# Exploiting algebraic structures in probing security.

Maxime Plançon*

IBM Research Europe, Zurich
ETH Zurich

**Abstract.** The so-called $\omega$-encoding, introduced by Goudarzi, Joux and Rivain (Asiacrypt 2018), generalizes the commonly used arithmetic encoding. By using the additionnal structure of this encoding, they proposed a masked multiplication gadget (GJR) with quasilinear (randomness and operations) complexity. A second contribution by Goudarzi, Prest, Rivain and Vergnaud in this line of research appeared in TCHES 2021. The authors revisited the aforementioned multiplication gadget (GPRV), and brought the IOS security notion for refresh gadgets to allow secure composition between probing secure gadgets.
In this paper, we propose a follow up on GPRV. Our contribution stems from a single Lemma, linking algebra and probing security for a wide class of circuits, further exploiting the algebraic structure of $\omega$-encoding. On the theoretical side, we weaken the IOS notion into the KIOS notion, and we weaken the usual $t$-probing security into the RTIK security. The composition Theorem that we obtain by plugging together KIOS, RTIK still achieves region-probing security for composition of circuits.
To substantiate our weaker definitions, we also provide examples of competitively efficient gadgets verifying our weaker security notions. Explicitly, we give 1) a refresh gadget that uses $d-1$ random field elements to refresh a length $d$ encoding that is KIOS but not IOS, and 2) multiplication gadgets asymptotically subquadratic in both randomness and complexity. While our algorithms outperform the ISW masked compiler asymptotically, their security proofs require a bounded number of shares for a fixed base field.

Keywords: Masking, Refresh Gadget, Multiplication Gadget, Probing Security

## 1 Introduction

Since their introduction in the late 90's by Kocher [KJJ99, Koc96], side-channel attacks have proven to be a major threat to cryptography. While cryptanalysis can evaluate the black-box security of cryptographic protocols, their security can be totally compromised by physical attacks. In a nutshell, side-channel attacks refer to any attack taking advantage of the implementation of a cryptographic

* Part of this work was done during an internship at PQShield in collaboration and under the supervision of Thomas Prest.

protocol, rather than only the public parameters and public communications. If a hardware device is manipulating carelessly a secret value, many observable signals (such as its temperature, power consumption, electromagnetic field, etc) are likely to leak secret information, and might even lead to a full-key recovery. These practical security flaws call for a solid non-ad hoc response.

Of all the side-channel adversary models such as the noisy leakage model [PR13, DDF14, DFS15] or the random probing model [ADF16], arguably the easiest to deal with is the so called (threshold) $t$-probing model [ISW03]. A $t$-probing adversary may choose adaptively and learn any $t$ intermediate values of the circuit.

Masking is a countermeasure that provably prevents recovering information when the adversary is snooping on the circuit. Informally, masking uses secret-sharing techniques to provide probing security to a circuit. A sensitive intermediate value $x$ of the cryptographic protocol is encoded into a vector of $d$ shares $(x_1, \ldots, x_d)$. While the knowledge of all $d$ shares allows to recover the secret it encodes, masking requires that any $d-1$ shares are independent of the secret value $x$. Any partial knowledge of the shares is therefore made useless in masking schemes, so as to provide $t$-probing security for $t < d$. The operations (additions, negations and multiplications for arithmetic circuits) then have to be performed *securely* in the encoded domain, so as to never manipulate secret variables directly. Each operation (or gate) of the circuit is transformed into a secure counterpart (or gadget), that takes as input encodings of the secrets, and outputs an encoding of the evaluation of the corresponding operation. Usually, masking schemes admit a coordinate-wise secure addition, leaving the multiplication the most challenging operation to perform securely in the encoded domain.

Replacing every gate with probing secure gadgets does not imply probing security for the whole circuit[BCPZ16, CPRR13], and extra efforts have to be put into composition security. Composition of gadgets is a line of research that has received a lot of attention, and is still an active field of research [ADF16, CS20, BCPZ16, GPRV21, BBD$^+$16].

The first masked multiplication was introduced in 2003 in [ISW03], and several variants achieving different trade-offs have been proposed [RP10, BBP$^+$16, BBP$^+$17]. The encoding used by ISW is the so called arithmetic masking (originally for boolean masking, but the arithmetic masking translation remains secure [RP10]), where the shares $\mathbf{x} = (x_1, \ldots, x_d)$ of some field element $x \in \mathbb{F}$ are such that $x_1 + \cdots + x_d = x$. Another way to interpret arithmetic masking is to say that the shares are the coefficients of a polynomial such that its evaluation in 1 is the secret. From a high level, the multiplication of two sharings $\mathbf{a}, \mathbf{b}$ of two secrets $a, b$ in ISW computes the coefficients of the polynomial $\mathbf{c} = \mathbf{ab}$ and rearranges the coefficients so as to have $\mathbf{c}$ of the same length $d$ as $\mathbf{a}$ and $\mathbf{b}$. This polynomial multiplication is performed following the schoolbook multiplication algorithm mixed up with some randomness for security. This yields a multiplication gadget running in $O(d^2)$ time with $O(d^2)$ randomness. The first attempt to build an asymptotically subquadratic secure masked multiplication is the multiplication from [GJR18], based on Fast Fourier Transform. GJR uses a different type

of encoding called $\omega$-encoding, where the $\mathbf{a}$'s evaluation is taken in some field element $\omega$ rather than 1. Arithmetic masking seems to be incompatible with the FFT since $a_1 + \cdots + a_d$ is an intermediate value of the FFT algorithm, which the adversary may therefore probe, and immediately break the masking scheme. There was a flaw in the original security proof of the GJR multiplication gadget, which was patched later in [GPRV21] and named GJR+. While GJR is a theoretical breakthrough, its range of application excludes AES for example. The security relies on the random choice of $\omega$, hence for reaching a reasonable level of security, GJR+ requires a gigantic underlying field, which limits its practical applications. Recently, [GPRV21] proposed a masked multiplication gadget for $\omega$-encodings also based on the FFT. This multiplication gadget which we will call GPRV achieves $O(d \log d)$ time and randomness complexity, where $d$ is the order of masking. One significant drawback of their construction is that the security proof relies on a non-standard ad-hoc assumption. This assumption, roughly speaking assumes that the computation of the FFT and inverse FFT of a polynomial are both probing secure. While one can check this hypothesis by exhaustive search, the computation becomes very costly as $d$ increases. The authors raise the open problem to build a strong theoretical foundation for replacing their assumption with a full proof.

The randomness complexity of a compiler (meaning the transformation of a circuit that replaces operation gates with secure masked gadgets) is of major importance. The predilection physical support for masked implementation is embedded systems, where randomness is expensive to produce. In this consideration, one of the goals in the field of masking is to achieve notions of security using as little randomness as possible. The authors of [GPRV21] give a generic composition Theorem that only requires $t$-probing security for the operation gadgets, and mask refreshing (they give such refresh algorithm verifying the desired Input-Output-Separation property) in between any two gadgets. This theorem ensures that the obtained compiler achieves the $r$-region-probing-security notion. Informally, region probing security means that the circuit can be split into independent regions, in which the side-channel adversary may probe a fixed ratio of the intermediate values yet learns no information on the secrets. The authors prove that a variant of the refresh gadget from [BCPZ16] achieves the IOS property and only requires $\frac{d \log d}{2}$ random field elements.

### 1.1 Our contribution

From a high level, we propose a retake on the circuit compiler from the recent paper [GPRV21]. Indeed, similarly as [GPRV21], we deal with polynomial encodings (i.e an encoding of $x \in \mathbb{F}$ is a degree $d-1$ polynomial $\mathbf{x}$ such that $x = \mathbf{x}(\omega)$) and our contributions can be summarized as

1. Definitions of security notions for operation gadgets (RTIK) and refresh gadgets (KIOS)
2. A composition Theorem linking the security notions from 1.
3. Examples of multiplication gadgets and refresh gadgets achieving the aforementioned notions

We detail separately each of these items in the following.

**Bridging algebra and probing security.** We deal with polynomial encodings, using some field element $\omega \in \mathbb{F}$, where the underlying field $\mathbb{F}$ is given by the cryptographic algorithm to be masked. The number of shares $d$ of this encoding is given by the number of probes that we tolerate from the adversary. The contributions of this paper stem from Lemma 3.6. Consider a circuit $\mathcal{C}$ taking as input an encoding $\mathbf{x}$. This lemma says, in a nutshell, that if there exists a subfield $K$ of $\mathbb{F}$ such that every intermediate value that can be probed by the adversary is a $K$-linear function of the input encoding $\mathbf{x}$, and if $d \leqslant [\mathbb{F} : K]$, then there exists a choice of $\omega$ for which $\mathcal{C}$ is $d-1$-probing secure. This choice of $\omega$ is actually any $\omega$ of algebraic degree greater than $d$ over $K$.

We consequently define security notions based on the aforementioned observation. We fix some subfield $K$ of $\mathbb{F}$ and write $k = [\mathbb{F} : K]$. The subfield $K$ has to be chosen such that, waving hands, the intermediate values involved in the gadget computations are $K$-linear. For the sake of simplicity, one may take $\mathbb{F} = \mathbb{F}_{p^k}$ for some prime $p$ and $K = \mathbb{F}_p$. The security notion that we introduce for gadgets is the so-called Reducible-To-Independent-$K$-Linear (RTIK) property, which requires a gadget $G$ to be such that for any set of probes $P$ that the adversary may chose, there exists another set of probes $Q$ that gives the adversary just as much or more information on the secret inputs of $G$, but those probes in $Q$ are $K$-linear. This way, we reduce the $t$-probing security game of $G$[1] to a setup that fits the requirements of our core Lemma whenever $d \leqslant k$. When the condition $d \leqslant k$ is satisfied, RTIK implies $d-1$-probing security, but we need a stronger assumption for composition. We note that similarly as the Probe-Isolating-Non-Interfering security notion [CS20], RTIK gadgets can be composed directly without refresh to form a $t$-probing secure circuit.

The security notion that we introduce for refresh gadgets is inspired by the Input-Output Separative (IOS) property. We briefly recall the idea behind the IOS property. Consider an IOS refresh gadget $R$ and two encodings $\mathbf{x}$ and $\mathbf{y}$ with $\mathbf{y} = R(\mathbf{x})$. Let us also assume that $\mathbf{x}$ is an output of some gadget $G_1$, and $\mathbf{y}$ is an input of some gadget $G_2$. We now let the $t$-probing adversary pick and learn $t$ intermediate variables in either $G_1, R$, or $G_2$. In this setting, the IOS property claims that any probe inside of the refresh gadget can be "moved" to a probe on a coordinate of $\mathbf{x}$ and/or a probe on a coordinate of $\mathbf{y}$. The probes on $\mathbf{x}$ are then considered as probes in $G_1$, the probes on $\mathbf{y}$ are then considered as probes on $G_2$, and $R$ itself is no more probed by the adversary. This reduces the security of the composition of the two gadgets $G_1, G_2$ to the individual security of each of the two gadgets. The security notion $\alpha$-KIOS that we define is identical to the IOS property, except the probes on $\mathbf{x}$ and $\mathbf{y}$ do not have to be coordinates, but any $K$-linear function of those inputs.[2] Executing the same reduction as the

[1] Actually, we use the slightly more general $r$-region probing security model.

[2] We also add a coefficient $\alpha$ to its definition, which upper bounds the ratio of $K$-linear probes on $\mathbf{x}$, $\mathbf{y}$ after the reduction and the count of initial probes in the KIOS gadget.

one explained above for IOS refresh gadgets, one ends up with $K$-linear probes on $\mathbf{x}$, $\mathbf{y}$, which in turn fall into the requirements of our core Lemma.

We finally give the equivalent of the composition Theorem from [GPRV21] that links $t$-probing security and IOS. This composition Theorem 3.16 links the RTIK property of gadgets and the KIOS property of refresh gadgets to the region-probing security of compound circuits.

**A 2-KIOS refresh gadget using $d-1$ randomness for length $d$ input encoding.** To substantiate the KIOS notion, we give examples of KIOS refresh gadgets. Notice that 1-KIOS is strictly weaker than IOS, and therefore any IOS refresh is an example of 1-KIOS refresh, including the one from [GPRV21] which uses $\frac{d \log d}{2}$ random elements. We also give an example of a 2-KIOS refresh gadget that is not IOS. This gadget is obtained by simply adding coordinate-wise an encoding of 0, obtained by running the algorithm PolyGenZero presented in Algorithm 4, which uses $d-1$ random field elements. We highlight that for security, we need the algebraic degree of $\omega$ over $K$ to be greater than $d$, and for PolyGenZero to be correct, we also need the algebraic degree of $\omega$ over $K$ to be less than $d$. In other words, we need $\omega$ to have algebraic degree exactly $d$ over $K$, and such choice of $\omega$ is only possible when $d$ divides $[\mathbb{F} : K]$. The intuition on the construction of this 2-KIOS gadget is detailed in Section 4.2.

**A tight compression algorithm.** The masked multiplication of two order $d$ encodings should remain an order $d$ encoding, but the computation of the polynomial product of two polynomials $\mathbf{a}, \mathbf{b}$ of degree $d-1$ yields a polynomial $\mathbf{z}$ of degree $2d-1$. The compression algorithm proposed in [GJR18, GPRV21] entails a loss of a factor 2 on the number of tolerated probes in the (region) probing security of the multiplication gadget. We define a folding algorithm that achieves the conversion of order $2d-1$ encoding into order $d$ encoding, and such that each of its intermediate values are $K$-linear. As a consequence, it can be composed without refresh and without tightness loss at the end of a multiplication gadget. Nonetheless, our folding algorithm is a bigger circuit (we left as an interesting open question estimating the count of operations in this algorithm depending on $\omega$ and $K$) than the compression algorithm from [GJR18, GPRV21], which mildly decreases the tolerated probing rate of the adversary.

**Multiplication gadgets with subquadratic randomness and multiplications.**[3] We propose two generic transformations that turn polynomial multiplication algorithms verifying some conditions into RTIK multiplication gadget. The polynomial multiplication algorithm to be transformed defines the subfield $K$, see Definition 5.1. For instance, if $\mathbb{F}$ is the finite field with $p^k$ elements, where $p$ is a prime, then Karatsuba induces the subfield $K = \mathbb{F}_p$.

The class of polynomial multiplication algorithms that are suited to our first transformation is broad: it contains the most common efficient algorithms such

[3] Please note that while we discuss about the asymptotic behaviour of the performances of our multiplication gadgets, their security only falls into our framework for bounded order of masking $d$, for a fixed $\mathbb{F}$.

as Karatsuba algorithm[KO62], all Toom-Cook variants [Too63, Coo66] and the Fast Fourier Transform. Let $\mathcal{M}$ be such a polynomial multiplication algorithm. We assume that $\mathcal{M}$ has identical time complexity and multiplication complexity $T(d)$, where $d$ is the degree of the inputs. The multiplication complexity of the gadget $\widehat{\mathcal{M}}$ is $T(d)$, and the randomness complexity of $\widehat{\mathcal{M}}$ is $T(d)\log T(d)/2$. When the instantiation of our transformation on the FFT is supported, our gadget achieves similar time and randomness complexities as the multiplication gadget from [GPRV21], but does not rely on a non-standard assumption for security, nor extensive precomputations.

The class of polynomial multiplications that are suited to our second transformation is narrower than the first one, but still contains Karatsuba for example. The transformation of a suitable $\mathcal{M}$ is written $\widetilde{\mathcal{M}}$, it has multiplication complexity $T(d)$ and randomness complexity $d\log d$. The transformation $\widetilde{\mathcal{M}}$ offers a different trade-off than $\widehat{\mathcal{M}}$: while both are RTIK, when projected to region-probing security, $\widehat{\mathcal{M}}$ splits into shorter circuits than $\widetilde{\mathcal{M}}$, and therefore has a higher tolerance in probing ratio. On the other hand, the number of random elements used per run in $\widetilde{\mathcal{M}}$ is lower than the number of random elements per run in $\widehat{\mathcal{M}}$.

**Limitations and open questions.**

*Lack of concreteness.* Our contribution mostly stands on the theoretical side. While we give examples of instantiations in Section 6, the concrete evaluation of the algorithms developed in this paper, even on their own, would deserve a thorough investigation that is left for future work. Determining if masking an actual cryptographic algorithm using our techniques can be more efficient than state-of-the-art masked implementation is another interesting open question.

*Range of applications.* Let us consider that when working in $\mathbb{F} = \mathbb{F}_{p^k}$ with $p$ a prime, we have $K = \mathbb{F}_p$. In this regime, we can perform masked multiplication with either transformation applied to Karatsuba, whenever the number $d$ of shares is at most $k$. Whenever there does not exist $\omega$ of degree exactly the desired $d$ (which happens, for example, when the desired $d$ is not a factor of $k$), both our 2-KIOS algorithm and our compression algorithm from Section 4 cannot apply, and we use their respective replacements Algorithm 2 and Algorithm 3 instead.

For example, in the AES field $\mathbb{F}_{256}$, we have $k = 8$, thus our masking scheme tolerates a number of shares $d$ up to 8, with possibility to use our 2-KIOS algorithm and our compression algorithm for $d \in \{2, 4, 8\}$, which may seem restrictive. We notice that [DKR+21] proposes a variant of AES running in $\mathbb{F}_{2^{32}}$, which would extend significantly the range of applications of our masking scheme. An example where this restriction is virtually absent is in the NTRUprime field [BCLV17]. This field is chosen as $\mathbb{F}_{p^q}$, where both $q$ and $p$ are primes, and $q$ is a few hundreds.

Gadget expansion [AIS18, BCP+20, BRTV21, BRT21] (which consists, waving hands, at applying a masking scheme on a circuit several times in a row) could be a solution to lift the upper bound on the number of shares of our masking scheme. We leave open the question to improve the upper bound on the number of shares of our masking scheme.

Finally, we believe that the new techniques and algorithms given in this paper may find other applications. In particular, with the standardization of lattice-based schemes [DKL+18, BDK+18], it is an interesting open question to see to what extent our techniques apply to rings. In particular for Kyber [BDK+18], there has been some interest in comparison gadgets (or equality-testing gadget)[DVBV22, CGMZ21, BC22] and it would be interesting to see if such gadget can be constructed from the building blocks presented in this paper.

## 2 Preliminaries

### 2.1 Notations

Throughout the paper, $\mathbb{F}$ denotes a field and $K \subset \mathbb{F}$ a subfield of $\mathbb{F}$. We write $\mathbb{F}_q$ the finite field with $q$ elements. Field elements are written in lower-case letters, vectors are written in bold lower-case letters and matrices are written in bold upper-case letters. Unless stated otherwise, vectors are column vectors, and for a vector $\mathbf{x}$, we denote $\mathbf{x}^T$ its transpose. We write $\odot$ the component-wise product of two vectors. We write $\mathbb{F}_d[X]$ the set of polynomials in $X$ of degree at most $d$ that have coefficients in $\mathbb{F}$. Abusing notation, we identify a polynomial to its list of coefficients and treat an element $\mathbf{a} \in \mathbb{F}^d$ as an element of $\mathbb{F}_{d-1}[X]$, e.g by writing $\mathbf{a}(\omega)$ the evaluation of the polynomial whose coefficients list is $\mathbf{a}$ in a field element $\omega$. We write $\pi_K(\omega)$ the minimal polynomial of $\omega$ over $K$, and we write $\deg_K(\omega)$ the degree of $\pi_K(\omega)$. For a distribution $D$, we do not have notation conventions whether the support of $D$ is a scalar or a vector, but rather rely on context. The notation $[n]$ shall denote the set $\{1, \ldots, n]$. For random variables $X, Y$, we write $X \perp Y$ when $X$ is independent of $Y$.

A circuit is a directed acyclic graph whose vertices are operations, and each edge is an intermediate value, intermediate variable or wire. We shall call internal randomness of a circuit the list $\boldsymbol{\rho}$ of the elements sampled by random gates in the circuit. This way, every intermediate value of the circuit is a deterministic function of its input and the internal randomness of the circuit. For a set of intermediate values $P = (p_1, \ldots, p_n)$ of a circuit with input $\chi$ and internal randomness $\boldsymbol{\rho}$, we write $P(\chi, \boldsymbol{\rho}) = (p_1(\chi, \boldsymbol{\rho}), \ldots p_n(\chi, \boldsymbol{\rho}))$. When $\boldsymbol{\rho}$ is not in the argument of $P$, we shall write $P(\chi)$ the random variable $P(\chi, \boldsymbol{\rho})$ for a uniformly random $\boldsymbol{\rho}$. We assume throughout the paper that the secret information manipulated by a circuit is a deterministic function of its input and internal randomness. For a circuit $\mathcal{C}$, we shall write $|\mathcal{C}|$ the number of intermediate variable of $\mathcal{C}$.

### 2.2 Masking

**Encodings** For a vector $\mathbf{v} \in (\mathbb{F}\backslash\{0\})^d$, a $\mathbf{v}$-linear sharing of an element $x \in \mathbb{F}$ is a vector $\mathbf{x}$ satisfying $\mathbf{v}^T\mathbf{x} = x$. Arithmetic masking is a particular case of $\mathbf{v}$-linear sharing, where $\mathbf{v} = (1 \ \ldots \ 1)$. For $\omega$ an element of $\mathbb{F}$, we let $\boldsymbol{\omega}_d = (\omega^i)_{0 \leqslant i \leqslant d-1}$. We say that a vector $\mathbf{x} \in \mathbb{F}^d$ is an $\boldsymbol{\omega}_d$-encoding of a field element $x \in \mathbb{F}$ when $\boldsymbol{\omega}^T\mathbf{x} = x$ (or equivalently $\mathbf{x}(\omega) = x$), which is also a particular case of linear

sharing. For $x \in \mathbb{F}$, the set of $\mathbf{v}$-encodings of $x$ is $H_x^{\mathbf{v}} = \{\mathbf{x} \in \mathbb{F}^d,\ \mathbf{v}^T\mathbf{x} = x\}$ and can be seen both as an affine hyperplane (with the convention $H_0^{\mathbf{v}} = H^{\mathbf{v}}$). We shall omit the supscript $\mathbf{v}$ when it is clear from context, and we notice that $H_x^{\boldsymbol{\omega}_d}$ can also be seen as the set of degree $d$ polynomials $\mathbf{x}$ such that $\mathbf{x}(\omega) = x$. We define $\mathcal{U}_{\mathbf{v}}(x)$ to be the uniform distribution over $H_x^{\mathbf{v}}$, and extend it coordinate-wise when applied on multiple entries.

We call an addition gadget (respectively a multiplication gadget) with respect to $\boldsymbol{\omega}_d$-encodings a circuit that takes as input two $\boldsymbol{\omega}_d$-encodings $\mathbf{a}, \mathbf{b}$ and returns an $\boldsymbol{\omega}_d$-encoding of $\boldsymbol{\omega}_d^T\mathbf{a} + \boldsymbol{\omega}_d^T\mathbf{b}$ (respectively $\boldsymbol{\omega}_d^T\mathbf{a} \cdot \boldsymbol{\omega}_d^T\mathbf{b}$). A correct refresh gadget with respect to $\boldsymbol{\omega}_d$-encodings is a circuit that takes as input an $\boldsymbol{\omega}_d$-encoding and returns an $\boldsymbol{\omega}_d$-encoding of the same secret. In general, for a gate $g$ in a circuit $\mathcal{C}$, we say that $G$ is a correct $\boldsymbol{\omega}_d$-encoding gadget for $g$ when $G$ takes as input $\boldsymbol{\omega}_d$-encodings of the sensitive inputs of $g$, and returns $\boldsymbol{\omega}_d$-encodings of the sensitive outputs of $g$.

### Security properties

**Definition 2.1 ($t$-probing security game).** *Let $n, t \geqslant 1$, $\mathcal{C}$ be a circuit inducing a set of intermediate variables $\mathcal{W}$, $\chi$ be the input random variable of $\mathcal{C}$ and $x_1, \ldots, x_n$ be secret variables. A $t$-probing adversary $\mathcal{A}$ on $(\mathcal{C}, \chi)$ against $x_1, \ldots, x_n$ plays the following game :*

1. *$\mathcal{A}$ chooses a set of probes $P \subset \mathcal{W}$ with $|P| \leqslant t$*
2. *The challenger runs $\mathcal{C}(\chi)$ and sends $P(\chi)$ to $\mathcal{A}$*
3. *$\mathcal{A}$ returns $(y_1, \ldots, y_n)$. He wins if $(y_1, \ldots, y_n) = (x_1, \ldots, x_n)$.*

A circuit $\mathcal{C}$ for which there is no unbounded adversary $\mathcal{A}$, playing the $t$-probing security game with respect to secrets $x_1, \ldots, x_n$, that has an advantage against an adversary who skips steps 1) and 2) is called $t$-probing secure. In the context of masking, the input $\chi$ of $\mathcal{C}$ contains encodings of the secret inputs, and the decoding of these are then hidden secrets of this circuit.

**Definition 2.2 ($r$-region probing security game).** *Let $n \geqslant 1$, $0 < r < 1$, $\mathcal{C}$ be a circuit, $\mathcal{C}_1, \ldots, \mathcal{C}_m$ be subcircuits of $\mathcal{C}$ such that $(\mathcal{C}_1, \ldots, \mathcal{C}_m)$ is a covering of $\mathcal{C}$, $\mathcal{W}_1, \ldots, \mathcal{W}_m$ be the induced sets of intermediate variables, $\chi$ be the input random variable of $\mathcal{C}$ and $x_1, \ldots, x_n$ be secrets. A $r$-region probing adversary against $(\mathcal{C}, \chi)$ with regions $\mathcal{C}_1, \ldots, \mathcal{C}_m$ plays the following game :*

1. *$\mathcal{A}$ chooses $m$ sets of probes $(P_i \subset \mathcal{W}_i)_{i \leqslant m}$ with $|P_i| \leqslant \lceil r|\mathcal{W}_i| \rceil$*
2. *The challenger runs $\mathcal{C}(\chi)$ and sends $(P_i(\chi))_{i \leqslant m}$ to $\mathcal{A}$*
3. *$\mathcal{A}$ returns $(y_1, \ldots, y_n)$. He wins if $(y_1, \ldots, y_n) = (x_1, \ldots, x_n)$.*

With identical input and secrets to hide, any $t$-probing secure circuit $\mathcal{C}$ is trivially $t/|\mathcal{C}|$-region probing secure. Conversely, if a circuit is $r$-region probing secure with $m = 1$, it is $\lfloor r|\mathcal{C}| \rfloor$-probing secure.

**Definition 2.3 ($t$-input-output separation).** *Let $\mathbf{v} \in (\mathbb{F} \backslash \{0\})^d$. A refresh gadget $G^R$ is called $t$-input-output separative when for any $\mathbf{x}, \mathbf{y}$ with $\mathbf{y} = G^R(\mathbf{x})$,*

*we have that* $\mathbf{y}$ *follows* $\mathcal{U}(\mathbf{v}^T\mathbf{x})$ *and for any set of intermediate values* $\mathcal{W}$ *with* $|\mathcal{W}| \leqslant t$*, we have that there exists a two-stage simulator* $\mathcal{S}_{G^R,\mathcal{W}} = (\mathcal{S}^1_{G^R,\mathcal{W}}, \mathcal{S}^2_{G^R,\mathcal{W}})$ *with the following properties.*

1. *The first one* $\mathcal{S}^1_{G^R,\mathcal{W}}$*, returns two sets of indices* $\mathcal{I}, \mathcal{J} \subset [d]$ *such that* $|\mathcal{I}|, |\mathcal{J}| \leqslant |\mathcal{W}|$*.*
2. *The second one* $\mathcal{S}^2_{G^R,\mathcal{W}}$*, ran on input* $\mathbf{x}_{|\mathcal{I}}, \mathbf{y}_{|\mathcal{J}}$*, returns an output identically distributed as* $\mathcal{W}(\mathbf{x}, \mathbf{r})$*, where* $\mathbf{r}$ *is the internal randomness of* $G^R$*,* $\mathbf{x}_{|\mathcal{I}}$ *is* $\mathbf{x}$ *restricted to the coordinates that appear in* $\mathcal{I}$ *and similarly for* $\mathbf{y}_{|\mathcal{J}}$*.*

The following composition Theorem claims that if a circuit $\mathcal{C}$ is split into $t$-probing secure subcircuits separated by $t$-IOS refresh gadgets, then the whole circuit is $r$-region probing secure for some ratio $r$. The statement of the Theorem deals with so-called standard masked compilers of arithmetic circuits, but similar proof techniques could aim for a more general claim.

**Theorem 2.4 (Composition Theorem, adapted from Theorem 1 [GPRV21]).** *Let* $\mathcal{C}$ *be an arithmetic circuit. If* $G^+$ *is a* $t^+$*-probing secure addition gadget,* $G^\times$ *is a* $t^\times$*-probing secure multiplication gadget and* $G^R$ *is a* $t^R$*-IOS refresh gadget, then the circuit* $\widehat{\mathcal{C}}$ *taking as input an encoding of the input of* $\mathcal{C}$ *obtained by replacing addition gates with* $G^+$*, multiplication gates by* $G^\times$ *and applying a refresh gadget* $G^R$ *to any input of an operation gadget is* $r$*-region probing secure, with*

$$r = \max_{t \leqslant t^R} \min \left( \frac{t^+ - 3t}{|G^+|}, \frac{t^\times - 3t}{|G^\times|}, \frac{t}{|G^R|} \right).$$

## 3 Polynomial masking probing security toolbox

In this section, we provide a series of tools - from elementary ones to masked compiler composition Theorem. The main result of this section is the latter composition Theorem 3.16. Towards this Theorem, we introduce some security notions such as RTC Definition 3.8, KIOS Definition 3.14 and RTIK Definition 3.15. These definitions are based on a somewhat new formalism of threshold probing security and probe sets introduced in Section 3.1, and are hinted by an algebraic approach to probing security introduced in Section 3.2.

### 3.1 Probabilistic approach to probing security

The $t$-probing security game, as defined in Definition 2.1, is usually translated as the simulatability of the leakage. In this subsection, we redefine $t$-probing security (as well as $r$-region probing security) in a formalism that relies on distributions rather than simulation. These probabilistic versions are a first step towards a reduction of probing security to algebra. The point of this subsection is to give formal tools, which allows to highlight the key arguments in our probing security proofs, and arguably makes those probing security proofs clearer. We first define a binary relation written $\leqslant$ on sets of probes, from which we derive that various elementary operations on sets of probes at least preserve the information learnt by the adversary.

**Definition 3.1 (Partial order of probe sets).** *Let $P, Q$ be two sets of probes on a circuit $\mathcal{C}$, taking as input a random variable $\chi$ and manipulating secret variables $x_1, \ldots, x_n$. Let $\phi$ be the sensitive part of $\chi$, i.e there exists a deterministic function $F$ such that $(x_1, \ldots, x_n) = F(\phi)$. We say that $Q$ contains more information than $P$, and we write $P \leqslant Q$, when*

$$(\phi|(P(\chi), Q(\chi))) = (\phi|Q(\chi)).$$

When $P \leqslant Q$, intuitively, all the sensitive information on the input $\chi$ of $\mathcal{C}$ carried by $P$ is also carried by $Q$. The binary relation $\leqslant$ verifies reflexivity and transitivity, but not antisymmetry. Since antisymmetry is irrelevant for our purposes, we chose to write this binary relation as a partial order relation.

We now provide an illustration of elementary operations on a set of probes $P_1$. The obtained sets $P_2, P_3$ are such that $P_3 \geqslant P_2 \geqslant P_1$, thus $P_3 \geqslant P_1$. Consider some circuit $\mathcal{C}$ that takes as input a sharing order $d = 2$ and two arithmetic encodings $(x_0, x_1)$, $(y_0, y_1)$. Assume that the secrets manipulated by the circuit are $x = x_0 + x_1$ and $y = y_0 + y_1$. Then, the sensitive part of the input is $\phi = (x_0, x_1, y_0, y_1)$. Now consider that a 3-probing adversary choses the set of probes $P_1 = (2x_0, y_0, x_0 + y_0)$. The first operation that we can do on this set of probes while preserving the information it contains is to remove the constant factor 2: with $P_2 = (x_0, y_0, x_0 + y_0)$, we have $P_2 \geqslant P_1$. Second, we can remove the redundancy : if the adversary learns $x_0$ and $y_0$, he might as well compute $x_0 + y_0$ himself. With $P_3 = (x_0, y_0)$, we have $P_3 \geqslant P_2$. Adding extra relations to a set of probes also yields that it contains more information. For instance if $Q_1 = (x_0 + y_0)$, then $Q_2 = (x_0, y_0)$ is such that $Q_2 \geqslant Q_1$. Examples of practical use can be found in the proofs of Proposition 4.1 and Theorem 5.3.

We now proceed to define $t$-probing security for masked circuit from a probability perspective.

**Definition 3.2 ($t$-probing security of linear-masked circuits, convenient version).** *Let $\mathbf{v} \in (\mathbb{F}\backslash\{0\})^d$, $\mathcal{C}$ be a circuit taking as input $\mathbf{v}$-encodings $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and $\mathcal{W}$ be the set of intermediate variables of $\mathcal{C}$. Then $\mathcal{C}$ is $t$-probing secure when $\forall P \subset \mathcal{W}$ with $|P| \leqslant t$, we have*

$$(\mathbf{v}^T\mathbf{x}_1, \ldots, \mathbf{v}^T\mathbf{x}_n) \perp P(\mathbf{x}_1, \ldots, \mathbf{x}_n).$$

**Definition 3.3 ($r$-region-probing security of linear-masked circuits, convenient version).** *Let $\mathbf{v} \in (\mathbb{F}\backslash\{0\})^d$, $0 < r < 1$, $\mathcal{C}$ be a circuit, $\mathcal{C}_1, \ldots, \mathcal{C}_m$ be subcircuits of $\mathcal{C}$ such that $(\mathcal{C}_1, \ldots, \mathcal{C}_m)$ is a covering of $\mathcal{C}$, $\mathcal{W}_1, \ldots, \mathcal{W}_m$ be the induced sets of intermediate variables of the subcircuits. We let $\mathbf{x}_1, \ldots \mathbf{x}_n$ be the input $\mathbf{v}$-encodings of $\mathcal{C}$. Then $\mathcal{C}$ is $r$-region-probing secure when $\forall P = (P_1, \ldots, P_m)$, with $P_i \subset \mathcal{W}_i$ and $|P_i| \leqslant \lceil r|\mathcal{C}_i| \rceil$, we have*

$$(\mathbf{v}^T\mathbf{x}_1, \ldots, \mathbf{v}^T\mathbf{x}_n) \perp P(\mathbf{x}_1, \ldots, \mathbf{x}_n).$$

In both definitions, the information learnt by the adversary (i.e $P(\mathbf{x}_1, \ldots, \mathbf{x}_n)$) is therefore independent of the secrets hidden in the circuit (i.e each sensitive

entry $x_i = \mathbf{v}^T\mathbf{x}_i$). Since there is information-theoretically no information learnt by the adversary by probing, if a masked circuit verifies one of the definitions above, it also verifies the corresponding usual game-based definition. The following Proposition links the relation $\leqslant$ to region probing security.

**Proposition 3.4.** *Let $\mathbf{v} \in (\mathbb{F}\backslash\{0\})^d$, $0 < r < 1$, $\mathcal{C}$ be a circuit taking as input $\mathbf{v}$-encodings $\mathbf{x}_1, \ldots, \mathbf{x}_n$. Assume that there exists a covering set of subcircuits $\mathcal{C}_1, \ldots, \mathcal{C}_m$, inducing sets of intermediate variables $(\mathcal{W}_1, \ldots, \mathcal{W}_m)$, such that for all set of probes $P = (P_1, \ldots, P_m)$ with $|P_i| \leqslant \lceil r|\mathcal{W}_i|\rceil$ for all $i \leqslant m$, there exists a set of probes $Q = (Q_1, \ldots, Q_m)$ such that*

1. $\forall\ i \leqslant m,\ P_i \leqslant Q_i$
2. $(\mathbf{v}^T\mathbf{x}_1, \ldots, \mathbf{v}^T\mathbf{x}_n) \perp Q(\mathbf{x}_1, \ldots, \mathbf{x}_n)$.

*Then $\mathcal{C}$ is $r$-region probing secure.*

*Proof.* Let $0 < r < 1$, $\mathcal{C}$ be a circuit taking as input $\mathbf{v}$-encodings $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and $\mathcal{C}_1, \ldots, \mathcal{C}_m$ be a covering set of subcircuits of $\mathcal{C}$. We take a set of probes $P = (P_1, \ldots, P_m)$ with $|P_i| \leqslant \lceil r|\mathcal{W}_i|\rceil$ for all $i \leqslant m$. Since $P$ verifies the requirements of the Proposition, we take $Q = (Q_1 \ldots, Q_m)$ verifying the conditions above. We have

$$(\mathbf{v}^T\mathbf{x}_1\ \ldots\ \mathbf{v}^T\mathbf{x}_n) = \big((\mathbf{v}^T\mathbf{x}_1\ \ldots\ \mathbf{v}^T\mathbf{x}_n)|Q(\mathbf{x}_1, \ldots, \mathbf{x}_n)\big) \quad (1)$$

$$= \big((\mathbf{v}^T\mathbf{x}_1\ \ldots\ \mathbf{v}^T\mathbf{x}_n)|(P(\mathbf{x}_1, \ldots, \mathbf{x}_n), Q(\mathbf{x}_1, \ldots, \mathbf{x}_n)\big), \quad (2)$$

where Equation (1) follows from independence and Equation (2) follows from the hypothesis of the proposition. It follows that $((\mathbf{v}^T\mathbf{x}_1\ \ldots\ \mathbf{v}^T\mathbf{x}_n)|P(\mathbf{x}_1, \ldots, \mathbf{x}_n)) = (\mathbf{v}^T\mathbf{x}_1\ \ldots\ \mathbf{v}^T\mathbf{x}_n)$ thus $\mathcal{C}$ is $r$-region-probing secure.

Using the correspondence between $t$-probing security and $r$-region probing security with $m = 1$, the Proposition above then implies that if for any set $P$ of $t$ probes on a circuit $\mathcal{C}$, there exists a set $Q$ with $P \leqslant Q$ and $Q$ is independent of the secrets, then the latter circuit is $\mathcal{C}$ is $t$-probing secure.

### 3.2 Probing security of $K$-linear circuits

This subsection contains three technical results Lemmas 3.5 to 3.7 that are building blocks for proving $t$-probing security of $\boldsymbol{\omega}$-masked circuits.

From a high level, the first Lemma 3.5 claims that when $\deg_K(\omega) \geqslant d$, the vector $\boldsymbol{\omega}_d$ is never in the span of $\ell < d$ vectors over $K$. The intuition of the connexion between this statement and probing security is as follows : This statement says, roughly speaking, that the probes are linearly independent of the decoding operation, and this statement is in turn used to prove the probabilistic independence between probes and secret in Lemma 3.6. The third Lemma 3.7 is only here for technical reasons, and it is very similar to Lemma 3.6.

To illustrate, consider a $t$-probing adversary against some circuit $\mathcal{C}$, taking as input a uniform $\boldsymbol{\omega}_d$-encoding of the secret. We assume that the adversary

has no prior knowledge on the secret $a = \boldsymbol{\omega}_d^T \mathbf{a}$ manipulated by $\mathcal{C}$, hence from the adversary's perspective, before probing, $\mathbf{a}$ is distributed uniformly over $\mathbb{F}^d$. Now, say we can force every intermediate value of our circuit $\mathcal{C}$ to be $K$-linear in $\mathbf{a}$. Then, when the adversary probes $t < d$ linearly independent equations on the encoding $\mathbf{a}$, he receives some values $\mathbf{v} \in \mathbb{F}^t$ of the form $\mathbf{v} = \mathbf{P}\mathbf{a}$ where $\mathbf{P} \in K^{t \times d}$. The probability that the secret is some $a' \in \mathbb{F}$, from the adversary's perspective, is then proportional to the number of solutions to the equations $\mathbf{v} = \mathbf{P}\mathbf{a}$ and $\boldsymbol{\omega}_d^T \mathbf{a} = a'$. When $\deg_K(\omega) \geqslant d$ is satisfied, Lemma 3.5 tells us that $\boldsymbol{\omega}_d \notin \text{Span } \mathbf{P}^T$, from which follows that the set of solutions to the latter equations is an affine subspace of dimension $d - t - 1$, of cardinality $|\mathbb{F}|^{d-t-1}|$ no matter what $a' \in \mathbb{F}$ is. In other words, the secret in the adversary's view is distributed uniformly random, therefore the adversary did not learn anything by probing, which is $t$-probing security.

We prove (in a slightly more general fashion) the result sketched above in Lemma 3.6. This Lemma is central in our framework : every security notion introduced in the next subsection points to it, and it is the last step in the proof of our main Theorem 3.16. We believe that Lemma 3.6 can be also very convenient for constructing other $\omega$-masking gadgets. For example, constructing efficient equality test gadgets, square gadgets, inverse gadget etc is still an open question as of today, and we believe that this result is a step towards these objectives.

**Lemma 3.5.** *Let $\mathbb{F}$ be a finite field, $K$ be a subfield of $\mathbb{F}$, $\mathbf{P} \in K^{\ell \times d}$ such that* rank $\mathbf{P} = t$ *and $\omega \in \mathbb{F}$. If* $\deg_K(\omega) \geqslant d$ *and $t < d$, then*

$$\text{rank} \begin{bmatrix} \mathbf{P} \\ \boldsymbol{\omega}_d^T \end{bmatrix} = t + 1.$$

*Proof.* Let us assume for one moment that rank $\begin{bmatrix} \mathbf{P} \\ \boldsymbol{\omega}_d^T \end{bmatrix} = t$, i.e $\boldsymbol{\omega}_d \in \text{Span } \mathbf{M}^T$. This means that there exists $t$ coefficients $\lambda_i \in \mathbb{F}^t$ such that $\mathbf{P}^T \boldsymbol{\lambda} = \boldsymbol{\omega}_d$. Now, since $t < d$, there exists vectors $\mathbf{p}_{t+1}, \ldots, \mathbf{p}_d$ with coefficients in $K$ that complete $\mathbf{P}$ into an invertible matrix. We let $\mathbf{Q}$ be its inverse, and we write $\mathbf{q}$ the last row of $\mathbf{Q}$. We have

$$\left[\mathbf{P}^T | \mathbf{p}_{t+1} | \ldots | \mathbf{p}_d\right] \begin{bmatrix} \boldsymbol{\lambda} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \boldsymbol{\omega}_d$$

$$\begin{bmatrix} \boldsymbol{\lambda} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \mathbf{Q}\boldsymbol{\omega}_d.$$

Taking the last row in the last equality, we get $\mathbf{q}^T \boldsymbol{\omega}_d = 0$. In other words, the polynomial with coefficients $\mathbf{q}$ cancels $\omega$ and has degree at most $d$, which is a contradiction with $\deg_K(\omega) \geqslant d$, and the claim follows.

**Lemma 3.6.** *Let $d$ be an order of masking, $\mathcal{C}$ be a circuit taking as input a uniform $\boldsymbol{\omega}_d$-encoding $\mathbf{x}$. If all the intermediate variables $p$ of $\mathcal{C}$ are of the form $p(\mathbf{x}) = \mathbf{p}^T\mathbf{x}$ for some vector $\mathbf{p} \in K^d$, then $\mathcal{C}$ is $d-1$-probing secure.*

*Proof.* Let $\mathcal{A}$ be a $d-1$-probing adversary against $\mathcal{C}$, probing a set $P$ of intermediate values of $\mathcal{C}$. Let $\chi$ be the distribution of the secret input $x$, inducing by uniformity a distribution $\bar{\chi}(\mathbf{x}) = \frac{1}{|\mathbb{F}|^{d-1}}\chi(\omega^T\mathbf{x})$. There exists a matrix $\mathbf{P} \in K^{(d-1)\times d}$ such that $P(\mathbf{x}) = \mathbf{P}\mathbf{x}$. We assume without loss of generality that $\mathbf{P}$ is full-rank. For $x \in \mathbb{F}, \mathbf{v} \in \mathbb{F}^{d-1}$, we have

$$\mathbb{P}(\boldsymbol{\omega}_d^T\mathbf{x} = x \cap P(\mathbf{x}) = \mathbf{v}) = \mathbb{P}(\boldsymbol{\omega}_d^T\mathbf{x} = x \cap \mathbf{P}\mathbf{x} = \mathbf{v}) \tag{3}$$

$$= \bar{\chi}\left(\ker\begin{bmatrix}\mathbf{P}\\ \boldsymbol{\omega}_d^T\end{bmatrix} + \mathbf{x}^*\right) \tag{4}$$

$$= \bar{\chi}\left(\mathbf{x}^*\right) = \frac{1}{|\mathbb{F}|^{d-1}}\chi(x) \tag{5}$$

$$= \mathbb{P}(P(\mathbf{x}) = \mathbf{v})) \cdot \mathbb{P}(\boldsymbol{\omega}_d^T\mathbf{x} = x), \tag{6}$$

where Equation (3) is the hypothesis of the Lemma, Equation (4) holds for some solution $\mathbf{x}^*$ to the equation $\begin{bmatrix}\mathbf{P}\\ \boldsymbol{\omega}_d^T\end{bmatrix} = \begin{bmatrix}\mathbf{v}\\ x\end{bmatrix}$, Equation (5) follows from Lemma 3.5 which implies that the matrix $\begin{bmatrix}\mathbf{P}\\ \boldsymbol{\omega}_d^T\end{bmatrix}$ is of rank $d$, therefore its kernel is 0, and Equation (6) holds because $\mathbb{P}(P(\mathbf{x}) = \mathbf{v}) = \mathbb{P}(\mathbf{x} \in D) = \frac{1}{|\mathbb{F}|^{d-1}}\sum_{y\in\mathbb{F}}\chi(y) = \frac{1}{|\mathbb{F}|^{d-1}}$, where $D$ is a one-dimensional affine space.

**Lemma 3.7.** *Let $d$ be an order of masking, $n, m$ positive integers with $n \geqslant m$, $\mathbf{M} \in K^{n\times m}$ be a tall full-rank matrix over $K$, and $\mathcal{C}$ be a circuit taking as input a uniform $\mathbf{M}\boldsymbol{\omega}_d$-encoding $\mathbf{x}$. If all the intermediate variables $p$ of $\mathcal{C}$ are of the form $p(\mathbf{x}) = \mathbf{p}^T\mathbf{x}$ for some vector $\mathbf{p} \in K^m$, then $\mathcal{C}$ is $d-1$-probing secure.*

*Proof.* We let $\mathbf{P}\mathbf{x}$ be the probes of the adversary, for some full-rank matrix $\mathbf{P} \in K^{(d-1)\times m}$. We want to prove that those probes on the vector $\mathbf{x}$ are independent of $(\mathbf{M}\boldsymbol{\omega}_d)^T\mathbf{x}$. Following the proof of Lemma 3.6, we only need to prove that $\begin{bmatrix}\mathbf{P}\\ \boldsymbol{\omega}_d^T\mathbf{M}^T\end{bmatrix}$ is full-rank to conclude that the distribution of $(\mathbf{M}\boldsymbol{\omega}_d)^T\mathbf{x}$ does not depend on the value of $\mathbf{P}\mathbf{x}$. Let us assume that the latter matrix is not full-rank. Since we assume without loss of generality that $\mathbf{P}$ is full-rank, this means that there exists $\boldsymbol{\lambda} \in \mathbb{F}^{d-1}$ such that $\mathbf{P}^T\boldsymbol{\lambda} = \mathbf{M}\boldsymbol{\omega}$. We multiply by the left on both sides with $\mathbf{M}^T$, then we multiply on both sides by the left by $(\mathbf{M}^T\mathbf{M})^{-1}$, and we obtain $\mathbf{P}'\boldsymbol{\lambda} = \boldsymbol{\omega}_d$, where $\mathbf{P}' = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{P}$ is a $d$ by $d-1$ matrix. We can now use Lemma 3.5 to conclude that such $\boldsymbol{\lambda}$ does not exist, hence the $d-1$-probing security.

### 3.3 Refreshing $\boldsymbol{\omega}_d$-encodings and composition of gadgets

For our own technical purposes (e.g the proof of Theorem 5.3) and for showing its close relation with Definition 3.14, we redefine the Input-Output Separation property introduced in [GPRV21]. The property Reducible-To-Coordinates

(RTC) for generators of $\mathbf{v}$-encodings of 0 is closely connected to the $\ell$-free property defined in the proof of Theorem 2 from [GPRV21] (from which the authors deduce the IOS property), thus we redefine the IOS property based on this RTC property. We prove that our new definition encompasses the original one, and show a template on how to build an IOS refresh gadget Algorithm 2 and Proposition 3.12 from an RTC generator of encodings of 0.

**Definition 3.8.** *(Reducible-To-Coordinates) Let $\mathbf{v} \in (\mathbb{F}\backslash\{0\})^d$, $t$ be an integer and $R$ be a gadget taking as input a dimension $d$, and returning a uniform $\mathbf{v}$-encoding $\mathbf{r}$ of 0. We say that $R$ is Reducible-To-Coordinates (RTC) when the distribution of $\mathbf{r}$ is uniform conditioned on $\mathbf{v}^T\mathbf{r} = 0$ and for every set of $t$ probes $P$ on $R$, there exists two sets of probes $Q_1, Q_2$ such that*

1. *$|Q_1| \leqslant t$*
2. *$(Q_1, Q_2) \leqslant P$*
3. *Every probe in $Q_1$ is a coordinate of $\mathbf{r}$*
4. *The distributions $Q_2$ and $(\mathbf{r}|Q_1)$ are independent*

Notice that in the definition above, the binary relation $\leqslant$ is taken with respect to the secret $r_0, \ldots, r_{d-1}$, i.e all the coordinates of the fresh vector $\mathbf{r}$, where for $t$-probing security of masked circuits we take the secrets to be the decoding of the masked inputs.

**Proposition 3.9.** *Algorithm 1 is RTC with $\mathbf{v} = (1, \ldots, 1)$.*

The Proposition above is a mild generalization of Theorem 2 from [GPRV21]. They prove that the refresh gadget obtained by adding coordinate-wise an encoding of 0 generated using ArithGenZero is IOS when $d$ is a power-of-two. We adapt their result from IOS to RTC, and extend it to any $d \geqslant 1$ by considering the refresh gadget from Appendix C [BCPZ16].

*Proof.* Uniformity. If $d = 1$, then the algorithm returns (0) and it is indeed a uniform arithmetic encoding of 0. If $d = 2$, then the algorithm returns $(r, -r)$ for some uniformly random $r$, which is also distributed uniformly among the arithmetic encodings of 0.

For $d \geqslant 3$, we assume by induction that the uniformity holds for every order less than $d - 1$. In particular, $\mathbf{r}_L = (r_0, \ldots, r_{\lfloor d/2 \rfloor - 1})$ and $\mathbf{r}_R = (r_{\lfloor d/2 \rfloor}, \ldots, r_{d-1})$ are uniform independent encodings of 0 of respective orders $\lfloor d/2 \rfloor$ and $\lceil d/2 \rceil$. Let $\mathbf{x} \in \mathbb{F}^d$. We let $\mathbf{t}_L = \mathbf{r}_L + \mathbf{s}$ and $\mathbf{t}_R = \mathbf{r}_R + \mathbf{s}$ and $u = \sum_{i=0}^{d/2-1} s_i$.

If $d$ is even, then $\mathbf{t}_L$ is distributed uniformly random among the arithmetic encodings of length $d/2$ of $u$. We have

$$\begin{aligned}
\mathbb{P}(\mathbf{t} = \mathbf{x}) &= \mathbb{P}(\mathbf{t}_L = \mathbf{x}_L \cap \mathbf{t}_R = \mathbf{x}_R) \\
&= \mathbb{P}(u = \sum_{i=0}^{d/2-1} (x_L)_i \cap \mathbf{t}_L = \mathbf{x}_L \cap \mathbf{t}_R = \mathbf{x}_R) \\
&= \mathbb{P}(u = \sum_{i=0}^{d/2-1} (x_L)_i \cap \mathbf{r}_L = \mathbf{x}_L - u \cap \mathbf{r}_R = \mathbf{x}_R + u)
\end{aligned}$$

First, we rule out the case $\sum_{i=0}^{d/2-1}(x_L)_i \neq -\sum_{i=0}^{d/2-1}(x_R)_i$. On one hand we have $\sum y_i = \sum(y_L)_i + \sum(y_R)_i = \sum(r_L)_i + \sum(r_R)_i = 0$, and on the other hand $\sum(y_L)_i + \sum(y_R)_i = \sum(x_L)_i + \sum(x_R)_i \neq 0$, therefore this event has probability 0. Otherwise, $\sum_{i=0}^{d/2-1}(x_L)_i = -\sum_{i=0}^{d/2-1}(x_R)_i$, hence $\mathbf{x}_L - u$ is in the domain of $\mathbf{r}_L$ and $\mathbf{x}_L - u$ is in the domain of $\mathbf{r}_R$. The random variables $u, \mathbf{r}_L, \mathbf{r}_R$ are uniform over their respective domains, mutually independent, hence $\mathbb{P}(\mathbf{t} = \mathbf{x})$ is constant uniform over the set of $\mathbf{x}$ such that $\sum x_i = 0$.

RTC. If $d = 1$, then $t = 0$ hence $Q_1 = Q_2 = \varnothing$, and 1) 2) 3) 4) are trivially verified. If $d = 2$, either $t = 0$ and 1) 2) 3) 4) are trivially verified, or $t = 1$. The one probe can only be $r$ or $-r$, hence $Q_1 = (r), Q_2 = \varnothing$ and 1) 2) 3) 4) are verified.

If $d \geqslant 3$, we assume by induction that ArithGenZero is RTC for all $3 \leqslant i \leqslant d-1$. We let $P$ be a set of probes with $|P| = t \leqslant d-1$, and split this set of probes into $(P_L, P_R, P_P)$, with respectively $P_L$ in the first recursive call and $|P_L| = t_L$, $P_R$ in the second recursive call and $|P_R| = t_R$ and $P_P$ with $|P_P| = t_P$ in the post-processing layer. We first deal with $P_P$, and more precisely we split $P_P$ into subsets $P_P^i$ for each $i \in [\![d/2]\!]$ as follows : $P_P^i$ contains the probes taken from the variables that are together in the $i$th step of the loop:

$$t_i = r_i + s_i \tag{7}$$

$$t_{\lfloor d/2 \rfloor + i} = r_{\lfloor d/2 \rfloor + i} - s_i. \tag{8}$$

For each of these $P_P^i$, we create a set $Q_P^i$, so as to have $Q_P^i \geqslant P_P^i$ and the probes in $Q_P^i$ are only coordinates of $\mathbf{t}$ and $\mathbf{r}$, except when the $s_i$ gives away no information. Explicitly, unless when $P_P^i = \{s_i\}$, we set $Q_P^i = P_P^i$, and replace $s_i$ with a variable among $\{t_i, t_{i+d/2}, r_i, r_{i+d/2}\}$ such that $s_i$ can be deduced from $Q_P^i$. When $P_P^i = \{s_i\}$, we set $Q_P^i = \{s_i\}$. Finally, we create $Q_P, P'_L, P'_R$ as follows : $Q_P$ is the concatenation of all the $Q_P^i$'s, $P'_L = P_L$, $P'_R = P_R$, and we move the probes of $Q_P$ of the form $r_i$ to $P'_L$ and $r_{i+d/2}$ to $P'_R$. Notice that for some integers $k_L, k_R$ such that $k_L + k_R \leqslant t_P$, we have $|Q_P| = t_P - k_L - k_R$, $P'_L = t_L + k_L$ and $P'_R = t_R + k_R$.

We then use the induction hypothesis on $P'_L$ and $P'_R$ and we obtain $Q_L^1, Q_L^2$ satisfying

1 $|Q_L^1| \leqslant t_L + k_L$,
2 $(Q_L^1, Q_L^2) \leqslant P'_L$,
3 Every probe in $Q_L^1$ is a coordinate of $r_L$,
4 The distributions $Q_L^2$ and $(\mathbf{r}_L | Q_L^1)$ are independent,

and similarly for $(Q_R^1, Q_R^2)$.

We now construct two sets of probes $Q_1, Q_2$ from the sets $Q_L^1, Q_L^2, Q_R^1, Q_R^2, Q_P$, and show that they verify 1) 2) 3) and 4). First, the sets of probes $Q_L^2, Q_R^2$ are added to $Q_2$. The probes in $Q_P$ that are coordinates of $\mathbf{t}$ are added to $Q_1$. Only remains probes that are coordinates of $\mathbf{r}$ and probes of the form $s_i$. For each probe of the form $s_i$, there exists two options. Either $r_i \in Q_L^1$ or $r_{i+d/2} \in Q_R^1$, in which case we add the $t_i$ and/or the $t_{i+d/2}$ that can be deduced to $Q_1$. Else,

we add $s_i$ to $Q_2$. The probes that are coordinates of $\mathbf{r}$ are added to $Q_2$, with one exception. When $r_i \in Q_L^1$, $r_{i+d/2} \in Q_R^1$ and $t_i \in Q_P$, then $t_{i+d/2}$ is added to $Q_1$(and similarly when $r_i \in Q_L^1$, $r_{i+d/2} \in Q_R^1$ and $t_{i+d/2} \in Q_P$, then $t_i$ is added to $Q_1$).

We now prove that $Q_1, Q_2$ verify the conditions 1) 2) 3) and 4) so Algorithm 1 is RTC. First, we count the number of probes in $Q_1$. These probes are either i) transferred directly from $Q_P$, ii) or computed from the knowledge of $s_i$ and some $r_i$, or iii) computed from $r_i, r_{i+d/2}$ and $t_{i+d/2}$. We write $k_S$ the number of probes in $Q_P$ that are not coordinates of $\mathbf{t}$. The number of probes that are added during i) is $t_P - k_L - k_R - k_S$. The number of probes that are added during ii) is bounded by $k_S$. The number of probes that are added during iii) is bounded by $\min(Q_L^1, Q_R^1)$. Thus we have $|Q_1| \leqslant t_P \leqslant |P|$. Second, $(Q_1, Q_2)$ are constructed so as to fulfil 2). Again by construction the probes in $Q_1$ are of the form $t_i$, and finally we carefully constructed $Q_2$ so it verifies 4), which completes the proof.

**Algorithm 1** ArithGenZero, adapted from Appendix C [BCPZ16]

**Require:** Masking order $d$
**Ensure:** $\mathbf{t} \in \mathbb{F}^d$ such that $\sum r_i = 0$

**if** $d = 1$ **then**
  **return** 0
**end if**
**if** $d = 2$ **then**
  $r \leftarrow \mathbb{F}$
  **return** $(-r, r)$
**end if**
$(r_0, \ldots, r_{\lfloor d/2 \rfloor - 1}) = \text{ArithGenZero}(\lfloor d/2 \rfloor)$
$(r_{\lfloor d/2 \rfloor}, \ldots, r_{d-1}) = \text{ArithGenZero}(\lceil d/2 \rceil)$
**for** $i = 0$ to $\lfloor d/2 \rfloor - 1$ **do**
  $s_i \leftarrow \mathbb{F}$
  $t_i = r_i + s_i$
  $t_{\lfloor d/2 \rfloor + i} = r_{\lfloor d/2 \rfloor + i} - s_i$
**end for**
**return t**

**Definition 3.10.** *(Input-Output Separative) Let $\mathbf{v} \in (\mathbb{F} \backslash \{0\})^d$, $t$ be an integer and $G$ be a gadget taking as input a $\mathbf{v}$-encoding $\mathbf{x}$, and returning an encoding $\mathbf{y}$ of the same secret as $\mathbf{x}$. We say that $G$ is $t$-IOS when the distribution of $\mathbf{y}$ is uniform conditioned on $\mathbf{v}^T\mathbf{y} = \mathbf{v}^T\mathbf{x}$ and for every set of $t$ probes $P$ on $G$, there exists three sets of probes $Q_x, Q_y, Q_2$ such that*

1. $|Q_x| \leqslant t$, $|Q_y| \leqslant t$
2. $(Q_x, Q_y, Q_2) \leqslant P$
3. *Every probe in $Q_x$ is a coordinate of $\mathbf{x}$ and every probe in $Q_y$ is a coordinate of $\mathbf{y}$*

*4. The distributions $Q_2$ and $((\mathbf{x}, \mathbf{y})|(Q_x, Q_y))$ are independent*

**Proposition 3.11.** *Let $\mathbf{v} \in (\mathbb{F}\backslash\{0\})^d$, $t$ be an integer and $G$ be a gadget taking as input a $\mathbf{v}$-encoding $\mathbf{x}$, and returning an encoding $\mathbf{y}$ of the same secret as $\mathbf{x}$. If $G$ is $t$-IOS according to Definition 3.10, then it is also $t$-IOS according to Definition 2.3.*

*Proof.* Let $G$ be a $t$-IOS gadget for Definition 3.10. First, the output distribution of $G$ is a uniform $\mathbf{v}$-encoding of $\mathbf{v}^T\mathbf{x}$, hence we only need to prove the existence of the simulator.

Let $P$ be a set of probes on $G$. There exists $(Q_x, Q_y, Q_2)$ that satisfy the conditions of Definition 3.10. From 3), the probes in $Q_x, Q_y$ define two sets of indices $\mathcal{I}, \mathcal{J}$, such that every probe in $Q_x$ is some $x_i$ for $i \in \mathcal{I}$ and every probe in $Q_y$ is some $y_j$ for $j \in \mathcal{J}$. From 1), both of these sets are such that $|\mathcal{I}| \leqslant t$ and $|\mathcal{J}| \leqslant t$. These sets are therefore valid outputs for the first simulator. From 2), the distribution of $P(\mathbf{x}, \mathbf{y})$ is determined by the distribution of $Q_x(\mathbf{x}), Q_y(\mathbf{y})$ and $Q_2(\mathbf{x}, \mathbf{y})$. From 4), $Q_2$ is independent of $((\mathbf{x}, \mathbf{y})|(Q_x, Q_y))$ (here $((\mathbf{x}, \mathbf{y})|(Q_x, Q_y))$ is the distribution of the remaining unknown coordinates of $\mathbf{x}$ and $\mathbf{y}$). Therefore, one way the second simulator can perfectly simulate the distribution of the probes is to first pick a uniform $\mathbf{y}'$ such that $y'_j = y_j$ for all $j \in \mathcal{J}$, then pick $\mathbf{x}'$ so that $\mathbf{x}'$ encodes the same element as $\mathbf{y}'$ and $x'_i = x_i$ for all $i \in \mathcal{I}$, and finally return a sample from the distribution $P(\mathbf{x}', \mathbf{y}')$.

**Algorithm 2** IOS refresh template

**Require:** Masking order $d$, $\mathbf{v} \in (\mathbb{F}\backslash\{0\})^d$, RTC generator of arithmetic encodings of 0 $R$, $\mathbf{v}$-encoding $\mathbf{x}$
**Ensure:** $\mathbf{y} \in \mathbb{F}^d$ such that $\mathbf{v}^T\mathbf{y} = \mathbf{v}^T\mathbf{x}$

1: $\mathbf{r} = R(d)$
2: **for** $i = 0$ to $d - 1$ **do**
3: $\quad s_i = v_i^{-1} r_i$
4: **end for**
5: $\mathbf{y} = \mathbf{x} + \mathbf{s}$
6: return $\mathbf{y}$

**Proposition 3.12.** *If $R$ is an RTC generator of arithmetic encodings of 0, then the refresh gadget obtained by instantiating Algorithm 2 with $R$ is an IOS refresh gadget for $\mathbf{v}$-encodings.*

*Proof.* Let $P$ be a set of $t$ probes on Algorithm 2 instantiated with $R$. These probes are either in $R$ or coordinates of $\mathbf{x}$, or coordinates of $\mathbf{y}$. We split $P$ into those three sets of probes $P_R, P_x, P_y$, and we have $|P_R| + |P_x| + |P_y| = t$. Because $R$ is assumed RTC, there exists $Q_1, Q_2$ such that

1. $|Q_1| \leqslant |P_R|$

2. $(Q_1, Q_2) \leqslant P_R$
3. Every probe in $Q_1$ is a coordinate of $\mathbf{r}$
4. The distributions $Q_2$ and $(\mathbf{r}|Q_1)$ are independent

We construct $(Q'_x, Q'_y, Q_3)$ that verify the conditions of Definition 3.10 as follows: for each probe of the form $r_i$ in $Q_1$, we add $x_i$ to $Q'_x$. We add every probe from $P_x$ to $Q'_x$. Similarly, we construct $Q'_y$ as the merge of $P_y$ and the probes $y_i$ for each $r_i$ in $Q_1$. Notice that we can remove $Q_1$ from the set of probes as they are now redundant with $(Q'_x, Q'_y)$. We set $Q_3 = Q_2$. We have 1) $|Q'_x| \leqslant |P_x| + |Q_1| \leqslant t$ and $|Q'_y| \leqslant |P_y| + |Q_1| \leqslant t$, 2) holds since we only used elementary operations on sets of probes as detailed in the early section Definition 3.1, 3) holds by construction and 4) holds under the RTC of $R$, which completes the proof.

We now move on to the definitions that exploit the algebraic structure of $\mathbb{F}/K$, starting with the translation of the Reducible-To-Coordinate property to the Reducible-To-K-Linear property. Notice that although RTK seems more appropriate to our techniques, we use the RTC property in the proof of Theorem 5.3 as the RTK property is insufficient in this case.

**Definition 3.13.** *(Reducible-To-K-Linear) Let $\omega \in \mathbb{F}$ and $K$ be a subfield of $\mathbb{F}$. Consider a gadget $R$ taking as input a dimension $d$ and returning an $\boldsymbol{\omega}_d$-encoding $\mathbf{r}$ of 0. Let $\alpha > 0$ be the slack factor of $R$. We say that $R$ is $\alpha$-Reducible-To-K-Linear (RTK) when the output distribution of $R$ is a uniform $\boldsymbol{\omega}_d$-sharing of 0, and for any set of independent probes $P$ on $R$ with $|P| = t < d$, there exists sets of probes $Q_1, Q_2$ such that*

*1) $|Q_1| \leqslant \alpha t$.*
*2) $(Q_1, Q_2) \leqslant P$*
*3) Every probe in $Q_1$ is $K$-linear in $\mathbf{r}$.*
*4) The distributions $Q_2$ and $(\mathbf{r}|Q_1)$ are independent.*

Notice that with this definition, if $R$ is RTC with respect to $\boldsymbol{\omega}_d$, then $R$ is 1-RTK. We now define the security notion achieved by the $\boldsymbol{\omega}_d$-encoding refresh gadget obtained by adding coordinate-wise a fresh $\boldsymbol{\omega}_d$-encoding of 0 to the input. The intuition why the KIOS security notion for refresh gadget brings composition security is similar to the one for IOS refresh gadgets. If we have $\mathbf{y} = \mathbf{r} + \mathbf{x}$, where $\mathbf{x}$ is some input $\boldsymbol{\omega}_d$-encoding and $\mathbf{r}$ is generated using an $\alpha$-RTK generator of encodings of 0, then we can reduce the probes in the $\alpha$-RTK to $K$-linear probes on $\mathbf{r}$, given by some matrix $\mathbf{P}$. In the next reduction step, we give to the adversary $\mathbf{Px}$ and $\mathbf{Py}$, which are still both $K$-linear. We can then remove the probes on $\mathbf{r}$ as they are redundant, and that way we achieve separation between $\mathbf{x}$ and $\mathbf{y}$.

**Definition 3.14.** *($K$-Input-Output Separative) Let $\omega \in \mathbb{F}$, $K$ be a subfield of $\mathbb{F}$, $\alpha > 0$ and $G$ be a gadget taking as input an $\boldsymbol{\omega}_d$-encoding $\mathbf{x}$, and returning an $\boldsymbol{\omega}_d$-encoding $\mathbf{y}$ of the same secret as $\mathbf{x}$. We say that $G$ is $K$-Input-Output Separative (KIOS) when the distribution of $\mathbf{y}$ is uniform conditioned on $\mathbf{y}(\omega) = \mathbf{x}(\omega)$ and for every set of $t$ probes $P$ on $G$, there exists three sets of probes $Q_x, Q_y, Q_2$ such that*

1. $|Q_x| \leqslant \alpha t, |Q_y| \leqslant \alpha t$
2. $(Q_x, Q_y, Q_2) \leqslant P$
3. *Every probe in* $Q_x$ *is* $K$*-linear in* $\mathbf{x}$*, and every probe in* $Q_y$ *is* $K$*-linear in* $\mathbf{y}$
4. *The distributions* $Q_2$ *and* $((\mathbf{x}, \mathbf{y})|(Q_x, Q_y))$ *are independent*

Finally, we plug together all the ideas of the section to define the RTIK property, and show how we can use this stronger notion of $t$-probing security for KIOS composition.

**Definition 3.15 (Reducible-To-Independent-K-Linear (RTIK)).** *Let* $\mathcal{C}$ *be a circuit taking as input* $n$ *uniform and independently generated*[4] $\boldsymbol{\omega}_d$*-encodings* $\mathbf{x}_1, \ldots, \mathbf{x}_n$*. We say that* $\mathcal{C}$ *is RTIK when for all set of probes* $P$*, there exists* $\mathbf{M}_i\boldsymbol{\omega}_d$*-encodings* $\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+m}$ *for some tall full-rank matrices* $\mathbf{M}_i$ *over* $K$ *and a set of probes* $Q = (Q_i)_{i \leqslant n+m}$ *such that*

1. *Any set of vectors* $(\mathbf{y}_i)_{i \leqslant n+m}$ *with* $\mathbf{y}_i$ *strict subset of* $\mathbf{x}_i$*, the* $(\mathbf{y}_i)_{i \leqslant n}$*'s are mutually independent*
2. *For all* $i \leqslant n + m$*,* $|Q_i| \leqslant |P|$
3. *For all* $i \leqslant n + m$*, every probe* $q \in Q_i$ *is of the form* $\mathbf{q}^T\mathbf{x}_i$ *for some* $q \in K^d$

Rephrasing (and simplifying) the definition above: an $\boldsymbol{\omega}_d$-masked circuit is said RTIK when any set of probes $P$ can be reduced to a set of probes $Q$ in which every probe is $K$-linear in a single $\boldsymbol{\omega}_d$-masked entry. Given this reduction, the straight-forward naive composition of RTIK circuits is then also RTIK. Since this definition directly falls into the requirements of Lemma 3.6, we can directly claim that when $\deg_K(\omega) \leqslant d$, any RTK circuit is $d-1$-probing secure, thus the composition of RTIK circuits is also $d-1$-probing secure.

We finally state in the Theorem below that placing a refresh in between RTIK circuits achieves region-probing security. The idea behind this composition Theorem is very similar to the intuition detailed in [GPRV21] on IOS composition. The basic idea is that when $C_2$ takes as input the output of some circuit $C_1$, one applies a KIOS refresh gadget on each input encoding of $C_2$. In the reduction, using the KIOS property, the leakage of the refresh is transferred to $K$-linear probes on $C_1$ and $C_2$. The leakage from the two subcircuits are then independent, and from the RTIK property, those leakages are $K$-linear, and Lemma 3.6 yields the region probing security.

**Theorem 3.16 (KIOS Composition Theorem, adapted from GPRV).** *Let* $N, d$ *be positive integers,* $\omega \in \mathbb{F}$ *such that* $\deg_K(\omega) \geqslant d$*,* $\mathcal{C}$ *be a circuit and* $(g_i)_{i \leqslant N}$ *be the list of gates of* $\mathcal{C}$*. Assume that for all* $i \leqslant N$*, there exists a correct* $\boldsymbol{\omega}_d$*-encoding gadget* $G_i$ *for* $g_i$*. Let* $G^R$ *be a correct refresh gadget with respect to* $\boldsymbol{\omega}_d$*-encodings. If the following properties hold :*

1. $\forall i \leqslant N$*,* $G_i$ *is RTIK*
2. $G^R$ *is* $\alpha$*-KIOS*

[4] Meaning that for any $(\mathbf{y}_i)_{i \in [n]}$ strict subsets $\mathbf{y}_i \subset \mathbf{x}_i$ of coordinates of the $\mathbf{x}_i$, the $(\mathbf{y}_i)_{i \in [n]}$ are uniform and mutually independent.

*the circuit $\mathcal{C}'$ obtained by replacing every gate $g_i$ by $G_i$, and applying $G^R$ to all inputs of the $G_i$'s is $r$-region-probing-secure, with $r \leqslant \frac{d-1}{\max_{j \leqslant N'} |\mathcal{C}_j|}$ and the $\mathcal{C}_j$'s are defined in the proof.*

*Proof.* We arrange the number $N'$ of refresh gadgets in $\mathcal{C}'$ into a list of $G_j^R$'s. Let us consider any set of probes $P = ((P_i)_{i \leqslant N}, (P_j^R)_{j \leqslant N'})$ on $\mathcal{C}'$, where the probes from $P_i$ are in gadget $G_i$, and the probes in $P_j^R$ are in $G_j^R$.

By using the $\alpha$-KIOS property of $G^R$, we obtain a set of probes $P' = (P'_i)_{i \leqslant N}$, with $P'_i$ on $G_i$ and $P' \geqslant P$. In other words, the probes from $P'$ are exclusively in the gadgets $G_i$, and due to the uniformity of $G^R$, we have that the probes in different gadgets are mutually independent.

We now use the RTIK property of the $G_i$'s, and we obtain a set of probes $Q = (Q_j)_{j \leqslant N'+M}$, with $Q \geqslant P'$, and $Q_j$ contains $K$-linear probes on either a $\boldsymbol{\omega}_d$-encoding $\mathbf{x}_j$ of the circuit, or a $\mathbf{M}_i\boldsymbol{\omega}_d$-encoding of the circuit for some tall matrix $\mathbf{M}_i$ with coefficients in $K$. Notice that thanks to 1. in Definition 3.15, the $Q_j$'s are still mutually independent.

Now, for all $j \leqslant N'$, we define a subcircuit $\mathcal{C}_j$ containing all the wires that would end up in $Q_j$ via the reduction from $P$ to $Q$. These subcircuits form a covering of $\mathcal{C}'$, hence we simply need to chose $r \leqslant \frac{d-1}{\max_{j \leqslant N'} |\mathcal{C}_j|}$, so that non of these subcircuits contains more than $d-1$ probes, after which Lemmas 3.6 and 3.7 complete the proof.

## 4 Elementary gadgets for polynomial masking

This section contains two polynomial masking building blocks. Both algorithms rely on a severe restriction on $d$ and $\deg_K(\omega)$: For security, we need $d \leqslant \deg_K(\omega)$ and for correctness, we need $d \geqslant \deg_K(\omega)$. In other words, we need $d$ to divide $[F : K]$, so that there exists an $\omega$ that satisfies the condition. The reason why we add the restriction $d \geqslant \deg_K(\omega)$ for correctness is that we will exploit the minimal polynomial $\pi_\omega$ of $\omega$, in ways that are detailed in the subsections below.

### 4.1 Folding gadget

This subsection is dedicated to a folding gadget that exploits the algebraic structure brought by $\boldsymbol{\omega}_d$-encodings. Folding gadgets are those that on input some $\boldsymbol{\omega}_{d_1}$-encoding $\mathbf{x}$ return an $\boldsymbol{\omega}_{d_2}$-encoding $\mathbf{y}$ of the same secret, where $d_1 \leqslant d_2$. Since we only need $(d_1, d_2) = (2d-1, d)$, we shall particularize to these specific values in the following. We first recall the so-called NaiveFold algorithm, as used in [GJR18, GPRV21]. This folding algorithm does not require any extra condition to be correct, but entails a factor two loss in probe tolerance.

As stated above, one problem with this compression is that in the current state-of-the-art methods for proving probing security, when the adversary probes some $x_i + \omega^d x_{d+i}$, we have to give away both $x_i$ and $x_{d+i}$. This doubles the number of probes of the adversary, hence in the end halves the number of probes tolerated in the region. While our folding matrix described below can tolerate

**Algorithm 3** NaiveFold

**Require:** $\boldsymbol{\omega}_{2d-1}$-encoding $\mathbf{x}$
**Ensure:** $\mathbf{y} \in \mathbb{F}^d$ such that $\mathbf{x}^T\boldsymbol{\omega}_{2d-1} = \mathbf{y}^T\boldsymbol{\omega}_d$

1: **for** $i = 0$ to $d-2$ **do**
2: $\quad y_i = x_i + \omega^d x_{d+i}$
3: **end for**
4: $y_{d-1} = x_{d-1}$
5: return $\mathbf{y}$

up to $d-1$ probes, it also entails more basic operations and therefore yields a mitigated gain in probing ratio.

The intuition of the construction is as follows: we define a full-rank folding matrix $\mathbf{F} \in K^{d\times(2d-1)}$, with coefficients in the subfield $K$, and mapping the $\boldsymbol{\omega}_{2d-1}$-encodings of some $x \in \mathbb{F}$ to the $\boldsymbol{\omega}_d$-encodings of this same $x$. This way, the computation of $\mathbf{y} = \mathbf{F}\mathbf{x}$ is $K$-linear and Lemma 3.6 applies. The existence of this matrix is only guaranteed when $\deg_K(\omega) \leqslant d$, therefore, so we can also use Lemma 3.6, we actually need the equality.

We now proceed to describe how to construct such a matrix, for a given $\omega$ and $d$. Suppose $\deg_K(\omega) = d$. Then, the minimal polynomial $\pi_\omega$ of $\omega$ over $K$ has degree $d$, therefore $\pi = \omega^d - \pi_\omega$ is of degree $d-1$ and is such that $\pi(\omega) = \omega^d$. In general, any $\omega^{d+i}$ for $0 \leqslant i \leqslant d-2$ is a polynomial in $\omega$ with coefficients in $K$ and degree $\leqslant d-1$. Let us therefore write $\boldsymbol{\pi}_i$ the column vector of coefficients of the $i$-th polynomial, for example $\boldsymbol{\pi}_0 = \pi$. One can check that the matrix

$$\mathbf{F} = \left[\mathbf{I}_d\ \boldsymbol{\pi}_0\ \boldsymbol{\pi}_1\ \ldots\ \boldsymbol{\pi}_{d-2}\right]$$

satisfies the equation $\mathbf{F}^T\boldsymbol{\omega}^d = \boldsymbol{\omega}^{2d-1}$. This implies that $\boldsymbol{\omega}_{2d-1}^T\mathbf{x} = \boldsymbol{\omega}_d^T\mathbf{F}\mathbf{x} = \boldsymbol{\omega}_d^T\mathbf{y}$.

Finally, we emphasize on the fact that one should chose $\omega$ so as to minimize the count of operations in the folding process, to in turn minimize the ratio of tolerated probes per gate in the region. The element $\omega$ has to be chosen from a fixed field $\mathbb{F}$, among the elements of given degree $d$ over some fixed subfield $K$ and it seems hard to make a general statement about the sparsity of the matrix $\mathbf{F}$. Nonetheless, in very specific cases, $\mathbf{F}$ can be very sparse. For example, if $K = \mathbb{F}_p$, and $d+1$ is a prime, one can chose $\omega$ to be a primitive $d$-th root of unity. This way, the minimal polynomial of $\omega$ is $1 + X + \cdots + X^d$, and $\omega^{d+1} = 1$. Then, for any $0 \leqslant d-3$, we have $\omega^{d+1+i} = \omega^i$ and $\omega^d = \sum_{i=0}^{d-1}\omega^i$. In this particular setting, the computation of $\mathbf{y} = \mathbf{F}\mathbf{x}$ takes approximately $3d$ operations in $\mathbb{F}$.

### 4.2 Refresh gadgets

In this subsection, we describe a 2-RTK generator of $\boldsymbol{\omega}_d$-encodings of 0 that only uses $d-1$ random field elements. We may recall that we are using the minimal polynomial $\pi_\omega$ of $\omega$, which can only be made possible if $d|[\mathbb{F} : K]$. On top of this condition, we also require that the greatest common divisor of $\omega^d - \pi_\omega$ and $X^d - \omega^d$ is $X - \omega$.

The intuition how Algorithm 4 works is as follows. First, the algorithm samples a uniformly random vector $\mathbf{x} \in \mathbb{F}^{d-1}$. Next, we compute $\mathbf{s} = \pi_\omega \mathbf{x}$, and we obtain a polynomial $\mathbf{s}$ of degree $d + d - 2$. The algorithm then returns $\mathbf{r}$ as the naive fold of $\mathbf{s}$ as described in the subsection above. The correctness is verified by construction: the evaluation of $\mathbf{r}$ in $\omega$ is 0 since $\pi_\omega$ divides $\mathbf{s}$ and the evaluation in $\omega$ is invariant through the naive fold. Remember that as explained in the previous section, the algorithm that takes as input an $\boldsymbol{\omega}_d$-encoding $\mathbf{x}$ and returns $\mathbf{y} = \mathbf{x} + \mathbf{r}$ where $\mathbf{r}$ is generated by such an $\alpha$-RTK generator of encodings of 0 is $\alpha$-KIOS.

**Algorithm 4** PolyGenZero

**Require:** Masking order $d$ with $d = \deg_K(\omega)$
**Ensure:** $\mathbf{r} \in \mathbb{F}^d$ such that $\mathbf{r}^T \boldsymbol{\omega}_d = 0$

1: $\mathbf{x} \leftarrow \mathbb{F}^{d-1}$
2: $\mathbf{s} = \pi_\omega \mathbf{x}$
3: $\mathbf{r} = \mathsf{NaiveFold}(\mathbf{s})$
4: return $\mathbf{r}$

**Proposition 4.1.** *If* $\deg_K(\omega) = d$ *and the greatest common divisor of* $\pi_\omega$ *and* $X^d - \omega^d$ *is* $X - \omega$*, then* PolyGenZero *is* 2*-RTK.*

*Proof.* Correctness : First, since $\mathbf{r} = \mathsf{NaiveFold}(s)$, we have $\mathbf{r}(\omega) = \mathbf{s}(\omega)$. Now since $\pi_\omega(\omega) = 0$, we have $\mathbf{s}(\omega) = \pi_\omega(\omega)\mathbf{x}(\omega) = 0$, which completes the proof of correctness.
Uniformity. One can check that the $\mathsf{NaiveFold}$ algorithm performs a reduction modulo $X^d - \omega^d$. This way, we have $\mathbf{r} = \pi_\omega \mathbf{x} \mod (X^d - \omega^d) = x \cdot (\pi_\omega \mod (X^d - \omega^d)$. If the greatest common divisor of $\omega^d - \pi_\omega$ and $X^d - \omega^d$ is $X - \omega$,[5] then as $\mathbf{x}$ varies across $\mathbb{F}^{d-1}$, $\mathbf{r}$ takes $|\mathbb{F}|^{d-1}$ different values, which completes the proof.

2-RTK: We consider a set of probes $P$ on PolyGenZero. We split the probes into three subsets : A set $P_1$ made of $t_1$ probes that are $K$-linear in $\mathbf{x}$, a set of $t_2$ probes $P_2$ that are coordinates of $\mathbf{s}$ and a set of $t_3$ probes $P_3$ made of probes that are coordinates of $\mathbf{r}$. We define an increasing sequence of sets of probes.

Set of probes 1: $(P_1, P_2, P_3)$, with $t_1 + t_2 + t_3 \leqslant t$. Any set of at most $t$ probes on PolyGenZero is of this form. Since $\pi_\omega$ has coefficients in $K$, $P_1$ is indeed $K$-linear in $\mathbf{x}$.

Set of probes 2: $(P_1', P_3)$, with $|P_1'| = t_1' \leqslant t_1 + t_2$. The set $P_1'$ is the concatenation of $P_1$ and $P_2$, where since $\pi_\omega$ has coefficients in $K$, each coordinate of $\mathbf{s}$ is $K$-linear in $\mathbf{x}$, therefore $P_1'$ is $K$-linear in $\mathbf{x}$.

[5] This condition seems to be always satisfied in finite fields, but we have no rigorous proof of that statement at the moment.

Set of probes 3: $(P_2', P_3)$, with $|P_1'| = t_2' \leqslant t_1'$. We transform the $K$-linear probes $P_1'$ on $\mathbf{x}$ to $K$-linear probes on $\mathbf{s}$. The probes from $P_1'$ are of the form $\mathbf{P}_1\mathbf{x} = \mathbf{v}$ with $\mathbf{P}_1 \in K^{t_1' \times (d-1)}$ and $\mathbf{v} \in \mathbb{F}^{t_1'}$. Multiplication with $\pi_\omega$ is a full-rank $K$-linear operation, therefore there exists a matrix $\mathbf{M} \in K^{(d-1)\times(2d-1)}$ such that $\mathbf{Ms} = \mathbf{x}$, hence the matrix $\mathbf{P}_2 = \mathbf{MP}_1$ yields a set of probes $P_2'$ over $\mathbf{s}$ such that $P_2' \geqslant P_1'$.

Set of probes 4: $(Q_2, P_3)$ with $|Q_2| \leqslant 4(t_1 + t_2)$ and $Q_2$ is $K$-linear in $\mathbf{s}$. We define $\mathbf{P}_L$ (respectively $\mathbf{P}_R$) the first $d-1$ columns of $\mathbf{P}_2$ (respectively the remaining $d$ columns of $\mathbf{P}_2$), and accordingly we define $\mathbf{s}_L, \mathbf{s}_R$. The set $Q_2$ is the concatenation of the probes $\mathbf{P}_L\mathbf{s}_L, \mathbf{P}_R\mathbf{s}_R, \mathbf{P}_L\mathbf{s}_R, \mathbf{P}_R\mathbf{s}_L$. The set $P_2'$ is determined by $\mathbf{P}_L\mathbf{s}_L + \mathbf{P}_R\mathbf{s}_R$ hence $Q_2 \geqslant P_2'$.

Set of probes $5 : Q_3$ with $|Q_3| \leqslant 2(t_1 + t_2) + t_3$. The set $Q_3$ is the concatenation of $P_3$ and $\mathbf{P}_L\mathbf{r}, \mathbf{Q}_L\mathbf{r}$, which is $K$-linear in $\mathbf{r}$ as required. We finish the proof by showing that $(\mathbf{r}|Q_3 \cap Q_2) = (\mathbf{r}|Q_3)$, i.e the extra probes from $Q_2$ do not give any information to the adversary. Due to the uniformity of $\mathbf{x}$, the distribution $((\mathbf{r}, \mathbf{s}_L, \mathbf{s}_R)|Q_3 \cap Q_2)$ is uniform over the set of solutions of

$$\begin{bmatrix} -\mathbf{I} & \mathbf{I} & \omega^{-d}\mathbf{I} \\ \mathbf{Q} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q} \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{s}_L \\ \mathbf{s}_R \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{v}' \\ \mathbf{v}_L \\ \mathbf{v}_R \end{bmatrix},$$

for some probed value vectors $\mathbf{v}', \mathbf{v}_L, \mathbf{v}_R$. In particular, the first row of the left hand side matrix is redundant, hence this matrix induces the same affine subspace of solutions as the matrix

$$\begin{bmatrix} \mathbf{Q} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q} \end{bmatrix}.$$

The latter matrix is block-wise diagonal, hence the distributions of $\mathbf{s}_L, \mathbf{s}_R$ are independent of the distribution of $\mathbf{r}$, which completes the proof.

## 5 Subquadratic multiplication gadgets

In this section, we give two generic transformations that turn a suitable polynomial multiplication algorithm into an RTIK multiplication gadget.

Both transformations entail different trade-offs: $\widehat{\ }$ yields a multiplication gadget that uses more random field elements, but a smaller circuit than $\check{\ }$, hence higher probing ratio eventually, as discussed in Section 6.

### 5.1 Generic GPRV-type Transformation

In this subsection, we show that (almost) any polynomial multiplication algorithm can be turned into a masked multiplication gadget. More precisely, the

polynomial multiplication gadgets that fit our transformation $\widehat{\ }$ are those algorithms that are based on evaluation-interpolation. This definition encompasses Karatsuba's algorithm, all Toom-Cook variants (which contains Karatsuba) and the FFT.

**Definition 5.1 (Evaluation-Interpolation-Based Polynomial Multiplication Algorithms).** *Let $\mathcal{M}$ be an algorithm taking as input two polynomials of degree $d-1$ that returns the product of the two inputs and $K$ a subfield of $\mathbb{F}$. We say that $\mathcal{M}$ is a $K$-Interpolation-Multiplication algorithm ($K$-IM for short) when there exists matrices $\mathbf{M}_1, \mathbf{M}_2$ with coefficients in $K$ such that for any $(\mathbf{a}, \mathbf{b}) \in \mathbb{F}_{d-1}[X]^2$, we have $\mathcal{M}(a, b) = \mathbf{M}_2 \cdot (\mathbf{M}_1\mathbf{a} \odot \mathbf{M}_1\mathbf{b})$.*

The architecture of our transformation applied to the FFT follows the blueprint from [GPRV21], whose security relies on a non-standard ad-hoc assumption. The security of our gadgets on the other hand is theoretically sound as it relies on no assumption, but rather a condition relating the multiplication algorithm $\mathcal{M}$, the order of masking $d$ and to some extent the size of $\mathbb{F}$ (we need $d \leqslant \log |\mathbb{F}|$). More precisely, we need Lemma 3.5 to apply, i.e we need to be able to pick $\omega$ such that $\deg_K(\omega) \geqslant d$ where the field $K$ is derived from $\mathcal{M}$. To be specific, $K$ is defined as the subfield $K$ such that $\mathcal{M}$ is a $K$-IM, as defined in Definition 5.1, meaning that $K$ is the smallest subfield of $\mathbb{F}$ such that the evaluation and interpolation operations induced by $\mathcal{M}$ are $K$-linear.

The transformation of a suitable multiplication algorithm $\mathcal{M}$ taking as input two polynomials $\mathbf{a}, \mathbf{b}$ into a secure multiplication gadget works as follows. Since $\mathcal{M}$ can be split into two phases, namely evaluation and interpolation, our gadget $\widehat{\mathcal{M}}$ starts by computing the evaluation of both polynomial entries $\mathbf{a}, \mathbf{b}$. This step is proven $d-1$-probing secure under Lemma 3.6 which is optimal. Then, $\widehat{\mathcal{M}}$ computes the evaluation $\mathbf{x}$ of the product $\mathbf{ab}$ by multiplying coordinate-wise their evaluations. In the proof, when the adversary probes a coordinate of $\mathbf{x}$, we give him both factors to keep no dependency between $\mathbf{a}$ and $\mathbf{b}$. Before proceeding to interpolation, we need to cut the bilinear dependencies. Splitting the evaluation and interpolation regions is done using the IOS refresh template Algorithm 2, with a suitably chosen $\mathbf{v}$ (that depends on the interpolation of $\mathcal{M}$) and ArithGenZero Algorithm 1. Notice that since the length of $\mathbf{x}$ is $T(d)$ (the multiplication complexity of $\mathcal{M}$), the cost of this refresh in randomness is $T(d) \log T(d)/2$.

The IOS property of the refresh ensures that in the security proof, we can split the probing security of the whole gadget into two (actually three) regions: before and after the refresh step independently (actually the evaluation region can be split further into a region on the first input and a region in the second input). $\widehat{\mathcal{M}}$ now computes the interpolation of the refreshed encoding $\mathbf{y}$, which yields the $2d-1$ coefficients of a polynomial $\mathbf{z}$ encoding $\mathbf{ab}$. Notice that if $\mathbf{a}(\omega) = a$, $\mathbf{b}(\omega) = b$, we want to find a polynomial $\mathbf{c}$ that encodes $ab$, for the same $\omega$ and masking order $d$. To this end, we multiply $\mathbf{z}$ with the folding matrix $\mathbf{F}$ so $\mathbf{c} = \mathbf{Fz}$

has degree $d-1$, and $\mathbf{c}(\omega) = \mathbf{z}(\omega) = \mathbf{a}(\omega)\mathbf{b}(\omega) = ab$, and the algorithm finally returns this $\mathbf{c}$. The construction of the matrix $\mathbf{F}$ is detailed in Section 4.1.[6]

**Algorithm 5** Multiplication gadget $\widehat{\mathcal{M}}(\mathbf{a}, \mathbf{b})$. The algorithm $\mathcal{R}$ on line 4 is Algorithm 2 instantiated with ArithGenZero

**Require:** A $K$-IM $\mathcal{M}$ with matrices $\mathbf{M}_1, \mathbf{M}_2$, folding matrix $\mathbf{F}$ (see Subsection 4.1) and two input encodings $\mathbf{a}, \mathbf{b} \in \mathbb{F}^d$
**Ensure:** $\mathbf{c} \in \mathbb{F}^d$ such that $\boldsymbol{\omega}_d^T \mathbf{a} \cdot \boldsymbol{\omega}_d^T \mathbf{b} = \boldsymbol{\omega}_d^T \mathbf{c}$

1: $\mathbf{a}' = \mathbf{M}_1 \mathbf{a}$ ▷ Evaluation of $\mathbf{a}$
2: $\mathbf{b}' = \mathbf{M}_2 \mathbf{b}$ ▷ Evaluation of $\mathbf{b}$
3: $\mathbf{x}' = \mathbf{a}' \odot \mathbf{b}'$ ▷ Component-wise multiplication of evaluations
4: $\mathbf{y}' = \mathcal{R}(\mathbf{x}', \mathbf{M}_2^T \boldsymbol{\omega}_{2d-1})$ ▷ Refresh
5: $\mathbf{z} = \mathbf{M}_2 \mathbf{y}'$ ▷ Interpolation of the product
6: $\mathbf{c} = \mathbf{F}\mathbf{z}$ ▷ Folding
7: return $\mathbf{c}$

**Theorem 5.2.** *Let $d$ be an order of masking, $K$ be a subfield of $\mathbb{F}$, $\mathcal{M}$ be a $K$-IM and $\omega \in \mathbb{F}$ such that $\deg_K(\omega) = d$. Then, the instantiation of Algorithm 5 with $\mathcal{M}$ is a correct RTIK multiplication gadget.*

*Proof.* **Correctness.** Let $\mathbf{a}, \mathbf{b} \in \mathbb{F}^d$. We have:

$$\boldsymbol{\omega}_d^T \mathcal{M}(\mathbf{a}, \mathbf{b}) = \boldsymbol{\omega}_d^T \mathbf{F}\mathbf{M}_2 \mathcal{R}(\mathbf{M}_1\mathbf{a} \odot \mathbf{M}_1\mathbf{b}, \mathbf{M}_2^T \boldsymbol{\omega}_{2d-1}) \tag{9}$$
$$= \boldsymbol{\omega}_d^T \mathbf{F}\mathbf{M}_2 (\mathbf{M}_1\mathbf{a} \odot \mathbf{M}_1\mathbf{b}) = \boldsymbol{\omega}_d^T \mathbf{F}(\mathbf{a} \cdot \mathbf{b}) \tag{10}$$
$$= \boldsymbol{\omega}_{2d-1}^T \mathbf{a} \cdot \mathbf{b}, \tag{11}$$

where Equation (9) is the definition of $\widehat{\mathcal{M}}(\mathbf{a}, \mathbf{b})$, Equation (10) follows from the correctness of $\mathcal{R}$ and $\mathcal{M}$, and Equation (11) holds since $\mathbf{F}$ is crafted so $\mathbf{F}\boldsymbol{\omega}_{2d-1} = \boldsymbol{\omega}_d$. Therefore $\widehat{\mathcal{M}}$ is a valid multiplication gadget.

**RTIK.** We consider the $\boldsymbol{\omega}_d$-encodings $\mathbf{a}, \mathbf{b}$, the $(\mathbf{F}\mathbf{M}_2)^T\boldsymbol{\omega}_d$-encoding $\mathbf{y}$, and let $P$ be a set of $t$ probes chosen by the adversary. Remind that $\mathbf{a}$ and $\mathbf{b}$ are the two inputs of the algorithm, thus are assumed to follow independently generated distributions of the form $\mathcal{U}_{\boldsymbol{\omega}_d}(H_x)$. Due to the uniformity of $\mathcal{R}$, $\mathbf{y}$ is also generated independently from a distribution of this form, and therefore 1. from Definition 3.15 is verified by $\mathbf{a}, \mathbf{b}, \mathbf{y}$. We now proceed to construct an increasing sequence of probes until we reach $Q$ that satisfies 2. and 3.

Set of probes 1: $P_1 = P = (P_a, P_b, P_x, P_R, P_y, P_z, P_c)$, where the subset of probes $P_X$ is a set of probes that are a function of $X$ for $X \in \{\mathbf{a}, \mathbf{b}, \mathbf{x}', \mathbf{y}', \mathbf{z}, \mathbf{c}\}$, and $P_R$

[6] We assume that the folding matrix exists i.e $d|[\mathbb{F} : K]$. If this condition is not verified, one can still use the NaiveFold at the cost of roughly halving the tolerated probing ratio.

is the subset of probes within the refresh $\mathcal{R}$. This is the set of $t$ probes chosen by the adversary.

Set of probes 2: $P_2 = (P_a, P_b, P'_x, P'_y, P_z, P_c)$. We obtain this set by using the IOS property Definition 3.10 on $\mathcal{R}$, and add the probes on $\mathbf{x}'$ to $P_x$ to obtain $P'_x$, and similarly for $\mathbf{y}'$.

Set of probes 3: $P_3 = (Q_a, Q_b, P'_y, P_z, P_c)$. We obtain this set of probes as follows. Notice that every probe from $P'_x$ is a coordinate of $\mathbf{x}'$. For each of these probes, we add the corresponding coordinate of $\mathbf{a}'$ to $P_a$ and similarly for $\mathbf{b}'$.

Set of probes 4: $Q = (Q_a, Q_b, Q_c)$. We obtain this set of probes by merging $P'_y, P_z, P_c$ into $Q_y$. One can check that $|Q_a|, |Q_b|, |Q_c| \leqslant |P|$, and that all these probes are indeed $K$-linear, which completes the proof.

### 5.2 Generic transformation with linear randomness

In this subsection, we detail a second transformation of $K$-IM into secure masked multiplication gadget. We still need $\deg_K(\omega) \geqslant d$, with equality for improved folding step (c.f Section 4.1). Unfortunately, we could only prove the security of this transformation for $K$-IM that satisfy an extra condition: we require that every intermediate value in the computation of $\mathbf{y} = \mathbf{M}_1\mathbf{x}$ for some $\mathbf{x} \in \mathbb{F}^d$ is a coordinate of $\mathbf{y}$. While this condition seems very restrictive, this nonetheless still includes Karatsuba's algorithm, while excluding the FFT. This construction offers a different trade-off between randomness and probing ratio.

The transformation presented in Section 5.1 yields a multiplication gadget running in the same time $O(T(d))$ as $\mathcal{M}$, and requiring $O(T(d)\log T(d))$ random field elements. The randomness cost of the multiplication comes solely from the use of ArithGenZero on the evaluation vector of the product. Intuitively, it may seem expensive to spend $T(d)\log T(d)/2$ random field elements on refreshing an encoding that masks the product of the two inputs, as those contain only linear entropy. More specifically, the vector $\mathbf{x}$ is computed as $\mathbf{M}_1\mathbf{a} \odot \mathbf{M}_1\mathbf{b}$, which can take at most $|\mathbb{F}|^{2d}$ different values (much less in practice), while the vector $\mathbf{y}$ obtained by using the IOS refresh template with $\mathbf{v} = \mathbf{M}_2^T\boldsymbol{\omega}_{2d-1}$ can take exactly $|\mathbb{F}|^{T(d)-1}$ values. The gadget described in Algorithm 6 takes advantage of this remark to reduce the randomness requirement to linear.

**Quasilinear randomness.**

We present the idea that allows us to cut the bilinear dependencies between $\mathbf{a}, \mathbf{b}$ with quasilinear randomness. In a nutshell, Algorithm 6 computes $\mathbf{a}' = \mathbf{M}_1\mathbf{a}$, $\mathbf{b}' = \mathbf{M}_1\mathbf{b}$ and coordinate-wise multiply them to get the vector $\mathbf{x}'$, just like the transformation in Subsection 5.1. The fork between both algorithms happens at this stage. Waving hands, in the place where we IOS refresh the whole vector $\mathbf{x}'$ in $\widetilde{\mathcal{M}}$, we instead refresh the masks of $\mathbf{a}$ and $\mathbf{b}$. More specifically, Algorithm 6 calls ArithGenZero and samples two $\boldsymbol{\omega}_d$-encodings of 0 $\mathbf{r}_1$ and $\mathbf{r}_2$, computes $\mathbf{r}'_1 = \mathbf{M}_1\mathbf{r}_1$ and $\mathbf{r}'_2 = \mathbf{M}_1\mathbf{r}_2$, and then sets $\mathbf{y}' = \mathbf{x}' + \mathbf{r}'_1 \odot \mathbf{b}' + \mathbf{a}' \odot \mathbf{r}'_2 + \mathbf{r}'_1 \odot \mathbf{r}'_2$. Rearranging the terms, we have $\mathbf{y}' = \mathbf{M}_1(\mathbf{a} + \mathbf{r}_1) \odot \mathbf{M}_1(\mathbf{b} + \mathbf{r}_2)$, where due to the RTC of Algorithm 4, $(\mathbf{a} + \mathbf{r}_1)$ and $(\mathbf{b} + \mathbf{r}_2)$ are IOS refreshed masks of the

inputs. Due to technical reasons detailed in the proof, we can only claim region probing security of this gadget if the probes in the left-multiplication by $\mathbf{M}_1$ are probes on the output of the latter multiplication.

**Algorithm 6** Multiplication gadget $\widetilde{\mathcal{M}}(\mathbf{a}, \mathbf{b})$

**Require:** $\mathbf{a}, \mathbf{b} \in \mathbb{F}^d$
**Ensure:** $\mathbf{c} \in \mathbb{F}^d$ such that $\boldsymbol{\omega}_d^T \mathbf{a} \cdot \boldsymbol{\omega}_d^T \mathbf{b} = \boldsymbol{\omega}_d^T \mathbf{c}$

1: $\mathbf{a}' = \mathbf{M}_1 \mathbf{a}$ ▷ Evaluation of $\mathbf{a}$
2: $\mathbf{b}' = \mathbf{M}_1 \mathbf{b}$ ▷ Evaluation of $\mathbf{b}$
3: $\mathbf{x}' = \mathbf{a}' \odot \mathbf{b}'$ ▷ Share-wise multiplication
4: $\mathbf{r}_1 \leftarrow \text{ArithGenZero}(d)$ ▷ Fresh encoding of 0 to refresh the mask $\mathbf{a}$
5: $\mathbf{r}_2 \leftarrow \text{ArithGenZero}(d)$ ▷ Fresh encoding of 0 to refresh the mask $\mathbf{b}$
6: $\mathbf{r}'_1 = \mathbf{M}_1 \mathbf{r}_1$ ▷ Evaluation of $\mathbf{r}_1$
7: $\mathbf{r}'_2 = \mathbf{M}_1 \mathbf{r}_2$ ▷ Evaluation of $\mathbf{r}_2$
8: $\mathbf{y}' = \mathbf{x}' + \mathbf{r}'_1 \odot \mathbf{b}' + \mathbf{a}' \odot \mathbf{r}'_2 + \mathbf{r}'_1 \odot \mathbf{r}'_2$ ▷ Refresh
9: $\mathbf{z} = \mathbf{M}_2 \mathbf{y}'$ ▷ Interpolation of the product
10: $\mathbf{c} = \mathbf{F}\mathbf{z}$ ▷ Folding
11: return $\mathbf{c}$

**Theorem 5.3.** *Let $d$ be an order of masking, $K$ be a subfield of $\mathbb{F}$, $\mathcal{M}$ be a $K$-IM and $\omega \in \mathbb{F}$ such that $\deg_K(\omega) = d$. Consider the circuit $\mathcal{C}_{\mathbf{M}_1}$ that takes a vector $\mathbf{x} \in \mathbb{F}^d$ and returns $\mathbf{M}_1\mathbf{x}$. Assume that $\mathcal{C}_{\mathbf{M}_1}$ is such that every intermediate value is a coordinate of its output. Then, the instantiation of Algorithm 6 with $\mathcal{M}$ is a correct RTIK multiplication gadget.*

*Proof.* Correctness. From the correctness of PolyGenZero, $\mathbf{r}'_1, \mathbf{r}'_2$ are $\boldsymbol{\omega}_d$ encodings of 0, hence $\mathbf{y}'$ is a $\mathbf{M}_2^T \boldsymbol{\omega}_{2d-1}$-encoding of 0. From the correctness of $\mathcal{M}$, $\mathbf{z}$ is a $\boldsymbol{\omega}_{2d-1}$-encoding of $\mathbf{a}^T \boldsymbol{\omega}_d \cdot \boldsymbol{\omega}_d^T \mathbf{b}$, and finally from the correctness of $\mathbf{F}$, $\mathbf{c}$ is a $\boldsymbol{\omega}_d$-encoding of $\boldsymbol{\omega}_d \cdot \boldsymbol{\omega}_d^T \mathbf{b}$.

**Reduction to $K$-linear independent probes.** Let $\mathcal{A}$ be a $t$-probing adversary against the multiplication gadget in Algorithm 6 probing a set $P = \{p_i,\ 1 \leqslant i \leqslant t\}$ of intermediate values in the circuit. In all generality, we split the set of probes into subsets $P_a, P_b, P_x, P_{r_1}, P_{r'_1}, P_{r_2}, P_{r'_2}, P_y, P_z, P_c$, where $P_a$ is the set of probes on Line 1, $P_b$ is the set of probes on Line 2, $P_x$ is the set of probes on Line 3, $P_{r_1}$ is the set of probes on Line 4, $P_{r_2}$ is the set of probes on Line 5, $P_{r'_1}$ is the set of probes on Line 6, $P_{r'_2}$ is the set of probes on Line 7, $P_y$ is the set of probes on Line 8, $P_z$ is the set of probes on Line 9 and $P_c$ is the set of probes on Line 10. We construct an increasing sequence of set of probes reaching $Q = (Q_a, Q_b, Q_y)$, which satisfies the conditions of Definition 3.15.

**Set of probes 1:** This is the set of probes

$$P_1 = (P_a, P_b, P_x, P_{r_1}, P_{r_2}, P_{r'_1}, P_{r'_2}, P_y, P_z, P_c)$$

chosen by the adversary.

**Set of probes 2:** This set of probes is

$$P_2 = (P_a, P_b, P_x, P'_{r_1}, P'_{r_2}, P_{r'_1}, P_{r'_2}, P_y, P_z, P_c),$$

where both $P'_{r_1}, P'_{r_2}$ are defined as follows. We use the RTC property of ArithGenZero to obtain sets of output-coordinate probes $P'_{r_1}, P'_{r_2}$ with $P'_{r_1} \geqslant P_{r_1}$ and $P'_{r_2} \geqslant P_{r_2}$.

**Set of probes 3:** This set of probes is

$$P_3 = (P_a, P_b, P_x, Q_{r'_1}, Q_{r'_2}, P_y, P_z, P_c),$$

where $Q_{r'_1}$ is the merge of $P'_{r_1}, P_{r'_1}$, and $Q_{r'_2}$ is the merge of $P'_{r_2}$ and $P_{r'_2}$. Notice that we have $\mathbf{r}'_1 = \mathcal{C}_{\mathbf{M}_1}(r_1)$, therefore $Q_{r'_1}$ only contains probes that are coordinates of $\mathbf{r}'_1$. Same thing for $Q_{r'_2}$. This step is made possible by the assumption on $\mathcal{C}_{\mathbf{M}_1}$.

**Set of probes 4:** This is the set of probes

$$P_4 = (P'_a, P'_b, P'_x, Q'_{r'_1}, Q'_{r'_2}, P'_y, P_z, P_c),$$

where the sets of probes $P'_a, P'_b, P'_x, Q'_{r'_1}, Q'_{r'_2}, P'_y$ are defined as follows. The computation on Line 8 is done coordinate-wise, meaning that any probe in $P_y$ is a deterministic function of the coefficient in a single common position $i$ of the vectors $\mathbf{a}', \mathbf{b}', \mathbf{x}', \mathbf{r}'_1, \mathbf{r}'_2, \mathbf{y}'$. For every probe in this line, we give away to the adversary the value of each of the vectors $\mathbf{a}', \mathbf{b}', \mathbf{x}', \mathbf{r}'_1, \mathbf{r}'_2, \mathbf{y}'$ in the corresponding position $i$, and update the sets of probes $P'_a, P'_b, P'_x, P'_{r_1}, P'_{r_2}, Q'_{r'_1}, Q'_{r'_2}, P'_y$ accordingly. Notice that $Q'_{r'_1}$ is still only made of probes that are coordinates of $\mathbf{r}'_1$, and similarly $Q'_{r'_2}$ is only made of probes that are coordinates of $\mathbf{r}'_2$.

**Set of probes 5:** This set of probes is

$$P_5 = (Q_a, Q_b, Q_x, Q_y, P_z, P_c),$$

where the sets $Q_a, Q_b, Q_x, Q_y$ are defined as follows. Every probe in $Q'_{r'_1}, Q'_{r'_2}$ is given by a coordinate in some position $i$ of $\mathbf{r}'_1$ or $\mathbf{r}'_2$. For each of these probes, we give away the $i$-th coordinate of $\mathbf{a}', \mathbf{b}', \mathbf{x}', \mathbf{y}'$, and update $Q_a, Q_b, Q_x, Q_y$ accordingly. To finish, we claim that with similar arguments as in the last step of the proof of Proposition 4.1, the information on the vectors $\mathbf{a}, \mathbf{b}, \mathbf{y}$ contained in the adversary's probes on $\mathbf{r}_1, \mathbf{r}_2$ is already contained in $Q_a, Q_b, Q_x, Q_y$, and we can therefore remove them from the set of probes.

**Set of probes 5:** This set of probes is

$$P_6 = (Q'_a, Q'_b, Q_y, P_z, P_c)$$

, where $Q'_a$ and $Q'_b$ are defined as follows. Every probe from $Q_x$ is by construction a coordinate of $\mathbf{x}'$. Those coordinates are of the form $x_i = a'_i b'_i$. For each of these probes, we give away both $a'_i$, $b'_i$ to the adversary, and remove $Q_x$ from the set of probes for redundancy.

**Set of probes 6:** This set of probes is $Q = (Q'_a, Q'_b, Q'_y)$, where $Q_y$ is the merge of $Q_y, P_z$ and $P_c$. We claim that through the sequence of probes, we achieve that $Q'_a$ is only made of $K$-linear probes on $\mathbf{a}$, $Q'_b$ is only made of $K$-linear probes on $\mathbf{b}$, and $Q'_y$ is only made of $K$-linear probes on $\mathbf{y}'$.

As a conclusion, we showed that if $\mathcal{A}$ probes a set $P$, there exists a set of probes $Q = (Q'_a, Q'_b, Q'_y)$ that verify the conditions 2. and 3. from Definition 3.15. Remains to prove that $Q_y$ is independent of $Q_a, Q_b$. We have $\mathbf{y}' = \mathbf{x}' + \mathbf{r}'_1 \odot \mathbf{b}' + \mathbf{a}' \odot \mathbf{r}'_2 + \mathbf{r}'_1 \odot \mathbf{r}'_2 = \mathbf{M}_1(\mathbf{a} + \mathbf{r}_1) \odot \mathbf{M}_1(\mathbf{b} + \mathbf{r}_2)$. From the uniformity of $\mathcal{R}$ in Lines 4 and 5, $(\mathbf{a} + \mathbf{r}_1)$ follows $\mathcal{U}_{\boldsymbol{\omega}_d}(\boldsymbol{\omega}_d^T \mathbf{a})$ and $(\mathbf{b} + \mathbf{r}_2)$ follows $\mathcal{U}_{\boldsymbol{\omega}_d}(\boldsymbol{\omega}_d^T \mathbf{b})$, thus $\mathbf{y}$ is independent of $\mathbf{a}, \mathbf{b}$ and the result follows.

**Note on the probing ratio.** The parameter $r$ for the region probing security of this gadget is given by $d-1$ divided by the size of the biggest subcircuit. There are 3 subcircuits defined from $Q_a, Q_b, Q_y$. The subcircuit of $Q_a$ is the one that contains all the probes from $P$ that ended up in $Q'_a$ through the sequence of sets of probes detailed above. In the end, the set of probes $Q_a$ gathers probes from the subcircuit computing Lines 1 and 3 to 8. We obtain that this subcircuit has size $10T(d)\log T(d) + 2d\log 2$. By symmetry, the subcircuit of $\mathbf{b}$ has the same size and probe tolerance. The last subcircuit's size depends on $F(d)$, that is, the number of gates in the computation of $\mathbf{z} = \mathbf{M}_2\mathbf{y}'$. This subcircuit has size $9T(d)\log T(d) + F(d) + 2d\log 2$.

## 6 Instantiations and performances

In this section, we give examples of simple and practical instantiations of the algorithms described in Sections 4 and 5. The $K$-IM that we use to instantiate our multiplication gadgets is Karatsuba's algorithm. The reason for this choice is threefold: the algorithm is very simple, its range of competitiveness against other multiplication algorithms is within the range of the number of shares in masking and its subfield $K$ is optimal. It seems that - depending on the metric - our multiplication algorithms are competitive with ISW multiplication around $d = 8$ and on, but more experiments have to be run to assess this statement.

We'll write Karatsuba's algorithm for polynomial multiplication $\mathcal{M}$. Again, Karatsuba's algorithm is a good candidate for our transformation as it is competitively fast for multiplying polynomials whose degree is in the masking range, and it can be used in any characteristic. We describe below the matrices $\mathbf{M}_1, \mathbf{M}_2$ that make Karatsuba a $\mathbb{F}_p$-IM, where $p$ is the characteristic of $\mathbb{F}$. As a consequence, if $\mathbb{F} = \mathbb{F}_{p^k}$, we'll assume that $d|k$ so we can use the Folding matrix, and in this case, $\widehat{\mathcal{M}}$ and $\widetilde{\mathcal{M}}$ both support $d \leqslant k$.

**Karatsuba matrices.**

We define recursively the matrices $\mathbf{M}_1, \mathbf{M}_2$ associated to Karatsuba algorithm. We will write $\mathbf{M}_1^d \in K^{d^{\log 3} \times d}, \mathbf{M}_2^d \in K^{(2d-1)\times \log 3}$ the matrices for degree $d-1$ input. Remind that here, $K$ is the smallest subfield of $\mathbb{F}$ that contains $-1, 0, 1$, that is $\mathbb{Z}/p\mathbb{Z}$ where $p$ is the characteristic of $\mathbb{F}$. We assume for simplicity

that $d = 2^\ell$ is a power of 2. Otherwise, one can fill the coefficients of the inputs with zeros until the degree indeed is a power of 2. For clearer exposition, we introduce another sequence of matrices $\mathbf{B}^d$.

We have $\mathbf{M}_1^1 = (1), \mathbf{M}_2^1 = (1)$ and for $d$ a power of two:

$$\mathbf{M}_1^{2d} = \begin{bmatrix} \mathbf{M}_1^d & \mathbf{0}_d \\ \mathbf{M}_1^d & \mathbf{M}_1^d \\ \mathbf{0}_d & \mathbf{M}_1^d \end{bmatrix} \mathbf{B}^{2d} = \begin{bmatrix} \mathbf{0}_d & \mathbf{0}_d & \mathbf{0}_d \\ -\mathbf{M}_2^d & \mathbf{M}_2^d & -\mathbf{M}_2^d \\ \mathbf{0}_d & \mathbf{0}_d & \mathbf{0}_d \end{bmatrix} \mathbf{M}_2^{2d} = \begin{bmatrix} \mathbf{M}_2^d & \mathbf{0}_d & \mathbf{0}_d \\ \mathbf{0}_d & \mathbf{0}_d & \mathbf{0}_d \\ \mathbf{0}_d & \mathbf{0}_d & \mathbf{M}_2^d \end{bmatrix} + \mathbf{B}^{2d}.$$

The two block columns of $\mathbf{M}_1^{2d}$ are of length $d$, and the block rows are of size $3^\ell$, so the dimensions of $\mathbf{M}_1^d$ are $3^\ell \times d$. The rows of $\mathbf{B}^{2d}$ are of length respectively $d, 2d-1, d$, while its columns are of length $3^\ell$. The matrix $\mathbf{M}_2^d$ has the same dimensions $(2d-1) \times 3^\ell$ as $\mathbf{B}^d$. With $\mathbf{a}, \mathbf{b}$ two polynomials of degree $d-1$, we have

$$\mathbf{a} \cdot \mathbf{b} = \mathbf{M}_2^d \left( \mathbf{M}_1^d \mathbf{a} \odot \mathbf{M}_1^d \mathbf{b} \right).$$

**Comparison of the performances of multiplication gadgets.**

In the graphs that can be found below, we measure the number of multiplications in $\mathbb{F}$, additions in $\mathbb{F}$, random elements from $\mathbb{F}$, and the probing ratio of an algorithm as the minimum ratio between the probe tolerance per subcircuit divided by the size of the subcircuit across all regions. We take into account for the size of the subcircuits operation gates, copy gates and randomness gates as 1 each. We mention that both $\widehat{\mathcal{M}}$ and $\widetilde{\mathcal{M}}$ are considered in their worse regime, that is, when the folding matrix does not exist. Doubling the probing ratio give the value to be expected when the folding matrix may be used.

We also remind the reader that the probing ratios are only indications, as the security proof of GPRV does not cover all the orders $d$ depicted, and similarly, without more precision on $\mathbb{F}$, $\widehat{\mathcal{M}}$ and $\widetilde{\mathcal{M}}$ may not have a security proof available.

## References


ADF16. Marcin Andrychowicz, Stefan Dziembowski, and Sebastian Faust. Circuit compilers with $o(1/log(n))$ leakage rate. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 586–615. Springer, 2016.

AIS18. Prabhanjan Ananth, Yuval Ishai, and Amit Sahai. Private circuits: A modular approach. In *Annual International Cryptology Conference*, pages 427–455. Springer, 2018.

BBD+16. 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 *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 116–129, 2016.

BBP+16. Sonia Belaïd, Fabrice Benhamouda, Alain Passelègue, Emmanuel Prouff, Adrian Thillard, and Damien Vergnaud. Randomness complexity of private circuits for multiplication. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 616–648. Springer, 2016.

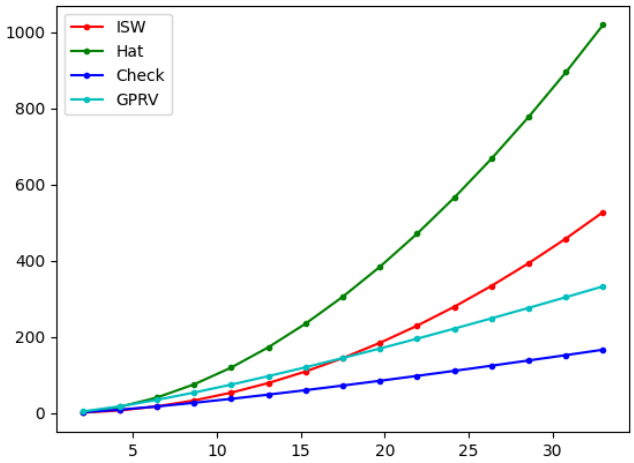


**Fig. 1.** Comparison of the number of random field elements required per run.

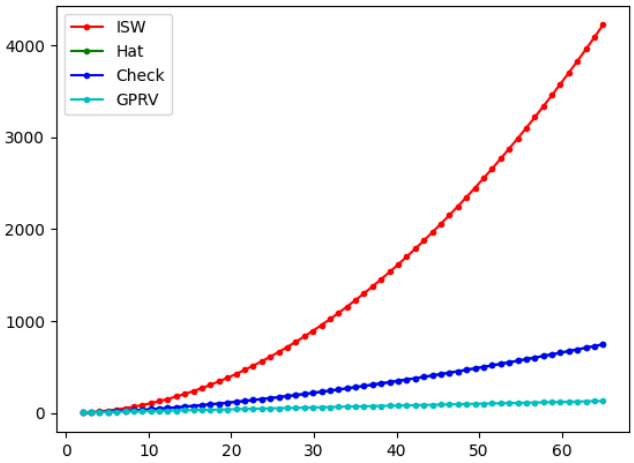


**Fig. 2.** Comparison of the number of multiplication between variables required per run.

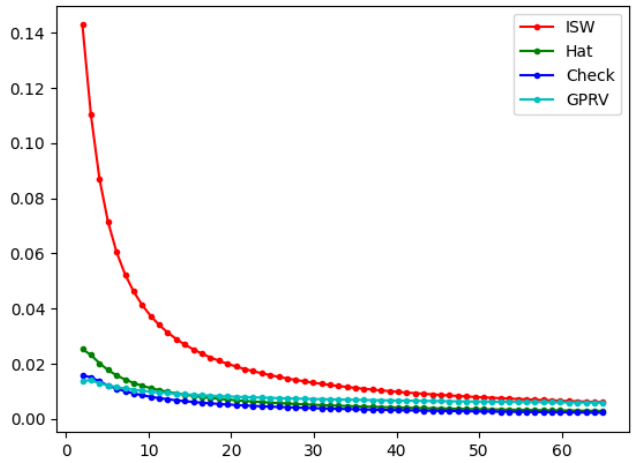


**Fig. 3.** Comparison of the probing ratio.

BBP+17. Sonia Belaïd, Fabrice Benhamouda, Alain Passelègue, Emmanuel Prouff, Adrian Thillard, and Damien Vergnaud. Private multiplication over finite fields. In *Annual International Cryptology Conference*, pages 397–426. Springer, 2017.

BC22. Olivier Bronchain and Gaëtan Cassiers. Bitslicing arithmetic/boolean masking conversions for fun and profit with application to lattice-based kems. *Cryptology ePrint Archive*, 2022.

BCLV17. Daniel J Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, and Christine van Vredendaal. Ntru prime: reducing attack surface at low cost. In *International Conference on Selected Areas in Cryptography*, pages 235–260. Springer, 2017.

BCP+20. Sonia Belaïd, Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, and Abdul Rahman Taleb. Random probing security: verification, composition, expansion and new constructions. In *Annual International Cryptology Conference*, pages 339–368. Springer, 2020.

BCPZ16. Alberto Battistello, Jean-Sébastien Coron, Emmanuel Prouff, and Rina Zeitoun. Horizontal side-channel attacks and countermeasures on the isw masking scheme. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 23–39. Springer, 2016.

BDK+18. Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-kyber: a cca-secure module-lattice-based kem. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 353–367. IEEE, 2018.

BRT21. Sonia Belaïd, Matthieu Rivain, and Abdul Rahman Taleb. On the power of expansion: more efficient constructions in the random probing model. In

*Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 313–343. Springer, 2021.

BRTV21. Sonia Belaïd, Matthieu Rivain, Abdul Rahman Taleb, and Damien Vergnaud. Dynamic random probing expansion with quasi linear asymptotic complexity. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 157–188. Springer, 2021.

CGMZ21. Jean-Sébastien Coron, François Gérard, Simon Montoya, and Rina Zeitoun. High-order polynomial comparison and masking lattice-based encryption. *Cryptology ePrint Archive*, 2021.

Coo66. Stephen A. Cook. *On the minimum computation time of functions*. PhD thesis, 1966. URL: `http://cr.yp.to/bib/entries.html#1966/cook`.

CPRR13. Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, and Thomas Roche. Higher-order side channel security and mask refreshing. In *International Workshop on Fast Software Encryption*, pages 410–424. Springer, 2013.

CS20. Gaëtan Cassiers and François-Xavier Standaert. Trivially and efficiently composing masked gadgets with probe isolating non-interference. *IEEE Transactions on Information Forensics and Security*, 15:2542–2555, 2020.

DDF14. Alexandre Duc, Stefan Dziembowski, and Sebastian Faust. Unifying leakage models: From probing attacks to noisy leakage. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 423–440. Springer, 2014.

DFS15. Stefan Dziembowski, Sebastian Faust, and Maciej Skorski. Noisy leakage revisited. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 159–188. Springer, 2015.

DKL$^+$18. Léo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-dilithium: A lattice-based digital signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 238–268, 2018.

DKR$^+$21. Christoph Dobraunig, Daniel Kales, Christian Rechberger, Markus Schofnegger, and Greg Zaverucha. Shorter signatures based on tailor-made minimalist symmetric-key crypto. *Cryptology ePrint Archive*, 2021.

DVBV22. Jan-Pieter D'Anvers, Michiel Van Beirendonck, and Ingrid Verbauwhede. Revisiting higher-order masked comparison for lattice-based cryptography: Algorithms and bit-sliced implementations. *Cryptology ePrint Archive*, 2022.

GJR18. Dahmun Goudarzi, Antoine Joux, and Matthieu Rivain. How to securely compute with noisy leakage in quasilinear complexity. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 547–574. Springer, 2018.

GPRV21. Dahmun Goudarzi, Thomas Prest, Matthieu Rivain, and Damien Vergnaud. Probing security through input-output separation and revisited quasilinear masking. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 599–640, 2021.

ISW03. Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In *Annual International Cryptology Conference*, pages 463–481. Springer, 2003.

KJJ99. Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *Annual international cryptology conference*, pages 388–397. Springer, 1999.

KO62. Anatolii Karatsuba and Yu Ofman. Multiplication of multidigit numbers on automata. *Soviet Physics Doklady*, 7:595, 12 1962.

Koc96. Paul C Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Annual International Cryptology Conference*, pages 104–113. Springer, 1996.

PR13. Emmanuel Prouff and Matthieu Rivain. Masking against side-channel attacks: A formal security proof. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 142–159. Springer, 2013.

RP10. Matthieu Rivain and Emmanuel Prouff. Provably secure higher-order masking of aes. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 413–427. Springer, 2010.

Too63. Andrei L Toom. The complexity of a scheme of functional elements realizing the multiplication of integers. In *Soviet Mathematics Doklady*, volume 3, pages 714–716, 1963.