First-order masking with only two random bits. In Begül Bilgin, Svetla Petkova-Nikova, and Vincent Rijmen, editors, *Proceedings of ACM Workshop on Theory of Implementation Security Workshop, TIS@CCS 2019, London, UK, November 11, 2019*, pages 10–23. ACM, 2019.
18. Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 463–481. Springer, Heidelberg, August 2003.
19. Pierre Karpman and Daniel S. Roche. New instantiations of the CRYPTO 2017 masking schemes. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part II*, volume 11273 of *LNCS*, pages 285–314. Springer, Heidelberg, December 2018.
20. Matthieu Rivain and Emmanuel Prouff. Provably secure higher-order masking of AES. In Stefan Mangard and François-Xavier Standaert, editors, *CHES 2010*, volume 6225 of *LNCS*, pages 413–427. Springer, Heidelberg, August 2010.
21. Kai Schramm and Christof Paar. Higher order masking of the AES. In David Pointcheval, editor, *CT-RSA 2006*, volume 3860 of *LNCS*, pages 208–225. Springer, Heidelberg, February 2006.

A Proofs

A.1 Proof of Theorem 1

We consider that we have ℓ compilers $CC_1, \dots, CC_\ell$, and we want to prove the following result:

Lemma 9. *Let $CC_1, \dots, CC_\ell$ RPE compilers with expanding functions $f_1, \dots, f_\ell$. The dynamic expanding compiler for $CC_1, \dots, CC_\ell$,which on input circuit C outputs the compiled circuit $CC_\ell \circ \dots \circ CC_1(C)$, is an RPE compiler with expanding function f such that*

$$f = f_\ell \circ \dots \circ f_1.$$

It can be seen that proving Lemma 9 implies proving the result of Theorem 1. Indeed, we can replace ℓ in the lemma by $k_1 + \ldots + k_\mu$ from Theorem 1 and consider the corresponding compilers with their expansion levels. Thus, we will prove in this appendix Lemma 9 and the proof of the Theorem will follow directly.

To prove the lemma, we first start by introducing some definitions from [9] for random probing expandability of level-ℓ with different sharing orders $n_1, \dots, n_\ell$ gadgets. First, we introduce a generalized definition of *adequate subsets of* $[n_1 \times \ldots \times n_\ell]$ as in [9]. For this, we define recursively a family $S_k \in \mathcal{P}([n_1 \times \ldots \times n_k])$ for $k \leq \ell$, where $\mathcal{P}([n_1 \times \ldots \times n_k])$ denotes the set of all subsets of $[n_1 \times \ldots \times n_k]$, as follows:

$$\begin{aligned} S_1(n,t) =& \{I \in [n],\ |I| \leq t\} \\ S_k(\{n_i\}_{i\in[k]}, \{t_i\}_{i\in[k]}) =& \{(I_1, \dots, I_{n_k}) \in (S_{k-1}(\{n_i\}_{i\in[k-1]}, \{t_i\}_{i\in[k-1]}) \cup [n_1 \times \ldots n_{k-1}])^{n_k}, \\ & I_j \in S_{k-1}\ \forall\ j \in [1, n_k] \text{ except at most } t_k\} \end{aligned}$$

In other words, a subset I belongs to S_k if among the n_k subset parts of I, at most t_k of them are full, while the other ones recursively belong to S_{k-1}. For simplicity, we will sometimes denote S_k without the parameters $(\{n_i\}_{i\in[k]}, \{t_i\}_{i\in[k]})$ which will be implicit in the notation. We will also denote for simplicity $N_i = n_1 \cdot \ldots \cdot n_i$ for $i \in \mathbb{N}$.

Then we recall the generalized definition of RPE with S_k for level-k gadgets.

Definition 12 (Random Probing Expandability with $\{S_k\}_{k\in\mathbb{N}}$). *Let $f : \mathbb{R} \to \mathbb{R}$ and $k \in \mathbb{N}$. An N_k-share gadget $G : \mathbb{K}^{N_k} \times \mathbb{K}^{N_k} \to \mathbb{K}^{N_k}$ is (S_k, f)-*random probing expandable (RPE) *if there exists a deterministic algorithm Sim_1^G and a probabilistic algorithm Sim_2^G such that for every input $(\widehat{x}, \widehat{y}) \in \mathbb{K}^{N_k} \times \mathbb{K}^{N_k}$, for every set $J \in S_k \cup [N_k]$ and for every $p \in [0, 1]$, the random experiment*

$$\begin{aligned} &W \leftarrow \mathsf{LeakingWires}(G, p) \\ &(I_1, I_2, J') \leftarrow \mathsf{Sim}_1^G(W, J) \\ &out \leftarrow \mathsf{Sim}_2^G(W, J', \widehat{x}|_{I_1}, \widehat{y}|_{I_2}) \end{aligned}$$

ensures that

1. *the failure events $\mathcal{F}_1 \equiv \big(I_1 \notin S_k\big)$ and $\mathcal{F}_2 \equiv \big(I_2 \notin S_k\big)$ verify*

$$\Pr(\mathcal{F}_1) = \Pr(\mathcal{F}_2) = \varepsilon \quad \textit{and} \quad \Pr(\mathcal{F}_1 \wedge \mathcal{F}_2) = \varepsilon^2 \tag{19}$$

   with $\varepsilon = f(p)$ (in particular $\mathcal{F}_1$ and $\mathcal{F}_2$ are mutually independent),

2. the set J' is such that $J' = J$ if $J \in S_k$, and $J' = [N_k] \setminus \{j^\star\}$ for some $j^\star \in [N_k]$ otherwise,
3. the output distribution satisfies

$$out \overset{id}{=} \left(\mathsf{AssignWires}(G, W, (\widehat{x}, \widehat{y}))\ ,\ \widehat{z}|_{J'}\right) \tag{20}$$

where $\widehat{z} = G(\widehat{x}, \widehat{y})$.

We are now ready to prove Lemma 9.

Proof (Lemma 9). We will prove the Lemma recursively. In other words, we will suppose that we have RPE compilers $CC_1, \ldots, CC_k$ with expanding functions $f_1, \ldots, f_k$ and (t_i, f_i)-RPE gadgets for each $i \leq k$, and we will prove that the gadgets of the expanding compiler $CC_k \circ \cdots \circ CC_1$ are (S_k, f)-RPE with $f = f_k \circ \cdots \circ f_1$. This will imply that the expanding compiler $CC_k \circ \cdots \circ CC_1$ is RPE with expanding function f.

The base case is one of the theorem hypotheses, namely for $k = 1$, the level-1 gadgets are (t_1, f_1)-RPE, which is equivalent to $(\mathcal{S}_1, f_1)$-RPE. We must then show the induction step: assuming that the level-k gadgets are $(S_k, f_k \circ \cdots \circ f_1)$-RPE, show that the level-$(k+1)$ gadgets are $(S_{k+1}, f_{k+1} \circ \cdots \circ f_1)$-RPE. For the sake of simplicity, we depict our proof by assuming that all the gadgets are 2-to-1 gadget (which is actually not the case for copy gadgets). The proof mechanism for the general case (with 2-to-1 and 1-to-2 gadgets) is strictly similar but heavier on the form. We also denote in the following

- $\varepsilon_k = f_k \circ \cdots f_1(p)$,
- G^{CC_k} to be a gadget of the expanding compiler CC_k,
- $G^{(k)}$ to be the gadget resulting from applying $CC_{k-1} \circ \ldots \circ CC_1(G^{CC_k})$, *i.e.* obtained by replacing each gate of the base gadget G^{CC_k} by the corresponding level-$(k-1)$ gadget $G^{(k-1)}$ and by replacing each wire of the base gadget by N_{k-1} wires carrying a N_{k-1}-linear sharing of the original wire.

In order to show that a gadget $G^{(k+1)}$ is $(S_{k+1}, \varepsilon_{k+1})$-RPE we must construct two simulators $\mathsf{Sim}_1^{G^{(k+1)}}$ and $\mathsf{Sim}_2^{G^{(k+1)}}$ that satisfy the conditions of Definition 12 for the set of subsets S_{k+1}. More precisely, we must construct two simulators $\mathsf{Sim}_1^{G^{(k+1)}}$ and $\mathsf{Sim}_2^{G^{(k+1)}}$ such that for every $(\widehat{x}^*, \widehat{y}^*) \in \mathbb{K}^{N_{k+1}} \times \mathbb{K}^{N_{k+1}}$, and for every set $J^* \in S_{k+1} \cup [N_{k+1}]$, the random experiment

$$\begin{aligned} &W^* \leftarrow \mathsf{LeakingWires}(G^{(k+1)}, p) \\ &(I_1^*, I_2^*, J^{*\prime}) \leftarrow \mathsf{Sim}_1^{G^{(k+1)}}(W^*, J^*) \\ &out \leftarrow \mathsf{Sim}_2^{G^{(k+1)}}(W^*, J^*, \widehat{x}^*|_{I_1^*}, \widehat{y}^*|_{I_2^*}) \end{aligned}$$

ensures that

1. the failure events $\mathcal{F}_1^* \equiv \left(I_1^* \notin S_{k+1}\right)$ and $\mathcal{F}_2^* \equiv \left(I_2^* \notin S_{k+1}\right)$ verify
$$\Pr(\mathcal{F}_1^*) = \Pr(\mathcal{F}_2^*) = \varepsilon_{k+1} \quad \text{and} \quad \Pr(\mathcal{F}_1^* \wedge \mathcal{F}_2^*) = \varepsilon_{k+1}^2 \tag{21}$$
2. the set $J^{*\prime}$ is such that $J^{*\prime} = J^*$ if $J^* \in S_{k+1}$ and $J^{*\prime} = [N_{k+1}] \setminus \{j^\star\}$ otherwise,
3. the output distribution satisfies
$$out \overset{\text{id}}{=} \left(\mathsf{AssignWires}(G^{(k+1)}, W, (\widehat{x}, \widehat{y}))\ ,\ \widehat{z}|_{J^{*\prime}}\right) \tag{22}$$

where $\widehat{z} = G^{(k+1)}(\widehat{x}, \widehat{y})$.

We distinguish two cases: either $J^* \in S_{k+1}$ (normal case), or $J^* = [N_{k+1}]$ (saturated case).

Normal case: $J^* \in S_{k+1}$. By definition of the expanding compiler, we have that a level-$(k+1)$ gadget $G^{(k+1)}$ is obtained by replacing each gate of the base gadget $G^{CC_{k+1}}$ of the compiler CC_{k+1} by the corresponding level-k gadget $G^{(k)}$ and by replacing each wire of the base gadget by N_k wires carrying a N_k-linear sharing of the original wire. In particular $G^{(k+1)}$ has N_{k+1} output wires which can be split in n_{k+1} groups of N_k wires, each group being the output of a different $G^{(k)}$ gadget. We split the set J^* accordingly so that $J^* = J_1^* \cup \dots \cup J_{n_{k+1}}^*$, where each set J_i^* pertains to the ith group of output wires. By definition of S_k, since $J^* \in S_{k+1}$, we must have $J_i^* \in S_k$ for all $1 \leq i \leq n_{k+1}$, except at most t_{k+1} of them for which $J_i^* = [N_k]$. We define J_{base} as the set of indexes i such that $J_i^* \notin S_k$. Therefore we must have $|J_{\text{base}}| \leq t_{k+1}$.

We first describe the simulator $\mathsf{Sim}_1^{G^{(k+1)}}$ that takes the leaking wires W^* and the output wires $J^* \in S_{k+1}$ to be simulated and produce the sets $I_1^* \subseteq [N_{k+1}]$ and $I_2^* \subseteq [N_{k+1}]$ of required inputs. The simulator $\mathsf{Sim}_1^{G^{(k+1)}}$ starts by defining a set W_{base} which is initialized to $\emptyset$; this will correspond to the set of leaking wires for the base gadget $G^{CC_{k+1}}$. Then the simulation goes through all the level-k gadgets composing $G^{(k+1)}$ from bottom to top *i.e.* starting with the level-k gadgets producing the output sharing up to the level-k gadgets processing the input sharings. Let us denote by $\{G_j^{(k)}\}_j$ these level-k gadgets. For each $G_j^{(k)}$, one runs the simulator Sim_1 from the $(S_k, f_k \circ \ldots \circ f_1)$-RPE property on input W_j and J_j defined as follows. The set of leaking wires W_j is defined as the subset of W^* corresponding to the wires of $G_j^{(k)}$. For the gadgets $G_j^{(k)}$ on the bottom layer, the set J_j is set to one of the J_i^* (with indices scaled to range in $[N_k]$). For all the other gadgets $G_j^{(k)}$ (which are not on the bottom layer), the set J is defined as the set I_1 or I_2 output from Sim_1 for the child gadget $G_{j'}^{(k)}$ (for which Sim_1 has already been run).

Whenever a failure event occurs for a $G_j^{(k)}$ gadget, namely when the set I (either I_1 or I_2) output from Sim_1 is such that $I \notin S_k$, we add the index of the wire corresponding to this input in the base gadget $G^{CC_{k+1}}$ to the set W_{base}. Once the Sim_1 simulations have been run for all the $G_j^{(k)}$ gadgets, ending with the top layers, we get the final sets I corresponding to the input shares. Each of these sets corresponds to an N_k-sharing as input of a $G_j^{(k)}$ gadget, which corresponds to a wire as input of the base gadget among the $2 \cdot n_{k+1}$ wires carrying the two input n_{k+1}-sharings of the base gadget. We denote by $I_{1,1}^*, \ldots, I_{1,n_{k+1}}^*$ and $I_{2,1}^*, \ldots, I_{2,n_{k+1}}^*$ the corresponding sets so that defining

$$I_1^* = I_{1,1}^* \cup \ldots \cup I_{1,n_{k+1}}^* \quad \text{and} \quad I_2^* = I_{2,1}^* \cup \ldots \cup I_{2,n_{k+1}}^* \ , \tag{23}$$

the tuple $\widehat{x}^*|_{I_1^*}$ and $\widehat{y}^*|_{I_2^*}$ contains the shares designated by the final I sets.

At the end of the $\mathsf{Sim}_1^{G^{(k+1)}}$ simulation, the set W_{base} contains all the labels of wires in the base gadget $G^{CC_{k+1}}$ for which a failure event has occurred in the simulation of the corresponding $G_j^{(k)}$ gadget. Thanks to the (S_k, ε_k)-RPE property of these gadgets, the failure events happen (mutually independently) with probability ε_k which implies

$$W_{\text{base}} \overset{\text{id}}{=} \mathsf{LeakingWires}(G^{CC_{k+1}}, \varepsilon_k) \tag{24}$$

Recall that $|J_{\text{base}}| \leq t_{k+1}$. We can then run $\mathsf{Sim}_1^{G^{CC_{k+1}}}$ to obtain:

$$(I_{1,\text{base}}, I_{2,\text{base}}) = \mathsf{Sim}_1^{G^{CC_{k+1}}}(W_{\text{base}}, J_{\text{base}}) \ . \tag{25}$$

For all $1 \leq i \leq n_{k+1}$, if $i \in I_{1,\text{base}}$, we force $I^*_{1,i} \leftarrow [N_k]$, so that the corresponding i-th input wire of the base gadget can be computed from the corresponding input wires in $I^*_{1,i}$. The simulator $\mathsf{Sim}_1^{G^{(k+1)}}$ then returns (I^*_1, I^*_2) as output.

The (t_{k+1}, f_{k+1})-RPE property of the base gadget $G^{CC_{k+1}}$ implies that the *base failure events* $|I_{1,\text{base}}| = n_{k+1}$ and $|I_{2,\text{base}}| = n_{k+1}$ are ε_{k+1}-mutually unlikely, where $\varepsilon_{k+1} = f_{k+1}(\varepsilon_k)$. We argue that for all $1 \leq i \leq n_{k+1}$, $I^*_{1,i} \notin S_k \iff i \in I_{1,\text{base}}$. Namely if a failure event has occurred for a set $I^*_{1,i}$ (*i.e.* $I^*_{1,i} \notin S_k$) then we must have $i \in I_{1,\text{base}}$. Indeed, if a failure event has occurred for a set $I^*_{1,i}$ then the label of the ith input wire (for the first sharing) of the base gadget $G^{CC_{k+1}}$ has been added to W_{base} and $\mathsf{Sim}_1^{G^{CC_{k+1}}}$ has no choice but to include this index to the set $I_{1,\text{base}}$ so that $\mathsf{Sim}_2^{G^{CC_{k+1}}}$ can achieve a perfect simulation of the wire assignment (as required by the RPE property of $G^{CC_{k+1}}$). Moreover if $i \in I_{1,\text{base}}$ then by construction we have set $I^*_{1,i} = [N_k]$ and therefore $I^*_{1,i} \notin S_k$. This implies that if $|I_{1,\text{base}}| \leq t_{k+1}$ then $I^*_1 \in S_{k+1}$ (and the same happens for I^*_2 w.r.t. $I_{2,\text{base}}$). We deduce that the failure events $\mathcal{F}^*_1$ and $\mathcal{F}^*_2$ are also ε_{k+1}-mutually unlikely, as required by the $(S_{k+1}, \varepsilon_{k+1})$-RPE property of $G^{(k+1)}$.

We now describe the simulator $\mathsf{Sim}_2^{G^{(k+1)}}$ that takes as input $\widehat{x}^*|_{I^*_1}$ and $\widehat{y}^*|_{I^*_2}$ and produces a perfect simulation of $\big(\mathsf{AssignWires}(G^{(k+1)}, W^*, (\widehat{x}^*, \widehat{y}^*)), \widehat{z}|_{J^*}\big)$ where $\widehat{z} = G^{(k+1)}(\widehat{x}, \widehat{y})$. Let $\widehat{x}^b$ and $\widehat{y}^b$ denote the n_{k+1}-linear sharings obtained by applying the linear decoding to each group of N_k shares in $\widehat{x}^*$ and $\widehat{y}^*$, so that the elements of $\widehat{x}^b$ and $\widehat{y}^b$ correspond to the input wires in the base gadget $G^{CC_{k+1}}$. The assignment expansion property implies that a perfect assignment of the wires of $G^{(k+1)}$ on input $\widehat{x}^*$ and $\widehat{y}^*$ can be derived from an assignement of the wires of the base gadget $G^{CC_{k+1}}$ on input $\widehat{x}^b$ and $\widehat{y}^b$. The simulator makes use of this property by first running

$$out_{\text{base}} \leftarrow \mathsf{Sim}_2^{G^{CC_{k+1}}}(W_{\text{base}}, J_{\text{base}}, \widehat{x}^b|_{I_{1,\text{base}}}, \widehat{y}^b|_{I_{2,\text{base}}}) \ , \tag{26}$$

Note that the input values $\widehat{x}^b|_{I_{1,\text{base}}}$ and $\widehat{y}^b|_{I_{2,\text{base}}}$ can be obtained from the corresponding shares in I^*_1 and I^*_2. Thanks to the (t_{k+1}, f_{k+1})-RPE property of $G^{CC_{k+1}}$ and by construction of $I_{1,\text{base}}$ and $I_{2,\text{base}}$, this outputs a distribution satisfying

$$out_{\text{base}} \overset{\text{id}}{=} \Big(\mathsf{AssignWires}(G^{CC_{k+1}}, W_{\text{base}}, (\widehat{x}^b, \widehat{y}^b)),\ \widehat{z}^b|_{J_{\text{base}}}\Big) \tag{27}$$

The simulator then goes through all the $G_j^{(k)}$ gadgets from input to output and for each of them runs the simulator Sim_2 of the RPE property on inputs W_j, J_j, $\widehat{x}|_{I_1}$ and $\widehat{y}|_{I_2}$ where W_j and J_j are the sets from the first phase of the simulation for the gadget $G_j^{(k)}$, I_1 and I_2 are the corresponding sets produced by the Sim_1 simulator for $G_j^{(k)}$, and $\widehat{x}$ and $\widehat{y}$ are the inputs of $G_j^{(k)}$ in the evaluation of $G^{(k+1)}(\widehat{x}^*, \widehat{y}^*)$. Provided that the partial inputs $\widehat{x}|_{I_1}$ and $\widehat{y}|_{I_2}$ are perfectly simulated, this call to Sim_2 produces a perfect simulation of $\big(\mathsf{AssignWires}(G_j^{(k)}, W_j, (\widehat{x}, \widehat{y}), \widehat{z}|_{J_j}\big)$ where $\widehat{z} = G_j^{(k)}(\widehat{x}, \widehat{y})$. In order to get perfect simulations of the partial inputs $\widehat{x}|_{I_1}$ and $\widehat{y}|_{I_2}$, the simulator proceeds as follows. For the top layer of $G^{(k)}$ gadgets (the ones processing the input shares) the shares $\widehat{x}|_{I_1}$ and $\widehat{y}|_{I_2}$ can directly be taken from the inputs $\widehat{x}^*|_{I^*_1}$ and $\widehat{y}^*|_{I^*_2}$. For the next gadgets the shares $\widehat{x}|_{I_1}$ and $\widehat{y}|_{I_2}$ match the shares $\widehat{z}|_J$ output from the call to Sim_2 for a parent gadget. The only exception occurs in case of a failure event.

In that case the simulation needs the full input $\widehat{x} = (x_1, \ldots, x_{N_k})$ (and/or $\widehat{y} = (y_1, \ldots, y_{N_k})$), while we have set $|I_1| = N_k - 1$ (and/or $|I_2| = (N_k - 1)$ to satisfy the RPE requirements of the parent gadget in the first simulation phase. Nevertheless, for such cases a perfect simulation of the plain value $x = \mathsf{LinDec}(\widehat{x})$ (and/or $y = \mathsf{LinDec}(\widehat{y})$) is included to out_{base} by construction of W_{base}. We can therefore perfectly simulate the missing share from the $N_k - 1$ other shares and the plain value x (or y). We thus get a perfect simulation of $\big(\mathsf{AssignWires}(G_j^{(k)}, W_j, (\widehat{x}, \widehat{y}), \widehat{z}|_{J_j}\big)$ for all the level-k gadgets $G_j^{(k)}$ which gives us a perfect simulation of $\big(\mathsf{AssignWires}(G^{(k+1)}, W^*, (\widehat{x}^*, \widehat{y}^*)), \widehat{z}|_{J^*}\big)$.

Saturated case: $\boldsymbol{J^* = [N_{k+1}]}$. The saturated case proceeds similarly. The difference is that we must simulate all N_{k+1} output shares of the level-$(k+1)$ gadget, except for one share index j^* that can be chosen by the simulator.

The simulator $\mathsf{Sim}_1^{G^{(k+1)}}$ is defined as previously. Since $J^* = [N_{k+1}]$, we must define $J_{\text{base}} = [1, n_{k+1}]$. Moreover we have $J_i^* = [N_k]$ for all $1 \le i \le n_{k+1}$. This implies that for the gadgets $G_j^{(k)}$ on the output layer, the sets J_j are all equal to $[N_k]$ as well. The set W_{base} is defined as previously, and the simulator $\mathsf{Sim}_1^{G^{(k+1)}}$ returns (I_1^*, I_2^*) as previously. The failure events $\mathcal{F}_1^*$ and $\mathcal{F}_2^*$ are still ε_{k+1}-mutually unlikely, as required by the $(S_{k+1}, \varepsilon_{k+1})$-RPE property of $G^{(k+1)}$.

The simulator $\mathsf{Sim}_2^{G^{(k+1)}}$ is defined as previously. In particular, from the running of the base gadget simulator $\mathsf{Sim}_2^{G^{CC_{k+1}}}$, we obtain a perfect simulation of the output wires $\widehat{z}^b|_{J'_{\text{base}}}$ for some J'_{base} with $|J'_{\text{base}}| = n_{k+1} - 1$. Combined with the perfect simulation of the output wires corresponding to the output sets J'_j from the gadgets $G_j^{(k)}$ on the output layer, with $|J'_j| = N_k - 1$, we obtain a subset J' of output wires for our level-$(k+1)$ gadget with $|J'| = N_{k+1} - 1$ as required. Eventually this gives us a perfect simulation of $\big(\mathsf{AssignWires}(G^{(k+1)}, W^*, (\widehat{x}^*, \widehat{y}^*)), \widehat{z}|_{J'}\big)$. This terminates the proof of Lemma 9. As stated earlier, proving Lemma 9 implies proving Theorem 1. Thus, this also terminates the proof for the theorem. □

A.2 Proof of Theorem 2

Proof. Let $\{\mathsf{CC}_i\}_i$ be a family of circuit compilers with complexity matrices $\{M_{\mathsf{CC}_i}\}_i$. Given a circuit C with its complexity vector N_C as described in Section 2.4, it can be verified that the complexity of the compiled circuit $\widehat{C} = \mathsf{CC}_\mu^{k_\mu} \circ \cdots\cdots \circ \mathsf{CC}_1^{k_1}(C)$ satisfies

$$N_{\widehat{C}} = M_{\mathsf{CC}_\mu}^{k_\mu} \cdot \ldots \cdot M_{\mathsf{CC}_1}^{k_1} \cdot N_C$$

If we denote $M_{\mathsf{CC}_i} = Q_i \cdot \Lambda_i \cdot Q_i^{-1}$ to be the eigen decomposition of the matrix M_{CC_i}, then we get

$$N_{\widehat{C}} = Q_\mu \cdot \Lambda_\mu^{k_\mu} \cdot Q_\mu^{-1} \cdot \ldots \cdot Q_1 \cdot \Lambda_1^{k_1} \cdot Q_1^{-1} \cdot N_C \tag{28}$$

We consider in the theorem that the expansion levels $\{k_i\}_i$ are the main parameters. We can also see from (28) that the complexity of the compiled circuit is expressed in terms of the eigen matrices to the powers k_i as $\Lambda_i^{k_i}$. The parameters $\{k_i\}_i$ do not affect the matrices $\{Q_i, Q_i^{-1}\}_i$. Then, if we denote $\lambda_i := \max\ \mathsf{eigenvalues}(M_{\mathsf{CC}_i})$ *i.e.* the maximum of the eigenvalues in Λ_i, then we get that in terms of the parameters $\{k_i\}_i$, the complexity of the compiled circuit $\widehat{C}$ can be expressed as

$$N_{\widehat{C}} = \mathcal{O}\Big(|\lambda_\mu|^{k_\mu} \cdot \ldots \cdot |\lambda_1|^{k_1}\Big) \cdot N_C$$

which gives

$$|\widehat{C}| = |C| \cdot \mathcal{O}\Big(\prod_{i=1}^{\mu} |\lambda_i|^{k_i}\Big)$$

which concludes the proof of Theorem 2. □

A.3 Proof of Theorem 3

To prove Theorem 3, we introduce the following lemma.

Lemma 10. *Let* CC_i *be an RPE circuit compiler of amplification order* d_i *and complexity* s_i*. For any probability*

$$p \leq \frac{1}{2} \cdot \frac{d_i + 1}{s_i - d_i} \tag{29}$$

the expanding function f_i *of* CC_i *is upper bounded by*

$$f_i(p) \leq 2\binom{s_i}{d_i} p^{d_i} \ . \tag{30}$$

Proof (Lemma 10). Let us first recall the following general bound on f_i:

$$f_i(p) \leq \sum_{j=d_i}^{s_i} \binom{s_i}{j} p^j \ , \tag{31}$$

for any $p \in [0, 1)$. From (29), for any $j \in [s_i]$, we get:

$$\binom{s_i}{j+1} p^{j+1} \leq \frac{1}{2} \binom{s_i}{j} p^j$$

which gives

$$f_i(p) \leq \sum_{j=d_i}^{s_i} \binom{s_i}{d_i} \left(\frac{1}{2}\right)^{j-d_i} p^{d_i} = \binom{s_i}{d_i} p^{d_i} \sum_{j=0}^{s_i - d_i} \left(\frac{1}{2}\right)^j \leq 2\binom{s_i}{d_i} p^{d_i} \ .$$

□

Proof (Theorem 3). We show that for every p satisfying

$$p < \frac{1}{\mathrm{e}} \left(\frac{1}{2\,\mathrm{e}}\right)^{\frac{1}{d_i - 1}} \left(\frac{d_i}{s_i}\right)^{1 + \frac{1}{d_i - 1}} \tag{32}$$

we have $f_i(p) < p$. Let us define

$$\bar{f}_i : p \mapsto 2\binom{s_i}{d_i} p^{d_i} \ .$$

(the upper bound on f_i from Lemma 10). The equation $\bar{f}_i(\gamma) = \gamma$ has the following solution

$$\gamma = \left(\frac{1}{2\binom{s_i}{d_i}}\right)^{\frac{1}{d_i - 1}}$$

which, from

$$\binom{s_i}{d_i} \leq \left(\frac{s_i \cdot \exp(1)}{d_i}\right)^{d_i} \ ,$$

further satisfies

$$\gamma \geq \frac{1}{\mathrm{e}} \left(\frac{1}{2\,\mathrm{e}}\right)^{\frac{1}{d_i - 1}} \left(\frac{d_i}{s_i}\right)^{1 + \frac{1}{d_i - 1}}$$

We deduce that (32) implies $p < \gamma$ which further implies $\bar{f}_i(p) < p$. Moreover (32) implies

$$p < \frac{1}{2}\left(\frac{d_i}{s_i}\right)^{1+\frac{1}{d_i-1}} < \frac{1}{2}\cdot\frac{d_i}{s_i} < \frac{1}{2}\cdot\frac{d_i+1}{s_i-d_i} \ ,$$

which, by Lemma 10, further implies $f_i(p) \leq \bar{f}_i(p)$. We hence deduce that (32) implies $f_i(p) < p$ which concludes the proof. $\square$

A.4 Proof of Corollary 1

Proof (Corollary 1). For any function $f(p) = c \cdot p^d$, we have

$$f^{(k)}(p) = c^{(d^{k-1}+d^{k-2}+\dots+1)} \cdot p^{d^k} \leq c^{\left(1+\frac{1}{d-1}\right)d^{k-1}} p^{d^k}.$$

When $c_i = 2\binom{s_i}{d_i}$, Equation (11) from Theorem 3 gives the first and the second inequalities. $\square$

A.5 Proof of Lemma 6

Proof. G_{cmult} has the exact same wires as the underlying G_{refresh} except for the extra input wires $\{a_1, \dots, a_n\}$ (the wires multiplied by the constant i.e $\{c \cdot a_1, \dots, c \cdot a_n\}$ are the input wires to G_{refresh}). So to simulate probes on G_{cmult}, we use the simulator of G_{refresh}. Each probe which is in the set $\{a_1, \dots, a_n\}$ will be replaced by the corresponding input share multiplied by the constant c, in the set of probes on G_{refresh}, which would lead to a probe on an input share of G_{refresh} of the form $c \cdot a_i$. It is clear that if we can perfectly simulate $c \cdot a_i$ in G_{refresh}, then we can perfectly simulate the input share a_i in G_{cmult}. Thus any set of probes on G_{cmult} is simulated using the simulator of G_{refresh} with the exact same number of probes. Hence, if G_{refresh} is (t, f)-(T)RPE n-share refresh gadget of amplification order d, then the gadget G_{cmult} is also (t, f')-(T)RPE of amplification order d. This concludes the proof. $\square$

A.6 Proof of Lemma 2

Proof. Let G_{refresh} be a (t, f)-TRPE refresh gadget for any $t \leq n-1$ with amplification order $d \geq \min(t+1, n-t)$ and which satisfies Definition 9. We will prove that the construction of G_{add} using G_{refresh} described in Section 4 is (t, f)-TRPE for any $t \leq n-1$ of amplification order $\min(t+1, n-t)$. This amounts to proving that:

1. Any set of leaking wires W such that $|W| < \min(t+1, n-t)$ can be simulated together with any set of outputs wires $J \subseteq [n]$ from sets of input wires I_1 on a and I_2 on b such that $|I_1| \leq \min(t, |W|)$ and $|I_2| \leq \min(t, |W|)$.
2. Any set of leaking wires such that $\min(t+1, n-t) \leq |W| < 2\min(t+1, n-t)$ can be simulated together with any set of outputs wires $J \subseteq [n]$ from sets of input wires I_1, I_2 such that $|I_1| \leq \min(t, |W|)$ or $|I_2| \leq \min(t, |W|)$ (because of the double failure, *i.e* failure on both inputs).

Indeed, this amplification order being the maximum one achievable by 2-input addition gadgets, it would conclude the proof.

We will denote $(e_1, \dots, e_n) = G_{\text{refresh}}(a_1, \dots, a_n)$ and $(f_1, \dots, f_n) = G_{\text{refresh}}(b_1, \dots, b_n)$. Then the gadget G_{add} consists in the sharewise addition $(e_1 + f_1, \dots, e_n + f_n)$ as described in Section 4.

We proceed by building the necessary simulators for G_{add} from the simulators that already exist for G_{refresh}. Concretely, we split each set W of leaking wires, into four subsets $W = W_1^r \cup W_1^a \cup W_2^r \cup W_2^a$ where W_1^r (resp. W_2^r) is the set of leaking wires during the computation of $G_{\text{refresh}}(a_1, \ldots, a_n)$ (resp. $G_{\text{refresh}}(b_1, \ldots, b_n)$), and W_1^a (resp. W_2^a) is the set of leaking wires of $(e_1, \ldots, e_n)$ (resp. $(f_1, \ldots, f_n)$). We can see that $W_1^r \cup W_1^a$ (resp. $W_2^r \cup W_2^a$) contains only leaking wires during the computation of $G_{\text{refresh}}(a_1, \ldots, a_n)$ (resp. $G_{\text{refresh}}(b_1, \ldots, b_n)$). We now demonstrate how we can simulate W when the output set J is of size less that t ((T)RPE1) and when it is of size strictly more than t ((T)RPE2).

- if $|J| \leq t$ ((T)RPE1): we prove both properties 1 and 2:
  1. we assume that $|W| < \min(t+1, n-t)$. We construct a new set of probes on $(e_1, \ldots, e_n)$ that we denote J_e such that $J_e = W_1^a \cup \{e_i \mid i \in J\}$. Similarly, we construct the set of probes on $(f_1, \ldots, f_n)$, $J_f = W_2^a \cup \{f_i \mid i \in J\}$. It is clear that if we can perfectly simulate W_1^r, W_2^r, J_e and J_f, then we can perfectly simulate W, and J (for each $i \in J$, we can perfectly simulate e_i in J_e and f_i in J_f so we can perfectly simulate $e_i + f_i$). We denote $|W_1^a| = m$ and $|W_2^a| = m'$. We have
  $$|W_1^r| \leq \min(t+1, n-t) - 1 - m \quad , \quad |J_e| \leq t + m$$
  and
  $$|W_2^r| \leq \min(t+1, n-t) - 1 - m', \quad , \quad |J_f| \leq t + m'$$
  From the (t, f)-TRPE property of G_{refresh} for any $t \leq n-1$ and specifically for $t' = t + m$ with amplification order at least $d' = \min(t+1+m, n-t-m)$, and since $|W_1^r| \leq \min(t+1, n-t) - 1 - m \leq d' - 1$, then there exists an input set of shares of a I_1 such that $|I_1| \leq \min(t+m, |W_1^r|) = |W_1^r| \leq |W|$ and I_1 perfectly simulates W_1^r and J_e.
  Similarly, there exists an input set of shares of b I_2 such that $|I_2| \leq \min(t+m', |W_2^r|) = |W_2^r| \leq |W|$ and I_2 perfectly simulates W_2^r and J_f.
  From these definitions, I_1 and I_2 together perfectly simulate W and J and are both of size less than $\min(t, |W|)$, which proves the first property in this scenario.
  2. we now assume that $\min(t+1, n-t) \leq |W| < 2\min(t+1, n-t)$. Without loss of generality, let us consider that $|W_1^r \cup W_1^a| < \min(t+1, n-t) \leq t$ (the proof is similar in the opposite scenario). As in the first property, we construct a new set of probes on $(e_1, \ldots, e_n)$ that we denote J_e such that $J_e = W_1^a \cup \{e_i \mid i \in J\}$. We fix the set of input shares I_2 on b as $I_2 = [n]$, so we can perfectly simulate all probes in W_2^r and W_2^a using the full input b. Next, we need to prove that we can perfectly simulate all probes in W_1^r and J_e similarly as before. We denote $|W_1^a| = m$. We have
  $$|W_1^r| \leq \min(t+1, n-t) - 1 - m \quad , \quad |J_e| \leq t + m$$
  From the (t, f)-TRPE property of G_{refresh} for any $t \leq n-1$ and specifically for $t' = t + m$ with amplification order at least $d' = \min(t+1+m, n-t-m)$, and since $|W_1^r| \leq \min(t+1, n-t) - 1 - m \leq d' - 1$, then there exists an input set of shares of a I_1 such that $|I_1| \leq \min(t+m, |W_1^r|) = |W_1^r| \leq |W|$ and I_1 perfectly simulates W_1^r and J_e.
  From these definitions, I_1 and I_2 together perfectly simulate W and J (J is simulated by perfectly simulating each $i \in J$ by using e_i in J_e and simulating f_i using the full input b), and we only have a failure on at most one of the inputs (b in this case). This concludes the proof for the second property.

At this point, we proved that G_{add} achieves an amplification order greater than or equal to $\min(t+1, n-t)$ for TRPE1. Since this amplification order is the maximum achievable by 2-input addition gadgets, then G_{add} achieves an amplification order exactly equal to $\min(t+1, n-t)$.

– if $|J| > t$ ((T)RPE2): we prove both properties 1 and 2:

  1. we assume that $|W| < \min(t+1, n-t)$. As before, we split W as $W = W_1^r \cup W_1^a \cup W_2^r \cup W_2^a$. We consider $J' = \{i \mid e_i \in W_1^a\} \cup \{i \mid f_i \in W_2^a\}$ so we have $|J'| \leq |W_1^a| + |W_2^a|$. We also construct the set W^r which contains the set of leaking wires on the first instance of G_{refresh} (on input a) in W_1^r, and all the wires that are leaking within the second instance of G_{refresh} in W_2^r. Hence, we have that $|W^r| \leq |W_1^r \cup W_2^r| < \min(t+1, n-t)$. Hence, we have $|W^r| + |J'| \leq \min(t+1, n-t) \leq n-1$, so by Definition 9 satisfied by G_{refresh}, there exists a set of output shares indices J such that $J' \subseteq J$ and $|J| = n-1$ such that W^r and J can be perfectly simulated from a set of input shares indices I such that $|I| \leq |W^r| + |J'|$. Thus, we can fix I_1 on a and I_2 on b such that $I_1 = I_2 = I$ and we fix the set of $n-1$ output shares indices on G_{add} as the same indices in J. Hence, we can perfectly simulate all wires in W^r and J, so we can perfectly simulate all wires in W_1^r and W_2^r and W_1^a and W_2^a as well as $n-1$ output shares of G_{add} using I_1 and I_2 such that $|I_1| = |I_2| \leq |W^r| + |J'| \leq |W| = \min(t, |W|)$. That concludes the proof for the first property.
  2. we now assume that $\min(t+1, n-t) \leq |W| < 2\min(t+1, n-t)$. Without loss of generality, let us consider that $|W_1^r \cup W_1^a| < \min(t+1, n-t)$ (the proof is similar in the opposite scenario).
     We fix $I_2 = [n]$ on input b, which allows us to perfectly simulate all wires and output shares on G_{refresh} instance with input sharing $(b_1, \ldots, b_n)$, including W_2^a and W_2^r. Next, we set $J' = \{i \mid e_i \in W_1^a\}$. Since $|W_1^r| + |J'| \leq n-1$, by Definition 9 satisfied by G_{refresh}, there exists a set of output shares indices J such that $J' \subseteq J$ and $|J| = n-1$ such that W_1^r and J can be perfectly simulated from a set of input shares indices I_1 on a such that $|I_1| \leq |W_1^r| + |J'| \leq |W_1^r| + |W_1^a| \leq |W|$. Thus, we can fix the set of $n-1$ output shares indices on G_{add} as the same indices in J. We can perfectly simulate all output shares indexed in J since for each $i \in J$, we can perfectly simulate e_i using I_1 and f_i using the full input b in I_2, so we can perfectly simulate $e_i + f_i$. Hence, we can perfectly simulate all wires in W as well as $n-1$ output shares of G_{add} using I_1 and I_2 such that $|I_1| \leq |W_1^r| + |W_1^a| \leq \min(t, |W|)$ and with a failure on input b with $I_2 = [n]$. That concludes the proof for the second property.

We thus proved that G_{add} achieves an amplification order greater than or equal to $\min(t+1, n-t)$ for TRPE2. Since $\min(t+1, n-t)$ is the maximum order achievable for TRPE2 for a 2-input gadget, then G_{add} achieves exactly the order $\min(t+1, n-t)$.

Since G_{add} has an amplification order equal to $\min(t+1, n-t)$ for TRPE1 and TRPE2, then G_{add} is a (t, f')-TRPE addition gadget for some function f' of amplification order $\min(t+1, n-t)$, which concludes the proof. □

A.7 Algorithm for the $\mathcal{O}(n \log n)$ Refresh Gadget

A.8 Proof of Lemma 3

Proof. We will prove that the gadget from Algorithm 3 is (t, f)-TRPE for any $t \leq n-1$ of amplification order $d \geq \min(t+1, n-t)$. For this, we will prove both properties TRPE1 and TRPE2.

Algorithm 3: QuasiLinearRefresh

```
Input  : (a_1, ..., a_n) input sharing
Output: (d_1, ..., d_n) such that d_1 + ··· + d_n = a_1 + ··· + a_n
if n = 1 then return a_1;
if n = 2 then
    r ← $;
    return (a_1 + r, a_2 − r);
end
for i ← 1 to ⌊n/2⌋ do
    r ← $;
    b_i ← a_i + r;
    b_{⌊n/2⌋+i} ← a_{⌊n/2⌋+i} − r;
end
if n mod 2 = 1 then b_n ← a_n;
(c_1, ..., c_{⌊n/2⌋}) ← QuasiLinearRefresh(b_1, ..., b_{⌊n/2⌋});
(c_{⌊n/2⌋+1}, ..., c_n) ← QuasiLinearRefresh(b_{⌊n/2⌋+1}, ..., b_n);
for i ← 1 to ⌊n/2⌋ do
    r ← $;
    d_i ← c_i + r;
    d_{⌊n/2⌋+i} ← c_{⌊n/2⌋+i} − r;
end
if n mod 2 = 1 then d_n ← c_n;
return (d_1, ..., d_n);
```

Proof for TRPE1
The gadget is proven to be $(n-1)$-SNI in [5], thus it is (t, f)-TRPE1 of amplification order $d \geq \min(t+1, n-t)$ thanks to Lemma 6 from [10]. Note that we can find failure sets of wires of size $t+1$ which require the knowledge of $t+1$ input shares (simply consider the leaking wires $\{a_1, \ldots, a_{t+1}\}$ on input a for instance), so $d \leq t+1$.

Proof for $(n-1)$-STRPE2 (which implies TRPE2)
We will first start by recalling the result of Lemma 5 in [5] which will be useful for our proof.

Lemma 5 from [5]. Let $a_1, a_2 \in \mathbb{K}$ be inputs, and let $r \xleftarrow{\$} \mathbb{K}$. Let V be a subset of the variables $\{a_1, a_2, r\}$ and $O \in \{\emptyset, \{a_1 + r\}\}$. Then the variables in $V \cup O \cup \{a2 - r\}$ can be perfectly simulated from $I \subset \{a_1, a_2\}$, with $|I| \leq |V| + 2 \cdot |O|$.

Proof of Lemma 5 from [5]. If $|O| = 1$ or $|V| \geq 2$, we can take $I = \{a_1, a_2\}$. If $|O| = 0$ and $|V| = 0$, we can simulate $a_2 - r$ with a random value. If $|O| = 0$ and $|V| = 1$, if $V = \{a_1\}$ we let $I = \{a_1\}$ and we can again simulate $a_2 - r$ with a random value; if $V = \{r\}$ or $V = \{a_2\}$ then we let $I = \{a_2\}$. □

We are now ready to prove our main result. For TRPE2, we will prove the slightly stronger property $(n-1)$-STRPE2. We can clearly see that $(n-1)$-STRPE2 implies TRPE2 of amplification order $d = t+1$ as shown in Remark 1. We will prove $(n-1)$-STRPE2 by recurrence on the number of shares $n \geq 2$.

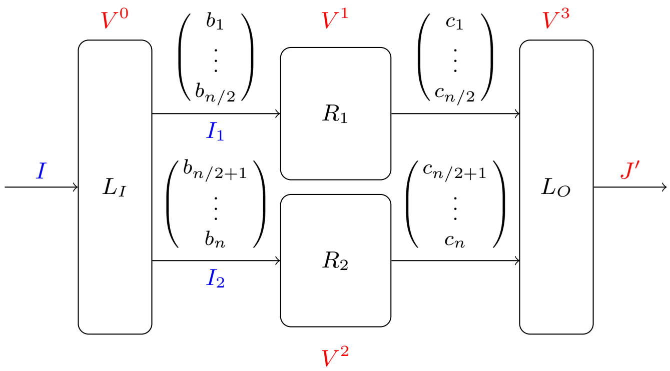


Fig. 4: $\mathcal{O}(n \log n)$ refresh gadget from [5]

The gadget in the base case ($n = 2$) gives the following output sharing:

$$d_1 \leftarrow a_1 + r$$
$$d_2 \leftarrow a_2 + r$$

The proof in this case is easy. Mainly, if $J' = \emptyset$, it is easy to see that we can choose J of size 1 such that we can perfectly simulate W and J from a set of input shares I on a such that $|I| \leq |W| \leq 1$. Otherwise, if $|J'| = 1$, then $|W| = 0$, and we choose $J = J'$ and in this case we have $|I| = 0$, since we can perfectly simulate any of the output shares alone by simply generating a freshly random value. This concludes the proof for the base case.

Next we suppose that the gadget is $(n' - 1)$-STRPE2 for any number of shares $n' < n$, and we prove the property for n shares.

To prove this, we split the gadget into four subgadgets as in Figure 4, where gadget L_I corresponds to the first loop in Algorithm 3 which adds $\lfloor n/2 \rfloor$ random values to the sharing, R_1 and R_2 gadgets correspond to the two recursive calls respectively, and L_O gadget corresponds to the second loop which also add $\lfloor n/2 \rfloor$ random values to the output sharing. We split any set of probes W on G_{refresh} into $W = V^0 \cup V^1 \cup V^2 \cup V^3$ on each of the subgadgets L_I, R_1, R_2 and L_O respectively. The gadget R_1 is a $\left\lfloor \frac{n}{2} \right\rfloor$-share gadget while R_2 is $\left\lceil \frac{n}{2} \right\rceil$-share gadget. We consider that there are no probes on the output shares of R_1 and R_2 as they can be probed through V^3. Similarly, we consider no output probes on L_I, since they can be probed through V^1 and V^2.

Let W bet the set of probes on G_{refresh} and J' be the set of output shares indices such that $|W| + |J'| \leq n - 1$. We will construct the sets J'_1 and J'_2 for output shares of the gadgets R_1 and R_2 as follows:

- for each $i \in J' \cap \left[\left\lfloor \frac{n}{2} \right\rfloor \right]$, add i to J'_1
- for each $i \in J' \cap \left[\left\lfloor \frac{n}{2} \right\rfloor + 1 : n \right]$, add i to J'_2
- for each $i \in \left[\left\lfloor \frac{n}{2} \right\rfloor \right]$ such that the input probe c_i to L_O is probed in V^3, add i to J'_1
- for each $i \in \left[\left\lfloor \frac{n}{2} \right\rfloor + 1 : n \right]$ such that the input probe c_i to L_O is probed in V^3, add i to J'_2

It can be seen that if we can perfectly simulate J'_1 and J'_2, then we can perfectly simulate J' and all probes in V^3 (V^3 is composed of input probes c_i and random variables r_i, since probes of the form $c_i + r_j$ are probed in J'). Observe that we also have $|J'_1| + |J'_2| \leq |V^3| + |J'|$.
In order for the recurrence hypothesis to hold, we need the following condition to hold for the gadget R_1:

$$|V^1| + |J'_1| \leq \left\lfloor \frac{n}{2} \right\rfloor - 1 \tag{33}$$

and the following for the gadget R_2:

$$|V^2| + |J'_2| \leq \left\lceil \frac{n}{2} \right\rceil - 1 \tag{34}$$

We consider three cases based on the sizes of the sets of probes:

– $|\mathbf{V^2}| + |\mathbf{J'_2}| \geq \left\lceil \frac{\mathbf{n}}{\mathbf{2}} \right\rceil$. Then we must have $|V^1|+|J'_1| \leq \left\lfloor \frac{n}{2} \right\rfloor -1$, because we have that $|W|+|J'| \leq n-1$ and $|J'_1| + |J'_2| \leq |V^3| + |J'|$.
 Since (33) holds, by the recurrence hypothesis on R_1, we can choose a set J_1 of size $\left\lfloor \frac{n}{2} \right\rfloor - 1$ such that $J'_1 \subseteq J_1$ and we can perfectly simulate J_1 and V^1 from a set of input shares I_1 on $(b_1, \ldots, b_{\lfloor n/2 \rfloor})$ such that $|I_1| \leq |V^1| + |J'_1|$. Since (34) does not hold for R_2, we can set $J_2 = \left[\left\lceil \frac{n}{2} \right\rceil : n \right]$ and $I_2 = \left[\left\lceil \frac{n}{2} \right\rceil : n \right]$, and finally set $J = J_1 \cup J_2$ of $n-1$ output shares on G_{refresh}. We can see that $J'_2 \subseteq J_2$ and J_2 and V^2 can be perfectly simulated from I_2 trivially (full input).
 Next, we show how to perfectly simulate the sets I_1, I_2 on intermediate variable b, and V^0. In fact, thanks to the properties of the L_I gadget, we can apply Lemma 5 from [5] for all $1 \leq i \leq \lfloor n/2 \rfloor$ on each set of intermediate variables $\{a_i, a_{\lfloor n/2 \rfloor + i}, r_i\}$ and output variable $b_i = a_i + r_i$, where all output variables $b_{\lfloor n/2 \rfloor + i} = a_{\lfloor n/2 \rfloor + i} - r_i$ must be simulated (since we fixed $I_2 = \left[\left\lceil \frac{n}{2} \right\rceil : n \right]$), and by summing the inequalities, we construct $I \subset [n]$ on n-share input a to perfectly simulate I_1, I_2 on intermediate variable b, and V^0 such that

$$|I| \leq |V^0| + 2|I_1| + (n \bmod 2) \leq |V^0| + 2(|V^1| + |J'_1|) + (n \mod 2)$$

 where $(n \bmod 2)$ comes from the fact that we need to perfectly simulate all shares of $(b_{\lceil n/2 \rceil}, \ldots, b_n)$ and if $n \mod 2 = 1$, then $b_n = a_n$ by construction of the gadget L_I.
 From (33) which holds in this case, observe that we have

$$|V^1| + |J'_1| + (n \bmod 2) \leq \left\lfloor \frac{n}{2} \right\rfloor \leq \left\lceil \frac{n}{2} \right\rceil \leq |V^2| + |J'_2|,$$

 then we have

$$|I| \leq |V^0| + 2(|V^1| + |J'_1|) + (n \mod 2) \leq |V^0| + |V^1| + |J'_1| + |V^2| + |J'_2|$$

 which gives

$$|I| \leq |W| + |J'|$$

 and using the input shares in I, we can perfectly simulate probes in V^0, I_1 and I_2, and using I_1 and I_2 we proved that we can perfectly simulate probes in V^1, V^2, J_1 and J_2, and so we can also perfectly simulate the chosen set of $n-1$ output shares J and probes in V^3. So we can perfectly simulate all internal probes plus the chosen set J of $n-1$ output shares from I. This proves the recurrence step in this case.

– $|\mathbf{V^1}| + |\mathbf{J_1'}| \geq \left\lfloor \frac{\mathbf{n}}{\mathbf{2}} \right\rfloor$. Then we must have $|V^2|+|J_2'| \leq \left\lceil \frac{n}{2} \right\rceil -1$, because we have that $|W|+|J'| \leq n-1$ and $|J_1'| + |J_2'| \leq |V^3| + |J'|$.
Since (34) holds, by the recurrence hypothesis on R_2, we can choose a set J_2 of size $\left\lceil \frac{n}{2} \right\rceil - 1$ such that $J_2' \subseteq J_2$ and we can perfectly simulate J_2 and V^2 from a set of input shares I_2 on $(b_{\lceil n/2 \rceil}, \ldots, b_n)$ such that $|I_2| \leq |V^2| + |J_2'|$. Since (33) does not hold for R_1, we can set $J_1 = [\, \left\lfloor \frac{n}{2} \right\rfloor \,]$ and $I_1 = [\, \left\lfloor \frac{n}{2} \right\rfloor \,]$, and finally set $J = J_1 \cup J_2$ of $n-1$ output shares on G_{refresh}. We can see that $J_1' \subseteq J_1$ and J_1 and V^1 can be perfectly simulated from I_1 trivially (full input).
Next, we show how to perfectly simulate the sets I_1, I_2 on intermediate variable b, and V^0. In fact, thanks to the properties of the L_I gadget, we can apply Lemma 5 from [5] for all $1 \leq i \leq \lfloor n/2 \rfloor$ on each set of intermediate variables $\{a_i, a_{\lfloor n/2 \rfloor + i}, r_i\}$ and output variable $b_{\lfloor n/2 \rfloor + i} = a_{\lfloor n/2 \rfloor + i} - r_i$, where all output variables $b_i = a_i + r_i$ must be simulated (since we fixed $I_1 = [\, \left\lfloor \frac{n}{2} \right\rfloor \,]$), and by summing the inequalities, we construct $I \subset [n]$ on n-share input a to perfectly simulate I_1, I_2 on intermediate variable b, and V^0 such that

$$|I| \leq |V^0| + 2|I_2| \leq |V^0| + 2(|V^2| + |J_2'|)$$

(in this case, we don't have the term ($n \mod 2$) anymore because we do not need the full input sharing $(b_{\lceil n/2 \rceil}, \ldots, b_n)$ for the simulation as before). Since (34) holds and (33) does not hold, we observe that

$$|V^2| + |J_2'| \leq \left\lceil \frac{n}{2} \right\rceil - 1 \leq \left\lfloor \frac{n}{2} \right\rfloor \leq |V^1| + |J_1'|$$

so we get

$$|I| \leq |V^0| + 2(|V^2| + |J_2'|) \leq |V^0| + |V^2| + |J_2'| + |V^1| + |J_1'|$$

which gives

$$|I| \leq |W| + |J'|$$

and using the input shares in I, we can perfectly simulate probes in V^0, I_1 and I_2, and using I_1 and I_2 we proved that we can perfectly simulate probes in V^1, V^2, J_1 and J_2, and so we can also perfectly simulate the chosen set of $n-1$ output shares J and probes in V^3. So we can perfectly simulate all internal probes plus the chosen set J of $n-1$ output shares from I. This proves the recurrence step in this case.

– $|\mathbf{V^1}| + |\mathbf{J_1'}| \leq \left\lfloor \frac{\mathbf{n}}{\mathbf{2}} \right\rfloor - \mathbf{1}$ and $|\mathbf{V^2}| + |\mathbf{J_2'}| \leq \left\lceil \frac{\mathbf{n}}{\mathbf{2}} \right\rceil - \mathbf{1}$. This case can be treated in the exact same way as the above cases. Namely, if we have $|V^1| + |J_1'| + (n \mod 2) \leq |V^2| + |J_2'|$, then we can consider the first case and treat it in the same way (by appyling the recursion hypothesis on gadget R_1 and setting $J_2 = [\, \left\lceil \frac{n}{2} \right\rceil : n]$ and $I_2 = [\, \left\lceil \frac{n}{2} \right\rceil : n]$.
Otherwise, if we have $|V^2|+|J_2'| \leq |V^1|+|J_1'|+(n \mod 2)$, then we can consider the second case and treat it in the same way (by appyling the recursion hypothesis on gadget R_2 and setting $J_1 = [\, \left\lfloor \frac{n}{2} \right\rfloor \,]$ and $I_1 = [\, \left\lfloor \frac{n}{2} \right\rfloor \,]$.
This also concludes the proof in this case.

By treating all possible cases on the probed wires, we conclude the recursive proof. This proves that for any n shares such that $|W| + |J'| \leq n-1$, we can choose a set J of $n-1$ output shares such that $J' \subseteq J$ and we can perfectly simulate J and W from a set of input shares I such that

$|I| \leq |W| + |J'|$. Thus, we conclude that the gadget G_{refresh} is $(n-1)$-STRPE2. Thus, it is also (t, f)-TRPE2 of amplification order $d = t+1$. This concludes the proof. $\square$

A.9 Proof of Lemma 4

Proof. We will prove in this appendix Lemma 4, i.e that the constructed multiplication gadget from the composition of G_{submult} satisfying $(d-1)$-NI and $(d-1, 2d-1)$-partial NI, and G_{compress} satisfying (t, f')-comp-TRPE results in a (t, f)-RPE gadget G_{mult} with amplification order $d = \min(t+1, n-t)$. First let us fix $t \leq n-1$. We will be splitting a set of probe W on the multiplication gadget into two sets of probes $W = W_m \cup W_c$ where W_m are probes on G_{submult} (internal and output wires) and W_c are probes G_{compress} (on internal wires only).

We start by proving RPE1. Let J bet a set of output shares such that $|J| \leq t$.

- Let W be a set of probes on the multiplication gadget such that $|W| = |W_m \cup W_c| \leq d-1$. We know in particular from the comp-TRPE property on G_{compress} that all wires in J and W_c can be simulated from a set of input shares I_c on the intermediate result c such that $|I_c| \leq |W_c|$ (since $|W_c| \leq d-1 < 2d$). Then, we have a set of probes $W'_m = W_m \cup I_c$ on G_{submult} which is of size $|W'_m| \leq |W_m| + |I_c| \leq |W_m| + |W_c| \leq d-1$, then from $(d-1)$-NI property of G_{submult} we know that all the probes in W'_m can be simulated from sets of input shares I_a and I_b such that $|I_a| \leq d-1 \leq t$ and $|I_b| \leq d-1 \leq t$. This proves that we can simulate all probes in the overall set of probes W and in J from at most t shares of a and t shares of b. this proves the first property for RPE1.
- Next let W be a set of probes on the multiplication gadget such that $d \leq |W| = |W_m \cup W_c| \leq 2d-1$. We need to show that we can simulate W and J with at most a failure on one of the inputs a or b. We know in particular from the comp-TRPE property on G_{compress} that all wires in J and W_c can be simulated from a set of input shares I_c on the intermediate result c such that $|I_c| \leq |W_c|$ (since $|W_c| \leq 2d-1 < 2d$). Then, we have a set of probes $W'_m = W_m \cup I_c$ on G_{submult} which is of size $|W'_m| \leq |W_m| + |I_c| \leq |W_m| + |W_c| \leq 2d-1$. Hence from $(d-1, 2d-1)$-partial NI property of G_{submult}, all the probes in W'_m can be simulated from sets of input shares I_a and I_b such that $|I_a| \leq d-1$ or $|I_b| \leq d-1 \leq t$. Since $d = \min(t+1, n-t)$, then this implies that we have a failure on at most one of the inputs.
  This proves that we can simulate all probes in the overall set of probes W and in J from at most t shares of at least one of the inputs a or b (in other words, if we need more than t shares of a, then we need at most t shares of b). This proves the second property for RPE1.

From the above two cases, we conclude that the multiplication gadget is (t, f_1)-RPE1 with amplification order $d = \min(t+1, n-t)$.

Next we prove the property RPE2.

- Let W be a set of probes on the multiplication gadget such that $|W| = |W_m \cup W_c| \leq d-1$. We know in particular from the comp-TRPE property on G_{compress} that there exists a set J of $n-1$ output shares such that all wires in W_c and J can be simulated from a set of input shares I_c on the intermediate result c such that $|I_c| \leq |W_c|$ (since $|W_c| \leq d-1 < 2d$). Then, we have a set of probes $W'_m = W_m \cup I_c$ on G_{submult} which is of size $|W'_m| \leq |W_m| + |I_c| \leq |W_m| + |W_c| \leq d-1$, then from $(d-1)$-NI property of G_{submult} we know that all the probes in W'_m can be simulated from sets of input shares I_a and I_b such that $|I_a| \leq d-1 \leq t$ and $|I_b| \leq d-1 \leq t$. This proves that there exists a set J of $n-1$ output shares such that we can simulate all probes in the overall set of probes W and in J from at most t shares of a and t shares of b. This proves the first property for RPE2.

– Next let W be a set of probes on the multiplication gadget such that $d \leq |W| = |W_m \cup W_c| \leq 2d-1$. We know in particular from the comp-TRPE property on G_{compress} that there exists a set J of $n-1$ output shares such that all wires in W_c and J can be simulated from a set of input shares I_c on the intermediate result c such that $|I_c| \leq |W_c|$ (since $|W_c| \leq 2d-1 < 2d$). Then, we have a set of probes $W'_m = W_m \cup I_c$ on G_{submult} which is of size $|W'_m| \leq |W_m| + |I_c| \leq |W_m| + |W_c| \leq 2d-1$. Hence as for RPE1, from $(d-1, 2d-1)$-partial NI property of G_{submult}, we have that all the probes in W'_m can be simulated from sets of input shares I_a and I_b such that $|I_a| \leq d-1$ or $|I_b| \leq d-1 \leq t$. Since $d = \min(t+1, n-t)$, then this implies that we have a failure on at most one of the inputs.
  This proves that there exists a set J of $n-1$ output shares such that we can simulate all probes in the overall set of probes W and in J from at most t shares of at least one of the inputs a or b (in other words, if we need more than t shares of a, then we need at most t shares of b). This proves the second property for RPE2.

From the above two cases, we conclude that the multiplication gadget is (t, f_2)-RPE2 with amplification order $d = \min(t+1, n-t)$.

Combining both properties RPE1 and RPE2 with the same amplification order d, we conclude that the multiplication gadget is (t, f)-RPE with $f = \max(f_1, f_2)$ and of amplification order $d = \min(t+1, n-t)$. This concludes the proof of lemma 4. ☐

A.10 Proof of Lemma 5

Proof. Let G_{compress} be the $[m:n]$-compression gadget from Algorithm 1 such that $m \geq 2n$ and let G_{refresh} be the m-share refresh gadget such that G_{refresh} is $(m-1)$-SNI and $(m-1)$-STRPE2. We will prove that G_{compress} $[m:n]$-compression gadget constructed with such G_{refresh} is (t, f)**-comp-TRPE**. Let us denote $(c_1, \ldots, c_m)$ the input shares of G_{compress}, $(d_1, \ldots, d_n)$ its output shares, and $(c'_1, \ldots, c'_m)$ the refreshed shares of $(c_1, \ldots, c_m)$ using G_{refresh}. We write m as $m = K.n + \ell$ for $K, \ell \in \mathbb{N}$ such that $K = \lfloor m/n \rfloor$. For each $1 \leq i \leq \ell$, we have $d_i = c'_i + \ldots + c'_{i+K.n}$, and for $\ell+1 \leq i \leq n$, we have $d_i = c'_i + \ldots + c'_{i+(K-1).n}$. We will prove that G_{compress} is (t, f)-comp-TRPE. This amounts to proving that $\forall\ W, |W| \leq 2d-1$ a set of probes on the internal wires of G_{compress} where $d = \min(t+1, n-t)$:

1. $\forall\ J, |J| \leq t$ a set of output shares of G_{compress}, J and W can be simulated from a set of input shares I of the input c of G_{compress}, such that $|I| \leq |W|$.
2. $\exists\ J', |J'| = n-1$ a set of output shares of G_{compress}, such that J' and W can be simulated from a set of input shares I of the input c of G_{compress}, such that $|I| \leq |W|$.

We will prove both points separately

1. Let J be a set of output shares indices on G_{compress} such that $|J| \leq t$ for a $t \leq n-1$ and let $d = \min(t+1, n-t)$. Let W be a set of probes on G_{compress} such that $|W| \leq 2d-1$. We need to prove that we can perfectly simulate W and J from input shares indices in I such that

$|I| \leq |W|$. For this, We will simulate W and J using probes on G_{refresh}. First let us consider $J^\star$ the set of probes such that $J^\star = \{i \mid c'_i \in W \cap \{c'_1, \dots, c'_m\}\}$.
We construct the set W' of probes on G_{refresh} as follows:

$$W' = \{p \mid p \in W \setminus \{c'_1, \dots, c'_m\}\} \tag{35}$$

In addition, we construct the set J' of output shares on G_{refresh} as follows:

$$J' = J^\star \cup \bigcup_{\substack{i \in J \\ i \leq \ell}} \{i, \dots, i + K.n\} \cup \bigcup_{\substack{i \in J \\ i > \ell}} \{i, \dots, i + (K-1).n\} \tag{36}$$

It is easy to see that if we can perfectly simulate W' and J', then we can perfectly simulate W and J since $W = W' \cup \{c'_i \mid i \in J^\star\}$ and by perfectly simulating $(c'_i, \dots, c'_{i+K.n})$ for $i \in J$ such that $i \leq \ell$, then we can perfectly simulate $d_i = c'_i + \ldots + c'_{i+K.n}$ and by perfectly simulating $(c'_i, \dots, c'_{i+(K-1).n})$ for $i \in J$ such that $i > \ell$, then we can perfectly simulate $d_i = c'_i + \ldots + c'_{i+(K-1).n}$; thus all output shares in J are perfectly simulate using shares in J'. Hence, we need to prove that we can perfectly simulate W' and J' using the G_{refresh} m-share gadget.
Observe that since $|J^\star| \leq |W \setminus W'|$, then

$$|J'| \leq |W \setminus W'| + K.|J| + \min(t, \ell) \leq K.t + \min(t, \ell) \tag{37}$$

where the term $\min(t, \ell)$ comes from the worst case where all output shares $i \in J$ are such that $i \leq \ell$, because in this case we add to J' all the indices $(i, \dots, i + K.n)$ instead of $(i, \dots, i + (K-1).n)$ according to (36). Also, according to (35), we have $|W'| \leq |W|$. Hence, we have

$$|W'|+|J'| \leq |W'|+|W \setminus W'|+K.|J|+\min(t,\ell) \leq |W|+K.|J|+\min(t,\ell) \leq 2d-1+K.t+\min(t,\ell)$$

, so

$$|W'| + |J'| \leq 2\min(t+1, n-t) - 1 + K.t + \min(t, \ell)$$

, then

$$|W'| + |J'| \leq 2(n-t) + K.t + \ell - 1 \leq 2n + (K-2).t + \ell - 1$$

, and from $t \leq n-1$ we get

$$|W'| + |J'| \leq K.n + \ell - 1 - (K-2) \ .$$

Since by hypothesis we have $m \geq 2n$, so $K \geq 2$ and $(K-2) \geq 0$, hence

$$|W'| + |J'| \leq K.n + \ell - 1 \leq m - 1$$

Then by the $(m-1)$-SNI property of m-share G_{refresh}, we can perfectly simulate the set of probes W' and output shares in J' from a set of input shares I such that $|I| \leq |W'|$, hence we have

$$|I| \leq |W'| \leq |W|$$

which completes the proof for the first point of comp-TRPE on gadget G_{compress}.

2. Let $t \leq n-1$ and let $d = \min(t+1, n-t)$. Let W be a set of probes on G_{compress} such that $|W| \leq 2d-1$. We need to prove that we can perfectly simulate W and a chosen set J of $n-1$ output shares from input shares indices in I such that $|I| \leq |W|$. For this, we will simulate W and choose the set J using probes on G_{refresh}. First let us consider $J^\star$ the set of probes such that $J^\star = \{i \mid c'_i \in W \cap \{c'_1, \ldots, c'_m\}\}$.
We construct the set W' of probes on G_{refresh} as follows:
$$W' = \{p \mid p \in W \setminus \{c'_1, \ldots, c'_m\}\} \tag{38}$$
In addition, we construct the set J' of output shares on G_{refresh} as follows:
$$J' = J^\star \tag{39}$$
Observe that
$$|W'| + |J'| \leq |W| \leq 2d-1 \leq 2\min(t+1, n-t) - 1$$
, so
$$|W'| + |J'| \leq 2n-1 \leq m-1$$
Then by the $(m-1)$-STRPE2 property of m-share G_{refresh}, there exists a set J'' such that $J' \subseteq J''$ and $|J''| = m-1$ and W' and J'' can be perfectly simulated from input shares indexed in I such that $|I| \leq |W'| + |J'|$. Since $W = W' \cup \{c'_i \mid i \in J'\}$ then $|I| \leq |W|$.
By perfectly simulating W' and J'', we can perfectly simulate W since $W = W' \cup \{c'_i \mid i \in J'\}$. In addition, we choose the set J of $n-1$ output shares on G_{compress} as follows:
$$J = \{i \mid i \leq \ell \text{ and } \{i, \ldots, i + K.n\} \subseteq J''\} \cup \{i \mid i > \ell \text{ and } \{i, \ldots, i + (K-1).n\} \subseteq J''\}$$
Since $|J''| = m-1$, then we are sure that $|J| = n-1$ since there is only 1 share of $(c'_1, \ldots, c'_m)$ missing from J''. And since we can perfectly simulate J'' then we can also perfectly simulate J like before.
This proves that we can choose a set J of $n-1$ output shares on G_{compress} using probes on the internal gadget G_{refresh} such that W and J can be perfectly simulated from input shares in I such that $|I| \leq |W'| + |J'| \leq |W|$ for any $|W| \leq 2d-1$. This concludes the proof for the second point of comp-TRPE on gadget G_{compress}.

Thus, we proved that G_{compress} from Algorithm 1 is (t, f)-STRPE2. This concludes the proof for Lemma 5. ☐

A.11 Proof of Lemma 7

Proof. Let $t \leq n-1$ where n is the number of shares such that $(n, t) \neq (2k+1, \lfloor\frac{n-1}{2}\rfloor)$ for $k \in \mathbb{N}$ (i.e n is even or $t \neq \lfloor\frac{n-1}{2}\rfloor$), and let $d = \min(t+1, n-t)$. We will prove that if both matrices γ and δ satisfy Condition 2, then G_{submult} from Lemma 7 is $(d-1)$-NI and $(d-1, 2d-1)$-partial NI.

Proof for $(d-1)$-NI:
If the matrices γ and δ satisfy Condition 2, then they also sastisfy Condition 1, since Condition 2

is stronger. Then, in [8], the authors prove that we have that if γ and δ satisfy Condition 1, then the gadget G_{submult} is $(n-1)$-NI. In addition, if G_{submult} is $(n-1)$-NI, then in particular it is also $(d-1)$-NI for any $t \leq n-1$ and $d = \min(t+1, n-t)$. This implies that if the matrices satisfy Condition 2, then the gadget G_{submult} is $(d-1)$-NI thanks to the proof from [8]. This concludes the proof for the first point of Lemma 7.

Proof for $(d-1, 2d-1)$-partial NI:
We need to prove that G_{submult} is $(d-1, 2d-1)$-partial NI where $d = \min(t+1, n-t)$. In other words, we need to consider a set of probes W of size $|W| \leq 2d-1 \leq n-1$ and show that W can be simulated from inputs shares I_a and I_b such that $|I_a| \leq d-1$ or $|I_b| \leq d-1$. For this, we will split the set W into 3 distinct subsets $W = W_1 \cup W_2 \cup W_3$ with respect to the form of the probes in W. In fact, The authors from [8] show that G_{submult} is $(n-1)$-NI if the matrices γ and δ satisfy certain conditions. In fact, all of the probes on the sub-gadget G_{submult} are of a form in one of the following sets:

Set 1: $a_1, a_i, r_i, r_i + a_i, \gamma_{j-1,i-1} r_i, \gamma_{j-1,i-1} r_i + a_i$ (for $2 \leq i \leq n$ and $2 \leq j \leq n$)
Set 2: $a_1 + \sum_{i=2}^{k}(r_i + a_i)$ (for $2 \leq k \leq n$)
Set 3: $a_1 + \sum_{i=2}^{k}(\gamma_{j-1,i-1} r_i + a_i)$ (for $2 \leq j \leq n$ and $2 \leq k \leq n$)
Set 4: $b_1, b_i, s_i, s_i + b_i, \delta_{j-1,i-1} s_i, \delta_{j-1,i-1} s_i + b_i$ (for $2 \leq i \leq n$ and $2 \leq j \leq n$)
Set 5: $b_1 + \sum_{i=2}^{k}(s_i + b_i)$ (for $2 \leq k \leq n$)
Set 6: $b_1 + \sum_{i=2}^{k}(\delta_{j-1,i-1} s_i + b_i)$ (for $2 \leq j \leq n$ and $2 \leq k \leq n$)
Set 7: $-r_i \times (b_1 + \sum_{j=2}^{n}(\delta_{i-1,j-1} s_j + b_j))$ (for $2 \leq i \leq n$)
Set 8: $-s_i \times (a_1 + \sum_{j=2}^{n}(\gamma_{i-1,j-1} r_j + a_j))$ (for $2 \leq i \leq n$)
Set 9: $(a_1 + \sum_{i=2}^{n}(r_i + a_i)) \times (b_1 + \sum_{i=2}^{n}(s_i + b_i))$

The matrix γ would be related to probes of the form 1,2 and 3, while the matrix δ is directly related to probes of the form 4,5 and 6.
So we split the set W into $W = W_1 \cup W_2 \cup W_3$ with respect to the form of each probe as follows:

- W_1 contains probes of the forms in the sets 1, 2 and 3.
- W_2 contains probes of the forms in the sets 4, 5 and 6.
- W_3 contains probes of the forms in the sets 7, 8 and 9.

This split means that the set W_1 only contains probes involving the input shares of a and the randoms r_i, while W_2 only contains probes involving the input shares of b and the randoms s_i. W_3 contains products of both of the probes of W_1 and W_2.
Next, we will construct two subsets of probes W_a and W_b from the set W and prove that we can simulate all probes in W from W_a and W_b. In other terms, we start with $W_a = W_1$ and $W_b = W_2$.

Suppose first that $W_3 = \emptyset$. Then we consider the sets $W_a = W_1$ and $W_b = W_2$ as before. Suppose that to simulate W_a, we need sets of input shares I_a such that $|I_a| \geq d$, and let M be the number of probes of the form in sets 2 and 3 in the set of probes W_a. Then from condition 2 on matrix γ we know that $|I_a| \leq |W_a| - M \leq |W_a|$ (because $|W_a| \leq 2d-1 \leq n-1$ since $t \leq n-1$ such that $\Big((n = 2k) \vee (t \neq \dfrac{n-1}{2})\Big)$), then in order to have $|I_a| \geq d$, we must have:

$$d \leq |I_a| \leq |W_a|$$

Hence, since $|W| \leq 2d-1$, then we must have $|W_b| \leq d-1$ (because $|W_a| + |W_b| \leq 2d-1$), then from condition 2 on matrix δ, we can perfectly simulate W_b from I_b such that $|I_b| \leq |W_b| - M' \leq$

$|W_b| \leq d-1$ where M' is the number of probes of the form in sets 5 and 6 in the set of probes W_b. Thus we showed that we can perfectly simulate W with $|W| \leq 2d-1 \leq n-1$ from W_a and W_b using I_a and I_b such that if $|I_a| \geq d$, then $|I_b| \leq d-1$, so we have $|I_a| \leq d-1$ or $|I_b| \leq d-1$. This concludes the proof in the case where $W_3 = \emptyset$.

Next, we suppose that $W_3 \neq \emptyset$ so there is at least one probe of one of the sets 7, 8 or 9 in W_3. We construct sets W_a and W_b as before starting with $W_a = W_1$ and $W_b = W_2$, and for each probe in W_3:

- If the probe is of the form $-r_i \times (b_1 + \sum_{j=2}^{n}(\delta_{i-1,j-1}s_j + b_j))$, then we do $W_a = W_a \cup \{-r_i\}$, $W_b = W_b \cup \{(b_1 + \sum_{j=2}^{n}(\delta_{i-1,j-1}s_j + b_j))\}$. We denote the set of these probes in W_3 as W_3^7.
- If the probe is of the form $-s_i \times (a_1 + \sum_{j=2}^{n}(\gamma_{i-1,j-1}r_j + a_j))$, then we do $W_a = W_a \cup \{(a_1 + \sum_{j=2}^{n}(\gamma_{i-1,j-1}r_j + a_j))\}$, $W_b = W_b \cup \{-s_i\}$. We denote the set of these probes in W_3 as W_3^8.
- if the probe is of the form $(a_1 + \sum_{i=2}^{n}(r_i + a_i)) \times (b_1 + \sum_{i=2}^{n}(s_i + b_i))$, then we do $W_a = W_a \cup \{(a_1 + \sum_{i=2}^{n}(r_i + a_i))\}$, $W_b = W_b \cup \{(b_1 + \sum_{i=2}^{n}(s_i + b_i))\}$. We denote the set of these probes in W_3 as W_3^9.

Suppose that in order to simulate W_a, we need the set I_a such that $|I_a| \geq d$. In addition, since $|W_a| \leq |W| \leq 2d-1 \leq n-1$ (because $t \leq n-1$ such that $\Big((n = 2k) \vee (t \neq \dfrac{n-1}{2})\Big)$), then we know from condition 2 on γ that W_a can be perfectly simulated from I_a such that $|I_a| \leq |W_a| - M$ where M is the number of probes in W_a of the form $(a_1 + \sum_{j=2}^{n}(\gamma_{i-1,j-1}r_j + a_j))$ or $(a_1 + \sum_{i=2}^{n}(r_i + a_i))$. Then, since probes in the sets W_3^8 and W_3^9 add to W_a probes of these forms, then we have $|I_a| \leq |W_a| - |W_3^8| - |W_3^9|$. Hence, in order to have $|I_a| \geq d$, we must have

$$d \leq |I_a| \leq |W_a| - |W_3^8| - |W_3^9| \leq |W_1| + |W_3^7|$$

Similarly, suppose that to simulate W_b we need $|I_b| \geq d$, then we also must have

$$d \leq |I_b| \leq |W_b| - |W_3^7| - |W_3^9| \leq |W_2| + |W_3^8|$$

Hence, in order to have $|I_a| \geq d$ and $|I_b| \geq d$ at the same time, we must have

$$2d \leq |I_a| + |I_b| \leq |W_1| + |W_3^7| + |W_2| + |W_3^8| \leq |W|$$

which holds a contradiction with the fact that $|W| \leq 2d-1$. Hence, we cannot have at the same time $|I_a| \geq d$ and $|I_b| \geq d$. So G_{submult} is $(d-1, 2d-1)$-partial NI in the case where $W_3 \neq \emptyset$. Hence, we conclude that G_{submult} is $(d-1, 2d-1)$-partial NI after proving the property in both cases $W_3 = \emptyset$ and $W_3 \neq \emptyset$.

We conclude that G_{submult} satisfies both $(d-1)$-NI and $(d-1, 2d-1)$-partial NI, which conludes the proof for Lemma 7. □

A.12 Proof of Proposition 1

We consider the following matrices

$$\mathbf{L} = \big[\mathbf{I}_n \big| \mathbf{0}_{n\times n} \big| \mathbf{I}_n \big| \mathbf{I}_n \big| \ldots \big| \mathbf{I}_n \big| \mathbf{T}_n \big| \mathbf{T}_n \big| \ldots \big| \mathbf{T}_n\big]$$

$$\mathbf{M} = \big[\mathbf{0}_{n\times n} \big| \mathbf{I}_n \big| \mathbf{I}_n \big| \mathbf{D}_{\gamma,1} \big| \ldots \big| \mathbf{D}_{\gamma,n} \big| \mathbf{T}_n \big| \mathbf{T}_{\gamma,1} \big| \ldots \big| \mathbf{T}_{\gamma,n}\big]$$

The matrices $\mathbf{L}$ and $\mathbf{M}$ have $z = (2n+4) \cdot n$ columns. We want to lower-bound the probability, for $\boldsymbol{\gamma}$ picked uniformly at random in $\mathbb{F}_q^{n \times n}$, that for any vector $v \in \mathbb{F}_q^z$ of Hamming weight $hw(v) \leq n$, and for any $i_1, \ldots, i_K \in [z]$ such that $v_{i_1} \neq 0, \ldots, v_{i_K} \neq 0$ and the corresponding columns $i_1, \ldots, i_K$ in $\mathbf{L}$ and in $\mathbf{M}$ have no zero coefficient (*i.e* there are K probes of the form $\sum_{i=1}^n (r_i + a_i)$ or $\sum_{j=1}^n (\gamma_{i,j} r_j + a_j)$ for any $i = 1, \ldots, n$), if $\mathbf{M}.v = 0$, then we have $hw(\mathbf{L}.v) \leq hw(v) - K$.

For any set $I \subseteq \{1, \ldots, z\}$, we denote by $\mathbf{L}_I$ the $n \times |I|$ submatrix of $\mathbf{L}$ obtained by only keeping the columns in $\mathbf{L}$ whose indices are in I and $\mathbf{M}_I$ is the $n \times |I|$ submatrix of $\mathbf{M}$ obtained by only keeping the columns in $\mathbf{M}$ whose indices are in I. We will lower-bound the probability that for any set $I \subseteq \{1, \ldots, z\}$ of cardinal n and any vector $\boldsymbol{v} \in \mathbb{F}_q^n$, if $hw(\mathbf{L}_I \cdot \boldsymbol{v}) \geq hw(v) - K + 1$ then $\mathbf{M}_I \cdot \boldsymbol{v} \neq \mathbf{0}_n$.

We consider different cases (in order of increasing generality) which depend on the columns selected with the set I:

1. $I \subseteq \{(n+4) \cdot n + 1, \ldots, z\}$, i.e., all columns in $\mathbf{M}_I$ are taken from the matrices $\mathbf{T}_{\boldsymbol{\gamma},i}$ for $i \in \{1, \ldots, n\}$;
2. $I \subseteq \{(n+3) \cdot n + 1, \ldots, z\}$, i.e., all columns in $\mathbf{M}_I$ are taken from the matrix $\mathbf{T}_n$ or the matrices $\mathbf{T}_{\boldsymbol{\gamma},i}$ for $i \in \{1, \ldots, n\}$;
3. $I \subseteq \{1, \ldots, n+1\} \cup \{(n+3) \cdot n + 1, \ldots, z\}$, i.e., all columns in $\mathbf{M}_I$ are taken from the null vectors, from the matrix $\mathbf{T}_n$ or the matrices $\mathbf{T}_{\boldsymbol{\gamma},i}$ for $i \in \{1, \ldots, n\}$;
4. $I \subseteq \{1, \ldots, z\}$, i.e., the columns in $\mathbf{M}_I$ can be taken arbitrarily.

Case 1. In order to analyze the probability in the first case, we recall the definition of a probability distribution on structured matrices introduced in [8]. In this distribution of structured matrices, a number of elements with known location are identically zero, and remaining elements are chosen uniformly at random independently of each other.

Definition 13. *Let n and m be two positive integers. Let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)$ be a non-decreasing finite sequence with $1 \leq \alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_m \leq n$.*

- *A matrix $\boldsymbol{\Theta} = (\theta_{i,j}) \in \mathbb{F}_q^{n \times m}$ is called a* progressive patterned matrix *with* pattern $\boldsymbol{\alpha}$ *if $\theta_{i,j} = 0$ for all $j \in \{1, \ldots, m\}$ and all $i \notin \{\alpha_{j-1} + 1, \ldots, \alpha_j\}$ (where $\alpha_0 = 0$).*
- *The* unitary progressive patterned matrix $\boldsymbol{\Upsilon}_{\boldsymbol{\alpha}} = (u_{i,j}) \in \mathbb{F}_q^{n \times m}$ *with* pattern $\boldsymbol{\alpha}$ *is defined by $u_{i,j} = 0$ for all $j \in \{1, \ldots, m\}$ and all $i \notin \{\alpha_{j-1} + 1, \ldots, \alpha_j\}$ and $u_{i,j} = 1$ for all $j \in \{1, \ldots, m\}$ and all $i \in \{\alpha_{j-1} + 1, \ldots, \alpha_j\}$.*
- *The distribution $\mathcal{D}_{\boldsymbol{\alpha}}$ is the probability distribution on* random progressive patterned matrix $\mathbf{S}_{\boldsymbol{\alpha}} = (s_{i,j}) \in \mathbb{F}_q^{n \times m}$ *whose elements $s_{i,j}$ for $(i,j) \in \{1, \ldots, n\} \times \{1, \ldots, m\}$ are sampled uniformly at random and independently according to:*

$$\Pr[s_{i,j} = s] = \begin{cases} 1 & \text{if } s = 0 \text{ and } u_{i,j} = 0 \\ 0 & \text{if } s \neq 0 \text{ and } u_{i,j} = 0 \\ q^{-1} & \text{for all } s \in \mathbb{F}_q \text{ if } u_{i,j} = 1 \end{cases}$$

where $\boldsymbol{\Upsilon}_{\boldsymbol{\alpha}} = (u_{i,j}) \in \mathbb{F}_q^{n \times m}$ is the unitary progressive patterned matrix with pattern α.

A matrix $\boldsymbol{\Theta}$ is thus a *progressive patterned matrix* with *pattern* $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)$ if it is of the form described in Figure 5 where the symbol $\star$ denotes an arbitrary value in $\mathbb{F}_q$. For the unitary progressive patterned matrix $\boldsymbol{\Upsilon}_{\boldsymbol{\alpha}}$, this symbol $\star$ is replaced by a 1 and for a random progressive patterned matrix $\mathbf{S}_{\boldsymbol{\alpha}}$ each symbol $\star$ is replaced by a value picked uniformly and independently at random in $\mathbb{F}_q$. Note that such a matrix can contain a null column (when $\alpha_i = \alpha_{i+1}$ for some $i \in \{1, \ldots, m-1\}$).

$$\begin{array}{r} \\ \\ \alpha_1 \\ \\ \alpha_2 \\ \alpha_3 \\ \\ \alpha_4 \\ \\ \\ \alpha_{m-1} \\ \\ \alpha_m \\ \\ \\ \\ \end{array} \left.\overbrace{\begin{pmatrix} \star & 0 & 0 & 0 & \cdots & 0 & 0 \\ \star & 0 & 0 & 0 & \cdots & 0 & 0 \\ \star & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \star & 0 & 0 & \cdots & 0 & 0 \\ 0 & \star & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \star & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \star & \cdots & 0 & 0 \\ 0 & 0 & 0 & \star & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \star & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & \star \\ 0 & 0 & 0 & 0 & \cdots & 0 & \star \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix}}^{m \text{ columns}}\right\} n \text{ rows}$$

Fig. 5: Form of a progressive patterned matrix with *pattern* $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)$

Belaïd *et al.* [8] also defined more generally block column matrices formed of progressive patterned matrices.

Definition 14. *Let n, m, t be three positive integers. Let $m_1, \ldots, m_t$ be positive integers such that $m_1 + \cdots + m_t = m$ and let $\boldsymbol{\alpha}^{(i)} = (\alpha_1^{(i)}, \ldots, \alpha_{m_i}^{(i)})$ be a non-decreasing finite sequence with $1 \leq \alpha_1^{(i)} \leq \alpha_2^{(i)} \leq \cdots \leq \alpha_{m_i}^{(i)} \leq n$ for all $i \in \{1, \ldots, t\}$. We suppose that there exists at least one $j \in \{1, \ldots, t\}$ such that $\alpha_{m_j}^{(j)} = n$.*

- *A matrix $\boldsymbol{\Theta} \in \mathbb{F}_q^{n \times m}$ is called a* block progressive patterned matrix *with* pattern $(\boldsymbol{\alpha}^{(1)}, \ldots, \boldsymbol{\alpha}^{(t)})$ *if there exist progressive patterned matrices $\boldsymbol{\Theta}^{(i)} \in \mathbb{F}_q^{n \times m_i}$ with pattern $\boldsymbol{\alpha}^{(i)}$ for all $i \in \{1, \ldots, t\}$ such that $\boldsymbol{\Theta} = (\boldsymbol{\Theta}^{(1)} | \ldots | \boldsymbol{\Theta}^{(t)})$.*
- *The* block unitary progressive patterned matrix $\boldsymbol{\Upsilon}_{\boldsymbol{\alpha}^{(1)}, \ldots, \boldsymbol{\alpha}^{(t)}} \in \mathbb{F}_q^{n \times m}$ *with* pattern $(\boldsymbol{\alpha}^{(1)}, \ldots, \boldsymbol{\alpha}^{(t)})$ *is $\boldsymbol{\Upsilon}_{\boldsymbol{\alpha}^{(1)}, \ldots, \boldsymbol{\alpha}^{(t)}} = (\boldsymbol{\Upsilon}_{\boldsymbol{\alpha}^{(1)}} | \ldots | \boldsymbol{\Upsilon}_{\boldsymbol{\alpha}^{(t)}})$.*
- *The distribution $\mathcal{D}_{\boldsymbol{\alpha}^{(1)}, \ldots, \boldsymbol{\alpha}^{(t)}}$ is the probability distribution on* block random progressive patterned matrix *in $\mathbb{F}_q^{n \times m}$ defined by*

$$\mathcal{D}_{\boldsymbol{\alpha}^{(1)}, \ldots, \boldsymbol{\alpha}^{(t)}} = (\mathcal{D}_{\boldsymbol{\alpha}^{(1)}} | \ldots | \mathcal{D}_{\boldsymbol{\alpha}^{(t)}}).$$

The main ingredient of the proof of Proposition 1 is the following technical lemma:

Lemma 11. *Let n, m, t be three positive integers with $m \geq n$ and let $\boldsymbol{\alpha}^{(i)}$ for $i \in \{1, \ldots, t\}$ be patterns for block progressive patterned matrix as in Definition 14. For a block random progressive patterned matrix $\mathbf{S}$ drawn following the distribution $\mathcal{D}_{\boldsymbol{\alpha}^{(1)}, \ldots, \boldsymbol{\alpha}^{(t)}}$, there exists a linear subspace of*

$\mathbb{F}_q^m$ *of dimension* $m - n$ *that contains* $\{\boldsymbol{v} \in \mathbb{F}_q^m$ *s.t.* $hw(\boldsymbol{v}) = m$ *and* $\mathbf{S}\boldsymbol{v} = \mathbf{0}\}$*, with probability at least* $1 - mq^{-1}$*.*

Proof (Lemma 11). We will prove this lemma by induction on m.

For $m = 1$, since $m \geq n \geq 1$, Definition 14 implies that the matrix $\mathbf{S}$ consists simply in a single entry $s_{1,1}$ which is picked uniformly at random in $\mathbb{F}_q$ and this entry is null with probability q^{-1}. The set $\{\boldsymbol{v} \in \mathbb{F}_q$ s.t. $hw(\boldsymbol{v}) = 1$ and $\mathbf{S} \cdot \boldsymbol{v} = \mathbf{0}\}$ is therefore the empty set with probability at least $1 - q^{-1}$ and it is thus included in the subspace of dimension 0 with probability at least $1 - q^{-1}$.

We now consider $m \geq 2$ and we suppose Lemma 11 proven for all block random progressive patterned matrix with strictly less than m columns.

We first assume that the matrix $\boldsymbol{\Upsilon}_{\boldsymbol{\alpha}^{(1)},\ldots,\boldsymbol{\alpha}^{(t)}} \in \mathbb{F}_q^{n\times m}$ is the matrix of ones $\mathbf{U}_{n\times m}$ (i.e., does not contain any zero). Then $\mathbf{S}$ is simply a matrix drawn from $\mathbb{F}_q^{n\times m}$ with the uniform distribution.

It is well known that the number of full-rank $n \times m$ matrices over $\mathbb{F}_q$ (with $m \geq n$) is:

$$(q^m - 1)(q^m - q)\cdots(q^m - q^{n-1})$$

and the probability that $\mathbf{S}$ is of full rank is thus equal to:

$$(1 - q^{-m})(1 - q^{-m+1})\ldots(1 - q^{-m+n-1})$$

which is greater than

$$1 - \sum_{i=m-n+1}^{m} q^{-i} \geq 1 - \sum_{i=m-n+1}^{\infty} q^{-i} = 1 - \frac{1}{q^{-m+n-1}(1 - 1/q)} \geq 1 - 2q^{n-m-1}.$$

The subspace $\{\boldsymbol{v} \in \mathbb{F}_q^m$ s.t. $\mathbf{S} \cdot \boldsymbol{v} = 0\}$ is therefore included in a linear subspace of dimension $m - n$ with probability at least $1 - 2q^{n-m-1}$ and the result follows (since $m \geq 2$).

We now assume that the matrix $\boldsymbol{\Upsilon}_{\boldsymbol{\alpha}^{(1)},\ldots,\boldsymbol{\alpha}^{(t)}} \in \mathbb{F}_q^{n\times m}$ contains some 0. By assumption, there exists some $j \in \{1, \ldots, t\}$ such that $\alpha_{m_j}^{(j)} = n$.

1. We first assume that $m_j > 1$ (*i.e.* that the column of index $m_1 + \cdots + m_j$ consists in $\alpha_{m_j-1}^{(j)} \geq 1$ zeroes followed by $\alpha_{m_j}^{(j)} - \alpha_{m_j-1}^{(j)} = n - \alpha_{m_j-1}^{(j)} \geq 1$ ones, see Figure 6). We consider the submatrix of $\boldsymbol{\Upsilon}_{\boldsymbol{\alpha}^{(1)},\ldots,\boldsymbol{\alpha}^{(t)}} \in \mathbb{F}_q^{n\times m}$ obtained by deleting the column of index $m_1 + \cdots + m_j$ and the rows of indices in the set $\{\alpha_{m_j-1}^{(j)} + 1, \ldots, \alpha_{m_j}^{(j)}\}$.
   It is easy to see that this submatrix is a block unitary progressive patterned matrix with $n' \leq n - 1$ rows and $m - 1$ columns, where some columns may possibly contain only zeroes (see Figure 6). We can thus apply the induction hypothesis to the submatrix $\mathbf{S}'$ of $\mathbf{S}$ obtained by deleting the same column and the same rows.
   By induction hypothesis, we know that with probability at least $1 - (m - 1)q^{-1}$, there exists a linear subspace $V' \subseteq \mathbb{F}_q^{m-1}$ of dimension $m-1-n'$ that contains the set $\{\boldsymbol{v} \in \mathbb{F}_q^{m-1}$ s.t. $hw(\boldsymbol{v}) = m - 1$ and $\mathbf{S}' \cdot \boldsymbol{v} = 0\}$.
   If V' is of dimension 0, then $\{\boldsymbol{v} \in \mathbb{F}_q^{m-1}$ s.t. $hw(\boldsymbol{v}) = m - 1$ and $\mathbf{S}' \cdot \boldsymbol{v} = 0\} \subseteq \{\mathbf{0}_{m-1}\}$ and this set is thus the empty set. We then have $\{\boldsymbol{v} \in \mathbb{F}_q^m, hw(\boldsymbol{v}) = m$ and $\mathbf{S} \cdot \boldsymbol{v} = 0\} = \emptyset$ with probability at least $1 - (m - 1)q^{-1} \geq 1 - mq^{-1}$, and so there exists a linear subspace V of dimension $m - n$ that contains this set.

$$
\begin{array}{c}
\overbrace{\qquad\qquad\quad}^{m_1}\ \overbrace{\qquad\quad}^{m_2}\qquad\quad\overbrace{\qquad\quad}^{m_j}\qquad\quad\overbrace{\qquad\qquad\quad}^{m_t}\\
\left(\begin{array}{cccccccccccccccc}
1 & 0 & 0 & 0 & 1 & 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 1 & 0 & 0 & 0\\
1 & 0 & 0 & 0 & 1 & 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 1 & 0 & 0 & 0\\
1 & 0 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 1 & 0 & 0\\
0 & 1 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 1 & 0 & 0\\
0 & 1 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 1 & 0 & 0\\
0 & 0 & 1 & 0 & 0 & 1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 & 1 & 0\\
0 & 0 & 0 & 1 & 0 & 0 & 1 & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 & 1 & 0\\
0 & 0 & 0 & 1 & 0 & 0 & 1 & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 & 1 & 0\\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 & 0 & 0 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 & 0 & 0 & 0
\end{array}\right)
\end{array}
$$

Fig. 6: Example of a matrix $\boldsymbol{\Upsilon}_{\boldsymbol{\alpha}^{(1)},\dots,\boldsymbol{\alpha}^{(t)}} \in \mathbb{F}_q^{n\times m}$. The column and the rows highlighted in red are deleted in order to apply the induction hypothesis.

If V' is of dimension $m-1-n' > 0$, we can assume without loss of generality that the column of $\mathbf{S}$ deleted to obtain $\mathbf{S'}$ was the last one (by permuting the blocks of the matrix). We have the following block-decomposition of $\mathbf{S}$

$$\mathbf{S} = \begin{pmatrix} \mathbf{S'} & \mathbf{0_{n'\times 1}} \\ \mathbf{S''} & \mathbf{u} \end{pmatrix}$$

where $\mathbf{S''}$ is a $(n-n')\times(m-1)$ matrix and $\mathbf{u}$ a column vector of dimension $(n-n')$. Note that $\boldsymbol{u}$ is a random vector in $\mathbb{F}_q^{n-n'}$ independent from $\mathbf{S'}$ and $\mathbf{S''}$. Let $\boldsymbol{v} \in \mathbb{F}_q^m$ such that $hw(\boldsymbol{v}) = m$ and $\mathbf{S}\boldsymbol{v} = 0$.

We write $\boldsymbol{v} = \begin{pmatrix} \boldsymbol{w} \\ \tau \end{pmatrix}$ where $\boldsymbol{w} \in \mathbb{F}_q^{m-1}$ and $\tau \in \mathbb{F}_q$ is a scalar. We have $hw(\boldsymbol{w}) = m-1$ and $\mathbf{S'}\boldsymbol{w} = 0$, and therefore $\boldsymbol{w} \in V'$. Since $\tau \neq 0$ by assumption, the vector $\boldsymbol{u}$ thus belongs to the image W of V' by $\mathbf{S''}$ (with probability at least $1-(m-1)q^{-1}$). Moreover, W has dimension at most $\max(m-1-n', n-n')$.

- If W is of dimension at most $n-n'-1$, since $\boldsymbol{u}$ is independent of $\mathbf{S'}$ and $\mathbf{S''}$ (and thus of W), $\boldsymbol{u}$ belongs to W with probability at most q^{-1}. Therefore, with probability at least $(1-q^{-1})\cdot(1-(m-1)q^{-1}) \geq 1-mq^{-1}$, $\{\boldsymbol{v} \in \mathbb{F}_q^m \text{ s.t. } hw(\boldsymbol{v}) = m \text{ and } \mathbf{S}\boldsymbol{v} = 0\} = \emptyset$.
- If W is of dimension $n-n'$, with probability $1-q^{-(n-n')} \geq 1-q^{-1}$, we have $\boldsymbol{u} \neq \mathbf{0}_{(\boldsymbol{n}-\boldsymbol{n'})\times\mathbf{1}}$ and we can construct a basis $\boldsymbol{u_1} = \boldsymbol{u}, \dots, \boldsymbol{u_{n-n'}}$ of W.

  All subspaces $V' \cap \mathbf{S''}^{-1}(\langle \boldsymbol{u_i} \rangle)$ are of dimension at least one and we have

$$V' = \bigoplus_{i=1}^{n-n'} V' \cap \mathbf{S''}^{-1}(\langle \boldsymbol{u_i} \rangle).$$

Therefore the linear subspace V defined as $V = V' \cap \mathbf{S''}^{-1}(\langle \boldsymbol{u_1} \rangle)$ satisfies

$$\begin{aligned}
\dim(V) &= \dim(V') - \sum_{i=2}^{n-n'} \dim\left(V' \cap \mathbf{S''}^{-1}(\langle \boldsymbol{u_i} \rangle)\right) \\
&\leq m - 1 - n' - (n - n' - 1) \\
&= m - n.
\end{aligned}$$

Moreover, we have $\{\boldsymbol{v} \in \mathbb{F}_q^m \text{ s.t. } hw(\boldsymbol{v}) = m \text{ and } \mathbf{S}\boldsymbol{v} = 0\} \subseteq V$ and since this occurs with probability at least $(1 - q^{-1})(1 - (m-1)q^{-1}) \geq 1 - mq^{-1}$, the result follows.

2. We now assume that $m_i = 1$ for all i such that $\alpha^{(i)}_{m_i} = n$ (*i.e.* that all the columns with a one in the last row consists only of ones, see Figure 7). Since the matrix $\mathbf{\Upsilon}_{\boldsymbol{\alpha}^{(1)},\dots,\boldsymbol{\alpha}^{(t)}} \in \mathbb{F}_q^{n \times m}$ contains some 0, there exists some $j \in \{1, \dots, t\}$ such that $m_j > 1$ and we consider such a $j \in \{1, \dots, t\}$ for which $\alpha_1^{(j)}$ is minimal (see Figure 7).
We consider the submatrix of $\mathbf{\Upsilon}_{\boldsymbol{\alpha}^{(1)},\dots,\boldsymbol{\alpha}^{(t)}} \in \mathbb{F}_q^{n \times m}$ obtained by deleting the column of index $m_1 + \cdots + m_{j-1} + 1$ and the rows of indices in the set $\{1, \dots, \alpha_1^{(j)} - 1\}$. It is easy to see that this submatrix is a block unitary progressive patterned matrix with $n' \leq n - 1$ rows and $m - 1$ columns (see Figure 7). We can thus apply the induction hypothesis to the submatrix $\mathbf{S}'$ of $\mathbf{S}$ obtained by deleting the same column and the same rows.

$$\begin{array}{c}
\begin{array}{ccccccc}
\overbrace{\hphantom{1\;0\;0\;0}}^{m_1} & \overbrace{\hphantom{1}}^{m_2} & \overbrace{\hphantom{1\;0\;0}}^{m_3} & \hphantom{\cdots} & \overbrace{\hphantom{1\;0\;0\;0}}^{m_j} & \hphantom{\cdots} & \overbrace{\hphantom{1\;0}}^{m_t}
\end{array} \\
\left(\begin{array}{cccccccccccccccc}
1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & \cdots & 1 & 0 & 0 & 0 & \cdots & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & \cdots & 1 & 0 & 0 & 0 & \cdots & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 & \cdots & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 & \cdots & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 & \cdots & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0
\end{array}\right)
\end{array}$$

Fig. 7: Example of a matrix $\mathbf{\Upsilon}_{\boldsymbol{\alpha}^{(1)},\dots,\boldsymbol{\alpha}^{(t)}} \in \mathbb{F}_q^{n \times m}$. The column and the rows highlighted in red are deleted in order to apply the induction hypothesis.

We know that with probability at least $1 - (m-1)q^{-1}$, there exists a linear subspace $V' \subseteq \mathbb{F}_q^{m-1}$ of dimension $m - 1 - n'$ that contains the set $\{\boldsymbol{v} \in \mathbb{F}_q^{m-1} \text{ s.t. } hw(\boldsymbol{v}) = m - 1 \text{ and } \mathbf{S}'\boldsymbol{v} = 0\}$.
If V' is of dimension 0, then $\{\boldsymbol{v} \in \mathbb{F}_q^{m-1} \text{ s.t. } hw(\boldsymbol{v}) = m - 1 \text{ and } \mathbf{S}'\boldsymbol{v} = 0\} \subseteq \{0\}$ and this set is thus the empty set. We then have $\{\boldsymbol{v} \in \mathbb{F}_q^m, hw(\boldsymbol{v}) = m \text{ and } \mathbf{S}\boldsymbol{v} = 0\} = \emptyset$ with probability at least $1 - (m-1)q^{-1} \geq 1 - mq^{-1}$, and so there exists a linear subspace V of dimension $m - n$ that contains this set.

If V' is of dimension $m-1-n' > 0$, we can assume without loss of generality that the column of $\mathbf{S}$ deleted to obtain $\mathbf{S}'$ was the last one (by permuting the blocks of the matrix). We have the following block-decomposition of $\mathbf{S}$

$$\mathbf{S} = \begin{pmatrix} \mathbf{S}'' & \mathbf{u} \\ \mathbf{S}' & \mathbf{0_{n'\times 1}} \end{pmatrix}$$

where $\mathbf{S}''$ is a $(n-n') \times (m-1)$ matrix and $\mathbf{u}$ a column vector of dimension $(n-n')$. Note that $\boldsymbol{u}$ is a random vector in $\mathbb{F}_q^{n-n'}$ independent from $\mathbf{S}'$ and $\mathbf{S}''$. Let $\boldsymbol{v} \in \mathbb{F}_q^m$ such that $hw(\boldsymbol{v}) = m$ and $\mathbf{S}\boldsymbol{v} = 0$.
We write $\boldsymbol{v} = \begin{pmatrix} \tau \\ \boldsymbol{w} \end{pmatrix}$ where $\boldsymbol{w} \in \mathbb{F}_q^{m-1}$ and $\tau \in \mathbb{F}_q$ is a scalar. We have $hw(\boldsymbol{w}) = m-1$ and $\mathbf{S}'\boldsymbol{w} = 0$, and therefore $\boldsymbol{w} \in V'$. Since $\tau \neq 0$ by assumption, the vector $\boldsymbol{u}$ thus belongs to the image W of V' by $\mathbf{S}''$ (with probability at least $1-(m-1)q^{-1}$). Moreover, W has dimension at most $\max(m-1-n', n-n')$.

- If W is of dimension at most $n-n'-1$, since $\boldsymbol{u}$ is independent of $\mathbf{S}'$ and $\mathbf{S}''$ (and thus of W), $\boldsymbol{u}$ belongs to W with probability at most q^{-1}. Therefore, with probability at least $(1-q^{-1}) \cdot (1-(m-1)q^{-1}) \geq 1-mq^{-1}$, $\{\boldsymbol{v} \in \mathbb{F}_q^m \text{ s.t. } hw(\boldsymbol{v}) = m \text{ and } \mathbf{S}\boldsymbol{v} = 0\} = \emptyset$.
- If W is of dimension $n-n'$ then $\mathbf{S}''$ is invertible. With probability $1-q^{-(n-n')} \geq 1-q^{-1}$, we have $\boldsymbol{u} \neq \mathbf{0_{(n-n')\times 1}}$ and we can construct a basis $\boldsymbol{u_1} = \boldsymbol{u}, \ldots, \boldsymbol{u_{n-n'}}$ of W.
  All subspaces $V' \cap \mathbf{S}''^{-1}(\langle \boldsymbol{u_i} \rangle)$ are of dimension at least one and we have

$$V' = \bigoplus_{i=1}^{n-n'} V' \cap \mathbf{S}''^{-1}(\langle \boldsymbol{u_i} \rangle).$$

  Therefore the linear subspace V defined as $V = V' \cap \mathbf{S}''^{-1}(\langle \boldsymbol{u_1} \rangle)$ satisfies

$$\begin{aligned} \dim(V) &= \dim(V') - \sum_{i=2}^{n-n'} \dim\left(V' \cap \mathbf{S}''^{-1}(\langle \boldsymbol{u_i} \rangle)\right) \\ &\leq m-1-n'-(n-n'-1) \\ &= m-n. \end{aligned}$$

  Moreover, we have $\{\boldsymbol{v} \in \mathbb{F}_q^m \text{ s.t. } hw(\boldsymbol{v}) = m \text{ and } \mathbf{S}\boldsymbol{v} = 0\} \subseteq V$ and since this occurs with probability at least $(1-q^{-1})(1-(m-1)q^{-1}) \geq 1-mq^{-1}$, the result follows.

This concludes the proof of Lemma 11.

Recall that we want to lower-bound the probability over the $\gamma \in \mathbb{F}_q^{n\times n}$, that for a given set $I \subseteq \{(n+4)\cdot n+1, \ldots, z\}$ of cardinal n, if $hw(\mathbf{L}_I \cdot \boldsymbol{v}) \geq n-K$ then $\mathbf{M}_I \cdot \boldsymbol{v} \neq \mathbf{0}_n$ for any vector $\boldsymbol{v} \in \mathbb{F}_q^n$. where K denotes the number of coordinates $i_1, \ldots, i_K \in [z]$ such that $v_{i_1} \neq 0, \ldots, v_{i_K} \neq 0$ and the corresponding columns $i_1, \ldots, i_K$ in $\mathbf{L}$ and in $\mathbf{M}$ have no zero coefficient.

Remark that the non-zero coefficients in the lower block of $\mathbf{L}_I$ and in $\mathbf{M}_I$ are at the same positions. If $K = 0$, then the matrices $\mathbf{M}_I$ and $\mathbf{L}_I$ have a null row. In this case, we have readily $hw(\mathbf{L}_I \cdot \boldsymbol{v}) \leq n-1 = n-K-1 < n-K$).

If $K \geq 1$, then the matrices $\mathbf{M}_I$ and $\mathbf{L}_I$ does not have a null row. The matrix $\mathbf{M}_I$ (up to some permutation of its columns) can be written as a block matrix where each block is of the form described in Figure 8 (on the left).

$$
\begin{pmatrix}
\gamma_{i,1} & \gamma_{i,1} & \cdots & \gamma_{i,1} & \cdots & \gamma_{i,1} \\
\vdots & & & & & \vdots \\
\gamma_{i,\alpha_1} & \gamma_{i,\alpha_1} & \cdots & \gamma_{i,\alpha_1} & \cdots & \gamma_{i,\alpha_1} \\
0 & \gamma_{i,\alpha_1+1} & \cdots & \gamma_{i,\alpha_1+1} & \cdots & \gamma_{i,\alpha_1+1} \\
\vdots & \vdots & & & & \vdots \\
0 & \gamma_{i,\alpha_2} & \cdots & \gamma_{i,\alpha_2} & \cdots & \gamma_{i,\alpha_2} \\
0 & 0 & & \gamma_{i,\alpha_2+1} & \cdots & \gamma_{i,\alpha_2+1} \\
\vdots & & & \vdots & & \vdots \\
0 & \cdots & 0 & \gamma_{i,\alpha_j} & \cdots & \gamma_{i,\alpha_j} \\
0 & \cdots & 0 & 0 & \cdots & \gamma_{i,\alpha_j+1} \\
\vdots & & & \vdots & & \vdots \\
0 & \cdots & \cdots & 0 & \cdots & \gamma_{i,\alpha_{m-1}} \\
0 & \cdots & \cdots & 0 & \cdots & \gamma_{i,\alpha_{m-1}+1} \\
\vdots & & & \vdots & & \vdots \\
0 & \cdots & \cdots & 0 & \cdots & \gamma_{i,\alpha_m} \\
0 & \cdots & \cdots & 0 & \cdots & 0 \\
\vdots & & & & & \vdots \\
0 & \cdots & \cdots & \cdots & \cdots & 0
\end{pmatrix}
\qquad
\begin{pmatrix}
\gamma_{i,1} & 0 & \cdots & \cdots & \cdots & 0 \\
\vdots & & & & & \vdots \\
\gamma_{i,\alpha_1} & 0 & \cdots & \cdots & \cdots & 0 \\
0 & \gamma_{i,\alpha_1+1} & \cdots & \cdots & \cdots & 0 \\
\vdots & \vdots & & & & \vdots \\
0 & \gamma_{i,\alpha_2} & \cdots & \cdots & \cdots & 0 \\
0 & 0 & & \gamma_{i,\alpha_{j-1}+1} & \cdots & 0 \\
\vdots & & & \vdots & & \vdots \\
0 & \cdots & 0 & \gamma_{i,\alpha_j} & \cdots & 0 \\
0 & \cdots & 0 & 0 & \cdots & 0 \\
\vdots & & & \vdots & & \vdots \\
0 & \cdots & \cdots & 0 & \cdots & 0 \\
0 & \cdots & \cdots & 0 & \cdots & \gamma_{i,\alpha_{m-1}+1} \\
\vdots & & & \vdots & & \vdots \\
0 & \cdots & \cdots & 0 & \cdots & \gamma_{i,\alpha_m} \\
0 & \cdots & \cdots & 0 & \cdots & 0 \\
\vdots & & & & & \vdots \\
0 & \cdots & \cdots & \cdots & \cdots & 0
\end{pmatrix}
$$

Fig. 8: Blocks appearing in matrices $\mathbf{M}_I$ and $\tilde{\mathbf{M}}_I$

From this matrix, one can construct another matrix $\tilde{\mathbf{M}}_I$ such that in each block, one substract each column to the following columns (i.e., one substract iteratively the i-th column to the columns of index in $\{i+1,\dots,m\}$ for $i \in \{1,\dots,m\}$). The blocks appearing in the matrix $\tilde{\mathbf{M}}_I$ are given in Figure 8 (on the right). Since we apply only elementary operations on the columns, if there exists a vector $\boldsymbol{v} \in \mathbb{F}_q^n$ such that $\mathbf{M}_I\boldsymbol{v} = 0$ then, there exists a vector $\boldsymbol{v'} \in \mathbb{F}_q^n$ such that $\tilde{\mathbf{M}}_I\boldsymbol{v'} = 0$.

Since $\mathbf{M}_I$ has no null row, we have $\alpha_m = n$ in one of this block (with the notation from Figure 8) and the matrix $\tilde{\mathbf{M}}_I$ is thus a block random progressive patterned matrix as defined in Definition 14. By Lemma 11, for each non-empty subset J of the n columns of $\tilde{\mathbf{M}}_I$, the probability over γ that there exists a vector $\boldsymbol{v'} \in \mathbb{F}_q^n$ with support J (i.e., set of non-zero coordinates) such that $\tilde{\mathbf{M}}_I\boldsymbol{v'} = 0$ is upper bounded by $n \cdot q^{-1}$. By the union bound over all supports, the probability over γ that there exists a vector $\boldsymbol{v'} \in \mathbb{F}_q^n$ such that $\tilde{\mathbf{M}}_I\boldsymbol{v'} = 0$ is thus upper-bounded by $2^n \cdot n \cdot q^{-1}$.

For the sets $I \subseteq \{(n+4)\cdot n+1,\dots,z\}$ of cardinal n, we have proved that with probability at least $1 - 2^n \cdot n \cdot q^{-1}$ (over the choice of $\gamma \in \mathbb{F}_q^{n\times n}$), we have $hw(\mathbf{L}_I \cdot \boldsymbol{v}) < n - K$ or $\mathbf{M}_I \cdot \boldsymbol{v} \neq \mathbf{0}_n$ for any vector $\boldsymbol{v} \in \mathbb{F}_q^n$.

Case 2. We now consider matrices $\mathbf{M}_I$ were all columns are taken from the matrix $\mathbf{T}_n$ or the matrices $\mathbf{T}_{\boldsymbol{\gamma},i}$ for $i \in \{1,\dots,n\}$ (i.e., $I \subseteq \{(n+3)\cdot n+1,\dots,z\}$). With the notation from Definition 14, we consider the modified distribution $\tilde{\mathcal{D}}_{\boldsymbol{\alpha}^{(1)},\dots,\boldsymbol{\alpha}^{(t)}}$ defined as the following probability distribution in $\mathbb{F}_q^{n\times m}$:

$$\tilde{\mathcal{D}}_{\boldsymbol{\alpha}^{(1)},\dots,\boldsymbol{\alpha}^{(t)}} = (\boldsymbol{\Upsilon}_{\boldsymbol{\alpha}^{(1)}}|\mathcal{D}_{\boldsymbol{\alpha}^{(2)},\dots,\boldsymbol{\alpha}^{(t)}}) = (\boldsymbol{\Upsilon}_{\boldsymbol{\alpha}^{(1)}}|\mathcal{D}_{\boldsymbol{\alpha}^{(2)}}|\dots|\mathcal{D}_{\boldsymbol{\alpha}^{(t)}})$$

(i.e., in which the first block is a fixed unitary progressive patterned matrix instead of being a random progressive patterned matrix). We can easily extend Lemma 11 to this distribution:

Lemma 12. *Let n, m, t be three positive integers with $m \geq n$ and let $\boldsymbol{\alpha}^{(i)}$ for $i \in \{1,\dots,t\}$ be patterns for block progressive patterned matrix as in Definition 14. For a block random progressive patterned matrix matrix $\mathbf{S}$ drawn following the distribution $\tilde{\mathcal{D}}_{\boldsymbol{\alpha}^{(1)},\dots,\boldsymbol{\alpha}^{(t)}}$, there exists a linear subspace of $\mathbb{F}_q^m$ of dimension $m-n$ that contains $\{\boldsymbol{v} \in \mathbb{F}_q^m$ s.t. $hw(\boldsymbol{v}) = m$ and $\mathbf{S}\boldsymbol{v} = 0\}$, with probability at least $1 - mq^{-1}$.*

Proof (Lemma 12). We will prove Lemma 12 by induction on m.

For $m = 1$, since $m \geq n \geq 1$, Definition 14 implies that the the matrix $\mathbf{S}$ either (1) consists simply in a single entry $s_{1,1}$ which is picked uniformly at random in $\mathbb{F}_q$ or (2) a constant non-null vector. In the first case, this vector is null with probability q^{-1} and in all cases the set $\{\boldsymbol{v} \in \mathbb{F}_q \text{ s.t. } hw(\boldsymbol{v}) = 1 \text{ and } \mathbf{S}\boldsymbol{v} = 0\}$ is therefore the empty set with probability at least $1 - q^{-1}$. It is thus included in the subspace of dimension 0 with probability at least $1 - q^{-1}$.

We now consider $m \geq 2$ and we assume Lemma 12 proven for all block random progressive patterned matrix matrix drawn from a distribution $\tilde{\mathcal{D}}_{\boldsymbol{\alpha}^{(1)},\ldots,\boldsymbol{\alpha}^{(t)}}$ with strictly less than m columns.

We first assume that the matrix $\boldsymbol{\Upsilon}_{\boldsymbol{\alpha}^{(1)},\ldots,\boldsymbol{\alpha}^{(t)}} \in \mathbb{F}_q^{n\times m}$ is the unitary matrix $\mathbf{U}_{n\times m}$ (i.e., does not contain any zero). Then, by assumption, we have $m_i = 1$ and $\alpha^{(i)} = n$ for $i \in \{1, \ldots, t\}$. The matrix $\mathbf{S}$ is thus the concatenation of the vector $\mathbf{1}_{n\times 1}$ and a matrix picked from $\mathbb{F}_q^{n\times m-1}$ with the uniform distribution. Using elementary operations on the columns of $\mathbf{S}$, one can obtain a matrix of the form

$$\begin{pmatrix} 1 & \mathbf{0_{1\times m-1}} \\ \boldsymbol{u_{n-1}} & \mathbf{S}' \end{pmatrix}$$

where $\boldsymbol{u_{n-1}} \in \mathbb{F}_q^{n-1}$ is the all-one vector and $\mathbf{S}'$ is a matrix drawn from $\mathbb{F}_q^{n-1\times m-1}$ with the uniform distribution. As in the proof of Lemma 11, the matrix $\mathbf{S}'$ is of full rank $n-1$ with probability at least $1 - 2q^{n-m-2}$. The matrix $\mathbf{S}$ is thus of full rank n with probability at least $1 - 2q^{n-m-2}$ and thus with probability at least $1 - mq^{-1}$.

We now assume that the matrix $\boldsymbol{\Upsilon}_{\boldsymbol{\alpha}^{(1)},\ldots,\boldsymbol{\alpha}^{(t)}} \in \mathbb{F}_q^{n\times m}$ contains some 0. By assumption, there exists $j \in \{1, \ldots, t\}$ such that $\alpha^{(j)}_{m_j} = n$ and in the following, it there exist two indices $j \in \{1, \ldots, t\}$ such that $\alpha^{(j)}_{m_j} = n$, we select one such index different from 1.

If $j = 1$, by assumption we have $\alpha^{(i)}_{m_i} < n$ for all $i \in \{2, \ldots, t\}$ and the last row of the matrix $\mathbf{S}$ has one coordinate equal to 1 and all other coordinates equal to 0. If $\boldsymbol{v} \in \mathbb{F}_q$ is of full Hamming weight $hw(\boldsymbol{v}) = m$, the last coordinate of the vector $\mathbf{S}\boldsymbol{v}$ is always non-null and the set $\{\boldsymbol{v} \in \mathbb{F}_q \text{ s.t. } hw(\boldsymbol{v}) = m \text{ and } \mathbf{S}\boldsymbol{v} = 0\}$ is therefore the empty set. It is thus included in the subspace of dimension 0 with probability at least $1 \geq 1 - mq^{-1}$. We therefore now assume that $j > 1$.

1. We first assume that $m_j > 1$ (*i.e.* that the column of index $m_1 + \cdots + m_j$ consists in $\alpha^{(j)}_{m_j-1} \geq 1$ zeroes followed by $\alpha^{(j)}_{m_j} - \alpha^{(j)}_{m_j-1} = n - \alpha^{(j)}_{m_j-1} \geq 1$ ones).
   We consider the submatrix of $\boldsymbol{\Upsilon}_{\boldsymbol{\alpha}^{(1)},\ldots,\boldsymbol{\alpha}^{(t)}} \in \mathbb{F}_q^{n\times m}$ obtained by deleting the column of index $m_1 + \cdots + m_j$ and the rows of indices i in $\{\alpha^{(j)}_{m_{j-1}+1}, \ldots, \alpha^{(j)}_{m_j}\}$. This submatrix is a block unitary progressive patterned matrix with $n' \leq n$ rows and $m-1$ columns. We can thus apply the induction hypothesis to the submatrix $\mathbf{S}'$ of $\mathbf{S}$ obtained by deleting the same column and the same rows. We know that with probability $1 - (m-1)q^{-1}$, there exist a linear subspace V' of dimension $m - 1 - n'$ that contains the set $\{\boldsymbol{v} \in \mathbb{F}_q^{m-1} \text{ s.t. } hw(\boldsymbol{v}) = m-1 \text{ and } \mathbf{S}'\boldsymbol{v} = 0\}$.
   If V' is of dimension 0, then $\{\boldsymbol{v} \in \mathbb{F}_q^{m-1} \text{ s.t. } hw(\boldsymbol{v}) = m-1 \text{ and } \mathbf{S}'\boldsymbol{v} = 0\} \subseteq \{0\}$ and the set is the empty set. We thus have $\{\boldsymbol{v} \in \mathbb{F}_q^{m}, hw(\boldsymbol{v}) = m \text{ and } \mathbf{S}\boldsymbol{v} = 0\} = \emptyset$ and with probability $1 - (m-1)q^{-1} \geq 1 - mq^{-1}$, there exist a linear subspace V of dimension $m - n$ that contains this set.
   If V' is of dimension $m - 1 - n' > 0$, we can assume without loss of generality that the deleted column of $\mathbf{S}$ to obtain $\mathbf{S}'$ was the last one in the last block (i.e., in a block where $\mathbf{S}$ is a random progressive patterned matrix since $j > 1$).

By permuting some rows and columns, we can write

$$\mathbf{S} = \begin{pmatrix} \mathbf{S}' & \mathbf{0_{n'\times 1}} \\ \mathbf{S}'' & \mathbf{u} \end{pmatrix}$$

where $\mathbf{S}'$ is a $(n-n') \times m-1$ matrix on which we can apply the induction hypothesis (since $m_j > 1$). Let $\boldsymbol{v} \in \mathbb{F}_q^m$ such that $hw(\boldsymbol{v}) = m$ and $\mathbf{S}\boldsymbol{v} = 0$.

We write $\boldsymbol{v} = \begin{pmatrix} \boldsymbol{w} \\ \tau \end{pmatrix}$ where $\boldsymbol{w} \in \mathbb{F}_q^{m-1}$ and $\tau \in \mathbb{F}_q$ is a scalar. We have $hw(\boldsymbol{w}) = m-1$ and $\mathbf{S}'\boldsymbol{w} = 0$, and therefore $\boldsymbol{w} \in V'$. Since $\tau \neq 0$ by assumption, the vector $\boldsymbol{u}$ thus belongs to the image W of V' by $\mathbf{S}''$ (with probability at least $1-(m-1)q^{-1}$). Since $j > 1$, note that $\boldsymbol{u}$ is a random vector in $\mathbb{F}_q^{n-n'}$ independent from $\mathbf{S}'$. We can then conclude as in the proof of Lemma 11.

2. We now assume that $m_i = 1$ for all i such that $\alpha_{m_i}^{(i)} = n$ for $i \in \{1,\ldots,t\}$ (*i.e.* that all the columns with a one in the last row consists only of ones).
   Since the matrix $\boldsymbol{\Upsilon}_{\boldsymbol{\alpha}^{(1)},\ldots,\boldsymbol{\alpha}^{(t)}} \in \mathbb{F}_q^{n\times m}$ contains some 0, there exists some $j \in \{2,\ldots,t\}$ such that $m_j > 1$ and we consider such a $j \in \{2,\ldots,t\}$ for which $\alpha_1^{(j)}$ is minimal.
   We consider the submatrix of $\boldsymbol{\Upsilon}_{\boldsymbol{\alpha}^{(1)},\ldots,\boldsymbol{\alpha}^{(t)}} \in \mathbb{F}_q^{n\times m}$ obtained by deleting the column of index $m_1 + \cdots + m_{j-1} + 1$ and the rows of indices in the set $\{1,\ldots,\alpha_1^{(j)}-1\}$. It is easy to see that this submatrix is a block unitary progressive patterned matrix with $n' \leq n-1$ rows and $m-1$ columns. We can thus apply the induction hypothesis to the submatrix $\mathbf{S}'$ of $\mathbf{S}$ obtained by deleting the same column and the same rows.
   We write $\boldsymbol{v} = \begin{pmatrix} \boldsymbol{w} \\ \tau \end{pmatrix}$ where $\boldsymbol{w} \in \mathbb{F}_q^{m-1}$ and $\tau \in \mathbb{F}_q$ is a scalar. We have $hw(\boldsymbol{w}) = m-1$ and $\mathbf{S}'\boldsymbol{w} = 0$, and therefore $\boldsymbol{w} \in V'$. Since $\tau \neq 0$ by assumption, the vector $\boldsymbol{u}$ thus belongs to the image W of V' by $\mathbf{S}''$ (with probability at least $1-(m-1)q^{-1}$). Since $j > 1$, note that $\boldsymbol{u}$ is a random vector in $\mathbb{F}_q^{n-n'}$ independent from $\mathbf{S}'$. We can then conclude as in the proof of Lemma 11.
   We know that with probability at least $1-(m-1)q^{-1}$, there exists a linear subspace $V' \subseteq \mathbb{F}_q^{m-1}$ of dimension $m-1-n'$ that contains the set $\{\boldsymbol{v} \in \mathbb{F}_q^{m-1} \text{ s.t. } hw(\boldsymbol{v}) = m-1 \text{ and } \mathbf{S}'\boldsymbol{v} = 0\}$.
   If V' is of dimension 0, then $\{\boldsymbol{v} \in \mathbb{F}_q^{m-1} \text{ s.t. } hw(\boldsymbol{v}) = m-1 \text{ and } \mathbf{S}'\boldsymbol{v} = 0\} \subseteq \{0\}$ and this set is thus the empty set. We then have $\{\boldsymbol{v} \in \mathbb{F}_q^m, hw(\boldsymbol{v}) = m \text{ and } \mathbf{S}\boldsymbol{v} = 0\} = \emptyset$ with probability at least $1-(m-1)q^{-1} \geq 1-mq^{-1}$, and so there exists a linear subspace V of dimension $m-n$ that contains this set.
   If V' is of dimension $m-1-n' > 0$, we can assume without loss of generality that the column of $\mathbf{S}$ deleted to obtain $\mathbf{S}'$ was the last one (by permuting the blocks of the matrix). We have the following block-decomposition of $\mathbf{S}$

$$\mathbf{S} = \begin{pmatrix} \mathbf{S}'' & \mathbf{u} \\ \mathbf{S}' & \mathbf{0_{n'\times 1}} \end{pmatrix}$$

   where $\mathbf{S}''$ is a $(n-n') \times (m-1)$ matrix and $\mathbf{u}$ a column vector of dimension $(n-n')$. Note that $\boldsymbol{u}$ is a random vector in $\mathbb{F}_q^{n-n'}$ independent from $\mathbf{S}'$ and $\mathbf{S}''$. Let $\boldsymbol{v} \in \mathbb{F}_q^m$ such that $hw(\boldsymbol{v}) = m$ and $\mathbf{S}\boldsymbol{v} = 0$. Since $j > 1$, note that $\boldsymbol{u}$ is a random vector in $\mathbb{F}_q^{n-n'}$ independent from $\mathbf{S}'$. We can then conclude as in the proof of Lemma 11.

This concludes the proof of the lemma. □

Using the same arguments as above for Case 1 (but replacing Lemma 11 by Lemma 12), we obtain that for any set $I \subseteq \{1, \dots, z\}$ of cardinal n such that $\mathbf{M}_I$ has no identically zero column vectors, with probability at least $1 - 2^n \cdot n \cdot q^{-1}$ over the choice of γ, we have $hw(\mathbf{L}_I \cdot \boldsymbol{v}) < n - K$ or $\mathbf{M}_I \cdot \boldsymbol{v} \neq \mathbf{0}_n$ for any vector $\boldsymbol{v} \in \mathbb{F}_q^n$ (where K denotes the number of coordinates $i_1, \dots, i_K \in [z]$ such that $v_{i_1} \neq 0, \dots, v_{i_K} \neq 0$ and the corresponding columns $i_1, \dots, i_K$ in $\mathbf{L}$ and in $\mathbf{M}$ have no zero coefficient). .

Case 3. We now consider the sets $I \subseteq \{1, \dots, n\} \cup \{(n+3) \cdot n+1, \dots, z\}$ of cardinal n for which $\mathbf{M}_I$ has some identically zero column vectors (i.e., $I \cap \{1, \dots, n\} \neq \emptyset$). For each $i \in I \cap \{1, \dots, n\} \neq \emptyset$, the i-th column in $\mathbf{L}$ is the i-th vector in the canonical basis of $\mathbb{F}_q^n$ (i.e., it corresponds to a probe of a value a_i). We can consider the submatrix of $\mathbf{M}_I$ and $\mathbf{L}_I$ in which we delete for each $i \in I \cap \{1, \dots, n\} \neq \emptyset$, the i-th column and the i-th row. We denote $\rho = \#I \cap \{1, \dots, n\} \neq \emptyset$.

Let us denote $\mathbf{M}'_I$ and $\mathbf{L}'_I$ the corresponding matrices (with $m' = m - \rho$ columns). These matrices are of the form handled in the previous *Case 2* (with $m' < m$). The previous argument shows therefore that with probability at least $1 - 2^n \cdot n \cdot q^{-1}$, we have $hw(\mathbf{L}'_I \cdot \boldsymbol{v}) < n - \rho - K$ or $\mathbf{M}'_I \cdot \boldsymbol{v} \neq \mathbf{0}_n$ for any vector $\boldsymbol{v} \in \mathbb{F}_q^{m'}$ (where K denotes the number of coordinates $i_1, \dots, i_K \in [z]$ such that $v_{i_1} \neq 0, \dots, v_{i_K} \neq 0$ and the corresponding columns $i_1, \dots, i_K$ in $\mathbf{L}$ and in $\mathbf{M}$ have no zero coefficient).

Going back to the original matrices $\mathbf{L}_I$ and $\mathbf{M}_I$ we have shown for any set $I \subseteq \{1, \dots, n\} \cup \{(n+3) \cdot n+1, \dots, z\}$ of cardinal n, with probability at least $1 - 2^n \cdot n \cdot q^{-1}$ over the choice of γ, we have $hw(\mathbf{L}_I \cdot \boldsymbol{v}) < n - K$ or $\mathbf{M}_I \cdot \boldsymbol{v} \neq \mathbf{0}_n$ for any vector $\boldsymbol{v} \in \mathbb{F}_q^n$ (indeed a vector $\boldsymbol{v}$ satisfies $\mathbf{M}_I \cdot \boldsymbol{v} = \mathbf{0}_n$ if an only if $\mathbf{M}'_I \cdot \boldsymbol{v'} = \mathbf{0}_n$ where $\boldsymbol{v'}$ denotes the restriction of $\boldsymbol{v}$ to the support $I \cap \{1, \dots, n\}$ and the Hamming weight of $hw(\mathbf{L}_I \cdot \boldsymbol{v})$ is at smaller than $hw(\mathbf{L}_I \cdot \boldsymbol{v'}) + \rho$ since at most ρ positions can be set arbitrarily..

Case 4. We now consider all sets $I \subseteq \{1, \dots, z\}$ (with no restrictions). Without loss of generality, we can assume that all not identically zero column vectors in $\mathbf{M}_I$ are pairwise distinct. Indeed, if two columns are equal, they come either from the two submatrices I_n of $\mathbf{M}$, or from the first column vectors of a submatrix I_n and the submatrix $\mathbf{T}_n$, or from the first column vectors of a submatrix $\mathbf{D}_{\boldsymbol{\gamma},i}$ for some $i \in \{1, \dots, n\}$ and the corresponding submatrix $\mathbf{T}_{\boldsymbol{\gamma},i}$. In all these cases, one can replace the index of the second vector in I by an index in $\{1, \dots, n-1\}$ (and modify the vector accordingly) in such a way that $\mathbf{M}_{I'}$ for the new set I' has a new null column vector for each duplicate in the original matrix $\mathbf{M}_I$.

We can now delete the columns corresponding to the null vectors as in Case 3 (i.e., for each $i \in I \cap \{1, \dots, n+1\} \neq \emptyset$, the i-th column and the i-th row in $\mathbf{M}_I$ and $\mathbf{L}_I$). The only difference occurs if a column in $\mathbf{M}_I$ is equal to the i-th vector in the canonical basis (for $i \geq 2$) or to the scalar multiplication of this vector by some element of the matrix $\gamma \in \mathbb{F}_q$ (corresponding to the cases $I \cap \{n+1, \dots, 2n\} \neq \emptyset$ and $I \cap \{2n+1, \dots, (n+3) \cdot n+1\} \neq \emptyset$ respectively). As in Case 3, we can delete the corresponding column and row in $\mathbf{M}_I$ and $\mathbf{L}_I$ (i.e., it corresponds to a probe of a value r_i, a value $a_i + r_i$ or a value $a_i + \gamma_{j,i} r_i$).

As above, if we denote $\mathbf{M}'_I$ and $\mathbf{L}'_I$ the corresponding matrices (with m' columns and $n' < n$ and $n'+1$ rows, respectively), the previous argument shows that with probability at least $1 - 2^n \cdot n \cdot q^{-1}$, we have $hw(\mathbf{L}'_I \cdot \boldsymbol{v}) < n' - K$ or $\mathbf{M}'_I \cdot \boldsymbol{v} \neq \mathbf{0}_n$ for any vector $\boldsymbol{v} \in \mathbb{F}_q^{m'}$ (where K denotes the number of coordinates $i_1, \dots, i_K \in [z]$ such that $v_{i_1} \neq 0, \dots, v_{i_K} \neq 0$ and the corresponding columns $i_1, \dots, i_K$ in $\mathbf{L}$ and in $\mathbf{M}$ have no zero coefficient).

Going back to the original matrices $\mathbf{L}_I$ and $\mathbf{M}_I$ we have shown for any set $I \subseteq \{1, \ldots, z\}$ of cardinal n, with probability at least $1 - 2^n \cdot n \cdot q^{-1}$ over the choice of γ, we have $hw(\mathbf{L}_I \cdot \boldsymbol{v}) < n - K$ or $\mathbf{M}_I \cdot \boldsymbol{v} \neq \mathbf{0}_n$ for any vector $\boldsymbol{v} \in \mathbb{F}_q^n$

Conclusion . By the union on all such sets, we obtain that the probability that, for $\boldsymbol{\gamma}$ picked uniformly at random in $\mathbb{F}_q^{n \times n}$, the matrix $\mathbf{M}$ satisfies Condition 3, i.e., for any vector $\boldsymbol{v} \in \mathbb{F}_q^z$ of Hamming weight $hw(\boldsymbol{v}) \leq n$ we have $hw(\mathbf{L} \cdot \boldsymbol{v}) < n - K$ or $\mathbf{M} \cdot \boldsymbol{v} \neq \mathbf{0}_n$ is at least

$$1 - \binom{z}{n} 2^n \cdot n \cdot q^{-1} = 1 - \binom{(2n+4) \cdot n + 1}{n} 2^n \cdot n \cdot q^{-1}.$$

The binomial coefficient in this lower-bound is always less than $(6n)^n$ (this can be checked by hand for small values of n and it follows for large values using the classical upper-bound $\binom{r}{s} \leq ((r \cdot \exp(1))/s)^s$). We thus obtain the claimed bounds and this concludes the proof. $\square$

A.13 Instantiations

In this paragraph, we present explicit matrices obtained following [19] that achieve our Condition 3 and can thus be used to instantiate our new multiplication gadget.

A first matrix for 3 shares can be used over the finite field $\mathbb{F}_{2^5}$ represented as $\mathbb{F}_2[X]/(X^5+X^2+1)$:

$$\boldsymbol{\gamma} = \begin{pmatrix} X+1 & X & X^2+1 \\ X & X^2+1 & X+1 \\ X^2+1 & X+1 & X \end{pmatrix}$$

Another matrix for 3 shares (denoted in hexadecimal by evaluating each polynomial at $X = 2$ and writing the result in base 16) can be used over the finite field $\mathbb{F}_{2^6}$ represented as $\mathbb{F}_2[X]/(X^6+X+1)$:

$$\boldsymbol{\gamma} = \begin{pmatrix} \texttt{36} & \texttt{30} & \texttt{1d} \\ \texttt{21} & \texttt{05} & \texttt{1a} \\ \texttt{35} & \texttt{31} & \texttt{1b} \end{pmatrix}$$

Another example for 4 shares can be instantiated using the following matrix (also denoted in hexadecimal) over the (AES) finite field $\mathbb{F}_{2^8}$ represented as $\mathbb{F}_2[X]/(X^8 + X^4 + X^3 + X + 1)$:

$$\boldsymbol{\gamma} = \begin{pmatrix} \texttt{2d} & \texttt{f5} & \texttt{2e} & \texttt{23} \\ \texttt{e1} & \texttt{c3} & \texttt{ac} & \texttt{30} \\ \texttt{bd} & \texttt{f6} & \texttt{fa} & \texttt{8a} \\ \texttt{e6} & \texttt{4a} & \texttt{4d} & \texttt{ab} \end{pmatrix}$$

Eventually, we present a matrix for 5 shares over the finite field $\mathbb{F}_{2^{10}}$ represented as $\mathbb{F}_2[X]/(X^{10}+X^3+1)$:

$$\boldsymbol{\gamma} = \begin{pmatrix} \texttt{225} & \texttt{2a9} & \texttt{0d0} & \texttt{224} & \texttt{2dd} \\ \texttt{254} & \texttt{11b} & \texttt{325} & \texttt{3a6} & \texttt{219} \\ \texttt{3d2} & \texttt{2bc} & \texttt{2bf} & \texttt{3a2} & \texttt{2a1} \\ \texttt{2af} & \texttt{311} & \texttt{295} & \texttt{26b} & \texttt{11d} \\ \texttt{16c} & \texttt{124} & \texttt{158} & \texttt{319} & \texttt{0b8} \end{pmatrix}$$