Exploiting algebraic structures in probing security.

Maxime Plançon*

ETH Zurich
IBM Research Europe, Zurich

Abstract. The so-called ω-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 follow-up 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, that is, a region-probing secure arithmetic circuit masked compiler. Our contribution stems from a single Lemma, linking algebra and probing security for a wide class of circuits, further taking advantage of the algebraic structure of ω-encoding, and the extension field structure of the underlying field $\mathbb{F}$ that was so far left unexploited. On the theoretical side, we propose a security notion for ω_d-masked circuits which we call Reducible-To-Independent-K-linear (RTIK). When the number of shares d is less than or equal to the degree k of $\mathbb{F}$, RTIK circuits achieve region-probing security. Moreover, RTIK circuits may be composed naively and remain RTIK. We also propose a weaker version of IOS, which we call KIOS, for refresh gadgets. This notion allows to compose RTIK circuits with a randomness/security tradeoff compared to the naive composition.
To substantiate our new definitions, we also provide examples of competitively efficient gadgets verifying the latter weaker security notions. Explicitly, we give 1) two refresh gadgets that use $d-1$ random field elements to refresh a length d encoding, both of which are KIOS but not IOS, and 2) a multiplication gadget with bilinear multiplication complexity $d^{\log 3}$ and uses d fresh random elements per run. Our compiler outperforms ISW asymptotically, but for our security proofs to hold, we do require that the number of shares d is less than or equal to the degree of $\mathbb{F}$ as an extension, so that there is sufficient structure to exploit.

Keywords: Masking, RTIK, Refresh Gadget, Multiplication Gadget

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

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

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 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. While t-probing security reduces to the more realistic models, the reductions are somewhat loose and depend more on the ratio t divided by the size of the circuit than t itself.

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 $(x_0, \ldots, x_{d-1})$. 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 unfortunately 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 for any number of shares 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 paper [GJR18], started a line of research towards constructing multiplication gadgets based on the Fast Fourier Transform. GJR uses a different type of encoding called ω-encoding, where $\mathbf{a}$'s evaluation is taken in some field element ω 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 ω, hence for reaching a reasonable level of security, GJR+ requires an underlying field of exponential size in the security parameter, which limits its practical applications. The follow-up paper [GPRV21] proposed a security proof for GJR+ for fields of smaller sizes. This 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 $x = \mathbf{x}(\omega)$), but we additionnally aim at exploiting the field extension structure of the field $\mathbb{F}$ over which our circuit is defined. Our contributions can be summarized as

1. Revisiting probing security from a probabilistic angle.
2. Introduction of new security notions tailored for circuits over extension fields: for operation gadgets (RTIK) and for refresh gadgets (RTK, KIOS)
3. Composition Theorems for RTIK gadgets and KIOS refresh gadgets.
4. Examples of competitively efficient multiplication gadgets and refresh gadgets achieving the aforementioned notions.

We detail separately each of these items in the following.

From game-based definitions to probabilistic definitions. The usual definition of t-probing security involves the existence of a simulator able to simulate the distribution of given wires with only partial knowledge of the secret. This simulation-based definition is inherited from the idea that a t-probing side-channel adversary plays a t-probing security game, in which the adversary learns some information on the wires $\mathcal{W}$ of the circuit $\mathcal{C}$, then wins if he guesses right the decoding of the sharings. The simulation argument definetly implies that the side-channel information yields no advantage. While simulators can be suitable tools for proving probing security, they do not seem to be a good fit with our techniques. We propose to take a different path and redefine probing security as the statistical independence of the leakage and the secrets. While this idea is nothing new, we believe that the formal definitions from Subsection 3.1 can be of independent interest. In particular, we formally define the intuitive idea that a given set of probes Q contains more information than some other set of probes P. This syntax enables "game hop"-based proof strategy. Informally, we let the adversary pick the initial set of probes P of his choice, then instead of proving some independence relation between P and the secrets directly, we reduce, via successive elementary game hops, the set of probes P to a set of probes Q that at least preserves the information of the adversary. At the end of this reduction from P to Q, the latter set of probes Q is such that our techniques apply and we manage to prove the independence of Q and the secrets, which in turn implies independence between P and the secrets.

Bridging algebra and probing security. We consider a circuit $\mathcal{C}$ over a finite field $\mathbb{F}$. We remind that our goal in this paper is to exploit the underlying field extension structure of $\mathbb{F}$, thus for the sake of clarity, we assume that $\mathbb{F} = \mathbb{F}_{p^k}$ is the finite field with p^k elements where p is a prime and $k \geqslant 2$. An even more concrete example is taking $\mathbb{F}$ to be the masking benchmark AES field $\mathbb{F}_{2^8}$. We deal with polynomial encodings, which is a special case of linear sharings where our decoding vector is chosen to be $\omega_d = (1, \omega, \ldots, \omega^{d-1})$, for some field element $\omega \in \mathbb{F}$. In other words, an ω_d-encoding $\mathbf{x} \in \mathbb{F}^d$ of some element x is such that

$$\omega_d^T \mathbf{x} = \sum_{i=0}^{d-1} x_i \omega^i = x.$$

The bridge relating the structure of $\mathbb{F}$ and probing security is a single Lemma 4.2. Consider that our circuit $\mathcal{C}$ takes as input an ω_d-encoding $\mathbf{x}$. In a nutshell, Lemma 4.2 says that under the conditions that

1. The number of shares is at most the degree of the extension: $d \leqslant k$
2. The intermediate values that the adversary can probe in $\mathcal{C}$ are of the form $\mathbf{p}^T\mathbf{x}$ with $\mathbf{p} \in \mathbb{F}_p$,

then there exists a choice of ω for which $\mathcal{C}$ is $d-1$-probing secure. This choice of ω is actually any ω of algebraic degree greater than d over $\mathbb{F}_p$. The geometry of this Lemma makes it intuitively more permissive than the usual definitions for t-probing, r-region-probing, (strong) non-interference and probe-isolating-non-interference. Indeed, the latter definitions (in probabilistic terms) require roughly speaking that the probes are independent of at least one coordinate of each sharings, while the former implies security regardless of the direction of the affine subspace given by the probes, provided that this subspace is directed by a matrix over the subfield and that its dimension is at least 1.

By following the rules for modifying the set of probes of the adversary, we can relax condition 2.: our circuit $\mathcal{C}$ is also $d-1$-probing secure if for all sets P of $d-1$ probes (that does not necessarily verify 2.), we can find a set of $d-1$ probes Q that contain at least as much information as P, but Q does verify 2.

The RTIK security notion (which stands for Reducible-To-K-Independent) for ω_d-masked circuits over extension fields roughly encompasses the circuits that fulfill the requirements of the above. The requirements for a circuit to be RTIK are slightly more general: the subfield K that contains the coefficients of the probes may be bigger than the prime field of $\mathbb{F}$, and the circuit $\mathcal{C}$ may take several encodings as input. In that case, we simply require that there exists some mutually independent encodings $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ and sets of probes $(Q_1, \ldots, Q_n)$ such that each Q_i is K-linear in $\mathbf{x}_i$. Notice that some of these encodings may not be inputs neither outputs of $\mathcal{C}$.

Since by construction, RTIK circuits over extension fields fall into the requirements of the core Lemma, it follows that RTIK circuits are $d-1$-probing secure. Actually, RTIK circuits are secure in the stronger r-region-probing model, where the adversary may place some number of probes in several different subcircuits. More precisely, for n the number of mutually independent encodings in the RTIK property, each of those encodings is associated with a set of probes Q_i, and each of those sets of probes Q_i defines a subcircuit of $\mathcal{C}$. By applying our core Lemma to each of these independent encodings, the adversary may place more than $d-1$-probes, provided that throughout the reduction, no Q_i ends up with more than $d-1$ probes. We note that similarly as the Probe-Isolating-Non-Interfering security notion [CS20], (all known) RTIK gadgets can be composed directly without refresh, in which case the composition of RTIK circuits remains RTIK, which in turn is r-region probing secure for some ratio r. We also mention that in terms of implementation, RTIK circuits seem rather stable, since as long as the wires are of the right K-linear form, the order of the operations does not affect security.

Although RTIK circuits may be composed directly and remain region-probing secure, the size of the probing regions of the composite circuits may increase and hence reduce the probing ratio, thus reduce the overall security of the implemen-

tation.[1] To mitigate this loss of security, we introduce a security notion for refresh gadgets 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 α-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 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. Applying a KIOS refresh to an encoding in between two RTIK circuits creates a new region at the cost of using random elements.

KIOS refresh gadgets 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](Actually, we prove the IOS property for a mild generalization of this algorithm) 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 ω over K to be greater than d, and for PolyGenZero to be correct, we also need the algebraic degree of ω over K to be less than d. In other words, we need ω to have algebraic degree exactly d over K, and such choice of ω is only possible when d divides $[\mathbb{F} : K]$. The intuition on the construction of this 2-KIOS gadget is detailed in Section 5.2.

We give a second example of KIOS refresh, which also uses $d-1$ random elements, and is 1-KIOS. The counterpart for this improvement is that it is slightly bigger than the previous one as a circuit. The intuition behind this algorithm is derived from the RTIK multiplication gadget Algorithm 7. In a nutshell, the idea is to sample a uniformly random vector $\mathbf{r}$, then multiply it using Karatsuba's algorithm with some fixed polynomial $\mathbf{u}$. Provided that the

[1] In fact, each of the $(Q_i)_{i\in[n]}$ of the composite RTIK circuit still tolerates $d-1$, but the number of regions is the number of fresh encodings involved in the circuit. This takes into account the input encodings, as well as the fresh random inputs used in multiplication gadgets and refresh gadgets.

[2] We also add a coefficient α 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.

only common factor of $\mathbf{u}$ and the minimal polynomial of ω is $X-\omega$ (which again requires $\deg_K(\omega) = d$), this algorithm generates ω_d-encodings of 0, which we can add coordinate-wise to obtain a 1-KIOS refresh gadget.

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 ω 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] The multiplication gadget GJR+ [GPRV21] has two security proofs, depending on the size of $\mathbb{F}$ (and to some extent d). When $|\mathbb{F}| \geqslant 2^\lambda$ for some security parameter λ a statistical argument based on the random choice of ω implies security in the random-probing model. When $|\mathbb{F}|$ is too small, the authors rely on a non-standard ad-hoc assumption that the circuit computing the FFT and its inverse are t-probing secure. Due to combinatorial explosion, it is only possible to test the assumption for small values of d, thus leaving a hole in the shape of the RTIK notion. Our first multiplication gadget is a generalization of GJR+, where one can use *any* evaluation-interpolation polynomial multiplication algorithm (not only the FFT), and turn it into a multiplication gadget. The regimes in which we can prove that [GPRV21]'s assumption hold is restricted to the tuples $(\mathbb{F}, d)$ such that $d \leqslant [\mathbb{F} : K]$. The subfield K for which the RTIK property holds is the smallest subfield that contains the coefficients of both evaluation and interpolation. Hence for maximizing the upper bound on d, one should chose the multiplication algorithm so that K is as small as possible, which is a first hint towards switching to Karatsuba's multiplication.

We also propose an optimized version of a multiplication gadget based on Karatsuba's algorithm. This Algorithm 7 uses d random field elements per run (which is most likely close to optimal), but does $d^{\log 3}$ bilinear multiplications. It verifies the RTIK property, thus it is composable without extra refreshing.[4] The intuition behind the optimizations is detailed in Section 6. We compare

[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 $\mathbb{F}$.

[4] The introduction of d random elements in the multiplication does increment the number of regions when composed with other circuits

the performances of our optimized multiplication gadget with a few existing constructions in Figure 1. We highlight that Algorithm 7 and ISW are the only multiplication gadgets that can be securely composed without extra refreshing. In terms of bilinear multiplication, Algorithm 7 is worse than GJR+ and Belaïd bil, but better than Belaïd rand and ISW. In terms of randomness, Algorithm 7 is close to optimal with d random elements, only beaten by Belaïd rand by one random element. Further details on this comparison can be found in Appendix A.

	ISW	Belaïd bil	Belaïd rand	GJR+	Algorithm 7
Bilinear mul	d^2	$2d-1$	d^2	$2d$	$d^{\log 3}$
Randomness	$\frac{d(d-1)}{2}$	$2(d-1)^2+\frac{(d-1)(d-2)}{2}$	$d-1$	$d\log(2d)$	d
t-threshold	$d-1$	$d-1$	$d-1$	$d/2-1$	$d-1$
Composable	YES	NO	NO	NO	YES

Fig. 1. Comparison table of multiplication gadgets for a number of shares d. ISW[ISW03] for arithmetic encodings, Belaid rand[BBP+17] Alg. 5, Belaid bil[BBP+17] Alg. 4, and GJR+[GPRV21]). Composable means whether refreshing is needed to be composed with other circuits.

1.2 Limitations and open questions.

Lack of concreteness. Our contribution mostly stands on the theoretical side. While we give performance comparisons Appendix A, 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. An extension field $\mathbb{F}/K$ of degree k is proven secure with our techniques up to $d = k$ shares. For example, in the AES field $\mathbb{F}_{256}$, we have $k = 8$, thus our masked compiler tolerates a number of shares d up to 8, with extra efficiency for $d|k$, i.e $d \in \{2, 4, 8\}$. The real world masked implementation are for the most part within this range, but it seems to be an interesting open question to lift the upper bound, especially for the extension field of lower degree, that have insufficient algebraic structure for our techniques to apply. 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 is, waving hands, aiming at boosting the security by repeating the masked compilation several times instead of just one, is an interesting direction which we leave for future work.

Masking lattice-based cryptography. We believe that part of the techniques and algorithms proposed in this paper may apply to the usual power-of-two cyclo-

tomic ring structure underlying lattice-based cryptography. It is also an interesting open question to know to what extent our constructions survive in the ring setting. Since the standardization of several lattice-based schemes, especially Kyber, constructing efficient equality-testing gadgets[DVBV22, CGMZ21, BC22] has received a lot of attention and the contributions of this paper may provide a different angle towards constructing efficient equality-test gadgets.

Formal verification of implementations. Maskverif [BBC$^+$18, BBC$^+$19] is a tool that, roughly speaking, when fed an implementation and an adversary model returns the level of security achieved by the input implementation against the given adversary model. The RTIK property seems like a nice property for automated testing, thus it is also an interesting open question to construct a verification tool for implementations.

The proofs Propositions and Theorems that are missing from the body of the paper can be found in the appendix, sorted by Sections in increasing order.

2 Background

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 $\mathbb{F}$. To ease the readability, we identify a polynomial to its list of coefficients, and use either notations interchangeably. An element $\mathbf{a} \in \mathbb{F}^d$ can be treated as an element of $\mathbb{F}_{d-1}[X]$ depending on context, e.g by writing $\mathbf{a}(\omega)$ the evaluation of the polynomial whose coefficients list is $\mathbf{a}$ in a field element ω, or multiplying two polynomials $\mathbf{ab}$ while keeping the vector notation. We write $\pi_K(\omega)$ the minimal polynomial of ω 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 χ 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 $\mathbf{v} = (1 \ \dots \ 1)$. For ω an element of $\mathbb{F}$, we let $\omega_d = (\omega^i)_{0\leqslant i\leqslant d-1}$. We say that a vector $\mathbf{x} \in \mathbb{F}^d$ is an ω_d-encoding of a field element $x \in \mathbb{F}$ when $\omega_d^T\mathbf{x} = x$ (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 $H_x^{\omega_d}$ 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 say that $(\mathbf{x}_1, \dots, \mathbf{x}_n)$ are mutually independent ω_d-encodings when for all $x_1, \dots, x_n$, the distributions $(\mathbf{x}_1|\omega_d^T\mathbf{x}_1 = x_1), \dots, (\mathbf{x}_n|\omega_d^T\mathbf{x}_n = x_n)$ are mutually independent.

We call an addition gadget (respectively a multiplication gadget) with respect to ω_d-encodings a circuit that takes as input two ω_d-encodings $\mathbf{a}, \mathbf{b}$ and returns an ω_d-encoding of $\omega_d^T\mathbf{a} + \omega_d^T\mathbf{b}$ (respectively $\omega_d^T\mathbf{a} \cdot \omega_d^T\mathbf{b}$). A correct refresh gadget with respect to ω_d-encodings is a circuit that takes as input an ω_d-encoding and returns an ω_d-encoding of the same secret. In general, for a gate g in a circuit $\mathcal{C}$, we say that G is a correct ω_d-encoding gadget for g when G takes as input ω_d-encodings of the sensitive inputs of g, and returns ω_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 $\mathcal{W}$, χ be the input random variable of $\mathcal{C}$ and $x_1, \dots, x_n$ be secret variables. A t-probing adversary $\mathcal{A}$ on $(\mathcal{C}, \chi)$ against $x_1, \dots, x_n$ 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 $(y_1, \dots, y_n)$. He wins if $(y_1, \dots, y_n) = (x_1, \dots, x_n)$.*

A circuit $\mathcal{C}$ for which there is no unbounded adversary $\mathcal{A}$, playing the t-probing security game with respect to secrets $x_1, \dots, x_n$, 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 χ 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, $\mathcal{C}_1, \dots, \mathcal{C}_m$ be subcircuits of $\mathcal{C}$ such that $(\mathcal{C}_1, \dots, \mathcal{C}_m)$ is a covering*

of $\mathcal{C}$, $\mathcal{W}_1, \ldots, \mathcal{W}_m$ be the induced sets of intermediate variables, χ 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 Probabilistic approach to probing security

In this section, we make our first step towards bridging probing security and algebra, which boils down to redefining from a probabilistic perspective the usual definitions of probing security, region-probing security and the IOS composition property. While the usual simulation-based definitions have their advantages, the probabilistic versions of the latter properties are a much better fit with our techniques. All the results, definitions and propositions in this section are stated for linear sharings ($\mathbf{v}$-encodings for any $\mathbf{v} \in (\mathbb{F} \backslash \{0\})^d$).

3.1 Redefining probing security through sets of probes and distribution of secrets.

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. From a high level, one can think of these probabilistic definitions as simply cutting the middle-man, where the middle-man is the simulator. Indeed, in a simulation-based proof, one has to define the simulator for any given set of probed wires (and maybe modify the probes of the adversary before doing so), and then justify that this simulator is actually giving samples of the right distribution. By relying directly on the distribution argument, we focus on proving that the leakage distribution is independent of the secrets, which in our mind highlights the key arguments of the proof and arguably makes it shorter.

We start off with 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 χ and manipulating secret variables $x_1, \ldots, x_n$. We say that Q contains more information than P, and we write $P \leqslant Q$, when*

$$((x_1, \ldots, x_n)|(P(\chi), Q(\chi))) = ((x_1, \ldots, x_n)|Q(\chi)).$$

When $\mathcal{C}$ contains random gates, and the outputs $(r_1, \ldots, r_m)$ of some random gates of $\mathcal{C}$ are sensitive, the condition becomes

$$\big((x_i)_{i\in[n]}, (r_j)_{j\in[m]}|P(\chi), Q(\chi)\big) = ((x_i)_{i\in[n]}, (r_j)_{j\in[m]}|Q(\chi)).$$

When $P \leqslant Q$, intuitively, all the sensitive information on the input χ 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. The point of this binary relation is to provide a formal justification for modifying the set of probes that the adversary initially choses in the probing security games. By using a few allowed elementary operations one after another, we are able to reduce any initial set of probes to another set of probes that has a shape that fits our techniques in the following sections.

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 two arithmetic encodings (x_0, x_1), (y_0, y_1). The secrets manipulated by the circuit are $x = x_0 + x_1$ and $y = y_0 + y_1$. 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 proofs that rely on an increasing sequence of sets of probes can be found in the proofs of Propositions 5.1 and 5.2 and Theorems 6.2 and 6.3.

We now proceed to define t-probing security and r-region probing security for masked circuit from a probabilistic 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 $\forall P = (P_1, \ldots, P_m) \subset \mathcal{W}_1 \times \cdots \times \mathcal{W}_m$, 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.

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 Revisiting Input-Output-Separation: Refreshing ω_d-encodings and composition of gadgets

For our own technical purposes (e.g the proof of Theorem 6.2) and for exposing the close relation between KIOS Definition 4.6 and IOS Definition 2.3, 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 ℓ-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 give explicitly the template to build an IOS refresh gadget Algorithm 2 and Proposition 3.9 from an RTC generator of encodings of 0.

Definition 3.5. *(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. $(Q_1, Q_2) \geqslant P$
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 3.6. *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].

Definition 3.7. *(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. $(Q_x, Q_y, Q_2) \geqslant P$
3. *Every probe in Q_x is a coordinate of $\mathbf{x}$ and every probe in Q_y is a coordinate of $\mathbf{y}$*

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

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

Proposition 3.8. *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 3.7, then it is also t-IOS according to Definition 2.3.*

Algorithm 2 RTC generator to 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 $\mathbf{v}^T\mathbf{y} = \mathbf{v}^T\mathbf{x}$

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

Proposition 3.9. *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.*

4 Algebraic approach in probing security for extension fields

In this section, we focus on the setting where $\mathbb{F}$ is an extension field over some subfield K. We only consider a specific type of encoding, which is ω_d-encoding, where $\omega_d = (1, \omega, \omega^2, \ldots, \omega^{d-1})$ is the vector with all the first d powers of some fixed field element $\omega \in \mathbb{F}$. Unless specified otherwise, ω is chosen so that its algebraic degree over the subfield K is at least the number of shares, in order to apply the core Lemmas from Section 4.1. We remind the reader that the notions detailed in this section exploit the algebraic structure of $\mathbb{F}$, and for our techniques to apply, the number of shares d cannot exceed $[\mathbb{F} : K]$.

In the first subsection, we state the core Lemmas that make the connexion between the extension field structure of $\mathbb{F}/K$ and probing security. In the second subsection, we introduce the RTIK security notion for circuits (a priori of any size between operation gadget to a full cryptographic algorithm implementation) that in turn implies region-probing security. In the last subsection, we show that RTIK circuits admit nice composition properties without refresh. We finally show that refreshing the encodings in between two RTIK circuits gives more security at the cost of randomness, and that the refresh gadget is still secure with a slightly weaker notion KIOS than the IOS notion.

4.1 Probing security of K-linear circuits

This subsection contains two technical results Lemmas 4.1 and 4.2 that are building blocks for proving t-probing security of ω_d-masked circuits.

From a high level, the first Lemma 4.1 claims that when $\deg_K(\omega) \geqslant d$, the vector ω_d is never in the span of $\ell < d$ vectors over K, where K is a subfield of $\mathbb{F}$. 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 4.2.

To illustrate, consider a t-probing adversary against some circuit $\mathcal{C}$, taking as input a uniform ω_d-encoding of the secret. We assume that the adversary has no prior knowledge on the secret $a = \omega_d^T \mathbf{a}$ 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 inner products of 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 $\omega_d^T \mathbf{a} = a'$. When $\deg_K(\omega) \geqslant d$ is satisfied, Lemma 4.1 tells us that $\omega_d \notin \text{Span } \mathbf{P}^T$, 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 4.2. 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 4.7. The convenient form of Lemma 4.2 makes it likely to find other applications in constructing efficient masked gadgets.

Lemma 4.1. *Let $\mathbb{F}$ be a finite field, K be a subfield of $\mathbb{F}$, $\mathbf{P} \in K^{t\times d}$ such that* rank $\mathbf{P} = t$ *and $\omega \in \mathbb{F}$. If $\deg_K(\omega) \geqslant d$ and $t < d$, then*

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

Proof. Let us assume for one moment that rank $\begin{bmatrix} \mathbf{P} \\ \omega_d^T \end{bmatrix} = t$, i.e $\omega_d \in$ Span $\mathbf{M}^T$. This means that there exists t coefficients $\lambda_i \in \mathbb{F}^t$ such that $\mathbf{P}^T\boldsymbol{\lambda} = \omega_d$. 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

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

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

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

Lemma 4.2. *Let d be an order of masking, $\mathcal{C}$ be a circuit taking as input a uniform ω_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 χ 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

$$\begin{aligned}
\mathbb{P}(\omega_d^T \mathbf{x} = x \cap P(\mathbf{x}) = \mathbf{v}) &= \mathbb{P}(\omega_d^T \mathbf{x} = x \cap \mathbf{P}\mathbf{x} = \mathbf{v}) && (1)\\
&= \bar{\chi}\left(\ker \begin{bmatrix} \mathbf{P} \\ \omega_d^T \end{bmatrix} + \mathbf{x}^*\right) && (2)\\
&= \bar{\chi}\left(\mathbf{x}^*\right) = \frac{1}{|\mathbb{F}|^{d-1}}\chi(x) && (3)\\
&= \mathbb{P}(P(\mathbf{x}) = \mathbf{v})) \cdot \mathbb{P}(\omega_d^T \mathbf{x} = x), && (4)
\end{aligned}$$

where Equation (1) is the hypothesis of the Lemma, Equation (2) holds for some solution $\mathbf{x}^*$ to the equation $\begin{bmatrix} \mathbf{P} \\ \omega_d^T \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ x \end{bmatrix}$, Equation (3) follows from Lemma 4.1 which implies that the matrix $\begin{bmatrix} \mathbf{P} \\ \omega_d^T \end{bmatrix}$ is of rank d, therefore its kernel is 0, and Equation (4) 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.

4.2 Weaker condition for region-probing security in extension fields

Rephrasing (and simplifying) the RTIK property: an ω_d-masked circuit $\mathcal{C}$ 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 ω_d-masked encoding.

Definition 4.3 (Reducible-To-Independent-K-Linear (RTIK)). *Let $\mathcal{C}$ be a circuit over a finite field $\mathbb{F}$, K be a subfield of $\mathbb{F}$ and $\mathcal{W}$ be the set of wires of $\mathcal{C}$. We say that $\mathcal{C}$ is RTIK when for all set of probes $P \subset \mathcal{W}$, there exists some number n of ω_d-encodings $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ and a set of probes $Q = (Q_1, \ldots, Q_n) \subset \mathcal{W}$ such that the following holds:*

1. *$(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ are mutually independent encodings*
2. *$Q \geqslant P$*
3. *$\forall i \in [n],\ |Q_i| \leqslant |P|$*
4. *For all $i \in [n]$, every probe in Q_i is a linear function of x_i over K.*

From RTIK to region-probing security. Let us consider an RTIK circuit $\mathcal{C}$. The subsets of probes $(Q_i)_{i\in[n]}$ naturally define a covering family of subcircuits $(\mathcal{C}_i)_{i\in[n]}$[5], which reveals the underlying region-probing model setting of $\mathcal{C}$. For all number of shares $d \leqslant [F : K]$, there exists a choice of ω for which Lemma 4.2 implies the individual and independent $d-1$-probing security of the subcircuits $(\mathcal{C}_i)_{i\in[n]}$. Notice that while condition 3. from the RTIK definition may imply that the count of probes that the reduction gives to the adversary can be as big as $n(d-1)$, the region-probing model mitigates this potential concern. In the

[5] The subcircuit $\mathcal{C}_i$ is defined as the smallest subcircuit such that any probe it contains is sent towards Q_i. Notice that two distinct subcircuits $\mathcal{C}_i, \mathcal{C}_j$ may overlap.

worse case scenario, these probes are evenly spread among all n subcircuits, in which case each region still tolerates up to $d-1$ probes, but also all subcircuits have a common overlap. In fact, we have that $\mathcal{C}$ is r-region probing secure for $r = \min_{i\in[n]} \frac{d-1}{|\mathcal{W}_i|}$, where $\mathcal{W}_i$ is the set of wires of $\mathcal{C}_i$. This means that in the worse case scenario, where all the circuits have a common overlap, the circuit is $d-1$-threshold probing secure.

On the encodings $\mathbf{x}_1, \ldots, \mathbf{x}_n$. RTIK circuits can be of any size, between a simple coordinate-wise gadget to a full masked implementation of a cryptographic algorithm. No matter what the size of the gadget is, the encodings $\mathbf{x}_1, \ldots, \mathbf{x}_n$ do not have to be unique, but their number n is unique. For example, the naive coordinate-wise addition gadget computing $\mathbf{c} = \mathbf{a} + \mathbf{b}$ is RTIK with respect to any two encodings among $(\mathbf{a}, \mathbf{b}, \mathbf{c})$. For these smaller circuits, such as gadgets, the encodings $\mathbf{x}_1, \ldots, \mathbf{x}_n$ can still be thought of as the input encodings and/or the output encodings, which is also the case for the multiplication gadgets defined in Section 6. For bigger gadgets, for example the benchmark round of AES, or even the full AES implementation, then the encodings $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ contain many more extra encodings than just the inputs and outputs. The number n of encodings (and thus the number of regions in the security model) is the number of fresh independent input encodings + the number of refresh gadgets in the masked circuit (The multiplication gadget Algorithm 7 also counts as a refresh here, as it somewhat contains a built-in refresh).

4.3 Composition notions for RTIK circuits in extension fields

We first show that some RTIK gadgets with a nice additionnal feature can be composed naively and still enjoy region-probing security.

Theorem 4.4. *Let $\mathcal{C}$ be a circuit over a finite field $\mathbb{F}$, and K be a subfield of $\mathbb{F}$. If $\mathcal{C}$ can be split into two disjoint subcircuits $\mathcal{C}_1, \mathcal{C}_2$ such that*

1. *$\mathcal{C}_1$ and $\mathcal{C}_2$ are RTIK circuits, inducing respectively encodings $(\mathbf{x}_1^1, \ldots, \mathbf{x}_n^1)$ and $(\mathbf{x}_1^2, \ldots, \mathbf{x}_m^2)$ for some integers n and m*
2. *The intersection of the input encodings of $\mathcal{C}_2$ and the output encodings of $\mathcal{C}_1$ is contained in both $(\mathbf{x}_1^1, \ldots, \mathbf{x}_n^1)$ and $(\mathbf{x}_1^2, \ldots, \mathbf{x}_m^2)$,*

then $\mathcal{C}$ is RTIK.

Proof. We start the proof by taking any set of probes P over $\mathcal{C}$. Since $\mathcal{C}$ is a disjointly covered by $\mathcal{C}_1$ and $\mathcal{C}_2$, P defines two disjoint subsets of probes P_1 over $\mathcal{C}_1$ and P_2 over $\mathcal{C}_2$. We first use the RTIK property of $\mathcal{C}_1$, which ensures the existence of encodings $(\mathbf{x}_1^1, \ldots, \mathbf{x}_n^1)$ and sets of probes $(Q_1, \ldots, Q_n) \geqslant P_1$ such that Q_i is K-linear in $\mathbf{x}_i^1$ and $|Q_i| \leqslant |P_1|$. We repeat the operation and use the RTIK property on $\mathcal{C}_2$, which ensures the existence of encodings $(\mathbf{x}_1^2, \ldots, \mathbf{x}_n^2)$ and sets of probes $(R_1, \ldots, R_n) \geqslant P_2$ such that R_i is K-linear in $\mathbf{x}_i^2$ and $|R_i| \leqslant |P_2|$.

We now write ℓ the number of output encodings of $\mathcal{C}_1$ that are also input encodings of $\mathcal{C}_2$, and assume without loss of generality that those encodings

are $(\mathbf{x}^1_{n-\ell+1}, \ldots, \mathbf{x}^1_n) = (\mathbf{x}^2_1, \ldots, \mathbf{x}^2_\ell)$. We now justify that the circuit $\mathcal{C}$ is RTIK with respect to the following encodings $(\mathbf{x}^1_1, \ldots, \mathbf{x}^1_n, \mathbf{x}^2_{\ell+1}, \ldots, \mathbf{x}^2_m)$. We define $S_1, \ldots, S_\ell$ as respectively $(Q_{n-\ell+1}, R_1), \ldots, (Q_n, R_\ell)$. The set of probes we consider is $Q = (Q_1, \ldots, Q_{n-\ell}, S_1, \ldots, S_\ell, R_{\ell+1}, \ldots, R_m)$. The encodings we consider are indeed mutually independent encodings, and we have that $Q \geqslant P$. We also have that $|Q_i| \leqslant |P_1| \leqslant |P|$ for $i \in [n-\ell]$, $|S_i| \leqslant |P_1| + |P_2| = |P|$ for $i \in [\ell]$ and $|R_i| \leqslant |P_2| \leqslant |P|$ for all $\ell + 1 \leqslant i \leqslant m$. Finally, the K-linearity property of the subcircuits $\mathcal{C}_1, \mathcal{C}_2$ is indeed passed on to the sets of probes for $\mathcal{C}$, which completes the proof.

On the extra condition for naive composition of RTIK circuits. The condition 2. from the Theorem above asks, roughly speaking, that when evaluating $\mathcal{C}_2$ on (part of) the output of $\mathcal{C}_1$, the encodings that are passed on from $\mathcal{C}_1$ to $\mathcal{C}_2$ are part of those vectors that define the RTIK property for both circuits. In practice, we are not aware of any combination of useful circuits that do not verify the aforementionned property. In all generality, we were not able to prove that this condition is always verified, but all our gadgets, as well as all coordinate-wise gadgets do verify the condition, and any circuit composed of our gadgets also verifies this condition.

Composition of more than two gadgets. As one would expect, it is possible to prove that the composition of several gadgets which enjoy the nice extra composability feature is RTIK. Indeed, by induction, one can step by step prove using Theorem 4.4 that the successive compositions are indeed RTIK, as the property propagates with no slack from two circuits to their composition. The fact that there is no slack is ensured by 3. from Definition 4.3. While it is possible to construct gadgets that verify 1. 2. and 4. as well as $|Q_i| \leqslant \alpha|P|$ for some slack factor α (e.g the NaiveFold algorithm defined in Section 5.1), we decide not to introduce this extra notation as the slack factor of a compound circuit grows exponentially with the number of subcircuits, and thus leads to rather inefficient constructions.

Why refreshing a secure circuit ? Again, the probing ratio r is given by the minimum of the individual $\frac{d-1}{|\mathcal{W}_i|}$, where $\mathcal{W}_i$ is the set of wires in the subcircuit $\mathcal{C}_i$. When one of the subcircuits is particularly large compared to the others, it may be beneficial to break it down into smaller independent subcircuits so as to increase the security of the compound circuit. This act of splitting a circuit into subcircuits can be done using an IOS refresh on the encodings, but the weaker notion of KIOS, more adapted to our RTIK circuits, is also suited. This notion is very similar to the IOS notion, thus we follow a similar path towards defining it.

Definition 4.5. *(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 ω_d-encoding $\mathbf{r}$ of 0. Let $\alpha > 0$ be the slack factor of R. We say that R is α-Reducible-To-K-Linear (RTK) when the output distribution of R is a uniform ω_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) $(Q_1, Q_2) \geqslant P$
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 ω_d, then R is 1-RTK. We now define the security notion achieved by the ω_d-encoding refresh gadget obtained by adding coordinate-wise a fresh ω_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 ω_d-encoding and $\mathbf{r}$ is generated using an α-RTK generator of encodings of 0, then we can reduce the probes in the α-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 4.6. *(*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* ω_d*-encoding* $\mathbf{x}$*, and returning an* ω_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*

We finally state in the Theorem below that placing a KIOS refresh in between RTIK circuits achieves region-probing security as well. 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 4.2 yields the region probing security.

Randomness/security tradeoffs of refreshing. As stated throughout the subsection, using KIOS refresh gadgets on the encodings increases the amount of encodings $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ in the RTIK definition, which in turn increases the number of subcircuits in the region-probing security of the latter circuit, and eventually increases the region-probing ratio r. One has to keep in mind that refreshing the shares of an encoding is costly in terms of randomness (and slightly increases the total number of wires in the circuit), thus one has to carefully optimize the amount of refreshing in a circuit to reach the desired security level. Notice that we assume that we use a KIOS refresh gadget in the statement of the KIOS composition Theorem with slack factor 1. Indeed, when the slack factor

of the KIOS refresh is 1, then the resulting circuit is RTIK, but when the slack factor $\alpha > 1$, the resulting circuit is not RTIK as it does not verify the property 3. of the RTIK definition, but it does verify the other ones 1. 2. and 4. When $\alpha > 1$, the resulting circuit remains r-region probing secure, but the number of tolerated probes per region is divided by α.

Theorem 4.7 (KIOS Composition Theorem, adapted from GPRV). *Let $\mathcal{C}$ be a circuit over a finite field $\mathbb{F}$, and K be a subfield of $\mathbb{F}$. If there exists two disjoint RTIK subcircuits $\mathcal{C}_1, \mathcal{C}_2$ of $\mathcal{C}$ such that $\mathcal{C}$ is the composition of $\mathcal{C}_1$ and $\mathcal{C}_2$, then the circuit $\widehat{\mathcal{C}}$ obtained by applying a 1-KIOS refresh to the outputs of $\mathcal{C}_1$ that are inputs of $\mathcal{C}_2$ is RTIK.*

Proof. We start with a set of probes P, which naturally defines P_1 over $\mathcal{C}_1$, P_2 over $\mathcal{C}_2$ and $R_1, \ldots, R_\ell$ where R_i corresponds to the leakage of the i-th refresh gadget, and ℓ is the number of encodings that are outputs of $\mathcal{C}_1$ and inputs of $\mathcal{C}_2$. We use the KIOS property of the refresh gadgets, which implies that for all $i \in [\ell]$, at most $|R_i|$ probes are added to both P_1 and P_2, and the probes from R_i are K-linear in the i-th encoding the probes in $\mathcal{C}_1$ and $\mathcal{C}_2$ for all $i \in [\ell]$. After this step, we can use the RTIK properties of $\mathcal{C}_1$ and $\mathcal{C}_2$, where the initial probes on $\mathcal{C}_1$ and $\mathcal{C}_2$ are respectively P_1 and P_2 augmented with the propagated probes from the refresh. Similarly as in the proof of Theorem 4.4, one can check that the two sets of probes Q_1 from $\mathcal{C}_1$ and Q_2 from $\mathcal{C}_2$ verify the conditions of RTIK for $\mathcal{C}$.

5 Miscellaneous RTIK and KIOS gadgets.

This section contains two ω_d-encodings building-block algorithms for constructinga masked compiler. Both algorithms rely on an additional restriction on d and $\deg_K(\omega)$: For security in our framework of RTIK gadgets, we need $d \leqslant \deg_K(\omega)$ and for correctness of the gadgets presented in this section, we also need $d \geqslant \deg_K(\omega)$. In other words, we need ω to be of degree *exactly* d. A classical result in algebra tells us that such a choice of ω is only possible when d is a factor of $[F : K]$. The reason why we add the restriction $d \geqslant \deg_K(\omega)$ for correctness is that we will exploit the minimal polynomial ω, which we write π_ω throughout the section, in ways that are detailed in the subsections below.

5.1 Folding gadget

This subsection is dedicated to a folding gadget that exploits the algebraic structure brought by ω_d-encodings. Folding gadgets are those that on input some ω_{d_1}-encoding $\mathbf{x}$ return an ω_{d_2}-encoding $\mathbf{y}$ of the same secret, where $d_1 \geqslant d_2$. Since we only need $(d_1, d_2) = (2d-1, d)$, we shall particularize to these specific values in the following, but our construction extends to $d_1 \geqslant 2d-1$. 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.

Algorithm 3 NaiveFold

Require: ω_{2d-1}-encoding $\mathbf{x}$
Ensure: $\mathbf{y} \in \mathbb{F}^d$ such that $\mathbf{x}^T\omega_{2d-1} = \mathbf{y}^T\omega_d$

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 $\mathbf{y}$

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 entails a slack factor of 2 that doubles the number of probes of the adversary, hence in the end halves the number of probes tolerated in the region. Evaluating our folding matrix is an RTIK circuit (in particular it has no slack factor), but it may also contain more wires than the NaiveFold algorithm, thus the gain in probing ratio is slightly fewer than a factor 2. We also remark that the NaiveFold algorithm computes the reduction modulo $(X^d - \omega^d)$, while the folding matrix computes the reduction modulo π_ω.

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 ω_{2d-1}-encodings of some $x \in \mathbb{F}$ to the ω_d-encodings of this same x. This way, the computation of $\mathbf{y} = \mathbf{F}\mathbf{x}$ is K-linear and the folding circuit is RTIK. The existence of this matrix is only guaranteed when $\deg_K(\omega) \geqslant d$, therefore, so we can also use Lemma 4.2, we actually need the equality.

We now proceed to describe how to construct such a matrix, for a given ω and d. Suppose $\deg_K(\omega) = d$. Then, the minimal polynomial π_ω of ω 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 ω^{d+i} for $0 \leqslant i \leqslant d-2$ is a polynomial in ω 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 $\mathbf{F}^T\omega^d = \omega^{2d-1}$. This implies that $\omega_{2d-1}^T\mathbf{x} = \omega_d^T\mathbf{F}\mathbf{x} = \omega_d^T\mathbf{y}$.

Optimizing the choice of ω. We emphasize on the fact that one should chose ω 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 ω 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 ω to be a primitive d-th root of unity. This way, the minimal polynomial of ω 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 $3d$ wires.

5.2 Refresh gadgets

In this subsection, we describe a 2-RTK generator of ω_d-encodings of 0 that only uses $d-1$ random field elements, as well as a 1-RTK generator of ω_d-encodings of 0 that uses $d-1$ random field elements. While the second one seems strictly better than the first one, it also contains more gates, and thus depending on the use-case and the metric to be optimized, the first one may yield a better efficiency. We may recall that we are using the minimal polynomial π_ω of ω, which can only be made possible if $d|[\mathbb{F} : K]$.

2-RTK algorithm. For the first construction, we require, on top of the condition $d|[\mathbb{F} : K]$, 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 ω is 0 since π_ω divides $\mathbf{s}$ and the evaluation in ω is invariant through the naive fold. Remember that as explained in the previous section, the algorithm that takes as input an ω_d-encoding $\mathbf{x}$ and returns $\mathbf{y} = \mathbf{x} + \mathbf{r}$ where $\mathbf{r}$ is generated by such an α-RTK generator of encodings of 0 is α-KIOS.

Algorithm 4 PolyGenZero

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

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

1-RTK algorithm.

The second RTK algorithm that we detail here is very similar to the refreshing procedure of Algorithm 7 that cuts the bilinear dependencies of our optimized RTIK multiplication gadget. We detail the instantiation of this RTK algorithm with Karatsuba's multiplication. More details on the associated evaluation matrix $\mathbf{M}_1$ and interpolation matrix $\mathbf{M}_2$ can be found in **??**. We start off by fixing a polynomial $\mathbf{u} \in \mathbb{F}^d$ with the following properties:

$$\text{The Karatsuba evaluation } \mathbf{u}' = \mathbf{M}_1 \mathbf{u} \text{ has all non-zero entries} \tag{5}$$

$$\text{The greatest common divisor of } \mathbf{u}(X) \text{ and } \pi_\omega(X) \text{ is } X - \omega. \tag{6}$$

We store the fix evaluation vector $\mathbf{u}'$. Then, Algorithm 5 samples a uniformly random polynomial $\mathbf{r} \in \mathbb{F}^d$, which therefore encodes a uniformly random value. We compute its Karatsuba evaluation of $\mathbf{r}' = \mathbf{M}_1\mathbf{r}$, and multiply this vector with $\mathbf{u}'$ coordinate-wise to obtain $\mathbf{x}' = \mathbf{r}' \odot \mathbf{u}'$. Finally, we return $\mathbf{s} = \mathbf{F}\mathbf{M}_2\mathbf{x}'$, which is the folding of the Karatsuba's interpolation of $\mathbf{x}'$.

Algorithm 5 KaratsubaRTK

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

1: $\mathbf{r} \leftarrow \mathbb{F}^{d-1}$
2: $\mathbf{r}' = \mathbf{M}_1\mathbf{r}$
3: $\mathbf{x}' = \mathbf{r}' \odot \mathbf{u}'$
4: $\mathbf{s} = \mathbf{F}\mathbf{M}_2(\mathbf{x}')$
5: return $\mathbf{s}$

Proposition 5.2. *If* $\deg_K(\omega) = d$ *and the vector* $\mathbf{u} \in \mathbb{F}^d$ *is such that Equations* (5) *and* (6) *hold, then Algorithm 5 is a* 1*-RTK generator of* ω_d*-encodings of* 0*.*

6 Subquadratic multiplication gadgets

In this section, we show that the FFT-based multiplication gadget from GPRV [GPRV21] can be proven secure in the region-probing model - provided that there is sufficient structure in $\mathbb{F}$ for the targeted number of shares. The framework that we prove secure in the first subsection is actually a generalization of GPRV, where the evaluation-interpolation polynomial multiplication algorithm used does not have to be the FFT, but *any* evaluation-interpolation-based multiplication gadget. There is a counterpart for using a polynomial multiplication with low bilinear multiplication complexity: roughly speaking, the fewer bilinear multiplications, the lower the upper bound on the available number of shares. In the second subsection, we detail an optimized version of the previous construction based on Karatsuba's multiplication. This masked multiplication gadget is RTIK (Thus in the proper setting, it is region-probing secure) and performs competitively well (see Appendix A for detailed comparison with existing gadgets.) The mutliplication gadgets presented in this section verify the extra composability condition from Theorem 4.4.

6.1 (Re)Revisited Quasilinear masked multiplication: Region-probing security proof for GPRV

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. The FFT instantiation of this transformation is GPRV's multiplication.

Definition 6.1 (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 the assumption that the circuits computing the evaluation and interpolation of the FFT are t-probing secure for some t. The assumption can be tested by exhausting the subsets of probes for a given size among the circuits, which is only possible for small number of shares. Our gadgets on the other hand are proven RTIK, which in turn yields region-probing security through Lemma 4.1. Our gadgets are thus theoretically sound, since they rely 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 $d \leqslant \log|\mathbb{F}|$). This condition is $d \leqslant k$ where $k = [\mathbb{F} : K]$, in order to apply Lemma 4.1. To be specific, K is defined as the subfield such that $\mathcal{M}$ is a K-IM, as defined in Definition 6.1. In other words, K is the smallest subfield of $\mathbb{F}$ such that the evaluation and interpolation operations induced by $\mathcal{M}$ are K-linear.

Intuition of the transformation. 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 $\mathbf{a}' = \mathbf{M}_1\mathbf{a}$ and $\mathbf{b}' = \mathbf{M}_1\mathbf{b}$. Then, $\widehat{\mathcal{M}}$ computes the evaluation $\mathbf{x}' = \mathbf{a}' \odot \mathbf{b}'$ of the product $\mathbf{ab}$ by multiplying coordinate-wise their evaluations. Before proceeding to interpolation, we need to cut the bilinear dependencies between $\mathbf{a}, \mathbf{b}$, which 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. $\widehat{\mathcal{M}}$ 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 ω 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 5.1.[6]

[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.

Intuition of the security proof. By definition of K, all the wires in the evaluation and interpolation subcircuits are K-linear. When the adversary probes an $x_i = a'_i b'_i$, the reduction gives him both factors a'_i, b'_i, which we recall are K-linear functions of $\mathbf{a}, \mathbf{b}$. The effect of the refresh is to create a third independent encoding $\mathbf{c}$ (the output of the gadget), together with a third probing region in which the probes are reducible to K-linear functions of $\mathbf{c}$. 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 $T(d) \log T(d)/2$. When the folding matrix $\mathbf{F}$ does not exist, one can use the NaiveFold algorithm instead. Probes in the NaiveFold of the form $(z_i + \omega^d z_{d+i})$ are reduced to (z_i, z_{d+i}), doubling the total number of probes of the adversary in the circuit.

Algorithm 6 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 5.1) and two input encodings $\mathbf{a}, \mathbf{b} \in \mathbb{F}^d$

Ensure: $\mathbf{c} \in \mathbb{F}^d$ such that $\omega_d^T \mathbf{a} \cdot \omega_d^T \mathbf{b} = \omega_d^T \mathbf{c}$

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: $\mathbf{x}' = \mathbf{a}' \odot \mathbf{b}'$ ▷ Component-wise multiplication of evaluations
4: $\mathbf{y}' = \mathcal{R}(\mathbf{x}', \mathbf{M}_2^T \omega_{2d-1})$ ▷ 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 6.2. *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 6 with $\mathcal{M}$ is a correct RTIK multiplication gadget.*

6.2 Efficient Karatsuba-based multiplication gadget

In this subsection, we detail an optimized version of the GPRV-type transformation from the previous subsection. The optimizations come from various technical improvements detailed below. We assume in the description of Algorithm 7 that d is a divisor of k, where k is the degree of $\mathbb{F}$ over its prime field. This assumption allows us to work with the degree d minimal polynomial π of ω over K, hence use the folding matrix Section 5.1.

Choice of Karatsuba's multiplication. Chosing particularly Karatuba's multiplication benefits our algorithm in several ways. Firstly, while Karatsuba's algorithm is asymptotically beaten by other algorithms, it is the most efficient multiplication in the masking range. Second, the subfield K associated to Karatuba's algorithm is $\mathbb{F}$'s prime field, which maximizes the degree k of $\mathbb{F}/K$. Remind that in our framework, the maximum number of probes per region is $k - 1$. Finally,

Karatsuba's algorithm verifies a crucial property for the randomness optimization detailed below.

Linear randomness. The transformation presented in Section 6.1 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. The encoding $\mathbf{x}'$ to be refreshed is even compressed into the ω_d-encoding $\mathbf{c}$, thus a single ω_d-encoding of 0 is enough entropy to mask $\mathbf{x}'$. To refresh $\mathbf{x}'$ into $\mathbf{y}'$, we compute $\mathbf{x}' = \mathbf{y}' + \mathbf{r}' \odot \mathbf{u}'$ as follows. We sample a completely uniform ω_d-encoding $\mathbf{r}$ from $\mathbb{F}^d$, and compute its Karatsuba's evaluation $\mathbf{r}' = \mathbf{M}_1\mathbf{r}$. We then multiply this vector $\mathbf{r}'$ coordinate-wise with a fixed vector $\mathbf{u}'$ and add this vector to $\mathbf{x}'$ to obtain $\mathbf{y}'$. This vector $\mathbf{u}'$ is the Karatsuba's evaluation of some fixed polynomial $\mathbf{u}$ satisfying the following two properties.

1. We require that $\mathbf{u}$ is such that its evaluation $\mathbf{u}'$ has all non-zero coefficients. This condition allows us to swap the probes of the form r'_i for probes of the form $r'_i u'_i$.
2. We require that the GCD of $\mathbf{u}(X)$ and $\pi(X)$ is $X-\omega$. The first consequence of the latter condition is that $\mathbf{u}(\omega) = 0$, thus $\mathbf{ru}(\omega) = 0$ from which we deduce the correctness of the gadget. The second consequence of this condition is that the reduction modulo (π) of the polynomial $\mathbf{ru}$ is therefore a uniformly random encoding of 0, from which we conclude the mutual independence of $\mathbf{a}, \mathbf{b}, \mathbf{c}$.

Special variant for $d = 2$. We mention that a variant of Algorithm 7, where $\mathbf{r}$ is sampled with an RTC generator of encodings of 0 such as ArithGenZero and $\mathbf{u}$ only has to be such that $\mathbf{u}'$ has all non-zero entries. This variant is also RTIK and uses $\frac{d\log d}{2}$ random elements. While $\frac{d\log d}{2}$ means more random elements than the d random elements needed for Algorithm 7 whenever $d \geqslant 3$, for $d = 2$, this variant uses only one random element versus 2 for Algorithm 7.

Theorem 6.3. *Let $\mathbb{F}$ be a finite field of degree k over its prime field K, $\omega \in \mathbb{F}$ be a fixed element of $\mathbb{F}$, π be the minimal polynomial of ω over K, d be the number of shares and $\mathbf{u} \in \mathbb{F}^d$ a fixed polynomial. Let $\mathbf{M}_1, \mathbf{M}_2$ be the evaluation and interpolation matrices of Karatsuba's multiplication.*

If we have the following three properties:

1. $\deg_K(\omega) = d$
2. $gcd(\mathbf{u}(X), \pi(X)) = X - \omega$
3. $\mathbf{M}_1\mathbf{u} = \mathbf{u}'$ *has all non-zero coefficients*

then karaopti *is a correct RTIK multiplication gadget for ω_d-encodings.*

Algorithm 7 Multiplication gadget $\mathsf{karaopti}(\mathbf{a}, \mathbf{b})$

Require: $\mathbf{a}, \mathbf{b} \in \mathbb{F}^d$ independent encodings
Ensure: $\mathbf{c} \in \mathbb{F}^d$ such that $\omega_d^T \mathbf{a} \cdot \omega_d^T \mathbf{b} = \omega_d^T \mathbf{c}$

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: $\mathbf{r} \leftarrow \mathbb{F}^d$ ▷ Fresh uniform encoding
5: $\mathbf{r}' = \mathbf{M}_1 \mathbf{r}$
6: $\mathbf{s}' = \mathbf{r}' \odot \mathbf{u}'$
7: $\mathbf{y}' = \mathbf{x}' + \mathbf{s}'$ ▷ Refresh
8: $\mathbf{z} = \mathbf{M}_2 \mathbf{y}'$ ▷ Interpolation of the product
9: $\mathbf{c} = \mathbf{F}\mathbf{z}$ ▷ Folding
10: return $\mathbf{c}$

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.

BBC+18. Gilles Barthe, Sonia Belaïd, Gaëtan Cassiers, Pierre-Alain Fouque, Benjamin Grégoire, and François-Xavier Standaert. maskverif: Automated analysis of software and hardware higher-order masked implementations. *Cryptology ePrint Archive*, 2018.

BBC+19. Gilles Barthe, Sonia Belaïd, Gaëtan Cassiers, Pierre-Alain Fouque, Benjamin Grégoire, and François-Xavier Standaert. maskverif: Automated verification of higher-order masking in presence of physical defaults. In *Computer Security–ESORICS 2019: 24th European Symposium on Research in Computer Security, Luxembourg, September 23–27, 2019, Proceedings, Part I 24*, pages 300–318. Springer, 2019.

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.

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.

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.

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.

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.

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.

A Performances comparison

In this section, we provide some extra implementation details, especially about Karatsuba's multiplication and compare our multiplication gadget with the existing multiplication gadgets.

In this section, we give examples of simple and practical instantiations of the algorithms described in Sections 5 and 6. 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 $d \leqslant k$.

A.1 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^ℓ, 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^ℓ. 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

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

With the matrix $\mathbf{M}_1$ definde this way, we can check that the property of Karatsuba's multiplication claimed in the discussion on linear randomness from the previous section is indeed true. When implementing the usual recursive Karatsuba's algorithm, the way the matrix-vector product $\mathbf{M}_1\mathbf{x}$ is evaluated respects the recursion defined above: to evaluate some $\mathbf{M}_1^{2d} \begin{bmatrix} \mathbf{x}_L \\ \mathbf{x}_R \end{bmatrix}$, the algorithm will first evaluate the three matrix-vector products $\mathbf{M}_1^d \mathbf{x}_L, \mathbf{M}_1^d(\mathbf{x}_L + \mathbf{x}_R), \mathbf{M}_1^d$ and so on until it reaches the maximum depth of recursion at the base case. To climb back the recursion, a single addition is necessary at each step, and all 3 evaluations per recursion are indeed added to the output.

A.2 Comparison of the performances of multiplication gadgets.

In this subsection, we give comparison tables of our multiplication gadget Algorithm 7 with the following multiplication gadgets:ISW[ISW03](we consider the arithmetic encoding variant from [RP10]), both the randomness optimized variant and the bilinear multiplications optimized variants of [BBP+17] denoted respectively Belaid rand and Belaid bil, and finally the multiplication gadget from [GJR18](we actually consider the so called gadget GJR+ from [GPRV21]). We then compare for $d \in \{2, 4, 8\}$ various metrics: the number of bilinear multiplications, the randomness cost (in field elements) and the estimated probing ratio. We chose $d \in \{2, 4, 8\}$ because in the benchmark setting of the AES field, these specific values of d are the most suited to our techniques (we can use the folding matrix), and thus we can highlight the efficiency of Algorithm 7. We nonetheless remind the reader that while ISW, Belaïd rand and Belaïd bil exist and have a security proof for any d, GJR+ is only secure when the underlying field is of exponential size (or if the probing security of the FFT and inverse FFT can be computationally checked) and Algorithm 7 requires $\mathbb{F}$ to be an extension field of degree at least d to have a security proof, and requires d to be a factor of the degree of this extension for improved efficiency.

	ISW	Belaïd bil	Belaïd rand	GJR+	GPRV	Algorithm 7
Bilinear mul	d^2	$2d-1$	d^2	$2d$		$d^{\log 3}$
Randomness	$\frac{d(d-1)}{2}$	$2(d-1)^2 + \frac{(d-1)(d-2)}{2}$	$d-1$	$d\log(2d)$		d
t-threshold	$d-1$	$d-1$	$d-1$	$d/2-1$	$d/2-1$	$d-1$
Condition*	YES	NO	NO	NO	NO	YES

Fig. 2. Comparison table of multiplication gadgets for a number of shares d.

$d=2$					
	ISW	Belaïd bil	Belaïd rand	GJR+	Algorithm 7
Bilinear mul	4	3	4	4	3
Randomness	1	2	1	8	2
Probing ratio	7.7%	4.7%	6.3%	3.1%	4.5%
$d=4$					
Bilinear mul	16	7	16	8	9
Randomness	6	21	3	24	4
Probing ratio	5.5%	2.2%	4.7%	1.9%	4.7%
$d=8$					
Bilinear mul	64	15	64	16	27
Randomness	28	119	7	64	8
Probing ratio	3.2%	1.0%	2.7%	1.2%	3.6%

Fig. 3. Comparison table of multiplication gadgets for a number of shares $d \in \{2, 4, 8\}$.

We mean by Bilinear mul in Figures 2 and 3 the number of products between two variables in $\mathbb{F}$ during a single run of the multiplication gadget. Randomness denotes the number of random field elements from $\mathbb{F}$ are required for a single run of the multiplication algorithm. t-threshold denotes the maximum number of tolerated probes (per region in the circuit), and the line condition* corresponds to the answer to the following question for the corresponding multiplication gadget: Can we compose the multiplication without refreshing its inputs and outputs ? Finally, in Figure 3, the Probing ratio is the estimation of the random probing probability p, which we compute as the ratio between the number of probes tolerated per region divided by the size of the largest region-probing subcircuit. The gadgets that are secure in t-probing model are thus considered as having a single subcircuit.

The probing ratio is mostly an indication of the level of security that these multiplication gadgets offer, but the concrete evaluation of the latter should be done in a much more involved way to be more than an indication. To estimate the number of wires in the largest subcircuit of Algorithm 7, we upper bound the number of operations to compute the matrix-vector product of the folding matrix $\mathbf{F}$ by $d + d^2$.

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.

B Proofs of Section 3

Proof of Proposition 3.4

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

$$(\mathbf{v}^T\mathbf{x}_1 \ \ldots \ \mathbf{v}^T\mathbf{x}_n) = \left((\mathbf{v}^T\mathbf{x}_1 \ \ldots \ \mathbf{v}^T\mathbf{x}_n)|Q(\mathbf{x}_1, \ldots, \mathbf{x}_n)\right) \tag{7}$$

$$= \left((\mathbf{v}^T\mathbf{x}_1 \ \ldots \ \mathbf{v}^T\mathbf{x}_n)|(P(\mathbf{x}_1, \ldots, \mathbf{x}_n), Q(\mathbf{x}_1, \ldots, \mathbf{x}_n)\right), \tag{8}$$

where Equation (7) follows from independence and Equation (8) 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.

Proof of Proposition 3.6

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 $i \in [\lfloor d/2 \rfloor]$ as follows : P_P^i contains the probes taken from the variables that are together in the ith step of the loop:

$$t_i = r_i + s_i \tag{9}$$

$$t_{\lfloor d/2 \rfloor + i} = r_{\lfloor d/2 \rfloor + i} - s_i. \tag{10}$$

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.

Proof of Proposition 3.8

Proof. Let G be a t-IOS gadget for Definition 3.7. 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 3.7. 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 $P(\mathbf{x}', \mathbf{y}')$.

Proof of Proposition 3.9

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 3.7 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.

C Proofs of Section 5

Proof of Proposition 5.1

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 $\mathbf{r} = \pi_\omega \mathbf{x} \bmod (X^d - \omega^d) = x \cdot (\pi_\omega \bmod (X^d - \omega^d)$. If the greatest common divisor of $\omega^d - \pi_\omega$ and $X^d - \omega^d$ is $X - \omega$,[7] 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 start with a general set of probes P on PolyGenZero chosen by the adversary. 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 aiming for a set of probes satisfying the conditions of Definition 4.5.

Set of probes 1:

$$(P_1, P_2, P_3),$$

with $t_1 + t_2 + t_3 \leqslant$. Any set of at most t probes on PolyGenZero is of this form, as argued above. Since π_ω 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 π_ω has coefficients in K, each coordinate of $\mathbf{s}$ is K-linear in $\mathbf{x}$, therefore P_1' is K-linear in $\mathbf{x}$.

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 $\mathbf{s}$ as follows. 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 $\mathbf{v} \in \mathbb{F}^{t_1'}$. Also, the adversary knows that $\mathbf{s}$ is computed as $\mathbf{s} = \pi_\omega \mathbf{x}$. As the multiplication with π_ω is a full-rank K-linear operation, there exists a matrix $\mathbf{M} \in K^{(d-1)\times(2d-1)}$ such that $\mathbf{M}\mathbf{s} = \mathbf{x}$. Summing up these facts, the adversary knows $\mathbf{M}\mathbf{s} = \mathbf{x}$ and $\mathbf{P}_1'\mathbf{x} = \mathbf{v}$. By substitution, the set of probes given by $\mathbf{P}_2'\mathbf{s} = \mathbf{v}$, where $\mathbf{P}_2' = \mathbf{P}_1\mathbf{M}$ is equivalent to P_1', thus we have $P_2' \geqslant P_1'$.

Set of probes 4:

$$(Q_2, P_3)$$

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

with $|Q_2| \leqslant 4(t_1 + t_2)$ and Q_2 is K-linear in $\mathbf{s}$. The adversary's knowledge out of set of probes 3 is given by the relations:

$$\begin{bmatrix} -\mathbf{I} & \mathbf{I} & \omega^d \mathbf{I} \\ \mathbf{P}_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_L & \mathbf{P}_R \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{s}_L \\ \mathbf{s}_R \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix},$$

where $\mathbf{v}_1, \mathbf{v}_2$ are the values learned by the adversary from his probes, $\mathbf{P}_L$ (respectively $\mathbf{P}_R$) is the first $d-1$ columns of $\mathbf{P}_2$ (respectively the remaining d columns of $\mathbf{P}_2$), and we define $\mathbf{s}_L, \mathbf{s}_R$ accordingly. The set Q_2 is defined as 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:

$$(Q_3, Q_2')$$

verifying the conditions from Definition 4.5. We introduce the notation

$$\mathbf{Q} = \begin{bmatrix} \mathbf{P}_L \\ \mathbf{P}_R \end{bmatrix}.$$

We define $Q_3 = (P_3, \mathbf{Q}\mathbf{r})$, and $Q_2' = \mathbf{Q}\mathbf{s}_L$, and proceed to prove all 4 items from Definition 4.5. We have $|Q_3| \leqslant 2(t_1+t_2)+t_3 \leqslant 2(t_1+t_2+t_3)$ thus 1) holds with $\alpha = 2$. The subset of probes Q_2 is given by $(\mathbf{Q}\mathbf{s}_L, \mathbf{Q}\mathbf{s}_R)$. The relation $\mathbf{Q}\mathbf{s}_R = \omega^{-d}(\mathbf{Q}\mathbf{s}_L - \mathbf{Q}\mathbf{r})$ implies that $\mathbf{Q}\mathbf{s}_R$ is redundant, hence $(Q_3, Q_2') \geqslant (Q_2, P_3)$. Using the sequence of sets of probes above, we conclude that 2) is also verified. The subset of probes Q_3 is K-linear in $\mathbf{r}$ as required by 3). We finish the proof by showing that $(\mathbf{r}|Q_3) \perp Q_2$. 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

$$\begin{bmatrix} -\mathbf{I} & \mathbf{I} & \omega^{-d} \mathbf{I} \\ \mathbf{P}_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{Q} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{s}_L \\ \mathbf{s}_R \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{v}_r \\ \mathbf{v}_r' \\ \mathbf{v}_L \end{bmatrix},$$

for some probed value vectors $\mathbf{v}_r, \mathbf{v}_r', \mathbf{v}_L$. It follows that the marginal distribution of $(\mathbf{r}|Q_3 \cap Q_2)$ is uniform over some affine subspace. 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.

Proof of Proposition 5.2

Proof. Correctness: From the correctness of Karatsuba's multiplication, we have that $\mathbf{M}_2\mathbf{x}' = \mathbf{ru}$, where the multiplication $\mathbf{ru}$ is the polynomial multiplication in $\mathbb{F}[X]$. From the correctness of the folding matrix, we have that $\mathbf{s}(\omega) = \mathbf{r}(\omega)\mathbf{u}(\omega)$, and since Equation (5) ensures that $X - \omega$ divides $\mathbf{u}$, we do have that $\mathbf{s}(\omega) = 0$.

Uniformity: From Equation (6), we have that $\mathbf{u} \wedge \pi_\omega = X - \omega$. It follows that the reduction modulo π_ω maps the dimension $d-1$ vector space of multiples of $\mathbf{u}$ of the form $\mathbf{ru}$ where $\mathbf{r}$ has degree $\leqslant d-2$ to a subspace of dimension $d-1$ of $\mathbb{F}[X]/\pi_\omega(X)$. Since we just proved that $\mathbf{s}(\omega) = 0$, this subspace is the hyperplane of ω_d-encodings of 0, thus by taking $\mathbf{r}$ uniformly random, we obtain a uniformly random output $\mathbf{s}$ over the ω_d-encodings of 0.

RTK: We start off with any set of t probes P, which we parse into $(P_r, P_{r'}, P_{x'}, P_s)$.

Set of probes 2: $P_2 = (P'_{r'}, P_{x'}, P_s)$. The set of probes $P'_{r'}$ is the concatenation of P_r and $P_{r'}$, which is only made of coordinate probes of the form r'_i due to the fact that intermediate values of Karatsuba's evaluation are all outputs(in particular the probes from P_r which are inputs of the evaluation are also outputs).

Set of probes 3: $P_2 = (P'_{x'}, P_s)$. The set $P'_{x'}$ is the concatenation of $P'_{r'}$ and $P'_{x'}$. More precisely, the probes from the set $P'_{r'}$ which are all of the form r'_i are modified into $x'_i = r'_i u'_i$. This is made possible by Equation (5).

Set of probes 4: $P_4 = Q_s$. The set Q_s contains the probes from P_s, as well as the probes from $P'_{x'}$. Since $\mathbf{M}_2$ has all its coefficients in the prime field of $\mathbb{F}$, its coefficients are in particular in K. Similarly, $\mathbf{F}$ has coefficients in K, thus Q_s contains only K-linear probes in $\mathbf{s}$, which completes the proof.

D Proofs from Section 6

Proof of Theorem 6.2

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

$$\omega_d^T \mathcal{M}(\mathbf{a}, \mathbf{b}) = \omega_d^T \mathbf{F}\mathbf{M}_2 \mathcal{R}(\mathbf{M}_1\mathbf{a} \odot \mathbf{M}_1\mathbf{b}, \mathbf{M}_2^T \omega_{2d-1}) \tag{11}$$

$$= \omega_d^T \mathbf{F}\mathbf{M}_2(\mathbf{M}_1\mathbf{a} \odot \mathbf{M}_1\mathbf{b}) = \omega_d^T \mathbf{F}(\mathbf{a} \cdot \mathbf{b}) \tag{12}$$

$$= \omega_{2d-1}^T \mathbf{a} \cdot \mathbf{b}, \tag{13}$$

where Equation (11) is the definition of $\widehat{\mathcal{M}}(\mathbf{a}, \mathbf{b})$, Equation (12) follows from the correctness of $\mathcal{R}$ and $\mathcal{M}$, and Equation (13) holds since $\mathbf{F}$ is crafted so $\mathbf{F}\omega_{2d-1} = \omega_d$. Therefore $\widehat{\mathcal{M}}$ is a valid multiplication gadget.

RTIK. We consider the ω_d-encodings $\mathbf{a}, \mathbf{b}, \mathbf{c}$, 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 be independent encodings. Due to the uniformity of $\mathcal{R}$, $\mathbf{y}'$ is a fresh re-encoding of $\mathbf{x}'$, and therefore $\mathbf{c} = \mathbf{F}\mathbf{M}_2\mathbf{y}', \mathbf{a}, \mathbf{b}$ are mutually independent. Hence 1. from Definition 4.3 is verified by $\mathbf{a}, \mathbf{b}, \mathbf{c}$. We now proceed

to construct an increasing sequence of probes until we reach Q that satisfies 2. 3. and 4.

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 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 3.7 on $\mathcal{R}$. We obtain the sets of probes P_a^R, P_b^R, P_2 verifying Definition 3.7. We add the probes from P_a^R on $\mathbf{a}'$ to P_a to obtain P'_a, and similarly we add the probes from P_b^R on $\mathbf{b}'$ to P_b to obtain P'_b. The probes from P_2 are independent from $((\mathbf{a}', \mathbf{b}')|P'_a, P'_b)$, thus are independent from $((\mathbf{a}', \mathbf{b}')|P'_a, P'_b)$ and may be discarded.

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 $\mathbf{x}'$. The sets of probes Q_a and respectively Q_b are initially defined as P'_a and respectively P'_b. Then, we remove each probe in P'_x, and replace it with two probes on the corresponding coordinate of $\mathbf{a}'$ and $\mathbf{b}'$. These probes are added to Q_a and Q_b respectively.

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.

Proof of Theorem 6.3

Proof. Correctness. We use the same notations as Algorithm 7. We have

$$\begin{aligned}
\omega_d^T \mathbf{c} &= \omega_{2d-1}^T \mathbf{z} && (14)\\
&= \omega_{2d-1}^T \mathbf{M}_2 \mathbf{y}' && (15)\\
&= \omega_{2d-1}^T \mathbf{M}_2 \mathbf{x}' + \omega_{2d-1}^T \mathbf{M}_2 \mathbf{s}' && (16)\\
&= \omega_d^T \mathbf{a} \cdot \omega_d^T \mathbf{b} + \omega_d^T \mathbf{r} \cdot \omega_d^T \mathbf{u} && (17)\\
&= \omega_d^T \mathbf{a} \cdot \omega_d^T \mathbf{b}, && (18)
\end{aligned}$$

where Equation (14) follows from the correctness of the folding matrix, Equation (15) follows form the definition of $\mathbf{z}$, Equation (16) follows from the definition of $\mathbf{y}'$, Equation (17) follows from the correctness of Karatsuba's multiplication, and Equation (18) follows from the fact that $(X - \omega)|\mathbf{u}$, thus $\omega_d^T \mathbf{u} = 0$.

RTIK. Let P be a set of probes. We parse P into

$$P_0 = (P_a, P_{a'}, P_b, P_{b'}, P_{x'}, P_r, P_{r'}, P_{s'}, P_{y'}, P_z, P_c),$$

where P_a contains the probes on $\mathbf{a}$, P'_a contains the probes on $\mathbf{a}'$, P_b contains the probes on $\mathbf{b}$, P'_b contains the probes on $\mathbf{b}'$, $P_{x'}$ contains the probes on $\mathbf{x}'$, P_r contains the probes on $\mathbf{r}$ and $\mathbf{r}'$, $P_{s'}$ contains the probes on $\mathbf{s}'$, $P_{y'}$ contains the probes on $\mathbf{y}'$, and P_z contains the probes on $\mathbf{z}$ and P_c contains the probes on $\mathbf{c}$.

One can check that the latter list of subsets of probes covers any set of probes P on the multiplication gadget that the adversary may select. We now construct sets of probes $(P_i)_{i\leqslant 5}$ such that $P_{i+1} \geqslant P_i$.

Set of probes 2:

$$P_2 = (P_a, P_{a'}, P_b, P_{b'}, P_{x'}, P'_{r'}, P_{s'}, P_{y'}, P_z, P_c).$$

The set of probes $P'_{r'}$ contains the probes from $P_{r'}$ as well as the probes from P_r. By following the usual evaluation algorithm of Karatsuba's multiplication, the computation of $\mathbf{r}' = \mathbf{M}_1\mathbf{r}$ is such that all the wires are coordinates of $\mathbf{r}'$. Thus it turns out that all the probes in $P'_{r'}$ are coordinates of $\mathbf{r}'$.

Set of probes 3:

$$P_3 = (P_a, P_{a'}, P_b, P_{b'}, P_{x'}, P_{s'}, P_{y'}, P_z, P_c).$$

The set of probes $P'_{s'}$ contains the probes from $P_{s'}$, as well as the probes $s'_i = r'_i u'_i$ for each probe in $P'_{r'}$ (which we recall are of the form r'_i). Since none of the u'_is are zero, we do preserve the information of the set of probes hence we have $P_3 \geqslant P_2$.

Set of probes 4:

$$P_4 = (P_a, P_{a'}, P_b, P_{b'}, P'_{x'}, P'_{y'}, P_z, P_c).$$

The set of probes $P'_{x'}$ contains the probes from $P_{x'}$, as well as the probe x'_i for each probe $s'_i \in P'_{s'}$. Similarly, the set of probes $P'_{y'}$ contains the probes from $P_{y'}$, as well as the probe y'_i for each probe $s'_i \in P'_{s'}$. Since we have $s'_i = y'_i - x'_i$, we have that $P_4 \geqslant P_3$.

Set of probes 5:

$$P_5 = (P_a, P'_{a'}, P_b, P'_{b'}, P'_{y'}, P_z, P_c).$$

The set of probes $P'_{a'}$ contains the probes from $P_{a'}$ as well as the probe a'_i for each probe $x'_i = a'_i b'_i$ in $P'_{x'}$. Similarly, $P'_{b'}$ contains all the probes from $P_{b'}$ as well as the probe b'_i for each probe $x'_i = a'_i b'_i$ in $P'_{x'}$. Since all the probes from $P'_{x'}$ are coordinates of $\mathbf{x}$, and that we have $x'_i = a'_i b'_i$, we can discard $P'_{x'}$ and still have $P_5 \geqslant P_4$.

Set of probes 6:

$$P_6 = (Q_a, Q_b, P'_{y'}, P_z, P_c).$$

The set of probes Q_a contains the probes from P_a, as well as the probes from $P'_{a'}$. The probes from $P'_{a'}$ are all coordinates of $\mathbf{a}'$, which are themselves K-linear functions of $\mathbf{x}$, thus Q_a contains only K-linear probes in $\mathbf{a}$. Similarly, the set of probes Q_b contains the probes from P_b as well as the probes from $P'_{b'}$, and the probes from Q_b are K-linear in $\mathbf{b}$.

Set of probes 7:

$$P_7 = (Q_a, Q_b, Q_c).$$

The set of probes Q_c contains all the probes from $P'_{y'}, P_z$ and P_c. Since $\mathbf{M}_2$ has coefficients in K, the probes from $P_{y'}$ are K-linear in $\mathbf{y}'$. Throughout the reduction, we only added to $P_{y'}$ probes that are coordinates of $\mathbf{y}'$ to construct $P'_{y'}$, thus $P'_{y'}$ contains only K-linear probes in $\mathbf{y}'$. Since $\mathbf{F}$ has coefficients in K, the probes from P_z are K-linear in $\mathbf{z}$. The probes on $\mathbf{c}$ are coordinates of $\mathbf{c}$, thus are also K-linear in $\mathbf{c}$. As we have $\mathbf{c} = \mathbf{F}\mathbf{M}_2\mathbf{y}'$, then K-linear probes on $\mathbf{y}'$ are also K-linear probes on $\mathbf{c}$. Similarly, $\mathbf{c} = \mathbf{F}\mathbf{z}$, thus K-linear probes on $\mathbf{z}$ are also K-linear probes on $\mathbf{c}$. In the end, Q_c contains only K-linear probes in $\mathbf{c}$.

Conclusion of the proof. To summarize: We started off with any set of probes P, and created a set of probes $P_7 = (Q_a, Q_b, Q_c) \geqslant P$, such that Q_a, Q_b, and Q_c are respectively K-linear in $\mathbf{a}, \mathbf{b}$ and $\mathbf{c}$. Throughout the reduction, we do increase the amount of probes of the adversary (In all generality we have $|P_7| \geqslant |P|$), but one can check that we still have $|Q_a|, |Q_b|, |Q_c| \leqslant |P|$. To finish the proof, we now argue that $\mathbf{a}, \mathbf{b}$ and $\mathbf{c}$ are mutually independent encodings.

Since $\mathbf{a}, \mathbf{b}$ are the two inputs of the gadgets, we can assume that these two encodings are mutually independent. To prove that $\mathbf{c}$ is independent of $\mathbf{a}, \mathbf{b}$, we notice the following:

$$\mathbf{c} = \mathbf{a}\mathbf{b} + \mathbf{r}\mathbf{u} \mod \pi. \tag{19}$$

Remind that the effect of the folding matrix is to reduce its input modulo π, where π is the minimal polynomial of ω over K. Since we chose $\mathbf{u}$ such that the GCD of $\mathbf{u}$ and π is $X - \omega$, then we have that $\mathbf{u}\mathbf{r} \mod \pi$ is distributed uniformly random among the ω_d-encodings of 0. Thus from Equation (19), we do have that $\mathbf{c}$ is a fresh ω_d-encoding of $\omega_{2d-1}^T \mathbf{a}\mathbf{b}$, which completes the proof.