On the Power of Expansion: More Efficient Constructions in the Random Probing Model

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

[1] CryptoExperts, France
[2] Sorbonne Université, CNRS, LIP6, F-75005 Paris, France

{sonia.belaid,matthieu.rivain,abdul.taleb}@cryptoexperts.com

Abstract. The random probing model is a leakage model in which each wire of a circuit leaks with a given probability p. This model enjoys practical relevance thanks to a reduction to the noisy leakage model, which is admitted as the right formalization for power and electromagnetic side-channel attacks. In addition, the random probing model is much more convenient than the noisy leakage model to prove the security of masking schemes. In a recent work, Ananth, Ishai, and Sahai (CRYPTO 2018) introduce a nice expansion strategy to construct random probing secure circuits. Their construction tolerates a leakage probability of 2^{-26}, which is the first quantified achievable leakage probability in the random probing model. In a follow-up work, Belaïd, Coron, Prouff, Rivain, and Taleb (CRYPTO 2020) generalize their idea and put forward a complete and practical framework to generate random probing secure circuits. The so-called expanding compiler can bootstrap simple base gadgets as long as they satisfy a new security notion called *random probing expandability* (RPE). They further provide an instantiation of the framework which tolerates a 2^{-8} leakage probability in complexity $\mathcal{O}(\kappa^{7.5})$ where κ denotes the security parameter.
In this paper, we provide an in-depth analysis of the RPE security notion. We exhibit the first upper bounds for the main parameter of a RPE gadget, which is known as the *amplification order*. We further show that the RPE notion can be made tighter and we exhibit strong connections between RPE and the *strong non-interference* (SNI) composition notion. We then introduce the first generic constructions of gadgets achieving RPE for any number of shares and with nearly optimal amplification orders and provide an asymptotic analysis of such constructions. Last but not least, we introduce new concrete constructions of small gadgets achieving maximal amplification orders. This allows us to obtain much more efficient instantiations of the expanding compiler: we obtain a complexity of $\mathcal{O}(\kappa^{3.9})$ for a slightly better leakage probability, as well as $\mathcal{O}(\kappa^{3.2})$ for a slightly lower leakage probability.

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

1 Introduction

Most commonly used cryptographic algorithms are assumed to be secure against *black-box* attacks, when the adversary is limited to the knowledge of some inputs and outputs. However, as revealed in the late nineties [18], their implementation on physical devices can be vulnerable to the more powerful *side-channel attacks*. The latter additionally exploit the physical emanations of the underlying device such as the execution time or the device temperature, power consumption, or electromagnetic radiations during the algorithm execution.

To counteract side-channel attacks which often only require cheap equipment and can be easily mounted in a short time interval, the cryptographic community has searched for efficient countermeasures. Among the different approaches, one of the most widely used is known as *masking*. Simultaneously introduced by Chari, Jutla, Rao and Rohatgi [10], and by Goubin and Patarin [16] in 1999, it happens to be strongly related to techniques usually applied in secure multi-party computation. In a nutshell, the idea is to split each sensitive variable of the implementation into n

shares such that $n-1$ of them are generated uniformly at random whereas the last one is computed as a combination of the original value and the random shares. Doing so, one aims to ensure that an adversary cannot recover the secret without knowledge of all the shares. When the shares are combined by bitwise addition, the masking is said to be *Boolean*, and it enjoys simple implementation for linear operations which can be simply applied on each share separately. However, things are trickier for non-linear operations for which it is impossible to compute the result without combining shares.

In order to reason about the security of these countermeasures, the community has introduced a variety of models. Among them, the *probing model* introduced by Ishai, Sahai, and Wagner in 2003 [17] is well suited to analyze the security of masked implementations. Basically, it assumes that an adversary is able to get the exact values of a certain number t of intermediate variables in an implementation. This way, it captures the increasing difficulty of combining noisy leakage to recover secrets. Despite its wide use by the community [20, 13, 11, 8, 12], the probing model raised a number of concerns regarding its relevance in practice. Therefore, in 2013, Prouff and Rivain introduced a general and practical model, known as the *noisy leakage model* [19]. This model well captures the reality of embedded devices by assuming that all the manipulated data leak together with some noise. Unfortunately, proving the security of a masking scheme in this model is rather tedious, which is why Duc, Dziembowski, and Faust provided in 2014 a reduction showing that a scheme secure in the probing model is also secure in the noisy leakage model [14].

This reduction is based on an intermediate leakage model, known as *random probing model*, to which the security in the noisy leakage model tightly reduces. In this model, every wire of a circuit is assumed to leak with some constant leakage probability. Then, a circuit is secure if there is a negligible probability that these leaking wires actually reveal information on the secrets. Compared to the probing model, the random probing model is closer to the noisy leakage model and, in particular, captures *horizontal attacks* which exploit the repeated manipulations of variables throughout the implementation. Classical probing secure schemes are also secure in the random probing model but the tolerated leakage probability (a.k.a. leakage rate) might not be constant which is not satisfactory from a practical viewpoint. Indeed, in practice, the leakage probability translates to some side-channel noise amount which might not be customizable by the implementer.

So far, only a few constructions [1, 3, 2, 9] tolerate a constant leakage probability. The two former ones [1, 3] are based on expander graphs and the tolerated probability is not made explicit. The third construction [2] is based on multi-party computation protocols and an expansion strategy. It reaches a tolerated leakage probability of around 2^{-26} for a complexity of $\mathcal{O}(\kappa^{8.2})$ for some security parameter κ, as computed by the authors of [9]. Finally, the more recent construction [9] relies on masking gadgets and a similar expansion strategy and reaches a tolerated leakage probability of 2^{-8} for a complexity of $\mathcal{O}(\kappa^{7.5})$. While obtaining such quantified tolerated leakage probability is of great practical interest, the obtained complexity is high which makes this construction hardly practical.

Besides their explicit construction, the authors of [9] provide a complete and practical framework to generate random probing secure implementations. Namely, they formalize the *expanding compiler* which produces a random probing secure version of any circuit from three base gadgets (for addition, copy, and multiplication) achieving a *random probing expandability* (RPE) property. The advantage of this approach is that it enables to bootstrap small gadgets (defined for a small number of shares) into a circuit achieving arbitrary security in the random probing model while tolerating a constant and quantified leakage probability. Although the concrete results of [9] in terms of complexity and

tolerated leakage probability are promising, the authors left open the analysis of this RPE property and the design of better gadgets in this paradigm.

Our contributions. In this paper, we provide an in-depth analysis of the random probing expandability security notion. We first provide some upper bounds for the *amplification order* of an RPE gadget, which is the crucial parameter in view of a low-complexity instantiation of the expanding compiler. We further show that the RPE notion can be made tighter and we exhibit strong relations between RPE and the *strong non-interference* (SNI) composition notion for probing-secure gadgets.

From these results, we introduce the first generic constructions of gadgets achieving RPE for any number of shares and with nearly optimal amplification orders. These generic gadgets are derived from the widely known Ishai-Sahai-Wagner (ISW) construction. We show that the obtained expanding compiler can approach a quadratic complexity depending on the leakage probability that must be tolerated: the smaller the leakage probability, the closer the complexity to $\mathcal{O}(\kappa^2)$. We further introduce a new multiplication gadget achieving the optimal amplification order, which allows us to improve the convergence to a quadratic complexity.

Finally, we provide new concrete constructions of copy, addition, and multiplication gadgets achieving maximal amplification orders for small numbers of shares. These gadgets yield much more efficient instantiations than all the previous schemes (including the analysed ISW-based constructions). While slightly improving the tolerated leakage probability to $p = 2^{-7.5}$, our 3-share instantiation achieves a complexity of $\mathcal{O}(\kappa^{3.9})$. For a slightly lower leakage probability, our 5-share instantiation drops the complexity to $\mathcal{O}(\kappa^{3.2})$.

We thus achieve a significant step forward in the quest for efficient random probing secure schemes that tolerate a quantified leakage probability. Besides our concrete instantiations, our work introduces several tools (new bounds, relations, and generic gadgets) that shall be instrumental for future constructions.

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 $\boldsymbol{x}|_I = (x_i)_{i \in I}$.

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 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. In this paper, we shall consider gadgets for three types of functions (corresponding to the three types of gates): the addition $g : (x_1, x_2) \mapsto x_1 + x_2$, the multiplication $g : (x_1, x_2) \mapsto x_1 \cdot x_2$ and the copy $g : x \mapsto (x, x)$. We shall generally denote such gadgets G_{add}, G_{mult} and G_{copy} respectively.

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 (and leaks nothing otherwise), 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 $\mathcal{W}$, denoted as

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

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

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

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

Definition 2 (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 3 (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* $\boldsymbol{x} \in \mathbb{K}^\ell$*:*

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

2.3 Expanding Compiler

In [2], Ananth, Ishai and Sahai propose an *expansion* approach to build a random-probing-secure circuit compiler from a secure multiparty 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 in the input circuit by the corresponding gadget. More specifically, assume we have three n-share gadgets G_{add}, G_{mult}, G_{copy}, for the addition, the multiplication, and the copy on $\mathbb{K}$. The base compiler CC simply consists in replacing each addition gate in the original gadget by G_{add}, each multiplication gate by G_{mult}, and each copy gate by G_{copy}, and by replacing each wire by n wires carrying a sharing of the original wire. One can derive three new n^2-share gadgets by simply applying CC to each gadget: $G^{(2)}_{\text{add}} = \mathsf{CC}(G_{\text{add}})$, $G^{(2)}_{\text{mult}} = \mathsf{CC}(G_{\text{mult}})$, and $G^{(2)}_{\text{copy}} = \mathsf{CC}(G_{\text{copy}})$. Doing so, we obtain n^2-share gadgets for the addition, multiplication, and copy on $\mathbb{K}$. 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 is replaced by a base gadget G_{add}, G_{mult}, or G_{copy}. The second output circuit $\widehat{C}_2$ is the original circuit C in which each gate is replaced by an n^2-share gadget $G^{(2)}_{\text{add}}$, $G^{(2)}_{\text{mult}}$, or $G^{(2)}_{\text{copy}}$ as defined above. 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.

This expanding compiler achieves random probing security if the base gadgets verify a property called *random probing expandability* [9].

2.4 Random Probing Expandability

We recall hereafter the original definition of the random probing expandability (RPE) property for 2-input 1-output gadgets.

Definition 4 (Random Probing Expandability [9]). *Let $f : \mathbb{R} \to \mathbb{R}$. An n-share 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*

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

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*

$$out \stackrel{id}{=} \left(\mathsf{AssignWires}(G, \mathcal{W}, (\widehat{x}, \widehat{y}))\ ,\ \widehat{z}|_{J'}\right) \tag{3}$$

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. Eventually, the RPE notion can also be extended to gadgets with an arbitrary number ℓ of inputs. The Sim_1^G simulator then produces ℓ sets $I_1, \ldots, I_\ell$ so that the corresponding failures ($|I_1| > t$), ..., ($|I_\ell| > t$) occur with probability ε and are additionally mutually independent. The Sim_2^G simulator then simply gets use of the shares of each input as designated respectively by the corresponding sets $I_1, \ldots, I_\ell$.

Note that as explained in [9], the requirement of the RPE notion on the mutual independence of the failure events might seem too strong. We can actually use the proposed relaxation referred to as *weak random probing expandability*. Namely, the equalities (Equation (2)) are replaced by inequalities as upper bounds are sufficient in our context. We refer the reader to [9] for the concrete reduction, which does not impact the amplification orders.

2.5 Complexity of the Expanding Compiler

We start by recalling the definition of the *amplification order* of a function and of a gadget.

Definition 5 (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 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 three gadgets used in the base compiler CC. If the latter achieves (t, f)-RPE with an amplification order d, the expanding compiler achieves $(p, 2^{-\kappa})$-random probing security with a complexity blowup of $\mathcal{O}(\kappa^e)$ for an exponent e satisfying

$$e = \frac{\log N_{\max}}{\log d} \tag{4}$$

with

$$N_{\max} = \max\left(N_{\mathrm{m,m}}\ ,\ \mathsf{eigenvalues}\left(\begin{pmatrix} N_{\mathrm{a,a}} & N_{\mathrm{c,a}} \\ N_{\mathrm{a,c}} & N_{\mathrm{c,c}} \end{pmatrix}\right)\right) \tag{5}$$

where $N_{\mathrm{x,y}}$ denotes the number of gates "x" in a gadget "y", with "m" meaning multiplication, "a" meaning addition, and "c" meaning copy. As an illustration, the instantiation proposed in [9] satisfies $N_{\max} = 21$ and $d = \frac{3}{2}$ which yields an asymptotic complexity of $\mathcal{O}(\kappa^{7.5})$.

Finally, we recall the notion of maximum *tolerated leakage probability* 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. The instantiation proposed in [9] tolerates a leakage probability up to $2^{-7.80}$.

3 Bounding the Amplification Order

As recalled above, the amplification order of a gadget is a crucial parameter of its random probing expandability. The higher the amplification order, the lower the asymptotic complexity of the expanding compiler, *ceteris paribus*. A natural question which was left open in [9] is to determine the best amplification order that can be hoped for given the different parameters of a gadget. In this section, we exhibit concrete upper bounds on the amplification order that can be achieved by a gadget depending on its input-output dimensions (ℓ, m), its number of shares n, and its RPE threshold t.

Before giving the bounds let us make a key observation on the amplification order of a gadget. Let G be a 2-to-1 n-share gadget achieving (t, f)-RPE. A subset $\mathcal{W}$ of the wires of G is said to be a *failure set* with respect to the first input (resp. the second input) if there exists a set $J \subseteq [n]$ such that $(I_1, I_2, J') \leftarrow \mathsf{Sim}_1^G(\mathcal{W}, J)$ implies $|I_1| > t$ (resp. $|I_2| > t$), namely if a leaking set $\mathcal{W}$ implies the failure event $\mathcal{F}_1$ (resp. $\mathcal{F}_2$) in the definition of RPE. One can check that G has amplification order $d \leq d_{up}$ if one of the two following events occurs:

1. there exists a failure set $\mathcal{W}$ w.r.t. the first input *or* the second input such that $|\mathcal{W}| = d_{up}$,
2. there exists a failure set $\mathcal{W}$ w.r.t. the first input *and* the second input such that $|\mathcal{W}| = 2d_{up}$.

In the former case, the existence of the failure set implies that the function $f(p)$ has a non-zero coefficient in $p^{d_{up}}$ and hence $d \leq d_{up}$. In the latter case, the existence of the double failure set implies that the function $f^2(p)$ has a non-zero coefficient in $p^{2d_{up}}$ and hence $d \leq d_{up}$. The case of a single-input gadget is simpler: it has amplification order $d \leq d_{up}$ if there exists a failure set $\mathcal{W}$ (w.r.t. its single input) such that $|\mathcal{W}| = d_{up}$.

We start by exhibiting a generic upper bound for the amplification order and then look at the particular case of what we shall call a *standard* multiplication gadget.

3.1 Generic Upper Bound

In the following we will say that a function $g : \mathbb{K}^\ell \to \mathbb{K}^m$ is *complete* if at least one of its m outputs is functionally dependent on the ℓ inputs. Similarly, we say that a gadget G is complete if its underlying function g is complete.

The following lemma gives our generic upper bound on the amplification order.

Lemma 1. *Let $f : \mathbb{R} \to \mathbb{R}$, $n \in \mathbb{N}$ and $\ell, m \in \{1, 2\}$. Let $G : (\mathbb{K}^n)^\ell \to (\mathbb{K}^n)^m$ be an ℓ-to-m n-share complete gadget achieving (t, f)-RPE. Then its amplification order d is upper bounded by*

$$\min((t+1), (3-\ell) \cdot (n-t)).$$

Proof. The first part of the bound on the amplification order $d \leq (t+1)$ is immediate since by probing $t+1$ shares of any input, the considered set will be a failure set of cardinality $t+1$. We then consider two cases depending on the number of inputs:

1. *1-input gadgets ($\ell = 1$):* We show that we can exhibit a failure set of size $2(n-t)$. Let us denote the output shares $z_1, \ldots, z_n$ (for two-output gadgets, *i.e.* $m = 2$, $z_1, \ldots, z_n$ can be any of the output sharings). In the evaluation of the (t, f)-RPE property, t shares among the z_i's (corresponding to the set J) must be simulated. Without loss of generality, let $z_1, \ldots, z_t$ be those shares (*i.e.* $J = [t]$). By including both input gates of each of the remaining output shares $z_{t+1}, \ldots, z_n$ in the set $\mathcal{W}$, the distribution to be simulated requires the knowledge of the full input (by completeness of the gadget). The set $\mathcal{W}$ is thus a failure set with $2(n-t)$ elements.
2. *2-input gadgets ($\ell = 2$):* Considering the same failure set as in the above case, the simulation of *out* requires the full two input sharings. Hence $\mathcal{W}$ is a failure set of size $2(n-t)$ with respect to the two inputs, and so the amplification order satisfies $d \leq (n-t)$.

We hence conclude that $d \leq \min((t+1), 2(n-t))$ for one-input gadgets, and $d \leq \min((t+1), (n-t))$ for two-input gadgets. □

Corollary 1 (One-input gadget). *The amplification order d of a one-input gadget achieving (t, f)-RPE is upper bounded by*

$$d \leq \frac{2(n+1)}{3} \ .$$

The above corollary directly holds from Lemma 1 for a RPE threshold $t = \frac{2n-1}{3}$ (which balances the two sides of the min).

Corollary 2 (Two-input gadget). *The amplification order d of a two-input gadget achieving (t, f)-RPE is upper bounded by*

$$d \leq \frac{n+1}{2} \ .$$

The above corollary directly holds from Lemma 1 for a RPE threshold $t = \frac{n-1}{2}$ (which balances the two sides of the min).

We deduce from the two above corollaries that for a circuit composed of addition, multiplication and copy gadgets, the amplification order is upper bounded

$$d \leq \min\left(\frac{2(n+1)}{3}, \frac{n+1}{2}\right) = \frac{n+1}{2} \ ,$$

which can only be achieved for an odd number of shares by taking $t = \frac{n-1}{2}$ as RPE threshold.

3.2 Upper Bound for Standard Multiplication Gadgets

The generic bound exhibited above is not tight in the special case of a standard multiplication gadget which computes cross products between the input shares, such as the ISW multiplication gadget [17]. We exhibit hereafter a tighter bound for such gadgets.

Formally, a n-share multiplication gadget G is a *standard multiplication gadget*, if on input $(\boldsymbol{x}, \boldsymbol{y}) \in (\mathbb{K}^n)^2$, G computes the cross products $x_i \cdot y_j$ for $1 \leq i, j \leq n$. Our upper bound on the amplification order for such gadgets is given in the following lemma.

Lemma 2. *Let $f : \mathbb{R} \to \mathbb{R}$ and $n \in \mathbb{N}$. Let G be an n-share standard multiplication gadget achieving (t, f)-RPE. Then its amplification order d is upper bounded by*

$$d \leq \min\left(\frac{t+1}{2}, (n-t)\right).$$

Proof. The second part of the bound $(n-t)$ holds directly from Lemma 1. We now prove the bound $(t+1)/2$ by exhibiting a failure set of size $t+1$ with t output shares, which will be a failure on both inputs. Let $\{m_{ij}\}_{0 \leq i,j \leq n}$ denote the cross products such that $m_{ij} = x_i \cdot y_j$. Consider a set $\mathcal{W}$ made of $t+1$ such variables $\{m_{ij}\}$ for which the indexes i and j are all distinct. Specifically, $\mathcal{W} = \{x_{i_1} \cdot y_{j_1}, \ldots, x_{i_{t+1}} \cdot y_{j_{t+1}}\}$ such that $\{i_\ell\}_{1 \leq \ell \leq t+1}$ and $\{j_\ell\}_{1 \leq \ell \leq t+1}$ are both sets of $(t+1)$ distinct indexes. Clearly, such a set is a failure set for both inputs $\boldsymbol{x}$ and $\boldsymbol{y}$ since it requires $t+1$ shares of each of them to be perfectly simulated (even without considering the output shares to be also simulated). We hence have a double failure set of cardinality $t+1$ which implies the $(t+1)/2$ upper bound on the amplification order. □

The above lemma implies that the highest amplification order for standard multiplication gadgets might be achieved for a RPE threshold $t = \frac{2n-1}{3}$ which yields the following maximal upper bound:

$$d \leq \frac{n+1}{3},$$

which is lower than the generic upper bound for 2-to-1 gadgets exhibited in Corollary 2. This loss suggests that better amplification orders could be achieved for multiplication gadgets that do not compute direct cross products of the input shares. We actually provide new constructions of multiplication gadgets avoiding this loss in Section 5.

4 A Closer Look at Random Probing Expandability

In this section, we give a closer look at the RPE notion. We first show that it naturally splits into two different notions, that we shall call RPE1 and RPE2, and further introduce a tighter variant which will be useful for our purpose. We then study the relations between (tight) RPE and the *Strong Non-Interference* (SNI) notion used for probing security. We exhibit strong connections between (tight) RPE1 and SNI, which will be very useful for our constructive results depicted in Section 5.

4.1 Splitting RPE

From Definition 4, we can define two sub-properties which are jointly equivalent to RPE. In the first one, designated by RPE1, the set J is constrained to satisfy $|J| \leq t$ and $J' = J$ (the simulator does not choose J'). In the second one, designated by RPE2, J' is chosen by the simulator such that $J' \subseteq [n]$ with $|J'| = n-1$ (and J does not matter anymore). For the sake of completeness, these two notions are formally defined in Appendix A.

This split is somehow a partition of the RPE notion since we have:

$$G \text{ is } (t, f)\text{-RPE} \iff G \text{ is } (t, f)\text{-RPE1 } \textit{and } G \text{ is } (t, f)\text{-RPE2}$$

for any gadget G. As a result of the above equivalence, we can show that a gadget achieves RPE1 and RPE2 independently in order to obtain RPE for this gadget. Formally, we use the following lemma.

Lemma 3. *An n-share gadget $G : \mathbb{K}^n \times \mathbb{K}^n \to \mathbb{K}^n$ which is (t, f_1)-RPE1 and (t, f_2)-RPE2 is also (t, f)-RPE with $f(p) \geq \max(f_1(p), f_2(p))$ for every $p \in [0, 1]$.*

We can refine the upper bounds introduced in Section 3 with respect to this split. In Lemma 1, the bound $d \leq t + 1$ applies to both RPE1 and RPE2, while the bound $d \leq (3 - \ell) \cdot (n - t)$ only applies to RPE1. Similarly, in Lemma 2, the bound $d \leq (t+1)/2$ applies to both RPE1 and RPE2, while the bound $d \leq (n - t)$ only applies to RPE1.

4.2 Tightening RPE

We introduce a tighter version of the RPE security property. The so-called *tight random probing expandability* (TRPE) is such that a failure occurs when the simulation requires more than t input shares (as in the original RPE notion) but also whenever this number of shares is greater than the size of the leaking set $\mathcal{W}$. Formally, the failure event $\mathcal{F}_j$ is defined as

$$\mathcal{F}_j \equiv \big(|I_j| > \min(t, |\mathcal{W}|)\big)$$

for every $j \in [\ell]$.

This tighter security property will be instrumental in the following to obtain generic RPE constructions. Similarly to the original RPE property, the TRPE property can be split into two intermediate properties, namely TRPE1 and TRPE2 and Lemma 3 also applies to the case of TRPE. Moreover the upper bounds on the amplification order for RPE in Lemmas 1 and 2 further apply to the amplification order for TRPE (which holds by definition). The formal TRPE, TRPE1, and TRPE2 definitions are given in Appendix B for the sake of completeness.

We show hereafter that the TRPE notion is actually equivalent to the RPE notion if and only if the function f is of maximal amplification order $t + 1$.

Lemma 4. *Let $t \in \mathbb{N}$, let $f : \mathbb{R} \to \mathbb{R}$ of amplification order d. Let G be a gadget.*

1. *If G achieves (t, f)-TRPE, then it achieves (t, f')-RPE for some $f' : \mathbb{R} \to \mathbb{R}$ of amplification order $d' \geq d$.*
2. *If G is of amplification order d with respect to t (i.e. d is the max amplification order of a function f for which G is (t, f)-RPE), then for all $f' : \mathbb{R} \to \mathbb{R}$ for which G achieves (t, f')-TRPE, f' is of amplification order $d' \leq d$.*
3. *If $d = t + 1$, then G achieves (t, f)-TRPE if and only if G achieves (t, f)-RPE.*

Proof. The proof for the first two points is easy. In particular, for the first point, if G achieves TRPE with an amplification order of d, then G achieves RPE with amplification order at least d, since a failure in the TRPE setting *i.e.* $|I_j| > \min(t, |\mathcal{W}|)$ does not necessarily imply a failure in the RPE setting *i.e.* $|I_j| > t$, meanwhile if there is no failure for TRPE for a leaking set of wires $\mathcal{W}$, then this implies that $|I_j| \leq \min(t, |\mathcal{W}|) \leq t$ so there is no failure in the RPE setting either.

As for the second point, the proof is similar: if G achieves an amplification of d in the RPE setting, then it achieves an amplification order of at most d in the TRPE setting, since a failure in the RPE setting *i.e.* $|I_j| > t$ immediately implies a failure in the TRPE setting $|I_j| > \min(t, |\mathcal{W}|)$. But also, even if there is no failure for a leaking set of wires $\mathcal{W}$ in the RPE setting we might still have a failure in the TRPE setting for the same set $\mathcal{W}$. This is mainly the case where $\mathcal{W}$ can be simulated with sets of input shares I_j such that $|\mathcal{W}| < |I_j| \leq t$, so we have $|I_j| \leq t$ (*i.e.* no failure

for RPE) and $|I_j| > \min(t, |\mathcal{W}|) = |\mathcal{W}|$ (*i.e.* failure on TRPE). This concludes the proof for the second point.

We will now prove the third point. Let $d = t + 1$. We will show that for every set $J' \subseteq [n]$ of output shares and every leaking set of wires $\mathcal{W}$, a failure occurs in the TRPE setting if and only if a failure also occurs in the RPE setting. If $|\mathcal{W}| \geq t$, then the two settings are equivalent since $\min(t, |\mathcal{W}|) = t$. We will thus only focus on the case $|\mathcal{W}| < t$. Clearly, a failure in the RPE setting, *i.e.* $|I_j| > t$, implies a failure in the TRPE setting, *i.e.* $|I_j| > \min(t, |\mathcal{W}|)$. Let us now show that the converse is also true.

We assume by contradiction that there exists J' and $\mathcal{W}$ implying a TRPE failure which is not an RPE failure, that is a set I_j satisfying $|\mathcal{W}| < |I_j| \leq t$. We then show that there exists a leaking set $\mathcal{W}'$ of size $|\mathcal{W}'| < t+1$ for which an RPE failure always occurs, which implies an amplification order strictly lower than $t + 1$ and hence contradicts the lemma hypothesis. This set $\mathcal{W}'$ is constructed as $\mathcal{W}' = \mathcal{W} \cup I'_j$ for some set $I'_j \subset [n] \setminus I_j$ such that $|I'_j| = t + 1 - |I_j|$. The simulation of $\mathcal{W}'$ and J' then requires the input shares from $I_j \cup I'_j$. However, we have

$$|I_j \cup I'_j| = |I_j| + |I'_j| = t + 1$$

implying an RPE failure, and

$$|\mathcal{W}'| = |\mathcal{W} \cup I'_j| \leq |\mathcal{W}| + |I'_j| = |\mathcal{W}| + t + 1 - |I_j| < |\mathcal{W}| + t + 1 - |\mathcal{W}| = t + 1.$$

Thus, we have built a failure set $\mathcal{W}'$ of size strictly less than the amplification order $t + 1$, which contradicts the hypothesis and hence concludes the proof. □

The above proof also applies to the case of the split notions, specifically for ((t, f)-RPE1, (t, f)-TRPE1) and for ((t, f)-RPE2, (t, f)-TRPE2).

4.3 Unifying (Tight) RPE and SNI

Strong non-interference (SNI) is a widely used notion to compose probing-secure gadgets [5]. In [9], the authors exhibit a relation between the SNI and the *random probing composability* (RPC) property in their Proposition 1. We go one step further and study the relation between SNI and (T)RPE.

We state hereafter some equivalence results between the (T)RPE1 and SNI notions, up to some constraints on the parameters. Let us first recall the definition of the SNI notion.

Definition 6 (Strong Non-Interference (SNI)). *Let n, ℓ and τ be positive integers. An n-share gadget $G : (\mathbb{K}^n)^\ell \to \mathbb{K}^n$ is τ-SNI if there exists a deterministic algorithm Sim_1^G and a probabilistic algorithm Sim_2^G such that for every set $J \subseteq [n]$ and subset $\mathcal{W}$ of wire labels from G satisfying $|\mathcal{W}| + |J| \leqslant \tau$, the following random experiment with any $\widehat{\boldsymbol{x}} \in (\mathbb{K}^n)^\ell$*

$$\begin{aligned} \boldsymbol{I} &\leftarrow \mathsf{Sim}_1^G(\mathcal{W}, J) \\ out &\leftarrow \mathsf{Sim}_2^G\big(\widehat{\boldsymbol{x}}|_{\boldsymbol{I}}\big) \end{aligned}$$

yields

$$|I_1| \leqslant |\mathcal{W}|, \ldots, |I_\ell| \leqslant |\mathcal{W}| \tag{6}$$

and

$$out \overset{id}{=} \big(\mathsf{AssignWires}(G, \mathcal{W}, \widehat{\boldsymbol{x}})\ ,\ \widehat{\boldsymbol{y}}|_J\big) \tag{7}$$

where $\boldsymbol{I} = (I_1, \ldots, I_\ell)$ and $\widehat{\boldsymbol{y}} = G(\widehat{\boldsymbol{x}})$.

We first formally show that (T)RPE1 implies SNI.

Lemma 5. *Let $t \in \mathbb{N}$ and $f : \mathbb{R} \to \mathbb{R}$ of amplification order $t+1$. Let G be a gadget which achieves (t, f)-TRPE1. Then G is also t-SNI.*

Proof. By definition of TRPE1 and by hypothesis on the amplification order, there exist input sets $I_1, \ldots, I_\ell$ which can perfectly simulate any leaking wires set $\mathcal{W}$ such that $|\mathcal{W}| \leq t$ and any set of output shares J such that $|J| \leq t$, satisfying $|I_1|, \ldots, |I_\ell| \leq |\mathcal{W}|$. Consequently, there exist input sets $I_1, \ldots, I_\ell$ which can perfectly simulate any leaking wires set $\mathcal{W}$ such that $|\mathcal{W}| = t_i \leq t$ and any set of output shares J such that $|\mathcal{W}| + |J| \leq t$ with $|I_1|, \ldots, |I_\ell| \leq t_i$. G is thus t-SNI. □

We now show that SNI implies TRPE1 up to some constraints on the parameters t and τ.

Lemma 6. *Let $\tau, \ell \in \mathbb{N}$. Let G be an ℓ-to-1 gadget which achieves τ-SNI. Then G satisfies (t, f)-TRPE1 for some $f : \mathbb{R} \to \mathbb{R}$ with an amplification order of*

$$d \geq \frac{1}{\ell} \min(t+1, \tau - t + 1) \ .$$

Proof. Since G is τ-SNI, then for any set of leaking wires $\mathcal{W}$ and output shares J such that $|\mathcal{W}| + |J| \leq \tau$, the wires indexed by $\mathcal{W}$ and the output shares indexed by J can be perfectly simulated from input shares indexed by $I_1, \ldots, I_\ell$ such that $|I_j| \leq |\mathcal{W}|$ for every $1 \leq j \leq \ell$. In the TRPE1 property, the set J of output shares can be any set of size $|J| \leq t$ so we can assume $|J| = t$ without loss of generality.

For a leaking set $\mathcal{W}$ of size $|\mathcal{W}| < \min(t+1, \tau - t + 1)$ no failure event occurs. Indeed τ-SNI and $|\mathcal{W}| < \tau - t + 1$ implies $|\mathcal{W}| + |J| \leq \tau$ and hence the existence of the sets $I_1, \ldots, I_\ell$ allowing the simulation with $|I_j| \leq |\mathcal{W}|$. And $|\mathcal{W}| < t + 1$ implies $|I_j| \leq \min(t, |\mathcal{W}|)$ for every j which implies the absence of failure. Then for a leaking set $\mathcal{W}$ of size $|\mathcal{W}| \geq \min(t+1, \tau - t + 1)$, no condition remains to rule out simulation failures and one could actually get a failure for every input. In the latter case, the amplification order would equal $\frac{1}{\ell} \min(t+1, n-t)$, but in all generality it could be higher (*i.e.* this value is a lower bound). □

An illustrative summary of the relations between RPE1, TRPE1 and SNI is depicted in Figure 1 (d denotes the amplification order of the function f). We hence observe an equivalence between the three notions up to some constraints on the parameters t, d, τ and ℓ.

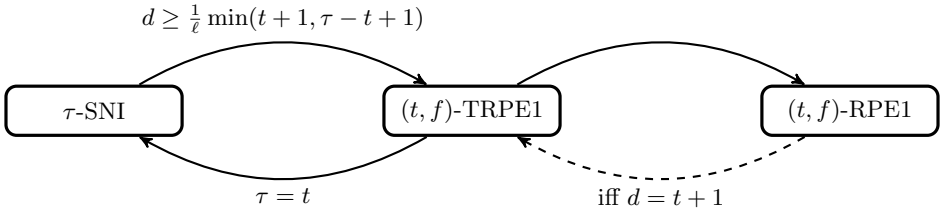

Fig. 1: Summary of relations between the different notions.

Relation and separation between (T)RPE2 and SNI. For a given n-share gadget G, the (T)RPE2 notion exclusively focuses on the simulation of a set of leaking intermediate variables together with a chosen set of $(n-1)$ output shares. If G is τ-SNI for $\tau < n-1$, then nothing can be claimed on the simulation of the latter sets. But if G is $(n-1)$-SNI, then any set of $(n-1)$ output shares can be perfectly simulated without the knowledge of any input share. Concretely, it implies that G is (t, f)-(T)RPE2 of amplification order at least 1 as a chosen output set of $(n-1)$ shares alone can be perfectly simulated without any additional knowledge on the input shares. Namely, we have

$$(n-1)\text{-SNI} \Rightarrow (t, f)\text{-(T)RPE2 of amplification order at least 1.}$$

Nevertheless, there is no relation from τ-SNI to (t, f)-(T)RPE2 for amplification orders strictly greater than 1 as (T)RPE2 would then consider leaking sets of size larger than or equal to n (for n-share gadgets, $\tau < n$). On the other side, there is no direct implication either from (t, f)-(T)RPE2 to τ-SNI since the former property does not consider all possible output sets of size $(n-1)$, but only a chosen one.

5 Generic Constructions

To the best of our knowledge, the only RPE gadgets in the literature are the ones designed in [9] which are restricted to a small number of shares, specifically $n \in \{2, 3\}$. A natural open question is the definition of RPE gadgets with good amplification orders, typically achieving or approaching the upper bounds exhibited in Section 3, for *any* number of shares n. In this section, we exhibit copy, addition, and multiplication gadgets derived from the widely known Ishai-Sahai-Wagner (ISW) construction [17]. Based on the results demonstrated in Section 4, we are able to show that these gadgets achieve RPE for any number of shares n with amplification orders close to the upper bounds (up to a small constant factor). We further provide an asymptotic analysis of the expanding compiler using these gadgets as well as a new multiplication gadget reaching the optimal amplification order hence improving the convergence to a better asymptotic complexity.

5.1 Generic Copy and Addition Gadgets

As intuitively proposed in [9] for small gadgets, copy and addition gadgets can be naturally derived from a refresh gadget. Such a gadget takes one sharing as input and outputs a new refreshed sharing of the same value. We formally introduce these natural constructions hereafter and show that their RPE security can be reduced to that of the underlying refresh gadget.

Generic Copy Gadget. Algorithm 1 displays the generic construction for the copy gadget from a refresh gadget. It simply consists in refreshing the input sharing twice to obtain two fresh copies.

Algorithm 1: Copy gadget G_{copy}

Input : $(a_1, \ldots, a_n)$ input sharing
Output: $(e_1, \ldots, e_n)$, $(f_1, \ldots, f_n)$ fresh copies of $(a_1, \ldots, a_n)$
$(e_1, \ldots, e_n) \leftarrow G_{\text{refresh}}(a_1, \ldots, a_n)$;
$(f_1, \ldots, f_n) \leftarrow G_{\text{refresh}}(a_1, \ldots, a_n)$;

We have the following lemma (see the proof in Appendix C).

Lemma 7. *Let $G_{refresh}$ be an n-share (t, f)-TRPE refresh gadget of amplification order d. Then, the copy gadget G_{copy} displayed in Algorithm 1 is (t, f')-TRPE also of amplification order d.*

As a consequence of this result, a TRPE refresh gadget directly yields a TRPE copy gadget achieving the same amplification order. Both gadgets can then reach the upper bound for 1-input gadgets whenever $t + 1 = 2(n - t)$ implying an amplification order $d = \frac{2(n+1)}{3}$.

Generic Addition Gadget. Algorithm 2 displays the generic construction for the addition gadget from a refresh gadget. It simply consists in refreshing both input sharings before adding them.

Algorithm 2: Addition Gadget G_{add}

Input : $(a_1, \ldots, a_n), (b_1, \ldots, b_n)$ input sharings
Output: $(c_1, \ldots, c_n)$ sharing of $a + b$
$(e_1, \ldots, e_n) \leftarrow G_{\text{refresh}}(a_1, \ldots, a_n)$;
$(f_1, \ldots, f_n) \leftarrow G_{\text{refresh}}(b_1, \ldots, b_n)$;
$(c_1, \ldots, c_n) \leftarrow (e_1 + f_1, \ldots, e_n + f_n)$;

We have the following lemma (see the proof in Appendix D).

Lemma 8. *Let $G_{refresh}$ be an n-share refresh gadget and let G_{add} be the corresponding addition gadget displayed in Algorithm 2. Then if $G_{refresh}$ is (t, f)-RPE (resp. (t, f)-TRPE) of amplification order d, then G_{add} is (t, f')-RPE (resp. (t, f')-TRPE) for some f' of amplification order $d' \geq \lfloor \frac{d}{2} \rfloor$.*

The above lemma shows that a (T)RPE refresh gadget of amplification order d directly yields a (T)RPE addition gadget of amplification order at least $\lfloor \frac{d}{2} \rfloor$. If the refresh gadget achieves the optimal $d = \frac{2(n+1)}{3}$, then the generic addition gadget has an amplification order at least $\lfloor \frac{n}{3} \rfloor$ which is not far from the upper bound for two-input gadgets of $\frac{n+1}{2}$.

We stress that the results of Lemma 7 and Lemma 8 are general and apply for any refresh gadget satisfying the (T)RPE property. In the rest of the section, we shall focus on a particular refresh gadget, namely the ISW-based refresh gadget. We show that this gadget achieves (T)RPE from which we obtain (T)RPE copy and addition gadgets for any number of shares n and with amplification orders close to the upper bound (up to a small constant factor).

5.2 ISW-based Copy and Addition Gadgets

As a basis of further constructions, we focus our analysis on the most deployed refresh gadget, which is based on the ISW construction [17].

ISW Refresh Gadget. This refresh can be seen as an ISW multiplication between the input sharing and the n-tuple $(1, 0, \ldots, 0)$. This is formally depicted in Algorithm 3.

Algorithm 3: ISW Refresh

Input : $(a_1, \ldots, a_n)$ input sharing, $\{r_{ij}\}_{1 \leq i < j \leq n}$ random values
Output: $(c_1, \ldots, c_n)$ such that $c_1 + \cdots + c_n = a_1 + \cdots + a_n$
for $i \leftarrow 1$ **to** n **do**
  $c_i \leftarrow a_i$;
end
for $i \leftarrow 1$ **to** n **do**
  for $j \leftarrow 1$ **to** $i-1$ **do**
    $c_i \leftarrow c_i + r_{ji}$;
  end
  for $j \leftarrow i+1$ **to** n **do**
    $c_i \leftarrow c_i + r_{ij}$;
  end
end
return $(c_1, \ldots, c_n)$;

We demonstrate through Lemma 9 that the ISW refresh gadget satisfies TRPE with an amplification order close to the optimal one. The proof is given in Appendix E.

Lemma 9. *Let $n \in \mathbb{N}$. For every $t \leq n-2$, the n-share ISW refresh gadget is (t, f_1)-TRPE1 and (t, f_2)-TRPE2 for some functions $f_1, f_2 : \mathbb{R} \to \mathbb{R}$ of amplification orders d_1, d_2 which satisfy:*

- $d_1 = \min(t+1, n-t)$ *for* f_1,
- $d_2 = t+1$ *for* f_2.

Corollary 3 then directly follows from Lemma 3 applied to TRPE and Lemma 9.

Corollary 3. *Let $n \in \mathbb{N}$. For every $t \leq n-2$, the n-share ISW refresh gadget is (t, f)-TRPE of amplification order*

$$d = \min(t+1, n-t).$$

According to Lemma 1, the upper bound on the amplification order of 1-input gadgets is $d \leq \min(t+1, 2(n-t))$ which gives $d \leq \frac{2n+2}{3}$ for $t = \frac{2n-1}{3}$. In contrast, the ISW refresh gadget reaches $d = \lfloor \frac{n+1}{2} \rfloor$ by taking $t = \lceil \frac{n-1}{2} \rceil$. While applying this result to the generic constructions of addition and copy gadgets introduced above, we obtain:

- a copy gadget of amplification order $d_c = \lfloor \frac{n+1}{2} \rfloor$ (Lemma 7),
- an addition gadget of amplification order at least $d_a = \lfloor \frac{n+1}{4} \rfloor$ (Lemma 8).

In the following, we demonstrate a tighter result than Lemma 8 for the ISW-based addition gadget (namely which does not imply the loss of a factor 2).

ISW-based Copy Gadget. The copy gadget G_{copy} that uses the n-share ISW refresh gadget as a building block in Algorithm 1 achieves the same amplification order as the ISW refresh for the TRPE setting, *i.e.* $d = \min(t+1, n-t)$. This is a direct implication from Lemma 7. Then, from Lemma 4, we have that ISW-based G_{copy} also achieves (t, f')-RPE with amplification order $d' \geq d$. We can actually prove that ISW-based G_{copy} achieves (t, f')-RPE with amplification order d' exactly equal to the amplification order in the TRPE setting, *i.e.* $d' = d = \min(t+1, n-t)$. This is stated in the following lemma which proof is given in Appendix F.

Lemma 10. *Let G_{copy} be the n-share copy gadget displayed in Algorithm 1 and instantiated with the ISW refresh gadget. Then for every $t \leq n-2$, G_{copy} achieves (t, f)-RPE with amplification order $d = \min(t+1, n-t)$.*

ISW-based Addition Gadget. The addition gadget G_{add} that uses the n-share ISW refresh gadget as a building block in Algorithm 2 achieves the same amplification order as the ISW refresh gadget, which is tighter than the bound from Lemma 8. This is stated in the following Lemma, which follows from Lemma 9, and from the fact that ISW refresh is $(n-1)$-SNI. The proof is given in Appendix G.

Lemma 11. *Let G_{add} be the n-share addition gadget displayed in Algorithm 2 and instantiated with the ISW refresh gadget. Then for every $t \leq n-2$, G_{add} achieves (t, f_1)-TRPE1 and (t, f_2)-TRPE2 for some functions $f_1, f_2 : \mathbb{R} \to \mathbb{R}$ of amplification orders d_1, d_2 which satisfy:*

- $d_1 = \min(t+1, n-t)$,
- $d_2 = t+1$.

Corollary 4 then directly follows from Lemma 11 by applying Lemma 3 (TRPE1 $\cap$ TRPE2 $\Rightarrow$ TRPE) and Lemma 4 (TRPE $\Rightarrow$ RPE).

Corollary 4. *Let $n \in \mathbb{N}$. For every $t \leq n-2$, the n-share gadget G_{add} displayed in Algorithm 2 and instantiated with the ISW refresh gadget is (t, f)-RPE of amplification order $d = \min(t+1, n-t)$.*

5.3 ISW Multiplication Gadget

In contrast to the copy and addition gadgets that are built from generic schemes with a refresh gadget as a building block, the multiplication gadget can be directly defined as the standard ISW multiplication, which is recalled in Algorithm 4.

Algorithm 4: ISW Multiplication

Input : $(a_1, \ldots, a_n)$,$(b_1, \ldots, b_n)$ input sharings, $\{r_{ij}\}_{1 \leq i < j \leq n}$ random values
Output: $(c_1, \ldots, c_n)$ sharing of $a \cdot b$
for $i \leftarrow 1$ **to** n **do**
  $c_i \leftarrow a_i \cdot b_i$;
end
for $i \leftarrow 1$ **to** n **do**
  for $j \leftarrow i+1$ **to** n **do**
    $c_i \leftarrow c_i + r_{ij}$;
    $r_{ji} \leftarrow (a_i \cdot b_j + r_{ij}) + a_j \cdot b_i$;
    $c_j \leftarrow c_j + r_{ji}$;
  end
end
return $(c_1, \ldots, c_n)$;

We have the following lemma (see the proof in Appendix H).

Lemma 12. *Let $n \in \mathbb{N}$. For every $t \leq n-2$, the n-share ISW multiplication gadget displayed in Algorithm 4 is (t, f_1)-RPE1 and (t, f_2)-RPE2 for some functions $f_1, f_2 : \mathbb{R} \to \mathbb{R}$ of amplification orders d_1, d_2 which satisfy:*

- $d_1 = \dfrac{\min(t+1, n-t)}{2}$,
- $d_2 = \dfrac{t+1}{2}$.

Corollary 5 then directly follows from Lemma 12 by applying Lemma 3 (RPE1 $\cap$ RPE2 $\Rightarrow$ RPE).

Corollary 5. *Let $n \in \mathbb{N}$. For every $t \leq n-2$, the n-share ISW multiplication gadget displayed in Algorithm 4 is (t, f)-RPE of amplification order*

$$d = \frac{\min(t+1, n-t)}{2} \ .$$

According to Lemma 2, the upper bound on the amplification order of a standard multiplication gadget (*i.e.* which starts with the cross-products of the input shares) is $d \leq \min((t+1)/2, (n-t))$ which gives $d \leq (n+1)/3$ for $t = (2n-1)/3$. In contrast, the ISW multiplication gadget reaches $d = \lfloor \frac{n+1}{4} \rfloor$ by taking $t = \lceil \frac{n-1}{2} \rceil$.

5.4 Application to the Expanding Compiler

As recalled in Section 2.5, instantiating the expanding compiler with three RPE base gadgets gives a $(p, 2^{-\kappa})$-random probing secure compiler (*i.e.* achieving κ bits of security against a leakage probability p) with a complexity blowup of $\mathcal{O}(\kappa^e)$ for an exponent e satisfying

$$e = \frac{\log N_{\max}}{\log d}$$

where $N_{\max}$ satisfies (5) and where d is the minimum amplification order of the three base gadgets.

We can instantiate the expanding compiler using the above ISW-based gadgets. Specifically, we use the ISW multiplication for the multiplication gadget G_{mult}, and the generic constructions of addition and copy gadgets based on the ISW refresh. From Lemmas 10, 11, and 12, the maximum amplification order achievable by the compiler is the minimum of the three gadgets, which is the order of the ISW multiplication gadget:

$$d = \frac{\min(t+1, n-t)}{2} \ .$$

Hence, for a given number of shares n, the maximum amplification order achievable is

$$d_{\max} = \left\lfloor \frac{n+1}{4} \right\rfloor$$

which is obtained for $t = \lceil \frac{n-1}{2} \rceil$. On the other hand, the value of $N_{\max}$ can be characterized in terms of the number of shares n from the ISW algorithm. Recall from Section 2.5 that

$$N_{\max} = \max\left(N_{\text{m,m}} \ , \ \mathsf{eigenvalues}\left(\begin{pmatrix} N_{\text{a,a}} & N_{\text{c,a}} \\ N_{\text{a,c}} & N_{\text{c,c}} \end{pmatrix} \right) \right).$$

In the case of the ISW-based gadgets, we have $N_{\mathrm{m,m}} = n^2$ and

$$\begin{pmatrix} N_{\mathrm{a,a}} & N_{\mathrm{c,a}} \\ N_{\mathrm{a,c}} & N_{\mathrm{c,c}} \end{pmatrix} = \begin{pmatrix} n(2n-1) & 2n(n-1) \\ n(n-1) & n^2 \end{pmatrix}.$$

The eigenvalues of the above matrix are $\lambda_1 = n$ and $\lambda_2 = 3n^2 - 2n$, implying $N_{\max} = 3n^2 - 2n$. Thus, the expanding compiler instantiated by our ISW-based gadgets has a complexity blowup $\mathcal{O}(\kappa^e)$ with exponent

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

Figure 2 (blue curve) shows the evolution of the value of this exponent with respect to the number of shares n (where we assume an odd n). The value of e clearly decreases as the number of shares grows, and this decrease is faster for a small number of shares ($5 \leq n \leq 10$). The exponent value reaches $e \approx 4$ for a number of shares around 25 and then slowly converges towards $e = 2$ as n grows. This is to be compared with the $\mathcal{O}(\kappa^{7.5})$ complexity achieved by the instantiation from [2, 9].

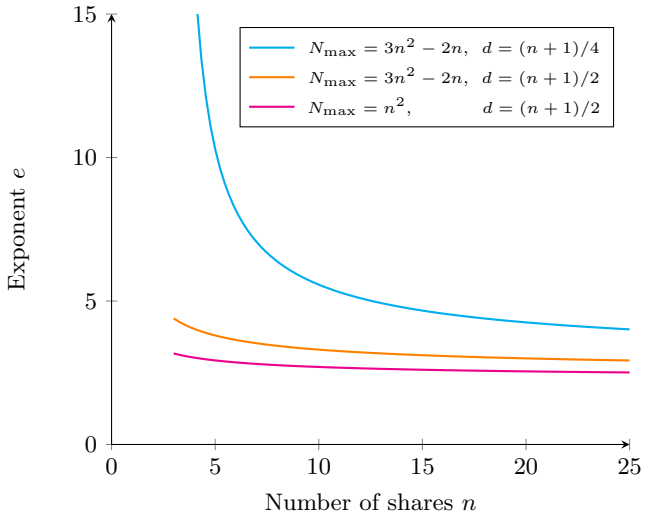

Fig. 2: Evolution of the complexity exponent $e = \log(N_{\max})/\log(d)$ with respect to the number of shares n. The blue curve matches the instantiation with the ISW-based gadgets; the orange curve assumes the optimal amplification order (*i.e.* an improvement of the multiplication gadget); the pink curve assumes a better complexity for addition and copy gadgets (so that $N_{\max}$ matches $N_{\mathrm{m,m}} = n^2$).

Towards a Better Complexity. Choosing gadgets which attain the upper bound $\min(t+1, n-t)$ on the amplification order from Lemma 1 allows the compiler to have the maximum amplification order $d = (n+1)/2$ and thus have the lowest complexity blowup. Our ISW-based copy and addition gadgets achieve this bound while the ISW multiplication gadget is limited to $(n+1)/4$ (Lemma 12). To reach the optimal amplification order, one would need a different multiplication gadget and in

particular a multiplication gadget which does not perform a direct product of shares (because of the bound from Lemma 2). We introduce such a multiplication gadget hereafter (see Section 5.5). Specifically, our new multiplication gadget achieves the upper bound on the amplification order $\min(t+1, n-t)$ by avoiding a direct product of shares using a prior refresh on the input sharings. The orange curve in Figure 2 shows the evolution of the value of the exponent when instantiating the expanding compiler with our previous addition and copy gadgets and this new multiplication gadget. For such an instantiation, the complexity exponent still slowly converges towards $e = 2$ but, as we can see from Figure 2, the exponent value is much better for small values of n. For example, we obtain $e \approx 3$ for $n = 20$.

Another possible direction for improvement would be to lower the complexity of the addition and copy gadgets, which is mainly dominated by the refreshing. Assume that we can design a (T)RPE refresh gadget in sub-quadratic complexity, *e.g.* as the refresh gadgets proposed in [20, 7, 15], then the eigenvalues of the matrix in (5) would also be sub-quadratic and the value of $N_{\max}$ from equation (5) would drop to $N_{\mathrm{m,m}} = n^2$ (if the multiplication gadget still requires n^2 multiplication gates). The pink curve in Figure 2 depicts the evolution of the exponent value under this assumption. We still have a slow convergence towards $e = 2$ but the exponent value is yet better for small values of n. For example, a complexity blowup of $\mathcal{O}(\kappa^{2.5})$ is obtained with 20 shares. We leave the task of finding such a sub-quadratic (T)RPE refresh gadget as an open question for further research.

The above analysis shows that the expanding compiler can theoretically approach a quadratic complexity at the cost of increasing the number of shares in the base gadgets. The downside of it is that the tolerated leakage probability is likely to decrease as the number of shares grow. For instance, the ISW construction is known to only tolerate a leakage probability $p = \mathcal{O}(1/n)$ [14]. The number of shares hence offers multiple trade-offs between the tolerated probability and the asymptotic complexity of the compiler. Starting from a target leakage probability p, one could determine the highest number of shares admissible from a generic construction (such as the ISW-based instantiation exhibited above) and thus deduce the best complexity exponent achievable. In Section 6, we exhibit concrete trade-offs that can be reached for small values of n.

5.5 Multiplication Gadget with Maximal Amplification Order

Constructing a multiplication gadget which achieves the upper bound on the amplification order from Lemma 1 is tricky. First, as a standard multiplication gadget (*i.e.* which computes the cross products of the input shares), the ISW multiplication cannot achieve the maximal amplification order (see Lemma 2). In order to reach the upper bound for two-input gadgets (see Corollary 2), we need a non-standard multiplication gadget, *i.e.* which does not perform a direct product between the input shares. As an additional observation, the addition, copy, and random gates are *virtually free* in a multiplication gadget since they do not impact the final complexity of the expanding compiler (see Section 2.5). This suggests that we can be greedy in terms of randomness to reach the maximal amplification order.

In the following, we will describe the construction of a new multiplication gadget which achieves the maximum amplification order $\min(t+1, n-t)$. We first describe our standard n-share multiplication gadget and then explain how we avoid the initial cross products of shares. First, the gadget

constructs the matrix of the cross product of input shares:

$$M = \begin{pmatrix} a_1 \cdot b_1 & a_1 \cdot b_2 & \cdots & a_1 \cdot b_n \\ a_2 \cdot b_1 & a_2 \cdot b_2 & \cdots & a_2 \cdot b_n \\ \vdots & \vdots & \ddots & \vdots \\ a_n \cdot b_1 & a_n \cdot b_2 & \cdots & a_n \cdot b_n \end{pmatrix}$$

Then, it picks n^2 random values which define the following matrix:

$$R = \begin{pmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,n} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n,1} & r_{n,2} & \cdots & r_{n,n} \end{pmatrix}$$

It then performs an element-wise addition between the matrices M and R:

$$P = M + R = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n,1} & p_{n,2} & \cdots & p_{n,n} \end{pmatrix}$$

At this point, the gadget randomizes each product of input shares from the matrix M with a single random value from R. In order to generate the correct output, the gadget adds all the columns of P into a single column V of n elements, and adds all the columns of the transpose matrix R^{T} into a single column X of n elements:

$$V = \begin{pmatrix} p_{1,1} + \cdots + p_{1,n} \\ p_{2,1} + \cdots + p_{2,n} \\ \vdots \\ p_{n,1} + \cdots + p_{n,n} \end{pmatrix}, \qquad X = \begin{pmatrix} r_{1,1} + \cdots + r_{n,1} \\ r_{1,2} + \cdots + r_{n,2} \\ \vdots \\ r_{1,n} + \cdots + r_{n,n} \end{pmatrix}$$

The n-share output is finally defined as $(c_1, \ldots, c_n) = V + X$.

In order to further increase the maximum amplification order attainable by the gadget, we need to avoid performing a direct product of shares (because of the bound proved in Lemma 2). For this, we add a pre-processing phase to the gadget using a refresh gadget G_{refresh}. Specifically, we refresh the input $(b_1, \ldots, b_n)$ each time it is used. In other terms, each row of the matrix M uses a fresh copy of $(b_1, \ldots, b_n)$ produced using the considered refresh gadget. This amounts to performing n independent refreshes of the input $(b_1, \ldots, b_n)$. The matrix M is thus defined as

$$M = \begin{pmatrix} a_1 \cdot b_1^{(1)} & a_1 \cdot b_2^{(1)} & \cdots & a_1 \cdot b_n^{(1)} \\ a_2 \cdot b_1^{(2)} & a_2 \cdot b_2^{(2)} & \cdots & a_2 \cdot b_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ a_n \cdot b_1^{(n)} & a_n \cdot b_2^{(n)} & \cdots & a_n \cdot b_n^{(n)} \end{pmatrix}$$

where $(b_1^{(j)}, \ldots, b_n^{(j)})$, $j \in [n]$, are the n independent refreshings of the input $(b_1, \ldots, b_n)$.

With this refreshing scheme, we avoid using the same share more than once for one of the two input sharings. As a consequence, the double failure set of size $t+1$ which is the reason behind the bound $(t+1)/2$ in Lemma 2, becomes a simple failure set (*i.e.* provoking a failure on a single input sharing). In addition, the computational overhead of these additional n refreshes is negligible compared to the joint contribution of the copy and addition gadgets to the complexity of the expanding compiler.

For the sake of completeness, we present the full algorithm for this multiplication gadget in Algorithm 5.

Algorithm 5: Our multiplication gadget

Input : $(a_1, \ldots, a_n)$,$(b_1, \ldots, b_n)$ input sharings, $\{r_{ij}\}_{1 \leq i \leq n, 1 \leq j \leq n}$ random values, refresh gadget G_{refresh}
Output: $(c_1, \ldots, c_n)$ sharing of $a \cdot b$
for $i \leftarrow 1$ **to** n **do**
  $(b_1^{(i)}, \ldots, b_n^{(i)}) \leftarrow G_{\text{refresh}}(b_1, \ldots, b_n)$;
end
for $i \leftarrow 1$ **to** n **do**
  for $j \leftarrow 1$ **to** n **do**
    $p_{i,j} \leftarrow a_i \times b_j^{(i)} + r_{i,j}$;
  end
end
$(v_1, \ldots, v_n) \leftarrow (0, \ldots, 0)$;
$(x_1, \ldots, x_n) \leftarrow (0, \ldots, 0)$;
for $i \leftarrow 1$ **to** n **do**
  for $j \leftarrow 1$ **to** n **do**
    $v_i \leftarrow v_i + p_{i,j}$;
    $x_i \leftarrow x_i + r_{i,j}$;
  end
end
for $i \leftarrow 1$ **to** n **do**
  $c_i \leftarrow v_i + x_i$;
end
return $(c_1, \ldots, c_n)$;

In the following lemma, we show that if the refresh gadget G_{refresh} achieves the TRPE1 property with the amplification order at least $d = \min(t+1, n-t)$ for any t, then the multiplication gadget depicted in Algorithm 5 achieves TRPE with the maximum amplification orders. The proof is given in Appendix I.

Lemma 13. *Let $t \leq n-1$. Let $G_{refresh}$ be a (t, f')-TRPE1 refresh gadget for some function $f' : \mathbb{R} \to \mathbb{R}$, and G_{mult} the n-share multiplication gadget from Algorithm 5. If f' is of amplification order $d' \geq d = \min(t+1, n-t)$, then G_{mult} achieves (t, f)-TRPE for some function $f : \mathbb{R} \to \mathbb{R}$ of amplification order $d = \min(t+1, n-t)$.*

Corollary 6 then directly follows from Lemma 13 by applying Lemma 4 (TRPE $\Rightarrow$ RPE).

Corollary 6. *Let $t \leq n-1$. Let $G_{refresh}$ be a (t, f')-TRPE1 refresh gadget for some function $f' : \mathbb{R} \to \mathbb{R}$, and G_{mult} the n-share multiplication gadget from Algorithm 5. If f' is of amplification order $d' \geq d = \min(t+1, n-t)$, then G_{mult} achieves (t, f)-RPE for some function $f : \mathbb{R} \to \mathbb{R}$ of the same amplification order $d = \min(t+1, n-t)$.*

6 Efficient Small Gadgets

This section displays our new constructions of small gadgets for copy, addition, and multiplication operations with a low number of shares. As explained in [9], we cannot achieve RPE security with relevant amplification orders for gadgets of less than 3 shares. Then, as explained in Section 3.1, the highest amplification orders can only be achieved for gadgets with an odd number of shares. We therefore omit 4-share gadgets and display our best trade-offs in terms of RPE security and complexity for 3-share and 5-share gadgets. Each one of these gadgets is experimentally verified using the `VRAPS` verification tool from [9].

Addition and Copy Gadgets. For the construction of small 3-share and 5-share addition and copy gadgets, we use the generic constructions depicted in Algorithms 1 and 2 (in Section 5) which naturally use a refresh gadget as a building block. We hence start by looking for refresh gadgets that have a good complexity in terms of gates count, and achieve the upper bound on the amplification order for the specific case of 3-share and 5-share constructions (but not necessarily for a higher number of shares).

Multiplication gadget. For the construction of small 3-share and 5-share multiplication gadgets, we use the generic construction depicted in Algorithm 5 from Section 5.5 which, to the best of our knowledge, is the only multiplication gadget which achieves the maximum amplification order for any number of shares, and specifically for 3-share and 5-share constructions. As for the refresh gadget G_{refresh} which is used to perform n refreshes on the second input, we use the same scheme as for the construction of small addition and copy gadgets (and which shall satisfy the necessary condition on G_{refresh} from Corollary 6).

While the multiplication gadget from Section 5.5 achieves the desired amplification order, we add another pre-processing phase to the gadget in order to further improve the tolerated leakage probability. In addition to the n refreshes performed on the second input b (see Algorithm 5), we add another single refresh of the input $(a_1, \ldots, a_n)$ before computing the cross-products, using the same refresh gadget G_{refresh}. Refreshing the input $(a_1, \ldots, a_n)$ before usage experimentally shows a further increase in the maximum tolerated leakage probability, by adding more randomness to the input shares before computing the cross-product matrix M in Algorithm 5. And since the refresh gadget G_{refresh} achieves the maximum amplification order, the amplification order achieved by G_{mult} is not affected by adding another refresh to the first input a.

The above construction achieves the maximum amplification order for 3-share ($d = 2$) and 5-share ($d = 3$) gadgets based on natural refresh gadgets detailed hereafter.

6.1 3-share Gadgets

We start with the construction of 3-share gadgets for our three base operations.

Copy and Addition Gadgets. We build our copy and addition gadgets from the instantiation of the generic constructions of Section 5 (Algorithms 1 and 2) with 3 shares. However, we do not use the ISW refresh gadget but the following more efficient construction with only two random values (instead of three):

$$\begin{aligned} G_{\text{refresh}} : c_1 &\leftarrow r_1 + a_1 \\ c_2 &\leftarrow r_2 + a_2 \\ c_3 &\leftarrow (r_1 + r_2) + a_3. \end{aligned}$$

This refresh is sufficient to reach the upper bounds on the amplification orders (from Lemma 1). From this basis, we obtain the following 3-share addition gadget with four random values:

$$\begin{aligned} G_{\text{add}} : c_1 &\leftarrow (r_1 + a_1) + (r_3 + b_1) \\ c_2 &\leftarrow (r_2 + a_2) + (r_4 + b_2) \\ c_3 &\leftarrow \big((r_1 + r_2) + a_3\big) + \big((r_3 + r_4) + b_3\big) \end{aligned}$$

and the following 3-share copy gadget with also four random values:

$$\begin{aligned} G_{\text{copy}} : c_1 &\leftarrow r_1 + a_1; & d_1 &\leftarrow r_3 + a_1 \\ c_2 &\leftarrow r_2 + a_2; & d_2 &\leftarrow r_4 + a_2 \\ c_3 &\leftarrow (r_1 + r_2) + a_3; & d_3 &\leftarrow (r_3 + r_4) + a_3. \end{aligned}$$

Multiplication Gadget. The following construction is a 3-share instantiation of the multiplication gadget described in Section 5.5. For the input refreshing, we use the 3-share refresh gadget described above with two uniformly random values. The construction achieves the bound on the amplification order from Lemma 1 with 17 random values:

$$\begin{aligned} G_{\text{mult}} : i_{1,1} &\leftarrow r_1 + b_1; & i_{1,2} &\leftarrow r_2 + b_2; & i_{1,3} &\leftarrow (r_1 + r_2) + b_3 \\ i_{2,1} &\leftarrow r_3 + b_1; & i_{2,2} &\leftarrow r_4 + b_2; & i_{2,3} &\leftarrow (r_3 + r_4) + b_3 \\ i_{3,1} &\leftarrow r_5 + b_1; & i_{3,2} &\leftarrow r_6 + b_2; & i_{3,3} &\leftarrow (r_5 + r_6) + b_3 \\ a'_1 &\leftarrow r_7 + a_1; & a'_2 &\leftarrow r_8 + a_2; & a'_3 &\leftarrow (r_7 + r_8) + a_3 \end{aligned}$$

$$\begin{aligned} c_1 &\leftarrow (a'_1 \cdot i_{1,1} + r_{1,1}) + (a'_1 \cdot i_{1,2} + r_{1,2}) + (a'_1 \cdot i_{1,3} + r_{1,3}) + (r_{1,1} + r_{2,1} + r_{3,1}) \\ c_2 &\leftarrow (a'_2 \cdot i_{2,1} + r_{2,1}) + (a'_2 \cdot i_{2,2} + r_{2,2}) + (a'_2 \cdot i_{2,3} + r_{2,3}) + (r_{1,2} + r_{2,2} + r_{3,2}) \\ c_3 &\leftarrow (a'_3 \cdot i_{3,1} + r_{3,1}) + (a'_3 \cdot i_{3,2} + r_{3,2}) + (a'_3 \cdot i_{3,3} + r_{3,3}) + (r_{1,3} + r_{2,3} + r_{3,3}). \end{aligned}$$

Results. Table 1 displays the results for the above gadgets obtained through the `VRAPS` tool. The second column gives the complexity, where N_a, N_c, N_m, N_r stand for the number of addition gates, copy gates, multiplication gates and random gates respectively. The third column provides the amplification order of the gadget. And the last column gives the maximum tolerated leakage probability. The last row gives the global complexity, amplification order, and maximum tolerated leakage probability for the expanding compiler using these three gadgets from the results provided in [9].

Table 1: Results for the 3-share gadgets for $(t = 1, f)$-RPE, achieving the bound on the amplification order.

Gadget	**Complexity** (N_a, N_c, N_m, N_r)	**Amplification order**	$\log_2$ **of maximum tolerated proba**
G_{refresh}	$(4, 2, 0, 2)$	2	-5.14
G_{add}	$(11, 4, 0, 4)$	2	-4.75
G_{copy}	$(8, 7, 0, 4)$	2	-7.50
G_{mult}	$(40, 29, 9, 17)$	2	-7.41
Compiler	$\mathcal{O}(\lvert C\rvert \cdot \kappa^{\mathbf{3.9}})$	**2**	$\mathbf{-7.50}$

Remark 1. The copy gadget G_{copy} instantiated in [9] which uses a refresh scheme with 3 randoms for each output, also reaches the amplification order 2. It tolerates a better leakage probability (*i.e.* $2^{-5.9}$) than the one provided here, but with a higher complexity of $(12, 9, 0, 6)$. If it is used to replace the 3-share copy gadget, the maximum tolerated leakage probability by the compiler from Table 1 would be of $2^{-7.4}$ slightly better than the current value of $2^{-7.5}$ but with a higher complexity of $\mathcal{O}(|C|\cdot\kappa^{4.08})$ instead of $\mathcal{O}(|C|\cdot\kappa^{3.9})$. Another copy gadget can be constructed by using the refresh scheme with 3 random values from [9] for one of the outputs, and the refresh scheme presented in this section with 2 random values for the second output. This gadget tolerates a maximum leakage probability of around $2^{-7.1}$ with a complexity of $(10, 8, 0, 5)$. Using it would bring the complexity of the compiler from Table 1 to $\mathcal{O}(|C|\cdot\kappa^4)$, while tolerating a leakage probability of $2^{-7.4}$, the same as that of the used multiplication gadget.

6.2 5-share Gadgets

We now present our 5-share gadgets for our three base operations, which reach the optimal amplification order from Lemma 1.

Copy and Addition Gadgets. As for the 3-share case, we use the generic constructions from Section 5. Instead of using the ISW refresh gadget which would require 10 uniformly random values for a 5-share construction, we use the *circular refresh gadget* described in [4, 6] (a.k.a. *block refresh gadget*):

$$\begin{aligned} G_{\text{refresh}} : c_1 &\leftarrow (r_1 + r_2) + a_1 \\ c_2 &\leftarrow (r_2 + r_3) + a_2 \\ c_3 &\leftarrow (r_3 + r_4) + a_3 \\ c_4 &\leftarrow (r_4 + r_5) + a_4 \\ c_5 &\leftarrow (r_5 + r_1) + a_5. \end{aligned}$$

This gadget only uses n randoms for an n-share construction, and while it does not achieve enough security in the generic case (unless the refresh block is iterated on the input a certain number of times [4, 6]), it proves to be more than enough to achieve the necessary amplification order for our

5-share constructions. We use a variant of the original version (also suggested in [4]): we choose to sum the random values first (thus obtaining a sharing of 0) before adding them to the input shares. The idea is to avoid using the input shares in any of the intermediate variables, so that input shares only appear in the input variables $\{a_i\}_{1\leq i\leq n}$ and the final output variables $\{c_i\}_{1\leq i\leq n}$. Intuitively, this trick allows to have less failure tuples in the gadget because there are less variables that could leak information about the input. This is validated experimentally where we obtain better results in terms of amplification order and tolerated leakage probability for small gadgets.

From this circular refresh, we obtain an addition gadget with ten random values which reaches the upper bound on the amplification order:

$$\begin{aligned} G_{\text{add}} : c_1 &\leftarrow \big((r_1 + r_2) + a_1\big) + \big((r_6 + r_7) + b_1\big) \\ c_2 &\leftarrow \big((r_2 + r_3) + a_2\big) + \big((r_7 + r_8) + b_2\big) \\ c_3 &\leftarrow \big((r_3 + r_4) + a_3\big) + \big((r_8 + r_9) + b_3\big) \\ c_4 &\leftarrow \big((r_4 + r_5) + a_4\big) + \big((r_9 + r_{10}) + b_4\big) \\ c_5 &\leftarrow \big((r_5 + r_1) + a_5\big) + \big((r_{10} + r_6) + b_5\big) \end{aligned}$$

and a copy gadget with also ten random values and which also reaches the upper bound on the amplification order:

$$\begin{aligned} G_{\text{copy}} : c_1 &\leftarrow (r_1 + r_2) + a_1; & d_1 &\leftarrow (r_6 + r_7) + a_1 \\ c_2 &\leftarrow (r_2 + r_3) + a_2; & d_2 &\leftarrow (r_7 + r_8) + a_2 \\ c_3 &\leftarrow (r_3 + r_4) + a_3; & d_3 &\leftarrow (r_8 + r_9) + a_3 \\ c_4 &\leftarrow (r_4 + r_5) + a_4; & d_4 &\leftarrow (r_9 + r_{10}) + a_4 \\ c_5 &\leftarrow (r_5 + r_1) + a_5; & d_5 &\leftarrow (r_{10} + r_6) + a_5. \end{aligned}$$

Multiplication Gadget. The following construction is a 5-share instantiation of the multiplication gadget described in Section 5.5. For the input refreshing, we use the 5-share circular refresh gadget described above. The gadget advantageously achieves the optimal amplification order (given by

Lemma 1) with 55 random values:

$$
\begin{aligned}
G_{\text{mult}} : i_{1,1} &\leftarrow (r_1 + r_2) + b_1; \quad i_{1,2} \leftarrow (r_2 + r_3) + b_2; \quad i_{1,3} \leftarrow (r_3 + r_4) + b_3; \\
& i_{1,4} \leftarrow (r_4 + r_5) + b_4; \quad i_{1,5} \leftarrow (r_5 + r_1) + b_5 \\
\\
& i_{2,1} \leftarrow (r_6 + r_7) + b_1; \quad i_{2,2} \leftarrow (r_7 + r_8) + b_2; \quad i_{2,3} \leftarrow (r_8 + r_9) + b_3; \\
& i_{2,4} \leftarrow (r_9 + r_{10}) + b_4; \quad i_{2,5} \leftarrow (r_{10} + r_6) + b_5 \\
\\
& i_{3,1} \leftarrow (r_{11} + r_{12}) + b_1; \quad i_{3,2} \leftarrow (r_{12} + r_{13}) + b_2; \quad i_{3,3} \leftarrow (r_{13} + r_{14}) + b_3; \\
& i_{3,4} \leftarrow (r_{14} + r_{15}) + b_4; \quad i_{3,5} \leftarrow (r_{15} + r_{11}) + b_5 \\
\\
& i_{4,1} \leftarrow (r_{16} + r_{17}) + b_1; \quad i_{4,2} \leftarrow (r_{17} + r_{18}) + b_2; \quad i_{4,3} \leftarrow (r_{18} + r_{19}) + b_3; \\
& i_{4,4} \leftarrow (r_{19} + r_{20}) + b_4; \quad i_{4,5} \leftarrow (r_{20} + r_{16}) + b_5 \\
\\
& i_{5,1} \leftarrow (r_{21} + r_{22}) + b_1; \quad i_{5,2} \leftarrow (r_{22} + r_{23}) + b_2; \quad i_{5,3} \leftarrow (r_{23} + r_{24}) + b_3; \\
& i_{5,4} \leftarrow (r_{24} + r_{25}) + b_4; \quad i_{5,5} \leftarrow (r_{25} + r_{21}) + b_5 \\
\\
& a'_1 \leftarrow (r_{26} + r_{27}) + a_1; \quad a'_2 \leftarrow (r_{27} + r_{28}) + a_2; \quad a'_3 \leftarrow (r_{28} + r_{29}) + a_3; \\
& a'_4 \leftarrow (r_{29} + r_{30}) + a_4; \quad a'_5 \leftarrow (r_{30} + r_{26}) + a_5
\end{aligned}
$$

$$
\begin{aligned}
c_1 \leftarrow & (a'_1 \cdot i_{1,1} + r_{1,1}) + (a'_1 \cdot i_{1,2} + r_{1,2}) + (a'_1 \cdot i_{1,3} + r_{1,3}) + (a'_1 \cdot i_{1,4} + r_{1,4}) \\
& + (a'_1 \cdot i_{1,5} + r_{1,5}) + (r_{1,1} + r_{2,1} + r_{3,1} + r_{4,1} + r_{5,1}) \\
c_2 \leftarrow & (a'_2 \cdot i_{2,1} + r_{2,1}) + (a'_2 \cdot i_{2,2} + r_{2,2}) + (a'_2 \cdot i_{2,3} + r_{2,3}) + (a'_2 \cdot i_{2,4} + r_{2,4}) \\
& + (a'_2 \cdot i_{2,5} + r_{2,5}) + (r_{1,2} + r_{2,2} + r_{3,2} + r_{4,2} + r_{5,2}) \\
c_3 \leftarrow & (a'_3 \cdot i_{3,1} + r_{3,1}) + (a'_3 \cdot i_{3,2} + r_{3,2}) + (a'_3 \cdot i_{3,3} + r_{3,3}) + (a'_3 \cdot i_{3,4} + r_{3,4}) \\
& + (a'_3 \cdot i_{3,5} + r_{3,5}) + (r_{1,3} + r_{2,3} + r_{3,3} + r_{4,3} + r_{5,3}) \\
c_4 \leftarrow & (a'_4 \cdot i_{4,1} + r_{4,1}) + (a'_4 \cdot i_{4,2} + r_{4,2}) + (a'_4 \cdot i_{4,3} + r_{4,3}) + (a'_4 \cdot i_{4,4} + r_{4,4}) \\
& + (a'_4 \cdot i_{4,5} + r_{4,5}) + (r_{1,4} + r_{2,4} + r_{3,4} + r_{4,4} + r_{5,4}) \\
c_5 \leftarrow & (a'_5 \cdot i_{5,1} + r_{5,1}) + (a'_5 \cdot i_{5,2} + r_{5,2}) + (a'_5 \cdot i_{5,3} + r_{5,3}) + (a'_5 \cdot i_{5,4} + r_{5,4}) \\
& + (a'_5 \cdot i_{5,5} + r_{5,5}) + (r_{1,5} + r_{2,5} + r_{3,5} + r_{4,5} + r_{5,5}).
\end{aligned}
$$

Results. Table 2 gives the results for the above gadgets obtained through the `VRAPS` tool.

Table 2: Results for the 5-share gadgets for $(t = 2, f)$-RPE, achieving the bound on the amplification order.

Gadget	**Complexity**	**Amplification order**	$\log_2$ **of maximum tolerated proba**
G_{refresh}	$(10, 5, 0, 5)$	3	-4.83
G_{add}	$(25, 10, 0, 10)$	3	$[-6.43, -3.79]$
G_{copy}	$(20, 15, 0, 10)$	3	$[-6.43, -5.78]$
G_{mult}	$(130, 95, 25, 55)$	3	$[-12.00, -6.03]$
Compiler	$\mathcal{O}(\lvert C\rvert \cdot \kappa^{\mathbf{3.23}})$	**3**	$[\mathbf{-12.00}, \mathbf{-6.03}]$

From Tables 1 and 2, we observe that the asymptotic complexity is better for the instantiation based on 5-share gadgets as they provide a better amplification order with limited overhead. While this result can seem to be counterintuitive, it actually comes from the fact that each gadget will be expended less in the second scenario. We stress that we could only obtain an interval $[2^{-12}, 2^{-6}]$ for the tolerated leakage probability because it was computationally too expensive to obtain a tighter interval from the `VRAPS` tool, but this could probably be improved in the future. Meanwhile, we can consider that our best complexity $\mathcal{O}(|C| \cdot \kappa^{3.2})$ comes at the price of a lower tolerated leakage probability of 2^{-12} (5-share gadget) compared to the $\mathcal{O}(|C| \cdot \kappa^{3.9})$ complexity and $2^{-7.5}$ tolerated leakage probability obtained for our 3-share instantiation.

In comparison, the previous instantiation of the expanding compiler [9] could only achieve a complexity of $\mathcal{O}(|C| \cdot \kappa^{7.5})$ for maximum tolerated probabilities of 2^{-8}, and the instantiation of the expanding approach with a multi-party computation protocol [2], could only achieve a complexity of $\mathcal{O}(|C| \cdot \kappa^{8.2})$ for maximum tolerated probabilities of 2^{-26}.

Acknowledgments. This work is partly supported by the French FUI-AAP25 VeriSiCC project.

References

1. Miklós Ajtai. Secure computation with information leaking to an adversary. In Lance Fortnow and Salil P. Vadhan, editors, *43rd Annual ACM Symposium on Theory of Computing*, pages 715–724, San Jose, CA, USA, June 6–8, 2011. ACM Press.
2. Prabhanjan Ananth, Yuval Ishai, and Amit Sahai. Private circuits: A modular approach. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 427–455, Santa Barbara, CA, USA, August 19–23, 2018. Springer, Heidelberg, Germany.
3. Marcin Andrychowicz, Stefan Dziembowski, and Sebastian Faust. Circuit compilers with $O(1/\log(n))$ leakage rate. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 586–615, Vienna, Austria, May 8–12, 2016. Springer, Heidelberg, Germany.
4. Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, François-Xavier Standaert, and Pierre-Yves Strub. Improved parallel mask refreshing algorithms: generic solutions with parametrized non-interference and automated optimizations. *Journal of Cryptographic Engineering*, 10(1):17–26, April 2020.
5. Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, Pierre-Yves Strub, and Rébecca Zucchini. Strong non-interference and type-directed higher-order masking. In Edgar R. Weippl,

Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016: 23rd Conference on Computer and Communications Security*, pages 116–129, Vienna, Austria, October 24–28, 2016. ACM Press.

6. Gilles Barthe, François Dupressoir, Sebastian Faust, Benjamin Grégoire, François-Xavier Standaert, and Pierre-Yves Strub. Parallel implementations of masking schemes and the bounded moment leakage model. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017, Part I*, volume 10210 of *Lecture Notes in Computer Science*, pages 535–566, Paris, France, April 30 – May 4, 2017. Springer, Heidelberg, Germany.
7. 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`.
8. 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, *Advances in Cryptology – EUROCRYPT 2016, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 616–648, Vienna, Austria, May 8–12, 2016. Springer, Heidelberg, Germany.
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, *Advances in Cryptology – CRYPTO 2020, Part I*, volume 12170 of *Lecture Notes in Computer Science*, pages 339–368, Santa Barbara, CA, USA, August 17–21, 2020. Springer, Heidelberg, Germany.
10. Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Michael J. Wiener, editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 398–412, Santa Barbara, CA, USA, August 15–19, 1999. Springer, Heidelberg, Germany.
11. Jean-Sébastien Coron. Higher order masking of look-up tables. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 441–458, Copenhagen, Denmark, May 11–15, 2014. Springer, Heidelberg, Germany.
12. Jean-Sébastien Coron, Aurélien Greuet, and Rina Zeitoun. Side-channel masking with pseudo-random generator. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020, Part III*, volume 12107 of *Lecture Notes in Computer Science*, pages 342–375, Zagreb, Croatia, May 10–14, 2020. Springer, Heidelberg, Germany.
13. Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, and Thomas Roche. Higher-order side channel security and mask refreshing. In Shiho Moriai, editor, *Fast Software Encryption – FSE 2013*, volume 8424 of *Lecture Notes in Computer Science*, pages 410–424, Singapore, March 11–13, 2014. Springer, Heidelberg, Germany.
14. Alexandre Duc, Stefan Dziembowski, and Sebastian Faust. Unifying leakage models: From probing attacks to noisy leakage. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 423–440, Copenhagen, Denmark, May 11–15, 2014. Springer, Heidelberg, Germany.
15. Stefan Dziembowski, Sebastian Faust, and Karol Zebrowski. Simple refreshing in the noisy leakage model. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology – ASIACRYPT 2019, Part III*, volume 11923 of *Lecture Notes in Computer Science*, pages 315–344, Kobe, Japan, December 8–12, 2019. Springer, Heidelberg, Germany.
16. Louis Goubin and Jacques Patarin. DES and differential power analysis (the "duplication" method). In Çetin Kaya Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES'99*, volume 1717 of *Lecture Notes in Computer Science*, pages 158–172, Worcester, Massachusetts, USA, August 12–13, 1999. Springer, Heidelberg, Germany.
17. Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481, Santa Barbara, CA, USA, August 17–21, 2003. Springer, Heidelberg, Germany.
18. Paul C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Neal Koblitz, editor, *Advances in Cryptology – CRYPTO'96*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113, Santa Barbara, CA, USA, August 18–22, 1996. Springer, Heidelberg, Germany.
19. Emmanuel Prouff and Matthieu Rivain. Masking against side-channel attacks: A formal security proof. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 142–159, Athens, Greece, May 26–30, 2013. Springer, Heidelberg, Germany.
20. Matthieu Rivain and Emmanuel Prouff. Provably secure higher-order masking of AES. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems – CHES 2010*, volume 6225

of *Lecture Notes in Computer Science*, pages 413–427, Santa Barbara, CA, USA, August 17–20, 2010. Springer, Heidelberg, Germany.

A Random Probing Expandability 1 & 2

Definition 7 (Random Probing Expandability 1). *Let $f : \mathbb{R} \to \mathbb{R}$. An n-share gadget $G : \mathbb{K}^n \times \mathbb{K}^n \to \mathbb{K}^n$ is (t, f)-RPE1 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 $J \subseteq [n]$, such that $|J| \leq t$, and for every $p \in [0, 1]$, the random experiment*

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

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*

$$\Pr(\mathcal{F}_1) = \Pr(\mathcal{F}_2) = \varepsilon \quad \textit{and} \quad \Pr(\mathcal{F}_1 \wedge \mathcal{F}_2) = \varepsilon^2 \tag{8}$$

   with $\varepsilon = f(p)$ (in particular $\mathcal{F}_1$ and $\mathcal{F}_2$ are mutually independent),
2. *the output distribution satisfies*

$$\mathit{out} \overset{id}{=} \big(\mathsf{AssignWires}(G, \mathcal{W}, (\widehat{x}, \widehat{y}))\ ,\ \widehat{z}|_J\big) \tag{9}$$

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

Definition 8 (Random Probing Expandability 2). *Let $f : \mathbb{R} \to \mathbb{R}$. An n-share gadget $G : \mathbb{K}^n \times \mathbb{K}^n \to \mathbb{K}^n$ is (t, f)-RPE2 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 $p \in [0, 1]$, the random experiment*

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

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*

$$\Pr(\mathcal{F}_1) = \Pr(\mathcal{F}_2) = \varepsilon \quad \textit{and} \quad \Pr(\mathcal{F}_1 \wedge \mathcal{F}_2) = \varepsilon^2 \tag{10}$$

   with $\varepsilon = f(p)$ (in particular $\mathcal{F}_1$ and $\mathcal{F}_2$ are mutually independent),
2. *J is such that $J \subseteq [n]$ with $|J| = n - 1$*
3. *the output distribution satisfies*

$$\mathit{out} \overset{id}{=} \big(\mathsf{AssignWires}(G, \mathcal{W}, (\widehat{x}, \widehat{y}))\ ,\ \widehat{z}|_J\big) \tag{11}$$

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

B Tight Random Probing Expandability

Definition 9 (Tight Random Probing Expandability). *Let $f : \mathbb{R} \to \mathbb{R}$. An n-share gadget $G : \mathbb{K}^n \times \mathbb{K}^n \to \mathbb{K}^n$ is* (t, f)-tight random probing expandable (TRPE) *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*

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

ensures that

1. *the failure events $\mathcal{F}_1 \equiv \big(|I_1| > \min(t, |\mathcal{W}|)\big)$ and $\mathcal{F}_2 \equiv \big(|I_2| > \min(t, |\mathcal{W}|)\big)$ verify*

$$\Pr(\mathcal{F}_1) = \Pr(\mathcal{F}_2) = \varepsilon \quad \textit{and} \quad \Pr(\mathcal{F}_1 \wedge \mathcal{F}_2) = \varepsilon^2 \tag{12}$$

   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*

$$\mathit{out} \overset{id}{=} \big(\mathsf{AssignWires}(G, \mathcal{W}, (\widehat{x}, \widehat{y}))\ ,\ \widehat{z}|_{J'}\big) \tag{13}$$

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

Definition 10 (Tight Random Probing Expandability 1). *Let $f : \mathbb{R} \to \mathbb{R}$. An n-share gadget $G : \mathbb{K}^n \times \mathbb{K}^n \to \mathbb{K}^n$ is* (t, f)-tight random probing expandable (TRPE) *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 $J \subseteq [n]$, such that $|J| \leq t$, and for every $p \in [0, 1]$, the random experiment*

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

ensures that

1. *the failure events $\mathcal{F}_1 \equiv \big(|I_1| > \min(t, |\mathcal{W}|)\big)$ and $\mathcal{F}_2 \equiv \big(|I_2| > \min(t, |\mathcal{W}|)\big)$ verify*

$$\Pr(\mathcal{F}_1) = \Pr(\mathcal{F}_2) = \varepsilon \quad \textit{and} \quad \Pr(\mathcal{F}_1 \wedge \mathcal{F}_2) = \varepsilon^2 \tag{14}$$

   with $\varepsilon = f(p)$ (in particular $\mathcal{F}_1$ and $\mathcal{F}_2$ are mutually independent),
2. *the output distribution satisfies*

$$\mathit{out} \overset{id}{=} \big(\mathsf{AssignWires}(G, \mathcal{W}, (\widehat{x}, \widehat{y}))\ ,\ \widehat{z}|_{J}\big) \tag{15}$$

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

Definition 11 (Tight Random Probing Expandability 2). *Let $f : \mathbb{R} \to \mathbb{R}$. An n-share gadget $G : \mathbb{K}^n \times \mathbb{K}^n \to \mathbb{K}^n$ is* (t, f)-tight random probing expandable (TRPE) *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 $p \in [0, 1]$, the random experiment*

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

ensures that

1. *the failure events $\mathcal{F}_1 \equiv \big(|I_1| > \min(t, |\mathcal{W}|)\big)$ and $\mathcal{F}_2 \equiv \big(|I_2| > \min(t, |\mathcal{W}|)\big)$ verify*

$$\Pr(\mathcal{F}_1) = \Pr(\mathcal{F}_2) = \varepsilon \quad \textit{and} \quad \Pr(\mathcal{F}_1 \wedge \mathcal{F}_2) = \varepsilon^2 \tag{16}$$

   with $\varepsilon = f(p)$ (in particular $\mathcal{F}_1$ and $\mathcal{F}_2$ are mutually independent),
2. *J is such that $J \subseteq [n]$ with $|J| = n - 1$*
3. *the output distribution satisfies*

$$\mathit{out} \overset{id}{=} \big(\mathsf{AssignWires}(G, \mathcal{W}, (\widehat{x}, \widehat{y}))\ ,\ \widehat{z}|_J\big) \tag{17}$$

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

C Proof of Lemma 7

Proof. To prove that G_{copy} is TRPE achieving the same amplification order d as the underlying refresh gadget G_{refresh}, we need to prove that any set of leaking wires $\mathcal{W}$ such that $|\mathcal{W}| \leq d - 1$ can be perfectly simulated together with any sets of outputs wires $J_1, J_2 \subseteq [n]$ (such that J_1 refers to the first output e and J_2 to the second output f) from a set of input wires I such that $|I| \leq \min(t, |\mathcal{W}|)$. In addition, we know from Lemma 1 that the maximal amplification order achievable in the TRPE setting is $d_{\max} \leq \min(t + 1, 2(n - t))$. Since we consider sets $\mathcal{W}$ of size at most $|\mathcal{W}| \leq d-1 \leq \min(t+1, 2(n-t))-1 \leq t$ then we need to prove that $|I| \leq \min(t, |\mathcal{W}|) = |\mathcal{W}|$.

The leaking set $\mathcal{W}$ can be split into two distinct subsets $\mathcal{W}_1$ and $\mathcal{W}_2$ such that $\mathcal{W} = \mathcal{W}_1 \cup \mathcal{W}_2$ where $\mathcal{W}_1$ (resp. $\mathcal{W}_2$) is the set of leaking wires of G_{refresh} for the output e (resp. f). Let $J_1, J_2 \subseteq [n]$. We consider four cases:

– $|J_1| \leq t, |J_2| \leq t$: since $|\mathcal{W}| \leq d - 1$, then $|\mathcal{W}_1|, |\mathcal{W}_2| \leq d - 1$. Since G_{refresh} achieves an amplification order d, then by definition of TRPE, the sets $\mathcal{W}_1$ and J_1 can be simulated with a set of input shares I_1 such that $|I_1| \leq \min(|\mathcal{W}_1|, t)$. Similarly, the sets $\mathcal{W}_2$ and J_2 can be simulated with a set of input shares I_2 such that $|I_2| \leq \min(|\mathcal{W}_2|, t)$. As a consequence, set I defined as $I = I_1 \cup I_2$ is enough to simulate $\mathcal{W} = \mathcal{W}_1 \cup \mathcal{W}_2$ and both output shares J_1 and J_2. Furthermore, we have

$$|I| \leq |I_1| + |I_2| \leq \min(|\mathcal{W}_1|, t) + \min(|\mathcal{W}_2|, t) \leq |\mathcal{W}| = \min(|\mathcal{W}|, t)$$

– $|J_1| > t, |J_2| > t$: in this case, we need to prove the existence of a set of input shares I such that $|I| \leq \min(t, |W|) = |\mathcal{W}|$ (since $|\mathcal{W}| \leq d-1 \leq t$) for which we can perfectly simulate $\mathcal{W}$ together with two chosen output sets J_1' and J_2' such that $|J_1'| = |J_2'| = n-1$. Since we have $\mathcal{W} = \mathcal{W}_1 \cup \mathcal{W}_2$ such that $|\mathcal{W}_1| \leq d-1, |\mathcal{W}_2| \leq d-1$, then by definition of TRPE, there exists $J_1', |J_1'| = n-1$ such that $\mathcal{W}_1$ and J_1' can be perfectly simulated from a set of inputs shares I_1 such that $|I_1| \leq \min(|\mathcal{W}_1|, t)$. Similarly, there exists $J_2', |J_2'| = n-1$ such that $\mathcal{W}_2$ and J_2' can be perfectly simulated from a set of inputs shares I_2 such that $|I_2| \leq \min(|\mathcal{W}_2|, t)$. By choosing such sets J_1', J_2', the overall simulation of G_{copy} can be done with the set of input shares $I = I_1 \cup I_2$, and we have

$$|I| \leq |I_1| + |I_2| \leq \min(|\mathcal{W}_1|, t) + \min(|\mathcal{W}_2|, t) \leq |\mathcal{W}| = \min(|\mathcal{W}|, t)$$

– $|J_1| \leq t, |J_2| > t$: Since $|J_1| \leq t$, by definition of TRPE, $\mathcal{W}_1$ and J_1 can be perfectly simulated from a set of input shares I_1 such that $|I_1| \leq \min(|\mathcal{W}_1|, t)$. In addition, for $|J_2| > t$, we also know that we can choose a set J_2' such that $|J_2'| = n-1$ that can be perfectly simulated with $\mathcal{W}_2$ from a set of input shares I_2 with $|I_2| \leq \min(|\mathcal{W}_2|, t)$. By choosing such a set J_2', the overall simulation of G_{copy} can be achieved with the set of input wires $I = I_1 \cup I_2$, and we have

$$|I| \leq |I_1| + |I_2| \leq \min(|\mathcal{W}_1|, t) + \min(|\mathcal{W}_2|, t) \leq |\mathcal{W}| = \min(|\mathcal{W}|, t)$$

– $|J_1| > t, |J_2| \leq t$: the proof is exactly the reflect of the previous one.

Since in the four cases, there is no failure tuple $\mathcal{W}$ of size $|\mathcal{W}| < d$, then the gadget G_{copy} achieves an amplification order d. Lemma 1 finally completes the proof. □

D Proof of Lemma 8

Proof. We need to prove that when G_{refresh} is (t, f)-RPE (resp. (t, f')-TRPE) of amplification order d, then G_{add} is (t, f')-RPE (resp. (t, f')-TRPE) of amplification order at least $\lfloor \frac{d}{2} \rfloor$. We will prove the property for the RPE setting, and the proof for the TRPE setting will be exactly the same except for the notion of failure event which changes. This amounts to proving that:

1. Any set of leaking wires $\mathcal{W}$ such that $|\mathcal{W}| < \lfloor \frac{d}{2} \rfloor$ 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 $|I_1| \leq t$ and $|I_2| \leq t$ (for TRPE we would have $|I_1| \leq \min(t, |\mathcal{W}|)$ and $|I_2| \leq \min(t, |\mathcal{W}|)$).
2. Any set of leaking wires such that $\lfloor \frac{d}{2} \rfloor \leq |\mathcal{W}| < d$ can be simulated together with any set of outputs wires $J \subseteq [n]$ from sets of input wires I_1, I_2 such that $|I_1| \leq t$ or $|I_2| \leq t$ (because of the double failure, *i.e* failure on both inputs) (for TRPE we would have $|I_1| \leq \min(t, |\mathcal{W}|)$ or $|I_2| \leq \min(t, |\mathcal{W}|)$).

We proceed by building the necessary simulators for G_{add} from the simulators that already exist for G_{refresh}. Concretely, we split each set $\mathcal{W}$ of leaking wires, into four subsets $\mathcal{W} = \mathcal{W}_1^r \cup W_1^a \cup \mathcal{W}_2^r \cup W_2^a$ where $\mathcal{W}_1^r$ (resp. $\mathcal{W}_2^r$) 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 $\mathcal{W}_1^a$ (resp. $\mathcal{W}_2^a$) is the set of leaking wires of $(e_1, \ldots, e_n)$ (resp. $(f_1, \ldots, f_n)$).

From these notations, we build a leaking set $\mathcal{W}'$ which contains $\mathcal{W}_1^r$ and $\mathcal{W}_2^r$ and also each input or pair of inputs of gates whose output is a wire in $\mathcal{W}_1^a$ or $\mathcal{W}_2^a$. We have that

$$|\mathcal{W}'| \leq |\mathcal{W}_1^r| + |\mathcal{W}_2^r| + 2|\mathcal{W}_1^a| + 2|\mathcal{W}_2^a| \leq 2|\mathcal{W}|.$$

The new set $\mathcal{W}'$ can be split into two subsets $\mathcal{W}'_1$ and $\mathcal{W}'_2$ such that $\mathcal{W}'_1$ (resp. $\mathcal{W}'_2$) 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 $\mathcal{W}'$ 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 $|\mathcal{W}| < \lfloor \frac{d}{2} \rfloor$. Then we consider the set $\mathcal{W}' = \mathcal{W}'_1 \cup \mathcal{W}'_2$ (as previously defined) such that
  $$|\mathcal{W}'| \leq 2|\mathcal{W}| < 2\lfloor \frac{d}{2} \rfloor \leq d$$
  and hence,
  $$|\mathcal{W}'_1| < d \quad \text{and} \quad |\mathcal{W}'_2| < d.$$
  From the (t, f)-RPE property of G_{refresh} and its amplification order, there exists an input set of shares of a I_1 such that $|I_1| \leq t$ (for TRPE we would have $|I_1| \leq \min(t, |\mathcal{W}|)$) and I_1 perfectly simulates $\mathcal{W}'_1$ and any set J_1 of up to t variables e_i. Similarly, there exists an input set of shares of b I_2 such that $|I_2| \leq t$ (for TRPE we would have $|I_2| \leq \min(t, |\mathcal{W}|)$) and I_2 perfectly simulates $\mathcal{W}'_2$ and any set J_2 of up to t variables f_i. J_1 and J_2 are chosen as the inputs e_i and f_i respectively of wires $e_i + f_i$ in set J. Namely $|J_1| = |J_2| = |J|$.
  From these definitions, I_1 and I_2 together perfectly simulate $\mathcal{W}'$ and J and are both of size less than t (less than $\min(t, |\mathcal{W}|)$ for TRPE), which proves the first property in this scenario.
  2. we now assume that $\lfloor \frac{d}{2} \rfloor \leq |\mathcal{W}| < d$. Then we consider the set $\mathcal{W}' = \mathcal{W}'_1 \cup \mathcal{W}'_2$ such that
  $$|\mathcal{W}'| \leq 2|\mathcal{W}| < 2d$$
  and hence,
  $$|\mathcal{W}'_1| < d \quad \text{or} \quad |\mathcal{W}'_2| < d.$$
  Without loss of generality, let us consider that $|\mathcal{W}'_1| < d$ (the proof is similar in the opposite scenario). From the (t, f)-RPE property of G_{refresh} and its amplification order, there exists an input set of shares of a I_1 such that $|I_1| \leq t$ (for TRPE we would have $|I_1| \leq \min(t, |\mathcal{W}|)$) and I_1 perfectly simulates $\mathcal{W}'_1$ and any set J_1 of up to t variables e_i. There also exists an input set of shares of b I_2 which perfectly simulates $\mathcal{W}'_2$ and any set J_2 of up to t variables f_i but not necessarily of size less than t (less than $\min(t, |\mathcal{W}|)$ for TRPE). If J_1 and J_2 are chosen as the inputs e_i and f_i respectively of wires $e_i + f_i$ in set J, then sets I_1 and I_2 together perfectly simulate $\mathcal{W}'$ and J. In this case, we only have a failure on at most one of the inputs (b in this case), which concludes the proof for the second property.

  At this point, we proved that G_{add} achieves an amplification order greater than or equal to $\lfloor \frac{d}{2} \rfloor$ for RPE1 (for TRPE1 in the TRPE setting).
- if $|J| > t$ ((T)RPE2): we prove both properties 1 and 2:
  1. we assume that $|\mathcal{W}| < \lfloor \frac{d}{2} \rfloor$. Then we consider the set $\mathcal{W}' = \mathcal{W}'_1 \cup \mathcal{W}'_2$ (as previously defined) such that
  $$|\mathcal{W}'| \leq 2|\mathcal{W}| < d.$$
  $\mathcal{W}'_1$ and $\mathcal{W}'_2$ both point to leaking wires in instances of G_{refresh}. We denote by $\mathcal{W}''_1$ the set of leaking wires on the first instance of G_{refresh} (on input a) such that $\mathcal{W}''_1$ contains $\mathcal{W}'_1$ and all the wires that are leaking within the second instance of G_{refresh} (designated by $\mathcal{W}'_2$ in this

second instance). Hence, we have that $|\mathcal{W}_1''| \leq |\mathcal{W}_1' \cup W_2'| < d$. From the (t, f)-RPE ((t, f)-TRPE in the TRPE setting) property of G_{refresh} and its amplification order, there exists an input set of shares of a I_1 such that $|I_1| \leq t$ (for TRPE we would have $|I_1| \leq \min(t, |\mathcal{W}|)$) and a set of output shares e_i J_1' of size $n-1$ such that the input shares of I_1 perfectly simulate the wires designated by $\mathcal{W}_1''$ and J_1'. Similarly, as both instances of G_{refresh} are identical, the same set I_2 but of input shares b perfectly simulates $\mathcal{W}_2''$ (defined as the equivalent of $\mathcal{W}_1''$ on the second instance) and J_2' which points to the same output shares than J_1 but on f_i instead of e_i. We thus have two input sets I_1 and I_2 of size less than t (less than $\min(t, |\mathcal{W}|)$ for TRPE2) whose shares perfectly simulate the wires $\mathcal{W}'$ and the elements $e_i + f_i$ of a set J' of size $n-1$ with $i \in J_1' = J_2'$. That concludes the proof for the first property.

2. we now assume that $\lfloor \frac{d}{2} \rfloor \leq |\mathcal{W}| < d$. Then we consider the set $\mathcal{W}' = \mathcal{W}_1' \cup \mathcal{W}_2'$ such that

$$|\mathcal{W}'| \leq 2|\mathcal{W}| < 2d.$$

Without loss of generality, let us consider that $|\mathcal{W}_1'| < d$ (the proof is similar in the opposite scenario). From the (t, f)-RPE ((t, f)-TRPE in the TRPE setting) property of G_{refresh} and its amplification order, there exists a set J_1' such that $|J_1'| = n-1$ and a set of input shares I_1 such that I_1 perfectly simulates $\mathcal{W}_1'$ and J_1' and $|I_1| \leq t$ (for TRPE we would have $|I_1| \leq \min(t, |\mathcal{W}|)$). Thus, we can select a set J' of outputs of G_{add} such that J' corresponds to the outputs of J_1 (for each element e_i designated by J_1, $e_i + f_i$ is pointed by J). Then, by choosing $I_2 = [n]$, we have two input sets I_1 and I_2 which perfectly simulate $\mathcal{W}'$ and an output set J' of size $n-1$ such that $|I_1| \leq t$ (for TRPE we would have $|I_1| \leq \min(t, |\mathcal{W}|)$). That concludes the proof for the second property.

We thus proved that G_{add} achieves an amplification order greater than or equal to $\lfloor \frac{d}{2} \rfloor$ for RPE2 (for TRPE2 in the TRPE setting).

Since G_{add} has an amplification order greater than or equal to $\lfloor \frac{d}{2} \rfloor$ for RPE1 and RPE2 (resp. TRPE1 and TRPE2), then G_{add} is a (t, f')-RPE (resp. (t, f')-TRPE) addition gadget for some function f' of amplification order $d' \geq \lfloor \frac{d}{2} \rfloor$, which concludes the proof. □

E Proof of Lemma 9

Proof. We start by proving the first property of the lemma, i.e the amplification order d_1 for TRPE1. The n-share ISW refresh gadget was proven to be $(n-1)$-SNI [5], hence it follows from Lemma 6 that $d_1 \geq \min(t+1, n-t)$. In addition, we know from the proof of Lemma 1 and as explained in section 4.1 that $d_1 \leq t+1$. It remains to show that $d_1 \leq n-t$. We thus have to exhibit a simulation failure by carefully choosing $n-t$ leaking variables (the leaking set $\mathcal{W}$) together with t leaking output variables (indexed by the set J). Consider the set of output shares indexed by $J = \{1, \ldots, t\}$, which corresponds to the first t shares $c_1, \ldots, c_t$ of the output. Next, we construct the set of leaking wires $\mathcal{W}$ of size $n-t$. First, observe that the partial sums of the output shares are of the form

$$c_{i,j} = \begin{cases} a_i + r_{1,i} + \cdots + r_{j,i} & \text{if } j < i \\ a_i + r_{1,i} + \cdots + r_{i-1,i} + r_{i,i+1} + \cdots + r_{i,j} & \text{otherwise.} \end{cases}$$

Then, let $\mathcal{W} = \{c_{t+1,t}, \ldots, c_{n,t}\}$. We can prove that the constructed set $\mathcal{W}$ along with the set of indexes of output shares $J = \{1, \ldots, t\}$ cannot be perfectly simulated with at most $\min(t, |\mathcal{W}|)$

input shares. For this, we consider a variable $s = c_1 + \cdots + c_t + c_{t+1,t} + \ldots + c_{n,t}$, the sum of the t output shares indexed in J, and the leaking variables from $\mathcal{W}$. Each of the output shares $\{c_i\}_{1\leq i\leq t}$ is the sum of exactly one input share a_i and $n-1$ random values. Each of the leaking variables $\{c_{i,t}\}_{t+1\leq i\leq n}$ is the sum of exactly one input share a_i and t random values. In addition, it can be observed that each random value appears exactly twice in the set of expressions of the variables $\{c_i\}_{1\leq i\leq t} \cup \{c_{i,t}\}_{t+1\leq i\leq n}$, so all of the random values are eliminated in the expression of the variable s, which is the sum of all of these variables. Since each of the variables has one input share a_i appearing in its expression, then we have $s = a_1 + \ldots + a_n = a$. Thus, simulating the variable s requires the knowledge of the full input, and hence the leaking variables indexed by $\mathcal{W}$ and J cannot be perfectly simulated without the knowledge of the full input. Hence, the set $\mathcal{W}$ of size $n-t$ represents a failure set with respect to TRPE1, and so the function f_1 cannot be of amplification order higher than $n-t$, that is $d_1 \leq n-t$. From the three inequalities $d_1 \geq \min(t+1, n-t)$, $d_1 \leq t+1$ and $d_1 \leq n-t$, we obtain $d_1 = \min(t+1, n-t)$.

Next, we demonstrate the second part of the lemma, *i.e.* the amplification order d_2 for TRPE2. Let $\mathcal{W}$ be a set of leaking wires such that $|\mathcal{W}| < t+1$. We aim to prove that there exists a set J indexing $n-1$ output shares such that the leaking variables indexed by both $\mathcal{W}$ and J can be perfectly simulated with the input shares indexed by a set I such that $|I| \leq \min(t, |\mathcal{W}|) = |\mathcal{W}|$. First, we observe that the leaking wires in $\mathcal{W}$ are of the following forms:

1. input share a_i
2. random variable r_{ij} $(i < j)$
3. partial sum $c_{ij} = \begin{cases} a_i + r_{1,i} + \cdots + r_{j,i} & \text{if } j < i \\ a_i + r_{1,i} + \cdots + r_{i-1,i} + r_{i,i+1} + \cdots + r_{i,j} & \text{otherwise.} \end{cases}$

We then build I from an empty set as follows. For every wire in $\mathcal{W}$ of the first or third form, we add index i to I. For every wire in $\mathcal{W}$ of the second form (r_{ij}), if $i \in I$, we add j to I, otherwise we add i to I. By construction we have $|I| \leq |\mathcal{W}| \leq t$. Moreover, the wires in the set $\mathcal{W}$ only depends of the input shares a_i with $i \in I$ which implies that we can perfectly simulate the variables indexed by $\mathcal{W}$ from the input shares indexed by I. We then build the set J as the union of two subsets J_1 and J_2 such that $J_1 = I$ and J_2 is any set satisfying $|J_2| = n-1-|I|$ and $J_1 \cap J_2 = \emptyset$. Now, we aim to show that the output shares determined by the indexes in $J = J_1 \cup J_2$ can be further perfectly simulated from the input shares indexed by I (namely given the previous simulation of the variables from $\mathcal{W}$). The simulation works as follows:

- each output share c_i such that $i \in J_1$ can be perfectly simulated with a_i (since $i \in I$) and $n-1$ uniformly random variables (the same generated r_{ij} can be reused for several c_i);
- for each output share c_i such that $i \in J_2$, we have $i \notin I$ and hence a_i is not available. Since by construction of J_1, all the variables observed through the set $\mathcal{W}$ are included in the set of variables observed through J_1, and since $|J_1| \leq |\mathcal{W}| \leq t \leq n-2$ and $|J_2| = n-1-|J_1|$, then each output wire c_i indexed in J_2 has at least one random value that does not appear in any other observation from $\mathcal{W}$ or J_1, so c_i can be assigned to a fresh random value. This produces a perfect simulation of all output wires indexed in J_2.

We thus obtain a perfect simulation of the output shares indexed by $J = J_1 \cup J_2$, such that $|J| = n-1$, together with the variables indexed by $\mathcal{W}$, from the input shares indexed by a set I of size $|I| \leq |\mathcal{W}| \leq t$, so $|I| \leq \min(t, |\mathcal{W}|)$. Hence the ISW refresh gadget is (t, f)-TRPE2 with an amplification order $d_2 \geq t+1$. In addition, we know from the proof of Lemma 1 and as explained in Section 4.1 that $d_2 \leq t+1$, hence $d_2 = t+1$ which concludes the proof. □

F Proof of Lemma 10

Proof. In order to prove that the amplification order d of G_{copy} instantiated with the ISW refresh gadget is equal to $\min(t+1, n-t)$, we first demonstrate that $d \geq \min(t+1, n-t)$ and then we show the existence of failure tuples to argue that $d \leq \min(t+1, n-t)$.

In fact, we already know that the ISW refresh gadget is (t, f_1)-TRPE of amplification order $d_1 = \min(t+1, n-t)$. Then from Lemma 7, we know that G_{copy} instantiated with ISW refresh is also (t, f_2)-TRPE of amplification order $d_2 = d_1 = \min(t+1, n-t)$. Then, from Lemma 4 we have that G_{copy} is (t, f)-RPE of amplification order $d \geq d_2 = \min(t+1, n-t)$. Next, we need to prove that $d \leq \min(t+1, n-t)$. In addition, we know from Lemma 1 that $d \leq t+1$. Hence, it remains to show that it is also upper bounded by $n-t$.

We know from the proof of Lemma 9 that, for the ISW refresh gadget, we can construct a set of leaking wires $\mathcal{W}$ of size $n-t$ along with a set of t indexes of output shares J such that a perfect simulation of both sets W and J requires the knowledge of the full input sharing *i.e.* $I = [n]$. Then, in the case of the copy gadget G_{copy}, let $\mathcal{W}$ be the set of leaking wires and $J_1, J_2 \subseteq [n]$ be the sets of output shares on the outputs e and f respectively. Then, we can split $\mathcal{W}$ into two distinct subsets $\mathcal{W}_1$ and $\mathcal{W}_2$ such that $\mathcal{W} = \mathcal{W}_1 \cup \mathcal{W}_2$, where $\mathcal{W}_1$ (resp. $\mathcal{W}_2$) is the set of leaking wires of ISW G_{refresh} for the output e (resp. f). Then, in the case where $|J_1| \leq t$, we can construct the set $\mathcal{W} = \mathcal{W}_1$ of size $n-t$ ($\mathcal{W}_2 = \emptyset$) in the exact same way as in the proof of Lemma 9, such that we have simulation failure of $\mathcal{W}_1$ along with the output shares indexed in J_1 on the input of the gadget. Otherwise, in the case where $|J_2| \leq t$, we can construct the set $\mathcal{W} = \mathcal{W}_2$ of size $n-t$ ($\mathcal{W}_1 = \emptyset$) in the exact same way, such that we have simulation failure of $\mathcal{W}_2$ along with the output shares indexed in J_2 on the input of the gadget. Hence, the amplification order d of G_{copy} is upper bounded by $n-t$.

From the three inequalities $d \geq \min(t+1, n-t)$, $d \leq t+1$ and $d \leq n-t$, we conclude that the copy gadget instantiated with ISW refresh is (t, f)-RPE of amplification order $d = \min(t+1, n-t)$. □

G Proof of Lemma 11

Proof. We proceed similarly to the proof of Lemma 8 to show that the function f_1 (resp. f_2) is of amplification order at least $\min(t+1, n-t)$ (resp. $(t+1)$). We split each set $\mathcal{W}$ of leaking wires, into four subsets $\mathcal{W} = \mathcal{W}_1^r \cup W_1^a \cup \mathcal{W}_2^r \cup W_2^a$ where $\mathcal{W}_1^r$ (resp. $\mathcal{W}_2^r$) 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 $\mathcal{W}_1^a$ (resp. $\mathcal{W}_2^a$) is the set of leaking wires of $(e_1, \ldots, e_n)$ (resp. $(f_1, \ldots, f_n)$). Then, we prove using previous lemmas that the amplification order for TRPE1 (resp. for TRPE2) at most $\min(t+1, n-t)$ (resp. $(t+1)$) by exhibiting failure tuples. Hence the final equalities.

- if $|J| \leq t$ (TRPE1): we prove two properties like in Lemma 8:
  1. we assume that $|\mathcal{W}| < \min(t+1, n-t)$, in particular $|\mathcal{W}_1^r| + |\mathcal{W}_1^a| < \min(t+1, n-t) \leq n-t$ and $|\mathcal{W}_2^r| + |\mathcal{W}_2^a| < \min(t+1, n-t) \leq n-t$. Since we have $|J| \leq t$, then $|\mathcal{W}_1^r| + |\mathcal{W}_1^a| + |J| \leq n-1$ and $|\mathcal{W}_2^r| + |\mathcal{W}_2^a| + |J| \leq n-1$. Then from the $(n-1)$-SNI property of the ISW refresh gadget, and by choosing J_1 and J_2 as the inputs e_i and f_i respectively of the wires $e_i + f_i$ in set J ($|J_1| = |J_2| = |J|$), we know that there exists an input set of shares of a I_1 such that $|I_1| \leq |\mathcal{W}_1^r| \leq |\mathcal{W}| \leq t$ and I_1 perfectly simulates $\mathcal{W}_1^r$, $\mathcal{W}_1^a$ and J_1. And there exists an

input set of shares of b I_2 such that $|I_2| \leq |\mathcal{W}_2^r| \leq |W| \leq t$ and I_2 perfectly simulates $\mathcal{W}_2^r$, $\mathcal{W}_2^a$ and J_2. Thus I_1 and I_2 together perfectly simulate $\mathcal{W}$ and J and are both of size less than $|\mathcal{W}| \leq t$ so less than $\min(t, |\mathcal{W}|)$. Hence, there is no failure on the inputs for any set of leaking wires $\mathcal{W}$ of size strictly less than $\min(t+1, n-t)$ along with a set J of at most t output shares.

2. we now assume that $\min(t+1, n-t) \leq |\mathcal{W}| < 2 \cdot \min(t+1, n-t)$. Without loss of generality, let us consider that $|\mathcal{W}_1^r| + |\mathcal{W}_1^a| < \min(t+1, n-t)$. We consider the set of input shares of b $I_2 = [n]$, which trivially simulates all of the wires in $\mathcal{W}_2^r$, $\mathcal{W}_2^a$ and the inputs f_i of the wires $e_i + f_i$ in set J. Next, since $|\mathcal{W}_1^r| + |\mathcal{W}_1^a| < \min(t+1, n-t) \leq n-t$, by choosing J_1 as the inputs e_i of the wires $e_i + f_i$ in set J, we know that $|\mathcal{W}_1^r| + |\mathcal{W}_1^a| + |J_1| \leq n-1$ and by the $(n-1)$-SNI property of ISW refresh gadget, there exists a set of input shares of a I_1 such that $|I_1| \leq |\mathcal{W}_1^r| \leq t$ so $|I_1| \leq \min(t, |\mathcal{W}_1^r|) \leq \min(t, |\mathcal{W}|)$ and I_1 perfectly simulates $\mathcal{W}_1^r$, $\mathcal{W}_1^a$ and J_1. Then I_1 and I_2 together perfectly simulate $\mathcal{W}$ and J, and we only have a failure on input b with $|I_2| = n > t$. Thus, for any set of leaking wires $\mathcal{W}$ such that $\min(t+1, n-t) \leq |\mathcal{W}| < 2 \cdot \min(t+1, n-t)$, we have a failure on at most one of the inputs.

From the above properties, we have that G_{add} is of amplification order $d_1 \geq \min(t+1, n-t)$ for TRPE1. Then, from the proof of Lemma 1, we know that there exists an immediate failure tuple of size $t+1$ (on the input shares), hence $d_1 \leq t+1$. From the same proof of Lemma 1, we also know that there exists a failure tuple of size $2(n-t)$ (with a set of t output shares). Since G_{add} has two inputs, then it results in the following lower bound: $d_1 \geq n-t$. From these three inequalities on d_1, we obtain $d_1 = \min(t+1, n-t)$.

– if $|J| > t$ (TRPE2): we also prove both properties 1 and 2:

1. we assume that $|\mathcal{W}| < t+1$ with $\mathcal{W} = \mathcal{W}_1^r \cup W_1^a \cup \mathcal{W}_2^r \cup W_2^a$. We need to prove that there exists a set J of $n-1$ output wires such that $\mathcal{W}$ and J can be perfectly simulated with sets of input shares I_1 and I_2 such that $|I_1| \leq \min(t, |\mathcal{W}|) = |\mathcal{W}|$ and $|I_2| \leq \min(t, |\mathcal{W}|) = |\mathcal{W}|$. Recall that all of the wires in $\mathcal{W}_1^r$ (resp. $\mathcal{W}_2^r$) are of the following forms:
   (a) input share a_i (resp. b_i).
   (b) random variable r_{ij} (resp. r'_{ij}) with $i < j$.
   (c) partial sum $e_{ij} = \begin{cases} a_i + r_{1,i} + \cdots + r_{j,i} & \text{if } j < i \\ a_i + r_{1,i} + \cdots + r_{i-1,i} + r_{i,i+1} + \cdots + r_{i,j} & \text{otherwise.} \end{cases}$
   resp. $f_{ij} = \begin{cases} b_i + r'_{1,i} + \cdots + r'_{j,i} & \text{if } j < i \\ b_i + r'_{1,i} + \cdots + r'_{i-1,i} + r'_{i,i+1} + \cdots + r'_{i,j} & \text{otherwise.} \end{cases}$

   In addition, the wires in $\mathcal{W}_1^a$ (resp. $\mathcal{W}_2^a$) are output wires of G_{refresh} of the form e_i (resp. f_i). We build I_1 and I_2 from empty sets as follows. For every wire in $\mathcal{W}_1^a \cup \mathcal{W}_2^a$, we add index i to I_1 and I_2. Next, for every wire in $\mathcal{W}_1^r \cup \mathcal{W}_2^r$ of the first or third form, we add index i both to I_1 and I_2. For every wire in $\mathcal{W}_1^r \cup \mathcal{W}_2^r$ of the second form, if $i \in I_1 (= I_2)$, we add j to I_1 and I_2. Otherwise, we add i to I_1 and I_2. It is clear that $I_1 = I_2$, and that $|I_1| = |I_2| \leq |\mathcal{W}| \leq t$. Following the t-SNI proof of the ISW refresh gadget, we can show that $\mathcal{W}_1^a$ and $\mathcal{W}_1^r$ are perfectly simulated using shares of indexes in I_1. Respectively, $\mathcal{W}_2^a$ and $\mathcal{W}_2^r$ are perfectly simulated using shares of indexes in I_2. So all wires in $\mathcal{W}$ are perfectly simulated using shares of indexes in I_1 and I_2. We then build the set J of $n-1$ indexes of output shares from two subsets J_1 and J_2. We define $J_1 = I_1 (= I_2)$, and J_2 as any set such that $J_2 = n - 1 - |J_1|$ and $J_1 \cap J_2 = \emptyset$. We now show that the output shares determined by the indexes in $J = J_1 \cup J_2$ can be perfectly simulated from I_1 and I_2:

- each output share $e_i + f_i$ such that $i \in J_1 \subseteq J$ can be perfectly simulated from e_i and f_i. Precisely, e_i can be perfectly simulated with a_i (since $i \in I_1$) and $n-1$ uniformly random variables. And each f_i can be perfectly simulated with b_i (since $i \in I_2$) and $n-1$ uniformly random variables.
- for each output share $e_i + f_i$ such that $i \in J_2$ so $i \notin I_1$, $i \notin I_2$, we show that we can still perfectly simulate e_i and f_i. By construction of the set J_1, all the variables observed through the set $\mathcal{W}$ are included in the set of variables observed through e_j and f_j for $j \in J_1$, and since $|J_1| = |I_1| \leq |\mathcal{W}| \leq t \leq n-2$ and $|J_2| = n-1-|J_1|$, then each of the wires e_i and f_i for which $e_i + f_i$ is indexed in J_2 has at least one random value that does not appear in any other observation from $\mathcal{W}$ or wires e_i and f_i for which $e_i + f_i$ is indexed in J_1. So e_i can be assigned to a fresh random value, and f_i can be assigned to a fresh random value. Thus $e_i + f_i$ is also assigned to a random value. This produces a perfect simulation of all output wires indexed in J_2.

Having a perfect simulation of J_1 and J_2, we conclude that we can perfectly simulate J along with $\mathcal{W}$ from the sets I_1 and I_2 with $|I_1| = |I_2| \leq |\mathcal{W}|$. So for every set of leaking wires $\mathcal{W}$ of size at most t, there exists a set of $n-1$ output wires J which can be perfectly simulated along with $\mathcal{W}$ from sets of input shares I_1 and I_2 of sizes at most $|\mathcal{W}| = \min(|\mathcal{W}|, t)$.

2. we now assume that $t+1 \leq |\mathcal{W}| < 2(t+1)$. Without loss of generality, let us consider that $|\mathcal{W}_1^r \cup \mathcal{W}_1^a| < t+1$ (the proof is similar in the opposite scenario). We consider the set of input shares of b $I_2 = [n]$ which trivially simulates all of the wires in $\mathcal{W}_2^r \cup \mathcal{W}_2^a$ and all of the inputs f_i of the output wires $e_i + f_i$. We construct the set I_1 for input shares of a similarly to the earlier construction (when $|\mathcal{W}| < t+1$), while only considering the sets $\mathcal{W}_1^r$ and $\mathcal{W}_1^a$. The corresponding set I_1 will produce a perfect simulation of $\mathcal{W}_1^r \cup \mathcal{W}_1^a$. So I_1 and I_2 perfectly simulate the set $\mathcal{W}$. Now we choose the set J from two subsets J_1 and J_2 such that $J_1 = I_1$ and J_2 as any set such that $J_2 = n-1-|J_1|$ and $J_1 \cap J_2 = \emptyset$. We now show that the output shares determined by the indexes in $J = J_1 \cup J_2$ can be perfectly simulated from I_1 and I_2:
   - each output share $e_i + f_i$ such that $i \in J_1 \subseteq J$ can be perfectly simulated from e_i and f_i. Precisely, e_i can be perfectly simulated with a_i (since $i \in I_1$) and $n-1$ uniformly random variables. And each f_i can be perfectly simulated with b_i (since $I_2 = [n]$) and $n-1$ uniformly random variables.
   - for each output share $e_i + f_i$ such that $i \in J_2$ so $i \notin I_1$, the input wire f_i can be perfectly simulated with b_i (since $I_2 = [n]$) and $n-1$ uniformly random variables. In addition, by construction of the set J_1, all the variables observed through the set $\mathcal{W}_1^r \cup \mathcal{W}_1^a$ are included in the set of variables observed through e_j for $j \in J_1$, and since $|J_1| = |I_1| \leq |\mathcal{W}| \leq t \leq n-2$ and $|J_2| = n-1-|J_1|$, then each of the wires e_i for which $e_i + f_i$ is indexed in J_2 has at least one random value that does not appear in any other observation from $\mathcal{W}_1^r \cup \mathcal{W}_1^a$ or wires e_i for which $e_i + f_i$ is indexed in J_1. So e_i can be assigned to a fresh random value. Thus $e_i + f_i$ is perfectly simulated from e_i and f_i. This produces a perfect simulation of all output wires indexed in J_2.

   Having a perfect simulation of J_1 and J_2, we conclude that we can perfectly simulate J along with $\mathcal{W}$ from the sets I_1 and I_2 with $|I_1| \leq t$ so $|I_1| \leq \min(t, |\mathcal{W}|) = t$ (recall that $|\mathcal{W}| \geq t+1$) and $|I_2| = n$, which is only a failure on one of the inputs b.

From the proof of both properties 1 and 2 for TRPE2, we thus have that G_{add} instantiated with ISW refresh achieves the amplification order $d_2 \geq t+1$ for TRPE2. In addition, we know from the proof of Lemma 1 and as explained in section 4.1 that $d_2 \leq t+1$. Hence $d_2 = t+1$.

We finally proved both amplification orders d_1 and d_2 for TRPE1 and TRPE2 respectively for G_{add} displayed in Algorithm 2 and instantiated with the n-share ISW refresh gadget, which concludes the proof. □

H Proof of Lemma 12

Proof. We start by proving the first property of the lemma. Since the n-share ISW multiplication gadget is $(n-1)$-SNI [5], then we know from Lemma 6 that

$$d_1 \geq \frac{\min(t+1, n-t)}{2}.$$

In addition, we know from the proof of Lemma 2 that

$$d_1 \leq \frac{t+1}{2}.$$

It remains to show that $d_1 \leq (n-t)/2$. In this purpose, we exhibit a simulation failure on both inputs by carefully choosing $n-t$ leaking variables, with t output variables. Consider the set of indexes of output shares $J = \{1, \ldots, t\}$, which corresponds to the first t output shares $c_1, \ldots, c_t$. Next, we construct the set of leaking wires $\mathcal{W}$ of size $n-t$. First, observe that the partial sums of the output shares are of the form

$$c_{i,j} = \begin{cases} a_i \cdot b_i + r_{i,1} + \cdots + r_{i,j} & \text{if } j < i \\ a_i \cdot b_i + r_{i,1} + \cdots + r_{i,i-1} + r_{i,i+1} + \cdots + r_{i,j} & \text{otherwise.} \end{cases}$$

Then, let $\mathcal{W} = \{c_{t+1,t}, \ldots, c_{n,t}\}$. We can prove that the constructed set $\mathcal{W}$ along with the set of output shares $J = \{1, \ldots, t\}$ cannot be perfectly simulated with at most t input shares. For this, we consider a variable $s = c_1 + \cdots + c_t + c_{t+1,t} + \ldots + c_{n,t}$, the sum of the t output shares indexed in J, and the leaking variables from $\mathcal{W}$. Each of the output shares $\{c_i\}_{1 \leq i \leq t}$ is the sum of

- one product of input shares $a_i \cdot b_i$
- $n-1$ random values,
- at most $n-1$ pairs of input shares products: $(a_i \cdot b_j,\ a_j \cdot b_i)$ with $i \neq j$.

Each of the leaking variables $\{c_{i,t}\}_{t+1 \leq i \leq n}$ is the sum of

- one product of input shares $a_i \cdot b_i$,
- t random values,
- at most t pairs of input shares products: $(a_i \cdot b_j,\ a_j \cdot b_i)$ with $i \neq j$.

In addition, each random value appears exactly twice in the set of expressions of the variables $\{c_i\}_{1 \leq i \leq t} \cup \{c_{i,t}\}_{t+1 \leq i \leq n}$, so all the random values are eliminated from the expression of the variable s, which is the sum of all of these variables. Hence, $s = a_1 \cdot b_1 + \ldots + a_n \cdot b_n + C$ where C is a variable containing other products of input shares of the form $a_i \cdot b_j$ and $a_j \cdot b_i$ with $i \neq j$. Thus, simulating the variable s requires the knowledge of the full inputs a and b. Since s is constructed from the set of leaking wires $\mathcal{W}$ and the output shares indexed in J, then $\mathcal{W}$ and J cannot be perfectly simulated without the knowledge of the full inputs a and b. Hence, the set $\mathcal{W}$ of size $n-t$ represents a failure tuple on both inputs, and so the function f_1 for RPE1 cannot be of amplification order higher than $(n-t)/2$. Thus, $d_1 \leq (n-t)/2$.

From the three inequalities $d_1 \geq \dfrac{\min(t+1, n-t)}{2}$, $d_1 \leq (t+1)/2$ and $d_1 \leq (n-t)/2$, we conclude that

$$d_1 = \frac{\min(t+1, n-t)}{2}.$$

Next, we demonstrate the second part of the lemma. Let $\mathcal{W}$ be a set of leaking wires such that $|\mathcal{W}| \leq t$. We aim to prove that there exists a set J of $n-1$ output wires such that $\mathcal{W}$ and J can be perfectly simulated with sets of input shares I_1 on a and I_2 on b such that $|I_1| \leq t$, $|I_2| \leq t$. First, observe that the leaking wires in $\mathcal{W}$ are of the following forms :

1. input shares a_i, b_i, product of shares $a_i \cdot b_i$.
2. partial sum $c_{i,j} = \begin{cases} a_i \cdot b_i + r_{i,1} + \cdots + r_{i,j} & \text{if } j < i \\ a_i \cdot b_i + r_{i,1} + \cdots + r_{i,i-1} + r_{i,i+1} + \cdots + r_{i,j} & \text{otherwise.} \end{cases}$
3. random variable r_{ij} for $i < j$, variable $r_{ji} = a_i \cdot b_j + r_{ij} + a_j \cdot b_i$ for $j > i$.
4. product of shares $a_i \cdot b_j$, or variable $a_i \cdot b_j + r_{ij}$ with $i \neq j$.

We build sets I_1 and I_2 from empty sets as follows. For every wire in $\mathcal{W}$ of the first or second form, we add index i to I_1 and I_2. For every wire in $\mathcal{W}$ of the third or fourth form, if $i \in I_1$, we add j to I_1, otherwise we add i to I_1, and if $i \in I_2$, we add j to I_2, otherwise we add i to I_2. Since $\mathcal{W}$ is of size at most t, then $|I_1| \leq t$ and $|I_2| \leq t$. Following the t-SNI property proof from [5], we can show that $\mathcal{W}$ is perfectly simulated using shares of indexes in I_1 and I_2. We now build the set J of $n-1$ indexes of output shares from two subsets J_1 and J_2. We define $J_1 = \{i \mid c_{i,j} \text{ is observed in } \mathcal{W}\}$. Next, we define J_2 as any set such that $|J_2| = n - 1 - |J_1|$ and $J_1 \cap J_2 = \emptyset$. Now, we show that the output shares determined by the indexes in $J = J_1 \cup J_2$ can be perfectly simulated from I_1 and I_2:

- First consider the output wires indexed in J_1, which have a partial sum observed. For each such variable c_i, the biggest partial sum which is observed is already simulated. For the remaining r_{ij} in c_i, if $i < j$, then r_{ij} is assigned to a fresh random value. Otherwise, if r_{ji} enters in the computation of any other internal observation, then $i, j \in I_1$ and $i, j \in I_2$, and so r_{ji} can be perfectly simulated from the input shares. If not, then r_{ji} is replaced by the random value r_{ij}. So all output wires indexed in J_1 are perfectly simulated from I_1 and I_2.
- Now consider the output wires indexed in J_2. None of the c_i indexed in J_2 has a partial sum observed. Meanwhile, each c_i indexed in J_2 is composed of $n-1$ random values, and at most one of them can enter in the expression of each other output wire c_j. Since by construction of J_1, all the variables observed through the set $\mathcal{W}$ are included in the set of variables observed through J_1, and since $|J_1| \leq |\mathcal{W}| \leq t \leq n-2$ and $|J_2| = n - 1 - |J_1|$, then each output wire c_i indexed in J_2 has at least one random value that does not appear in any other observation from $\mathcal{W}$ or J_1, so c_i can be assigned to a fresh random value. This produces a perfect simulation of all output wires indexed in J_2.

We conclude that the set J of $n-1$ wires is perfectly simulated along with $\mathcal{W}$ from the constructed sets I_1 and I_2 of sizes $|I_1| \leq |\mathcal{W}| \leq t$ and $|I_2| \leq |\mathcal{W}| \leq t$. So there is no failure set of observations of size at most t for RPE2 on any of the inputs. Hence $d_2 \geq (t+1)/2$. In addition, we know from the proof of Lemma 2 and as explained in section 4.1 that $d_2 \leq (t+1)/2$. Hence, $d_2 = (t+1)/2$, which concludes the proof for RPE2. □

I Proof of Lemma 13

Recall the procedure of the gadget. We consider that we have an n-share (t, f')-TRPE refresh gadget G_{refresh} achieving the amplification order $d \geq \min(t+1, n-t)$. First, the gadget G_{mult} performs n executions of the gadget G_{refresh} on the input sharing $(b_1, \ldots, b_n)$ to produce:

$$\begin{aligned}(b_1^{(1)}, \ldots, b_n^{(1)}) &\leftarrow G_{\text{refresh}}(b_1, \ldots, b_n)\\ &\cdots\\ (b_1^{(n)}, \ldots, b_n^{(n)}) &\leftarrow G_{\text{refresh}}(b_1, \ldots, b_n)\end{aligned}$$

then, the gadget constructs the matrix of the cross product of input shares using the refreshed input shares of b:

$$M = \begin{pmatrix} a_1 \cdot b_1^{(1)} & a_1 \cdot b_2^{(1)} & \cdots & a_1 \cdot b_n^{(1)} \\ a_2 \cdot b_1^{(2)} & a_2 \cdot b_2^{(2)} & \cdots & a_2 \cdot b_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ a_n \cdot b_1^{(n)} & a_n \cdot b_2^{(n)} & \cdots & a_n \cdot b_n^{(n)} \end{pmatrix}.$$

Then, it picks n^2 random values which define the following matrix:

$$R = \begin{pmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,n} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n,1} & r_{n,2} & \cdots & r_{n,n} \end{pmatrix}.$$

It then performs an element-wise addition between the matrices M and R:

$$P = M + R = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n,1} & p_{n,2} & \cdots & p_{n,n} \end{pmatrix}.$$

At this point, the gadget randomized each product of input shares from the matrix M with a single random value from R. In order to generate the correct output, the gadget adds all the columns of P into a single column V of n elements, and adds all the columns of the transpose matrix R^{T} into a single column X of n elements:

$$V = \begin{pmatrix} p_{1,1} + \cdots + p_{1,n} \\ p_{2,1} + \cdots + p_{2,n} \\ \vdots \\ p_{n,1} + \cdots + p_{n,n} \end{pmatrix}, \qquad X = \begin{pmatrix} r_{1,1} + \cdots + r_{n,1} \\ r_{1,2} + \cdots + r_{n,2} \\ \vdots \\ r_{1,n} + \cdots + r_{n,n} \end{pmatrix}$$

The n-share output is finally defined as $(c_1, \ldots, c_n)^T = V + X$ such that

$$\begin{aligned} c_1 &= V_1 + X_1 \\ &\cdots \\ c_n &= V_n + X_n. \end{aligned}$$

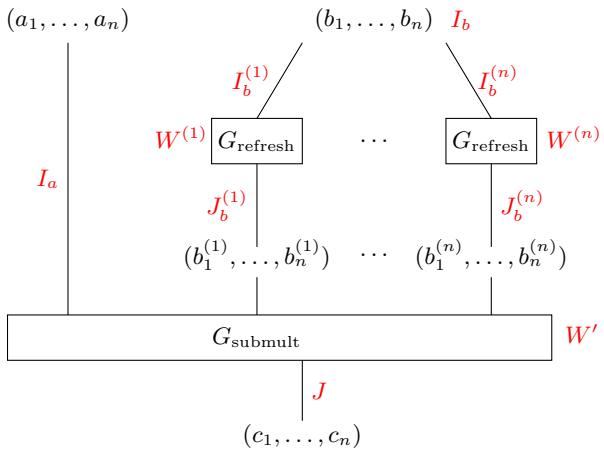

Fig. 3: G_{mult} gadget from Section 5.5.

Figure 3 represents the G_{mult} gadget from a high-level, composed of several blocks. First, a refresh gadget G_{refresh} is executed n independent times on the input sharing of b to produce n fresh copies $b^{(1)}, \ldots, b^{(n)}$. Then, the gadget G_{submult} takes as input $(a_1, \ldots, a_n)$ and the outputs of the refreshing gadgets $b^{(1)}, \ldots, b^{(n)}$ to produce the output of G_{mult}.

In the following proofs, we will denote W to be any set of probes on the global gadget G_{mult}, then W can be split as $W = W' \cup W^{(1)} \cup \ldots \cup W^{(n)}$ where $W^{(i)}$ is the set of probes on the internal wires of the execution of G_{refresh} for the fresh sharing $b^{(i)}$ of b, and W' is the set of probes on the internal wires of G_{submult}. We will also denote J to be any set of output wires of G_{mult} (which are the output wires of G_{submult}), and $J_b^{(i)}$ (resp. $I_b^{(i)}$) any set of output wires (resp. input wires) of the execution of G_{refresh} for the fresh sharing $b^{(i)}$ of b. Observe that any probe on the output wires of G_{refresh} for any sharing $b^{(i)}$ can be obtained through internal probes in W' on G_{submult}, so in the beginning we always consider that $J_b^{(i)} = \emptyset$ for all $i \in [n]$.

Observe that any probe in the set W' on the internal wires of G_{submult} is of one of the following forms:

(a) $a_i,\ b_j^{(i)},\ a_i \cdot b_j^{(i)},\ r_{i,j},\ p_{i,j} = a_i \cdot b_j^{(i)} + r_{i,j}$,
(b) $V_{i,j}$ partial sum of the first j terms of V_i. Observe that $V_{i,n} = V_i$,
(c) $X_{i,j}$ partial sum of the first j terms of X_i. Observe that $X_{i,n} = X_i$.

Also observe that each random value $r_{i,j}$ only appears in the expression of the wires $r_{i,j}$, $p_{i,j}$, $V_{i,j}$, or $X_{j,i}$ (so also $c_i = V_i + X_i$ and $c_j = V_j + X_j$), and does not appear anywhere else in the wires.

We will first start by proving some simple claim.

Claim 1 *Let J be a set of output shares of G_{mult} and $W = W' \cup W^{(1)} \cup \ldots \cup W^{(n)}$ be a set of leaking wires as described above such that $|J| + |W'| \leq n - 1$ (we only consider the set W' of probes on the internal wires to $G_{submult}$). Then, for any $i \in J$ such that $V_{i,j} \notin W$ for any $j \in [n]$, the output wire c_i can be perfectly simulated by generating a uniform random value without knowing any of the input shares.*

Proof. Let $i \in J$ such that $V_{i,j} \notin W'$ for any $j \in [n]$. Then we know that the expression of V_i in $c_i = V_i + X_i$ contains $n - 1$ random values since $V_i = p_{i,1} + \ldots + p_{i,n}$ and each $p_{i,j} = a_i \cdot b_j^{(i)} + r_{i,j}$

(without counting the random $r_{i,i}$ because it is cancelled out in c_i as it appears in V_i and X_i and $c_i = V_i + X_i$). Observe that each random value $r_{i,k}$ in V_i appears in exactly one other output share $c_k = V_k + X_k$ that comes from the expression of $X_k = r_{1,k} + \ldots + r_{i,k} + \ldots + r_{n,k}$. In other terms, each output share c_k has exactly one random value in common with V_i in c_i. Then, by probing $|J|$ output shares in J including c_i, there are at least $n - |J|$ remaining random values in V_i that do not appear in any other expression of the output shares. In addition, observe that any probed variable in W' can have in its expression at most one random value in common with V_i (because each random value $r_{i,j}$ appears exactly once in each of the wires $p_{i,j}$, $r_{i,j}$ or X_j). Then, since $|W'| \leq n - |J| - 1$ (because $|J| + |W'| \leq n - 1$), there is at least $n - |J| - (n - |J| - 1) = 1$ remaining random value $r_{i,\ell}$ where $\ell \in [n]$ in V_i, that does not appear in any other expression of the probed values in W' or J. So $c_i = V_i + X_i$ can be perfectly simulated by generating the uniform random value $r_{i,\ell}$, which concludes the proof.

In the following, we will separately prove the TRPE1 then the TRPE2 property on G_{mult} via Lemmas 14 and 17 to demonstrate Lemma 13.

I.1 Proof for TRPE1 property

Lemma 14. *The multiplication gadget G_{mult} is (t, f_1)-TRPE1 of amplification order $d = \min(t + 1, n - t)$*

Proof. We proceed in two steps through the following two lemmas 15 and 16, considering the leaking wires in two distinct ranges.

Lemma 15. *Let J be a set of at most t output shares of G_{mult}. Let W be a set of leaking wires as described above such that $|W| \leq d - 1 \leq t$. Then W and J can be perfectly simulated from at most $\min(t, |W|) = |W|$ shares of each of the inputs a and b.*

Proof. Let J be the set of t output shares of G_{mult} (i.e of G_{submult}), and let $W = W' \cup W^{(1)} \cup \ldots \cup W^{(n)}$ with $|W| \leq d - 1 \leq t$ be the set of probes on the global gadget G_{mult} and decomposed as explained earlier. We organize the proof in two steps:

1. We first identify the set of input shares I_a and the sets $J_b^{(i)}$ for $i \in [n]$ which are necessary to perfectly simulate J and W' in G_{submult}.
2. Then, we show that we can perfectly simulate the sets $J_b^{(i)}$ and $W^{(i)}$ for $i \in [n]$ using the simulator of the gadget G_{refresh}. This will determine the sets $I_b^{(i)}$ necessary for each of the n simulations of G_{refresh}, and thus determine the set I_b of input shares on b as $I_b = I_b^{(1)} \cup \ldots \cup I_b^{(n)}$.

Using I_b, we will be able to perfectly simulate $J_b^{(i)}$ and $W^{(i)}$ for $i \in [n]$. Then using I_a and $J_b^{(i)}$ for $i \in [n]$, we will be able to perfectly simulate W' and J. This will lead to a perfect simulation of all probes W and output shares in J on the global gadget G_{mult}.

We first start by constructing the set of input shares indices I_a and the sets $J_b^{(k)}$ for $k \in [n]$ depending on the probes in the set W' as follows[3]:

(a) For probes of form (a), we add index i to I_a, and index j to $J_b^{(k)}$ for $k \in [n]$.

[3] We consider that all $J_b^{(k)}$ are empty at first since all the output shares of G_{refresh} can be probed directly in W'.

(b) For probes of form (b), we add index i to I_a and to $J_b^{(k)}$ for $k \in [n]$.
(c) For probes of form (c), we add index i to $J_b^{(k)}$ for $k \in [n]$.

Observe that since $|W| \leq d-1$, then in particular $|W'| \leq d-1 \leq t$, then $|I_a| \leq |W'| \leq |W| \leq \min(t, |W|)$ so we have no failure on the input a. We also have $|J_b^{(k)}| \leq |W'| \leq t$.

Simulation of W': probes of the form (a) can be perfectly simulated from the corresponding input shares in I_a and $J_b^{(k)}$, and by generating uniformly random values $r_{i,j}$ when necessary. Probes of the form (c) are also perfectly simulated by simply generating uniformly random values, since $X_{i,j} = r_{1,i} + \ldots + r_{j,i}$. As for probes of the form (b), we know that $i \in I_a$ and $i \in J_b^{(i)}$, then we look at each of the terms $p_{i,j'}$ for $j' \in [j]$ in $V_{i,j} = p_{i,1} + \ldots p_{i,j}$. In particular, if $j \geq i$, the term $p_{i,i}$ is in the partial sum $V_{i,j}$ and is perfectly simulated using the input shares a_i and $b_i^{(i)}$ and by generating the random value $r_{i,i}$. Next, for each $p_{i,j'}$ such that $j' \neq i$, if $j' \in J_b^{(i)}$, then $p_{i,j'}$ can be perfectly simulated from the corresponding input shares and by generating uniformly at random $r_{i,j'}$. Otherwise, if $j' \notin J_b^{(i)}$, then that means that the wires $p_{i,j'}, r_{i,j'}$ and $X_{j'}$ are not probed in W' because otherwise j' would have been added to all $J_b^{(k)}$ for $k \in [n]$. Since the random value $r_{i,j'}$ only appears in the expression of the wires $p_{i,j'}, r_{i,j'}$ and $X_{j'}$ (besides $V_{i,j}$ which is already probed), and of the output wire $c_{j'} = V_{j'} + X_{j'}$, we need to consider two cases:

- $j' \notin J$: in this case, the random value $r_{i,j'}$ can be used to mask the expression of $p_{i,j'}$ in the partial sum $V_{i,j}$, perfectly simulating it without the need to the share $b_{j'}^{(i)}$.
- $j' \in J$: $c_{j'} = V_{j'} + X_{j'}$, and $r_{i,j'}$ is the one of the summed terms in the expression of $X_{j'}$. We know that $V_{j',k} \notin W'$ for any $k \in [n]$ since otherwise j' would have been added to $J_b^{(i)}$. Since in addition we have $|J| + |W| \leq t + d - 1 \leq t + n - t - 1 \leq n - 1$, by claim 1, the output share c'_j can be masked by some random value $r_{j',\ell}$. Thus, $X_{j'}$ is masked and $r_{i,j'}$ does not appear anymore in $c_{j'}$. So $r_{i,j'}$ can be used to mask the expression of $p_{i,j'}$ in the partial sum $V_{i,j}$. This brings us to a perfect simulation of $p_{i,j'}$ simply by generating at random $r_{i,j'}$.

By perfectly simulating each of the terms $p_{i,j'}$ for $j' \in [j]$ in the probed wire $V_{i,j}$ independently, we can perfectly simulate their sum and thus perfectly simulate $V_{i,j}$. This brings us to a perfect simulation of the set W'.

Simulation of J: Let $i \in J$.

- if $V_{i,j} \notin W'$ for any $j \in [n]$, then by claim 1, c_i is perfectly simulated by simply generating a uniform random value $r_{i,\ell}$ for some $\ell \in [n]$.
- if $V_{i,j} \in W'$ for at least one $j \in [n]$, then let $V_{i,j'}$ be the largest of the probed partial sums. All of the partial sums including $V_{i,j'}$ are perfectly simulated as described earlier. Then, let us consider $c_i + V_{i,j'} = p_{i,j'+1} + \ldots + p_{i,n} + X_i$. The wire X_i can be perfectly simulated by generating uniform random values. As for each of the terms $p_{i,j'+1}, \ldots, p_{i,n}$, they can each be perfectly simulated in the exact same way each of the terms in $V_{i,j'}$ are simulated independently.

  In the particular case where $j' \leq i$ then the term $p_{i,i} = a_i \cdot b_i^{(i)} + r_{i,i}$ appears in the expression of $c_i + V_{i,j'}$, and in this case, the random value $r_{i,i}$ is cancelled out in the expression of $c_i + V_{i,j'}$ since it appears in both $p_{i,i}$ and X_i, and $c_i + V_{i,j'} = p_{i,j'+1} + \ldots + p_{i,i} + \ldots + p_{i,n} + X_i$. So to simulate the term $p_{i,i}$ in $c_i + V_{i,j'}$ we need both input shares a_i and $b_i^{(i)}$. This is already the case by construction because we assume that $V_{i,j'} \in W'$.
  Thus, by perfectly simulating $V_{i,j'}$ and $c_i + V_{i,j'}$, the output share c_i is also perfectly simulated.

Also, since $|J_b^{(k)}| \leq |W'| \leq t$ and $|W^{(k)}| \leq d-1$, and since G_{refresh} is (t, f')-TRPE achieving the amplification order d, then we can perfectly simulate sets $J_b^{(k)}$ and $W^{(k)}$ from the set of input shares $I_b^{(k)}$ such that $|I_b^{(k)}| \leq |W^{(k)}| \leq t$ for $k \in [n]$. Thus, we can let $I_b = I_b^{(1)} \cup \ldots \cup I_b^{(n)}$ and we have $|I_b| \leq |W^{(1)}| + \ldots + |W^{(n)}| \leq |W| \leq \min(|W|, t)$, so we have no failure on the input b either. Until now, we have shown that we can simulate all sets $W^{(k)}$ and $J_b^{(k)}$ from I_b of size at most $\min(|W|, t)$. It remains to show that we can also perfectly simulate the sets W' and J from I_a and $J_b^{(k)}$ for $k \in [n]$.

We have shown that we can perfectly simulate any set of t output shares J and any set of probes W of size at most $d-1$, with at most $\min(|W|, t)$ shares of each of the inputs a and b. This concludes the proof of Lemma 15. □

Remark 2. We can observe that for this lemma to apply on G_{mult}, we do not need the pre-processing phase of the refresh on input b. In fact, we can see that during the construction of the sets $J_b^{(k)}$, we add each index to all of the sets for all $k \in [n]$. However, executing n refreshings on the input b is necessary to prove the next result, specifically when we consider W such that $d \leq |W| \leq 2d-1$.

To get back to the proof of Lemma 14, we also need the following result.

Lemma 16. *Let J be a set of at most t output shares of G_{mult}. Let W be a set of leaking wires as described above such that $d \leq |W| \leq 2d-1$. Then W and J can be perfectly simulated from the sets of input shares I_a and I_b such that $|I_a| \leq \min(|W|, t)$ **or** $|I_b| \leq \min(|W|, t)$. In other terms, we have a simulation failure on at most one of the inputs a or b.*

Proof. Recall that the set W can be split into subsets $W = W' \cup W^{(1)} \cup \ldots \cup W^{(n)}$ as described above. We can consider two cases.

Case 1: $|\mathbf{W'}| \leq \mathbf{d-1}$. This case is similar to the case of Lemma 15, so we can construct the set I_a in the same way as in the proof of Lemma 15, and we can eventually consider $I_b = [n]$. We know that $|I_a| \leq |W'| \leq d-1 \leq t$, so there is no failure on the input a. And all probes in W' can be simulated like in the proof of Lemma 15 with I_a and trivially with $I_b = [n]$. Also, all probes in $W^{(1)} \cup \ldots \cup W^{(n)}$ can be trivially simulated since we have access to the full input b. As for output shares in J, whenever $i \in J \cap I_a$, then $c_i = V_i + X_i$ is easily simulated using $I_b = [n]$. If $i \in J$ but $i \notin I_a$, then $V_{i,j} \notin W'$ for any $j \in [n]$ and since $|J| + |W'| \leq t + d - 1 \leq t + n - t - 1 \leq n - 1$, c_i is perfectly simulated by a single random value thanks to claim 1. Thus, W and J are perfectly simulated with at most $|W'| \leq \min(|W|, t)$ shares of a and eventually n shares of b.

Case 2: $|\mathbf{W'}| \geq \mathbf{d}$ *(and thus* $|W^{(1)} \cup \ldots \cup W^{(n)}| \leq d-1$*).* In this case, we will construct the sets I_a and $J_b^{(k)}$ from empty sets, in a way that we will have a simulation failure on at most one of the inputs a or b, and we will be able to perfectly simulate W' and output shares in J using I_a and $J_b^{(k)}$. We will also show how to perfectly simulate all $J_b^{(k)}$ and $W^{(k)}$ using a set of input shares I_b.

First, we construct the sets I_a and $J_b^{(k)}$ depending on the probes in W' as follows:

(a) For probes of form (a), we add index i to I_a, and index j only to $J_b^{(i)}$.
(b) For probes of form (b), we add index i to I_a and only to $J_b^{(i)}$.
(c) For probes of form (c), we add index i to $J_b^{(k)}$ for all $k \in [n]$.

In the rest of the proof, we will show that if we have a failure on one of the inputs, we can still perfectly simulate W and J without a failure on the other input. In this purpose, we will consider two cases: in the first case (2.1), we will have a failure on input a (*i.e.*, more than $\min(t, |W|)$ shares of a are added to I_a) and in the second case (2.2), we won't have a failure on input a, and so we will eventually have a failure on input b.

Case 2.1: Simulation failure on input a. Notice that by construction we always have $|I_a| \leq |W'| \leq |W|$. Thus, a simulation failure on input a for TRPE1 means that the set I_a is of size $|I_a| \geq t+1 \geq d$. We will first start by showing that the sets $W^{(k)}$ and $J_b^{(k)}$ can be perfectly simulated using the simulator of G_{refresh} without a failure on the input b. Next, we will show that W' and output shares in J can be perfectly simulated using I_a and $J_b^{(k)}$.

Since we only add shares indices to I_a when we have probes of the form (a) or (b), this means that we have at least $t+1$ probes of these two forms with $t+1$ different values for the index i. In addition, since we have at least $t+1$ probes (a) or (b) with distinct values for the index i, then this also means that each of the sets $J_b^{(i)}$ built from these probes has at most one share of $b^{(i)}$ added to it by construction. In other terms, when we only consider probes of the form (a) and (b) with distinct i, we have $|J_b^{(k)}| \leq 1$ for each $k \in [n]$.

Now let us consider the remaining probes in W which are either in W' of the form (c), in W' of the form $(a)/(b)$ for which $i \in I_a$ or in $W^{(1)} \cup \ldots \cup W^{(n)}$. Since $|I_a| \geq t+1 \geq d$, then there are at most $d-1$ of these remaining probes. Without loss of generality, we consider that there are exactly $d-1$ instead of at most $d-1$ probes. Let m be the number of probes in $W^{(1)} \cup \ldots \cup W^{(n)}$ and $d-1-m$ the remaining in W' of the form (c) or of the form $(a)/(b)$ for which $i \in I_a$.

Since each wire in W' of the form (c) or of the form $(a)/(b)$ for which $i \in I_a$ results in adding at most one more share index to each $J_b^{(k)}$ for $k \in [n]$, then we have $|J_b^{(k)}| \leq 1 + (d-1-m) = d-m$. And $|W^{(1)} \cup \ldots \cup W^{(n)}| \leq m$, in particular $|W^{(k)}| \leq m$ for any $k \in [n]$.

- if $m = 0$, then $W^{(k)} = \emptyset$ for any $k \in [n]$, and $|J_b^{(k)}| \leq d \leq \min(t+1, n-t) \leq \lfloor \frac{n+1}{2} \rfloor \leq n-1$, so by the TRPE property of G_{refresh} for any $t \leq n-1$, all of the $J_b^{(k)}$ sets can be perfectly simulated with no knowledge of the input shares of b since $W^{(k)} = \emptyset$, so $I_b^{(k)} = \emptyset$. Hence, $I_b = I_b^{(1)} \cup \ldots \cup I_b^{(n)} = \emptyset$ and we have no simulation failure on the input b.
- if $m > 0$, then $|J_b^{(k)}| \leq d-1 \leq t$ and $|W^{(1)} \cup \ldots \cup W^{(n)}| \leq d-1$, in particular $|W^{(k)}| \leq d-1$ for each $k \in [n]$. Thus, by the (t, f)-TRPE property of the refresh gadget G_{refresh} achieving the amplification order d for any $t \leq n-1$, we can perfectly simulate both sets for each $k \in [n]$ with $I_b^{(k)}$ such that $|I_b^{(k)}| \leq |W^{(k)}|$. Thus, we can let $I_b = I_b^{(1)} \cup \ldots \cup I_b^{(n)}$ so we can have $|I_b| \leq |W^{(1)} \cup \ldots \cup W^{(n)}| \leq d-1 \leq t$, and we can perfectly simulate $W^{(1)} \cup \ldots \cup W^{(n)}$ along with $J_b^{(1)} \cup \ldots \cup J_b^{(n)}$ from the set I_b without a simulation failure on input b.

So far we proved that if we have $|I_a| \geq t+1$, then we must have $|I_b| \leq |W^{(1)} \cup \ldots \cup W^{(n)}| \leq d-1 \leq t$, and $W^{(1)} \cup \ldots \cup W^{(n)}$ can be perfectly simulated along with $J_b^{(1)} \cup \ldots \cup J_b^{(n)}$ from the set I_b. Next we need to prove that we can perfectly simulate W' and J from these sets I_a and $J_b^{(1)} \cup \ldots \cup J_b^{(n)}$.

Case 2.1.1: $I_a = [n]$. This only occurs by construction in the case where $|W| = 2d-1 = n$ so when $d = d_{\max} = \lfloor \frac{n+1}{2} \rfloor$ for $t = \lceil \frac{n-1}{2} \rceil$. In this case, since $|W| \leq 2d-1 \leq (n+1)-1 \leq n$, then all probes in W are all in W' of the form (a) or (b) with n distinct values for the index i and so

$|J_b^{(i)}| \leq 1$ for all $i \in [n]$. In other words, for each $i \in [n]$ there is exactly one probe in W' of the form (a) or (b) and no probe of the form (c) i.e $X_{i,j}$ nor probes in $W^{(1)} \cup \ldots \cup W^{(n)}$. We will prove that all the probes in W and in J can be perfectly simulated from these constructed sets I_a and $J_b^{(i)}$ for $i \in [n]$. For this, for each $i \in [n]$ we consider three cases:

- $V_{i,j} \notin W'$ for any $j \in [n]$, then we know that there exists a probe of the form (a) in W' with index i, in other terms, $a_i \in W'$, or $\exists!\ j \in [n]$ such that $b_j^{(i)}$ or $a_i \cdot b_j^{(i)}$ or $r_{i,j}$ or $p_{i,j} = a_i \cdot b_j^{(i)} + r_{i,j}$ is probed in W'. The corresponding probe is perfectly simulated by construction of the sets I_a and $J_b^{(i)}$.
  If we also have $i \in J$, then we know that we only have one probe of the form (a) for the considered index i in W' and no probe of the form (b) or any probe of the form (c). And since there are t output shares probed in J, then there are at least $n - t - 1 > 1$ (since $t = \lceil \frac{n-1}{2} \rceil$) remaining random values which only appear in the expression of c_i, and any of them can be used to perfectly simulate c_i without the knowledge of the input shares (*i.e.*, to mask c_i).
- $V_{i,n} \in W'$ then $V_{i,n}$ contains in its expression n random values $r_{i,1}, \ldots, r_{i,n}$. Since there are no probes of the form (a) for the index i, and no probes of the form (c), then each of these random values appears at most once in each of the expressions of the probed outputs c_j in J. With t probed output shares, there are $n - t > 1$ (since $t = \lceil \frac{n-1}{2} \rceil$) remaining random values which only appear in the expression of $V_{i,n}$ and any of them can be used to perfectly simulate $V_{i,n}$, *i.e.*, mask $V_{i,n}$.
  If in addition we have $i \in J$, then the output share c_i is perfectly simulated by simulating $V_{i,n}$ and simulating $c_i + V_{i,n} = X_i$ which is perfectly simulated by generating uniform random values.
- $V_{i,j} \in W'$ for some $j \in [n]$ such that $1 < j < n$ ($j > 1$ because otherwise it would be the wire $p_{i,1}$ which is probed). Thus, $V_{i,j}$ is the sum of at least two wires p_{i,j_1} and p_{i,j_2}.
  - If $i \notin J$, then c_i is not probed and $V_{i,j}$ is the sum of at most $n - 1$ terms of the form $p_{i,1} = a_i \cdot b_1^{(i)} + r_{i,1}, \ldots p_{i,j} = a_i \cdot b_j^{(i)} + r_{i,j}$. We have that $i \in I_a$ by construction and $j \in J_b^{(i)}$. In fact we can reconstruct $J_b^{(i)}$ into $J_b^{(i)} = \{1, \ldots, j\}$ such that $|J_b^{(i)}| \leq n - 1$ and since $W^{(i)} = \emptyset$, then by the (t, f)-TRPE1 property of G_{refresh} for any $t \leq n - 1$, we still have no failure on the input b and we still have $|I_b^{(i)}| \leq |W^{(i)}| = 0$. In addition, we can perfectly simulate this way all of the summed terms in $V_{i,j}$ by using the corresponding input shares and thus we can perfectly simulate $V_{i,j}$. Since we have no probes of the form (a) for this same index i, then reconstructing $J_b^{(i)}$ does not affect the simulation of the probes.
  - If $i \in J$, then we consider $V_{i,j}$ and $c_i + V_{i,j}$. Since we have no probes of the form (a) for the index i, then as proven before, with t probed output shares, there are at least $n - t > 1$ remaining random values which only appear in the expression of $V_{i,j}$ or $c_i + V_{i,j}$. Any of these random values can be used to mask the expression of $V_{i,j}$ or $c_i + V_{i,j}$. In the case where the expression of $V_{i,j}$ is masked, then we can reconstruct as before the set $J_b^{(i)}$ with at most $n - 1$ output shares of $b^{(i)}$ in order to perfectly simulate all the terms $p_{i,k}$ in $c_i + V_{i,j}$ including the shares of $b^{(i)}$ and thus perfectly simulate $c_i + V_{i,j}$ (the rest of the terms are just random values to be generated uniformly at random). In the other case where the expression of $c_i + V_{i,j}$ is masked, we can also reconstruct the set $J_b^{(i)}$ with at most $n - 1$ output shares of $b^{(i)}$ in order to perfectly simulate all the summed terms in $V_{i,j}$. In either case, by perfectly simulating one term ($V_{i,j}$ or $c_i + V_{i,j}$) masked by a random value, and perfectly simulating

the remaining one with $i \in I_a$ and the reconstructed set $J_b^{(i)}$, we can perfectly simulate both $V_{i,j}$ and $c_i + V_{i,j}$ and hence also perfectly simulate the output share c_i.

So we proved that we can perfectly simulate the sets W' and J from the constructed set I_a and from sets $J_b^{(i)}$ such that $|J_b^{(i)}| \leq n-1$ for all $i \in [n]$. Furthermore, from the TRPE property of G_{refresh} for any $t \leq n-1$ and the fact that $W^{(i)} = \emptyset$ for all $i \in [n]$, we have no simulation failure on the input b. This concludes the simulation of W and output shares in J for the case where $I_a = [n]$.

Case 2.1.2: $I_a \subset [n]$ *with* $|I_a| \leq n-1$. In this case, we have at least one index $k \in [n] \setminus I_a$ for which there are no probes in W' of the form (a) or (b). In other terms, no partial sum of V_k is probed, no product of shares $a_k \cdot b_j^{(k)}$ or $p_{k,j}$ is probed, and no random value $r_{k,j}$ is probed since otherwise we would have $k \in I_a$ by construction.

On another hand, since $|I_a| \geq t+1$, there are at most $d-1 \leq n-t-1$ remaining probes of the form (c) in W', and since we have t output shares in the set J, there exists at least one wire X_ℓ such that $\ell \notin J$ and for which there is no partial sum $X_{\ell,j}$ probed in W'.

These two wires X_ℓ and V_k for $\ell, k \in [n]$ will be very important for the simulation of the sets W' and J. In particular, we need the two following claims.

Claim 2 *Let $i \in J$. Suppose that $i \notin I_a$. Then the expression of $c_i = V_i + X_i$ can be masked by the random value $r_{i,\ell}$, in other terms $c_i \leftarrow r_{i,\ell}$.*

Proof. This claim can be proved easily, since we suppose that $i \notin I_a$ so the random value $r_{i,\ell}$ and $p_{i,\ell}$ are not probed in W'. In addition, since $\ell \notin J$ and $X_{\ell,j} \notin W'$ for all $j \in [n]$, then the random value $r_{i,\ell}$ does not appear in any other probed wire expression except in c_i, then c_i can be masked by the random value $r_{i,\ell}$. □

Claim 3 *Let $i \in J$. Suppose that $X_{i,j} \notin W'$ for any $j \in [n]$. Suppose that $i \in I_a$. Then the expression of $c_i = V_i + X_i$ can be masked by the random value $r_{k,i}$, in other terms $c_i \leftarrow r_{k,i}$.*

Proof. Since we suppose that $k \notin I_a$, then the random value $r_{k,i}$ or $p_{k,i}$ or $V_{k,j}$ for all $j \in [n]$ are not probed in W'. Then, if $k \notin J$, then the random value $r_{k,i}$ does not appear in the expression of any other probed wire in W' or J and c_i can be masked by the random value $r_{k,i}$. Otherwise, if $k \in J$, then by Claim 2, $c_k = V_k + X_k$ can be masked by $r_{k,\ell}$ and so c_i can also be masked by $r_{k,i}$ since $i \neq \ell$ (because $i \in J$ and $\ell \notin J$). □

From these two claims, we are now ready to show that W' and J can be perfectly simulated with the sets I_a and $J_b^{(1)} \cup \ldots \cup J_b^{(n)}$ as constructed earlier with respect to the probes in the set W'. Recall that all probes in $W^{(1)} \cup \ldots \cup W^{(n)}$ and $J_b^{(1)} \cup \ldots \cup J_b(n)$ are perfectly simulated using I_b and the simulator of G_{refresh}.

Simulation of W'. Probes of the form (a) and (c) are trivially simulated by construction of the sets of input shares and by generating uniformly at random the necessary random values. Let us now check the probes of the form (b). Let $V_{i,j} = p_{i,1} + \ldots + p_{i,j}$ be such a probe. Let us consider each of the terms $p_{i,j'}$ for $j' \in [j]$. if $j' = i$, then by construction $p_{i,i}$ is perfectly simulated using a_i and $b_i^{(i)}$ and by generating the random value $r_{i,i}$ if needed. Otherwise, let $j' \neq i$. If $j' \in J_b^{(i)}$ then the simulation of $p_{i,j'}$ is straightforward. Otherwise if $j' \notin J_b^{(i)}$, then we know that none of the wires

$r_{i,j'}$ or $p_{i,j'}$ or $X_{j',s}$ for all $s \in [n]$ are probed in W'. Thus, $r_{i,j'}$ can be eventually used to mask the expression of $p_{i,j'}$ without the need of the share $b^{(i)}_{j'}$ for the simulation. Meanwhile, we still need to check if $j' \in J$, since $X_{j'}$ appears in the expression of $c_{j'} = V_{j'} + X_{j'}$. Then we consider two cases:

- If $j' \notin I_a$, then by claim 2, $c_{j'}$ can be masked by the random value $r_{j',\ell}$ and so $r_{i,j'}$ does not appear in the expression of $X_{j'}$ in $c_{j'}$ anymore, and $r_{i,j'}$ can be used to mask $p_{i,j'}$.
- Otherwise, if $j' \in I_a$, then by claim 3, $c_{j'}$ can be masked by the random value $r_{k,j'}$ and so $r_{i,j'}$ does not appear in the expression of $X_{j'}$ in $c_{j'}$ anymore, and $r_{i,j'}$ can be used to mask $p_{i,j'}$ (since $i \notin k$).

Thus, each term $p_{i,j'}$ in $V_{i,j}$ can be perfectly simulated and thus $V_{i,j} = p_{i,1} + \ldots + p_{i,j}$ can be perfectly simulated. This concludes the simulation of the set W'.

Simulation of J. Let $i \in J$. If $i \notin I_a$, then by claim 2, c_i is perfectly simulated by generating the random value $r_{i,\ell}$. Otherwise, let $i \in I_a$. If $X_{i,j} \notin W$ for any $j \in [n]$, then by claim 3, c_i is perfectly simulated by generating the random value $r_{k,i}$. Otherwise, we can show that we can perfectly simulate each term in $c_i = V_i + X_i$. In particular, each term in X_i can be simulated by generating the underlying random value uniformly. For each term in the sum V_i, we know in particular that $a_i \cdot b^{(i)}_i$ is perfectly simulated since $X_{i,j} \in W'$ for at least one $j \in [n]$ so $i \in J^{(i)}_b$ by construction. For the other terms in V_i, they can be perfectly simulated in the exact same way as we simulated the probes $V_{i,j}$ of the form (b) in the set W'. So c_i is perfectly simulated by summing all the perfectly simulated terms. This concludes the simulation proof for the set J.

Up until now, we have concluded that if we have a constructed set I_a of size at least $t+1$, then we can perfectly simulate the sets W and J without having a simulation failure on the input b. In the rest of the proof, we will consider that $|I_a| \leq t$ (along with $|I_a| \leq |W|$ by construction meaning that we have no failure on input a), and we will prove that we can perfectly simulate W and J with at most a simulation failure on b. Recall that we are also considering that $|W'| \geq d$ and $|W^{(1)} \cup \ldots \cup W^{(n)}| \leq d-1$.

Case 2.2: $|I_a| \leq t$. This means that the number of probes of the form (a) or (b) in W' with distinct values for the index i is at most t.

First, let us consider that $|I_a| \geq d$ (this is the case where $d = n - t \leq t+1$). Then, as proved earlier, and with t additional output shares in J of the form $c_i = V_i + X_i$, there are at least one X_ℓ remaining such that $\ell \notin J$ and $X_{\ell,j} \notin W$ for all $j \in [n]$. In this case, we can set $I_b = [n]$ and I_a as constructed with respect to the probes in W'. It is clear that all probes in $W^{(1)} \cup \ldots \cup W^{(n)}$ and $J^{(1)}_b \cup \ldots \cup J^{(n)}_b$ are trivially simulated using $I_b = [n]$. In addition, all probes in W' are also perfectly simulated by construction of the set I_a and using $I_b = [n]$ and generating the necessary random values. This means that we can perfectly simulate all of the set of probes $W = W' \cup W^{(1)} \cup \ldots \cup W^{(n)}$. As for the set of output shares indexed in J. Let $i \in J$. If $i \in I_a$, then c_i is perfectly simulated using the share a_i and $I_b = [n]$, and by generating the necessary random values. Otherwise, if $i \notin I_a$, then in the same way as in claim 2, c_i can be masked by the random value $r_{i,\ell}$ (because $\ell \notin J$ and $X_{\ell,j} \notin W$ for all $j \in [n]$), so a_i is not needed for the simulation of c_i. This proves that we can perfectly simulate the output shares in J with I_a and $I_b = [n]$.

In the rest, we suppose that $|I_a| \leq d-1 \leq n-t-1$, *i.e.*, the number of probes of the form (a) or (b) in W' with distinct values for the index i is at most $d-1 \leq n-t-1$. In this case, and with t

additional output shares, we have at least one index k such that $k \notin J$ and for which there are no probes in W' of the form (a) or (b). In other terms, no partial sum of V_k is probed, no product of shares $a_k \cdot b_j^{(k)}$ or $p_{k,j}$ is probed, and no random value $r_{k,j}$ is probed. Now we reason on the number of probes of the form (c) in W':

- We first consider the special case where the number of $X_{i,j}$ probed (of form (c) in W') for distinct values of i is equal to n. In other terms, we have probes $X_{1,j_1}, \ldots, X_{n,j_n}$ for certain values $j_1, \ldots, j_n$. Since the set of probes W satisfies $|W| \leq n$ (because $2d-1 \leq n$), then this means that there are no remaining probes in the set W except for the n probes of the form (c) in W'. This is an easy case since we can let $I_b = [n]$ and $I_a = J$ (always without a failure on a since in the case where $|W| = n$, we have $d = t+1 = n-t$ so $|I_a| = |J| \leq \min(t, |W|)$ where $t \leq |W|$). This allows us to trivially simulate all output wires indexed in J, and since the remaining wires in W are just sums of random values, we can simulate them by generating the corresponding random values.
- Next, we consider that there is at least one index ℓ such that $X_{\ell,j} \notin W$ for all $j \in [n]$ (in other terms, the number of probes of the form $X_{i,j}$ for distinct values of i is at most $n-1$). Notice that this case is slightly different than the case of claims 2 and 3, since ℓ can be in the set J. In this case, we can let $I_b = [n]$ so that we can perfectly simulate all wires in $W^{(1)} \cup \ldots \cup W^{(n)}$ and $J_b^{(1)} \cup \ldots \cup J_b^{(n)}$ using $I_b = [n]$, and we can perfectly simulate all wires in W' using I_a by construction and $I_b = [n]$ and generating the necessary random values. Next, we need to prove that we can perfectly simulate all output shares in J. Let $i \in J$. If $i \in I_a$, then c_i is perfectly simulated using a_i and $I_b = [n]$ and generating the necessary random values. Next, if $i \notin I_a$, then if $\ell \notin J$, we can use claim 2 to prove that we can replace the expression of $c_i = V_i + X_i$ by the random value $r_{i,l}$ and so the share a_i is not needed for the simulation of c_i (even if $X_{i,j}$ for a certain j is probed, the expression of V_i is still masked by $r_{i,l}$ and a_i is not needed to simulate X_i which is a sum of random values). Meanwhile, if $\ell \in J$, then we cannot directly use the random value $r_{i,\ell}$ to mask the expression of c_i. But since $X_{\ell,j} \notin W$ for all $j \in [n]$, and since $r_{k,\ell} \notin W$ because $k \notin I_a$ by assumption, then c_ℓ can be masked by the random value $r_{k,\ell}$, i.e $c_\ell = V_\ell + X_\ell \leftarrow r_{k,\ell}$. Since $i \in J$ and $k \notin J$, then $i \neq k$ and the random value $r_{i,\ell}$ does not appear anymore in X_ℓ in the expression of c_ℓ. Since $i \notin I_a$ then $r_{i,\ell}$ can be used to mask the expression of the output share c_i indexed in J and so the share a_i is not needed for the simulation of c_i. This proves that we can perfectly simulate all shares in J with the constructed sets I_a and $I_b = [n]$.

We managed to show that whenever the construction of the set I_a gives $|I_a| \leq t$, then we can perfectly simulate the sets W and J with at most a failure on input b and while still having $|I_a| \leq t$ and $|I_a| \leq |W|$.

By considering both cases $|I_a| \geq t+1$ and $|I_a| \leq t$, we covered all the cases for the simulation, and we proved that we can always perfectly simulate the set of probes W along with the set of output shares J while having a failure on at most one of the inputs. This concludes the proof of Lemma 16. □

I.2 Proof for TRPE2 property

Lemma 17. *The above multiplication gadget is (t, f_2)-TRPE2 of amplification order $d \geq \min(t+1, n-t)$*

Proof. To prove the lemma, we proceed in two steps through the following two lemmas 18 and 19.

Lemma 18. *Let W be a set of leaking wires as described above such that $|W| < \min(t+1, n-t)$. Then there exists a set J of $n-1$ output shares, such that W and J can be perfectly simulated from at most $\min(|W|, t) = |W|$ shares of each of the inputs a and b.*

Proof. We will construct the set of input shares indices I_a and the sets of output shares $J_b^{(k)}$ for $k \in [n]$ depending on the probes in the set W' (recall that $W = W' \cup W^{(1)} \cup \ldots \cup W^{(n)}$) as follows (we consider that all $J_b^{(k)}$ are empty at first since all the output shares of G_{refresh} can be probed directly in W'):

(a) For probes of form (a), we add index i to I_a, and index j to $J_b^{(k)}$ for $k \in [n]$.
(b) For probes of form (b), we add index i to I_a and to $J_b^{(k)}$ for $k \in [n]$.
(c) For probes of form (c), we add index i to $J_b^{(k)}$ for $k \in [n]$.

Observe that since $|W| < \min(t+1, n-t)$, then in particular $|W'| \leq \min(t+1, n-t) - 1 \leq t$, then $|I_a| \leq |W'| \leq |W| \leq t$ so we have no failure on the input a. Also, since $|J_b^{(k)}| \leq |W'| \leq t$ and $|W^{(k)}| < \min(t+1, n-t)$, then by the (t, f')-TRPE1 property of G_{refresh}, we will be able to simulate sets $J_b^{(k)}$ and $W^{(k)}$ from the set of input shares $I_b^{(k)}$ such that $|I_b^{(k)}| \leq |W^{(k)}| \leq t$ for $k \in [n]$. Thus, we can let $I_b = I_b^{(1)} \cup \ldots \cup I_b^{(n)}$ and we have $|I_b| \leq |W^{(1)} \cup \ldots \cup |W^{(n)}| \leq |W| \leq t$, so we have no failure on the input b either. Until now, we have shown that we can simulate all sets $W^{(k)}$ and $J_b^{(k)}$ from I_b of size at most $\min(|W|, t) = |W|$. It remains to show that we can also perfectly simulate the set W' and a well chosen set J of $n-1$ output shares, from I_a and $J_b^{(k)}$ for $k \in [n]$. We will choose the set J from two subsets $J = J_1 \cup J_2$, where $J_1 = \{i \mid i \in J_b^{(k)} \text{ for any } k \in [n]\}$, and $J_2 \subset [n]$ is any set such that $J_1 \cap J_2 = \emptyset$ and $|J_1 \cup J_2| = n-1$. Let $\ell \in [n]$ be the index such that $\ell \notin J$. Since $|W| \leq \min(t+1, n-t) - 1 \leq n-1$, then by construction of the sets $J_b^{(k)}$, we have that $|J_b^{(1)} \cup \ldots \cup J_b^{(n)}| \leq n-1$, then for the index ℓ, we have that $\ell \notin J_b^{(k)}$ for all $k \in [n]$, then $X_{\ell,j} \notin W$ for any $j \in [n]$ by construction of the sets $J_b^{(k)}$. The value of X_ℓ will be useful to use the following claim.

Claim 4 *Let $i \in J$. Suppose that $V_{i,j} \notin W$ for all $j \in [n]$. Then the expression of $c_i = V_i + X_i$ can be masked by the random value $r_{i,\ell}$, in other terms $c_i \leftarrow r_{i,\ell}$.*

Proof. The proof of this claim is quite straightforward since we suppose that $V_{i,j} \notin W$ for all $j \in [n]$, so none of the partial sums $V_{i,j}$ has been probed. Then V_i in c_i contains $n-1$ random values. In particular, we know that $r_{i,\ell}$ and $p_{i,\ell}$ only appear in the expression of the probed output c_i, because if they were probed in W then we would have $\ell \in J_b^{(i)}$ by construction, but we suppose that $\ell \notin J_b^{(k)}$ for all $k \in [n]$. In addition, since $X_{\ell,j} \notin W$ for all $j \in [n]$ (because otherwise then by construction $\ell \in J_b^{(k)}$ which does not hold), then $r_{i,\ell}$ does not appear in any other expression of the probed wires in W, so we can simply use it to perfectly simulate c_i. □

We can now show that the sets W' and J can be perfectly simulated from the constructed sets I_a and $J_b^{(k)}$.

Simulation of W'. Probes of the form (a) can be perfectly simulated from the corresponding input shares in I_a and $J_b^{(k)}$, and by generating uniformly random values $r_{i,j}$ when necessary. Probes of the form (c) are also perfectly simulated by simply generating uniformly random values, since $X_{i,j} = r_{1,i} + \ldots + r_{j,i}$. As for probes of the form (b), we know that $i \in I_a$, then we look at each of the terms $p_{i,j'}$ for $j' \in [j]$ in $V_{i,j} = p_{i,1} + \ldots p_{i,j}$. For each $p_{i,j'}$, if $j' \in J_b^{(i)}$, then $p_{i,j'}$ can be perfectly simulated from the corresponding input shares and by generating uniformly at random $r_{i,j'}$. Otherwise, if $j' \notin J_b^{(i)}$, then that means that the wires $p_{i,j'}, r_{i,j'}$ and $X_{j'}$ are not probed in W'. That means that we can potentially replace $p_{i,j'}$ by a random value $r_{i,j'}$ since $r_{i,j'}$ does not appear in any other expression of the variables probed in W'. Meanwhile, we also need to check the case where $j' \in J$, since $c_{j'} = V_{j'} + X_{j'}$, and $r_{i,j'}$ is the one of the summed terms in the expression of $X_{j'}$:

- If $j' \notin J$, then we can replace $p_{i,j'}$ by a random value $r_{i,j'}$ since $r_{i,j'}$ does not appear in any other expression of the variables probed in W' and is not probed either through $c_{j'}$.
- If $j' \in J$, then we also know that $V_{j'} \notin W'$ (because otherwise we would have by construction $j' \in J_b^{(k)}$ for $k \in [n]$ which does not hold), then we know from claim 4 that $c_{j'}$ can be masked by the random value $r_{j',\ell}$, which masks $V_{j'} + X_{j'}$. Since $\ell \neq j'$ (because $\ell \notin J$ while $j' \in J$), then $r_{i,j'}$ does not appear anymore in any other wire expression of the probed variables in W or J except in the term $p_{i,j'}$ of $V_{i,j}$, so $r_{i,j'}$ can be used to mask the expression of $p_{i,j'}$.

By perfectly simulating each term $p_{i,j'}$ in $V_{i,j}$, we can perfectly simulate $V_{i,j}$. Thus, we can perfectly simulate all wires in W'.

Simulation of J. Let $i \in J$. Let us first consider the case where $V_{i,j} \notin W'$ for any $j \in [n]$, then by claim 4, the output share c_i can be masked by the random variable $r_{i,\ell}$, so c_i is perfectly simulated by generating a fresh random value. Otherwise, if $V_{i,j} \in W'$ for a certain $j \in [n]$, then we know that the value of $V_{i,j}$ is perfectly simulated as proven above. Now, let us check each term $p_{i,j'}$ for $j' \in [j+1, n]$. Actually, we can also perfectly simulate each of these terms like the terms $p_{i,j'}$ for $j' \in [j]$. Plus, the term $p_{i,i}$ is perfectly simulated by construction of the sets I_a and $J_b^{(i)}$ (because $V_{i,j} \in W'$). In addition, all terms in X_i in $c_i = V_i + X_i$ can be perfectly simulated by generating a fresh random value. Thus, c_i can be perfectly simulated by summing all of the perfectly simulated terms in it. This brings us to a perfect simulation of all output shares in J. We have shown that we can perfectly simulate any set of probes W of size at most $\min(t+1, n-t) - 1$ with a chosen set J of $n-1$ output shares, with at most $\min(|W|, t) = |W|$ shares of each of the inputs a and b. This concludes the proof of Lemma 18. □

Remark 3. We can observe that for this lemma to apply on G_{mult}, we don't need the pre-processing phase of the refresh on input b. In fact, you can see that during the construction of the sets $J_b^{(k)}$, we add each index to all of the sets for all $k \in [n]$. However, executing n refreshings on the input b will be necessary for the proof of the next result, specifically when we consider W such that $\min(t+1, n-t) \leq |W| < 2 \cdot \min(t+1, n-t)$.

To get back to the proof of Lemma 17, we also need the following result.

Lemma 19. *Let W be a set of leaking wires as described above such that $\min(t+1, n-t) \leq |W| < 2 \cdot \min(t+1, n-t)$. Then there exists a set J of $n-1$ output shares such that W and J can be perfectly simulated from sets of input shares I_a and I_b such that $|I_a| \leq \min(|W|, t)$ **or** $|I_b| \leq \min(|W|, t)$. In other terms, we have a simulation failure on at most one of the inputs a or b.*

Proof. Recall that the set W can be split into subsets $W = W' \cup W^{(1)} \cup \ldots \cup W^{(n)}$ as described above. We consider two cases.

Case 1: $|W'| < \min(t+1, n-t)$. This case is similar to the case of Lemma 18, so we can construct the set I_a in the same way as in the proof of Lemma 18, and we can eventually consider $I_b = [n]$. We know that $|I_a| \leq |W'| \leq \min(t+1, n-t) - 1 \leq t$, so there is no failure on the input a. And all probes in W' can be simulated like in the proof of Lemma 18 with I_a and trivially with $I_b = [n]$. Also, all probes in $W^{(1)} \cup \ldots \cup W^{(n)}$ can be trivially simulated since we have access to the full input b. In addition, we choose the set J of size $n-1$ in the same way as in Lemma 18. Whenever $i \in J$ and $V_{i,j} \in W'$ for some $j \in [n]$, then $c_i = V_i + X_i$ is easily simulated using $I_b = [n]$ and the share a_i. If $i \in J$ but $V_{i,j} \notin W'$ for all $j \in [n]$, then as in the proof of Lemma 18, c_i in this case can be masked by the random value $r_{i,\ell}$ (because $|W'| < \min(t+1, n-t)$) and so simulating c_i amounts to generating uniformly at random the corresponding random value. Thus, W and J are perfectly simulated with at most $\min(|W|, t)$ shares of a and eventually the full input b.

Case 2: $|W'| \geq \min(t+1, n-t)$ *(and thus* $|W^{(1)} \cup \ldots \cup W^{(n)}| < \min(t+1, n-t)$*).* In this case, we will construct the sets I_a and $J_b^{(k)}$ from empty sets, in a way that we will have a simulation failure on at most one of the inputs a or b. We construct the mentioned sets depending on the probes in W' as follows:

(a) For probes of form (a), we add index i to I_a, and index j only to $J_b^{(i)}$.
(b) For probes of form (b), we add index i to I_a and only to $J_b^{(i)}$.
(c) For probes of form (c), we add index i to $J_b^{(k)}$ for all $k \in [n]$.

In the rest of the lemma, we will prove that if we have a failure on one of the inputs, we can still perfectly simulate W and a chosen set J of $n-1$ output shares without a failure on the other input. For this, we will consider two cases, the first where we have a failure on input a, the second where we don't have a failure on input a, and so we can eventually have a failure on input b.

Case 2.1: simulation failure on input a, i.e. $I_a > t$. This means that the set I_a is of size $|I_a| \geq t+1 \geq \min(t+1, n-t)$ (this is because by construction $|I_a| \leq |W|$, so to have $|I_a| > \min(|W|, t)$, we must have $|I_a| > t$). We will first start by showing that the sets $W^{(k)}$ and $J_b^{(k)}$ can be perfectly simulated using the simulator of G_{refresh} without a failure on the input b. Next, we will show that W' and a well chosen set of $n-1$ output shares in J can be perfectly simulated using I_a and $J_b^{(k)}$.

Since we only add shares indices to I_a, when we have probes of the form (a) or (b), this means that we have at least $t+1$ probes of these two forms with $t+1$ different values for the index i. In addition, since we have at least $t+1$ probes (a) or (b) with distinct values for the index i, then this also means that each of the sets $J_b^{(i)}$ has at most one share of $b^{(i)}$ added to it. In other terms, $|J_b^{(k)}| \leq 1$ for each $k \in [n]$ (from the probes (a) and (b) with distinct indices i).

Now let us consider the remaining probes in W which are either in W' of the form (c) or in $W^{(1)} \cup \ldots \cup W^{(n)}$ or of the forms (a) or (b) with $i \in I_a$. Since $|I_a| \geq t+1 \geq \min(t+1, n-t)$, then there are at most $\min(t+1, n-t) - 1$ of these remaining probes. Without loss of generality, we consider that there are exactly $\min(t+1, n-t) - 1$ instead of at most $\min(t+1, n-t) - 1$ probes. Let m be the number of probes in $W^{(1)} \cup \ldots \cup W^{(n)}$ and $d-1-m$ the remaining probes in W' of the form (c) or (a)/(b) with $i \in I_a$. Since each wire in W' of the form (c) or (a)/(b) with $i \in I_a$ results in adding at most one more share index to each $J_b^{(k)}$ for $k \in [n]$, then we have

$|J_b^{(k)}| \leq 1 + (\min(t+1, n-t) - 1 - m) = d - \min(t+1, n-t)$. And $|W^{(1)} \cup \ldots \cup W^{(n)}| \leq m$, in particular $|W^{(k)}| \leq m$ for any $k \in [n]$.

- if $m = 0$, then $W^{(k)} = \emptyset$ for any $k \in [n]$, and $|J_b^{(k)}| \leq \min(t+1, n-t) \leq \lfloor \frac{n+1}{2} \rfloor \leq n-1$, so by the TRPE property of G_{refresh} for any $t \leq n-1$, all of the $J_b^{(k)}$ sets can be perfectly simulated with no knowledge of the input shares of b since $W^{(k)} = \emptyset$, so $I_b^{(k)} = \emptyset$. Hence, $I_b = I_b^{(1)} \cup \ldots \cup I_b^{(n)} = \emptyset$ and we have no simulation failure on the input b.
- if $m > 0$, then $|J_b^{(k)}| \leq \min(t+1, n-t) - 1 \leq t$ and $|W^{(1)} \cup \ldots \cup W^{(n)}| < \min(t+1, n-t)$, in particular $|W^{(k)}| \leq \min(t+1, n-t) - 1$ for each $k \in [n]$. Thus, by the (t, f)-TRPE property of the refresh gadget G_{refresh} achieving the amplification order d, we can perfectly simulate both sets for each $k \in [n]$ with $I_b^{(k)}$ such that $|I_b^{(k)}| \leq |W^{(k)}|$. Thus, we can let $I_b = I_b^{(1)} \cup \ldots \cup I_b^{(n)}$ so we can have $|I_b| \leq |W^{(1)} \cup \ldots \cup W^{(n)}| \leq \min(t+1, n-t) - 1 \leq t$, and we can perfectly simulate $W^{(1)} \cup \ldots \cup W^{(n)}$ along with $J_b^{(1)} \cup \ldots \cup J_b^{(n)}$ from the set I_b without a simulation failure on input b.

So far we proved that if we have $|I_a| > t$, then we must have $|I_b| \leq t$, and $W^{(1)} \cup \ldots \cup W^{(n)}$ can be perfectly simulated along with $J_b^{(1)} \cup \ldots \cup J_b^{(n)}$ from the set I_b. Next we need to prove that we can perfectly simulate W' and a chosen set J of $n-1$ output shares, from these sets I_a and $J_b^{(1)} \cup \ldots \cup J_b^{(n)}$. We consider two sub-cases.

Case 2.1.1: $I_a = [n]$. In this case, since $|W| \geq n$ (from I_a) and $|W| < 2\min(t+1, n-t) \leq n+1$, then $|W| = n$ and all probes in W are all in W' of the form (a) or (b) with n distinct values for the index i. We neither have probes in $W^{(1)} \cup \ldots \cup W^{(n)}$ nor in W' of the form (c). Thus, we can reconstruct each $|J_b^{(k)}|$ of size at most $n-1$ without having a failure on the input b (since G_{refresh} is (t', f')-TRPE for any $t' \leq n-1$ achieving $d' = \min(t'+1, n-t')$ and all $W^{(k)}$ are empty). We consider two cases:

- Suppose that for each $i \in [n]$, we have at least one probe in W' of the form $r_{k,i}$ or $p_{k,i}$ for some $k \in [n]$, note this probe $q_{k,i} \in \{r_{k,i}, p_{k,i}\}$. Since also $I_a = [n]$, this means that we have probes $q_{k_1,1}, \ldots, q_{k_n,n}$, such that $k_1 \neq \ldots \neq k_n$. Because $|W'| = n$, then all probes in W' are of the form (a) (specifically $q_{k,i}$), and we have no probes of the form $V_{i,j}$ for any $i, j \in [n]$. In this case, the simulation of the probes in W' is straightforward by construction of the sets I_a and $J_b^{(k)}$. As for the set J, we let $J \subset [n]$ such that $|J| = n-1$ (any set of $n-1$ shares works), and let $\ell \in [n]$ such that $\ell \notin J$. Observe that out of all the random values $r_{i,\ell}$ in X_ℓ in the expression of $c_\ell = V_\ell + X_\ell$, only the random value $r_{k_\ell,\ell}$ appears in the expression of the probe $q_{k_\ell,\ell}$ in the set W', and all other random values $r_{i,\ell}$ for $i \neq k_\ell$ do not appear in any other probed variable in W' (since $W' = \{q_{k_1,1}, \ldots, q_{k_\ell,\ell}, \ldots, q_{k_n,n}\}$, such that $k_1 \neq \ldots \neq k_n$). Then, for each $i \in J$ such that $i \neq k_\ell$, the expression of $c_i = V_i + X_i$ can be masked by the random value $r_{i,\ell}$, so simulating c_i amounts to generating a fresh random value $r_{i,\ell}$. Now let's check $i = k_\ell \in J$. Since $q_{k_\ell,\ell}$ is probed, then we cannot mask the expression of c_{k_ℓ} using $r_{k_\ell,\ell}$. However, for each $i \in J$ with $i \neq k_\ell$, we have that c_i is masked by $r_{i,\ell}$. Since $c_i = V_i + X_i$, and the random value $r_{k_\ell,i}$ is one of the terms in X_i, then $r_{k_\ell,i}$ does not appear anymore in the expression of c_i. And since $W' = \{q_{k_1,1}, \ldots, q_{k_\ell,\ell}, \ldots, q_{k_n,n}\}$, such that $k_1 \neq \ldots \neq k_n$ and $q_{k_\ell,\ell} \in W'$, then $q_{k_\ell,i} \notin W'$ and $r_{k_\ell,i}$ only appears in the expression of c_{k_ℓ}, so c_{k_ℓ} can be masked by the random value $r_{k_\ell,i}$. Thus, we proved that we can perfectly simulate the sets W' and J using the sets I_a and $J_b^{(k)}$.

– Next, we suppose that there exists $\ell \in [n]$ such that we have no probes in W' of the form $q_{k,\ell} \in \{r_{k,\ell}, p_{k,\ell}\}$. In this case, we choose $J = [n] \setminus \{\ell\}$. Next, we show that we can perfectly simulate all probes in W' and output shares in J for each $i \in I_a = [n]$. For this, first let $i \in [n] \setminus \{\ell\}$ (notice that we automatically have $i \in J$):
  - if for the considered i, the probe in W' is of the form (a) i.e $a_i, b_j^{(i)}, a_i \cdot b_j^{(i)}, p_{i,j} = a_i \cdot b_j^{(i)} + r_{i,j}$, then the simulation of this probe is trivial by construction of the sets I_a and $J_b^{(i)}$. In addition, we know that $r_{i,\ell}$ and $p_{i,\ell}$ are not probed in W' by assumption, and since $X_{\ell,j} \notin W'$ for all $j \in [n]$, then the random value $r_{i,\ell}$ only appears in the expression of $c_i = V_i + X_i$ (specifically in V_i), and so can be used to mask c_i. So simulating c_i amounts to generating a fresh random value.
  - if for the considered i, the probe in W' is of the form (b), i.e $V_{i,j} \in W'$ for a certain $j \in [n]$ (there is a unique probe of this form), then:
    * either $j < \ell$, and so the random value $r_{i,\ell}$ can be used as before to mask the expression of $c_i + V_{i,j}$, and since in this case $V_{i,j}$ contains less than $n-1$ terms $p_{i,j'}$, then we can add all the necessary shares of $b^{(i)}$ to $J_b^{(i)}$without having a failure on b (recall that $W^{(i)} = \emptyset$). So we can perfectly simulate $V_{i,j}$ and $c_i + V_{i,j}$, and hence also simulate c_i.
    * or $j \geq \ell$, and so the random value $r_{i,\ell}$ can be used in this case to mask the expression of $V_{i,j}$ so simulating $V_{i,j}$ amounts to generating a fresh random value, and since $V_{i,j}$ is the sum of at least two terms of the form $p_{i,j'}$, then $c_i + V_{i,j}$ can be simulated with at most $n-1$ shares of $b^{(i)}$, so there is no simulation failure on input $b^{(i)}$. So we can perfectly simulate $V_{i,j}$ and $c_i + V_{i,j}$, and hence also simulate c_i.

  Next we consider the case of the probe $V_{\ell,j}$:
  - either $j < \ell$, and so in this case $V_{\ell,j}$ contains less than $n-1$ terms $p_{\ell,j'}$, then we can add all the necessary shares of $b^{(\ell)}$ to $J_b^{(\ell)}$without having a failure on b (recall that $W^{(\ell)} = \emptyset$). So we can perfectly simulate $V_{\ell,j}$ using the input share a_ℓ, the input shares of $b^{(\ell)}$ and by generating necessary random values.
  - or $j \geq \ell$, and so the random value $r_{\ell,\ell}$ can be used in this case to mask the expression of $V_{\ell,j}$ so simulating $V_{\ell,j}$ amounts to generating a fresh random value.

  Thus, also in this case, we can perfectly simulate W' and a chosen set of $n-1$ output shares without a failure on input b, using $I_a = [n]$.

This concludes the simulation for the special case where $I_a = [n]$.

Case 2.1.2: $I_a \subset [n]$ *such that* $|I_a| \leq n-1$. Let k such that $k \notin I_a$. Recall that $|I_a| \geq t+1 \geq \min(t+1, n-t)$ and $|W| < 2 \cdot \min(t+1, n-t)$, then there are at most $\min(t+1, n-t) - 1 \leq t \leq n-1$ probes remaining either in $W^{(1)} \cup \ldots \cup W^{(n)}$, of the form (c) in W', or of the form (a)/(b) with $i \in I_a$. Thus, there exists at least one index $\ell \in [n]$ such that $X_{\ell,j} \notin W'$ for all $j \in [n]$. In this case, we choose $J = [n] \setminus \{\ell\}$. Next, we will prove that we can perfectly simulate the sets W' and J from the constructed sets I_a and $J_b^{(k)}$, using the following claims.

Claim 5 *Let $i \in J$. Suppose that $i \notin I_a$. Then the expression of $c_i = V_i + X_i$ can be masked by the random value $r_{i,\ell}$, in other terms $c_i \leftarrow r_{i,\ell}$.*

Proof. This claim can be proved easily, since we suppose that $i \notin I_a$ so the random value $r_{i,\ell}$ and $p_{i,\ell}$ are not probed in W'. In addition, since $\ell \notin J$ and $X_{\ell,j} \notin W$ for all $j \in [n]$, then the random value $r_{i,\ell}$ does not appear in any other probed wire expression except in c_i, then c_i can be masked by the random value $r_{i,\ell}$. □

Claim 6 *Let $i \in J$. Suppose that $X_{i,j} \notin W$ for any $j \in [n]$. Suppose that $i \in I_a$. Then the expression of $c_i = V_i + X_i$ can be masked by the random value $r_{k,i}$, in other terms $c_i \leftarrow r_{k,i}$.*

Proof. Since we suppose that $k \notin I_a$, then the random value $r_{k,i}$ or $p_{k,i}$ or $V_{k,j}$ for all $j \in [n]$ are not probed in W. Then, if $k \notin J$, then the random value $r_{k,i}$ does not appear in the expression of any other probed wire in W or J and c_i can be masked by the random value $r_{k,i}$ (Recall that $c_i = V_i + X_i$ and $X_i = r_{1,i} + \ldots + r_{n,i}$). Otherwise, if $k \in J$, then by Claim 5, $c_k = V_k + X_k$ can be masked by $r_{k,\ell}$ and so c_i can also be masked by $r_{k,i}$ since $i \neq \ell$ (because $i \in J$ and $\ell \notin J$) and $i \neq k$ (because $i \in I_a$ and $k \notin I_a$). □

Probes of the forms (a) or (c) in W' are trivially simulated using the constructed sets of input shares, and generating the necessary random values. Let us now check the probes of the form (b). Let $V_{i,j} = p_{i,1} + \ldots + p_{i,j}$ be such a probe. Let us consider each of the terms $p_{i,j'}$ for $j' \in [j]$. if $j' = i$, then by construction $p_{i,i}$ is perfectly simulated using a_i and $b_i^{(i)}$ and by generating the random value $r_{i,i}$ if needed. Otherwise, let $j' \neq i$. If $j' \in J_b^{(i)}$ then the simulation of $p_{i,j'}$ is straightforward. Otherwise if $j' \notin J_b^{(i)}$, then we know that none of the wires $r_{i,j'}$ or $p_{i,j'}$ or $X_{j',s}$ for all $s \in [n]$ are probed in W'. Thus, $r_{i,j'}$ can be eventually used to mask the expression of $p_{i,j'}$ without the need of the share $b_{j'}^{(i)}$ for the simulation. Meanwhile, we still need to check if $j' \in J$, since $r_{i,j'}$ appears in $X_{j'}$ in the expression of $c_{j'} = V_{j'} + X_{j'}$.

- If $j' \in J$ and $j' \notin I_a$, then by claim 5, $c_{j'}$ can be masked by the random value $r_{j',\ell}$ and so $r_{i,j'}$ does not appear in the expression of $X_{j'}$ in $c_{j'}$ anymore, and $r_{i,j'}$ can be used to mask $p_{i,j'}$.
- Otherwise, if $j' \in J \cap I_a$, then by claim 6 $c_{j'}$ can be masked by the random value $r_{k,j'}$ and so $r_{i,j'}$ does not appear in the expression of $X_{j'}$ in $c_{j'}$ anymore, and $r_{i,j'}$ can be used to mask $p_{i,j'}$ (since $i \notin k$).

Thus, each term $p_{i,j'}$ in $V_{i,j}$ can be perfectly simulated and thus $V_{i,j} = p_{i,1} + \ldots + p_{i,j}$ can be perfectly simulated. This concludes the simulation of the set W'.

We now focus on the simulation of J. Let $i \in J$. If $i \notin I_a$, then by claim 5, c_i is perfectly simulated by generating the random value $r_{i,\ell}$. Otherwise, let $i \in I_a$. If $X_{i,j} \notin W$ for any $j \in [n]$, then by claim 6, c_i is perfectly simulated by generating the random value $r_{k,i}$. Otherwise, we can show that we can perfectly simulate each term in $c_i = V_i + X_i$. In particular, each term in X_i can be simulated by generating the underlying random value uniformly. For V_i, we know in particular that $a_i \cdot b_i^{(i)}$ is perfectly simulated since $X_{i,j} \in W'$ for at least one $j \in [n]$ so $i \in J_b^{(i)}$ by construction. For the other terms in V_i, they can be perfectly simulated in the exact same way as we simulated the probes $V_{i,j}$ of the form (b) in the set W'. So c_i is perfectly simulated by summing all the perfectly simulated terms. This concludes the simulation proof for the set J.

Up until now, we have concluded that if we have a constructed set I_a of size at least $t+1$, then we can perfectly simulate the sets W and a chosen set J of $n-1$ output shares, without having a simulation failure on the input b. In the rest of the proof, we will consider that $|I_a| \leq t$, and we will prove that we can perfectly simulate W and J with at most a simulation failure on b. Recall that we are also considering that $|W'| \geq d$ and $|\mathbf{W^{(1)}} \cup \ldots \cup \mathbf{W^{(n)}}| \leq \mathbf{d - 1}$.

Case 2.2: $|I_a| \leq t$. This means that the number of probes of the form (a) or (b) in W' with distinct values for the index i is at most t.

First, let us check the special case where the number of probes of the form (c) in W' with different values for the index i is equal to n (notice that this cannot occur when we have $|I_a| \geq t+1$). Since $|W| \leq 2 \cdot \min(t+1, n-t) - 1 \leq n$, then we have $W = \{X_{1,j_1}, \ldots, X_{n,j_n}\}$ for certain $j_1, \ldots, j_n \in [n]$. So we can let $J_b^{(k)} = [n]$ for all $k \in [n]$ and $I_b = [n]$, and by construction $I_a = \emptyset$. In this case, we choose $J = [n-1]$. The simulation of the set W is straightforward since all wires of the form (c) are just sums of random values. Then, let us consider the output shares in J.

- If for at least one $\ell \in J$, we have $X_{\ell,n} \in W$, we can mask the expression of $X_{\ell,n}$ by the random value $r_{n,\ell}$ (because there are no probes of the form (a) or (b) in W and $n \notin J$, so $r_{n,\ell}$ only appears in the expression of $X_{\ell,n}$). Recall that $X_{\ell,n} = r_{1,\ell} + \ldots + r_{n,\ell}$, so $r_{n,\ell}$ masks all random values $r_{j,\ell}$ for $j \in [n-1]$. Each of the random values $r_{j,\ell}$ for $j \in [n-1] \setminus \{\ell\}$ can be used to mask the corresponding output share c_j for $j \in J$ because there are no probes of the form (a) or (b) in W and $X_{\ell,n}$ is already masked by $r_{n,\ell}$, so $r_{j,\ell}$ only appears in the expression of $c_j = V_j + X_j$, so $c_j \leftarrow r_{j,\ell}$. As for the output c_ℓ, we can let $I_a = \{\ell\}$ and we can perfectly simulate c_ℓ using a_ℓ and $I_b = [n]$. Since $|W| = n$, so $\min(t+1, n-t) > \dfrac{n}{2} \geq 1$, so we have no failure on the input a, and we can perfectly simulate the chosen set J and the set of probes W.
- Now we consider that for all $W = \{X_{1,j_1}, \ldots, X_{n,j_n}\}$, we have $j_1 < n, \ldots, j_n < n$. In this case, the set W is also trivially simulated by generating random values, and we let $J = [n-1]$. Since, $n \notin J$ and there are no probes of the form (a) or (b) in W, then the random values $r_{n,i}$ for $i \in [n-1]$ only appear in the expression of the output share $c_i = V_i + X_i$ each. And since all probes of the form $X_{i,j}$ are such that $j < n$, then we can let $r_{n,i}$ be used to mask the expression of $c_i + X_{i,j}$ because $r_{n,i}$ does not appear in $X_{i,j}$ for $j < n$, i.e $c_i + X_{i,j} \leftarrow r_{n,i}$. By perfectly simulating the masked expression of $c_i + X_{i,j}$ and the sum of random values $X_{i,j}$, we can perfectly simulate c_i. Thus, simulating all output shares in J amounts to generating random values uniformly. So we can perfectly simulate sets W and J from $I_a = \emptyset$ and $I_b = [n]$.

Next, we suppose that the number of probes of the form (c) in W' with different values for the index i is strictly smaller than n. So, there is at least one index ℓ such that $X_{\ell,j} \notin W'$ for all $j \in [n]$. We let $J = [n] \setminus \{\ell\}$. We also let $I_b = [n]$ and we keep the set I_a as constructed according to the probes in the set W'. Observe that all probes in W' are perfectly simulated by easily using the set I_a and $I_b = [n]$. As for the output shares in J, observe that for each $i \in J$ such that $i \notin I_a$, we can use claim 5 to mask the expression of c_i by $r_{i,\ell}$, and so the share a_i is not needed for the simulation of c_i. Otherwise, if $i \in J \cap I_a$, then c_i is perfectly simulated using a_i and $I_b = [n]$.

This proves that whenever $i \notin I_a$, the output share c_i can be simulated without the need of the share a_i. Since we suppose that $|I_a| \leq t$, then we conclude that we can perfectly simulate W and a chosen set of $n-1$ output shares J with at most a simulation failure on input b.

By considering both cases $|I_a| \geq t+1$ and $|I_a| \leq t$, we covered all the cases for the simulation, and we proved that we can always perfectly simulate the set of probes W along with a chosen set of $n-1$ output shares J while having a failure on at most one of the inputs. This concludes the proof of Lemma 19. □

From Lemmas 18 and 19, we conclude that G_{mult} is (t, f_2)-TRPE2 of amplification order $d \geq \min(t+1, n-t)$. This concludes the proof of Lemma 17. □