



Exploiting Algebraic Structures in Probing Security

Maxime Plançon^{1,2}(✉)

¹ IBM Research Europe, Zurich, Switzerland
mpl@zurich.ibm.com

² ETH Zurich, Zurich, Switzerland

Abstract. The so-called ω -encoding, introduced by Goudarzi, Joux and Rivain (Asiacrypt 2018), generalizes the commonly used arithmetic encoding. By using the additional 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

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

1 Introduction

Since their introduction in the late 90's by Kocher [KJJ99, Koc96], side-channel attacks have proven to be a major threat to cryptography. While cryptanalysis can evaluate the black-box security of cryptographic protocols, their security can be totally compromised by physical attacks. In a nutshell, side-channel attacks refer to any attack taking advantage of the implementation of a cryptographic 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, \dots, 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, CPR13], 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 $\mathbf{x} = (x_1, \dots, x_d)$ of some field element $x \in \mathbb{F}$ are such that $x_1 + \dots + x_d = x$. 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 $\mathbf{c} = \mathbf{a}\mathbf{b}$ 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 $a_1 + \dots + a_d$ 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 Results and Technical Overview

From a high level, this paper is a retake on the circuit compiler from [GPRV21], and proposes a region-probing secure masked compiler for arithmetic circuits over extension fields. The contributions of this paper are listed in 4 categories:

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, and security reductions from the latter notions to region-probing security.
4. Examples of competitively efficient multiplication gadgets and refresh gadgets achieving the aforementioned notions, constituting our masked compiler.

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 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 Subject. 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 $k \geq 2$. An even more concrete example is taking \mathbb{F} to be the 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 $\omega_d = (1, \omega, \dots, \omega^{d-1})$, 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 the single Lemma 2. Consider that our circuit \mathcal{C} takes as input an ω_d -encoding \mathbf{x} . In a nutshell, Lemma 2 says that under the conditions that

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

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 or equal to 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. The former on the other hand implies security regardless of the direction of the affine subspace in which the encoding lies, provided that the latter subspace is directed by the kernel of 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-Independent- K -Linear) 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 $(\mathbf{x}_1, \dots, \mathbf{x}_n)$ and sets of probes (Q_1, \dots, 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. 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 implementation. 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.¹ 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 Sect. 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 8. 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 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

¹ 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.

\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.² The multiplication gadget GJR+ [GPRV21] has two security proofs, depending on the size of \mathbb{F} (and to some extent d). When $|\mathbb{F}| \geq 2^\lambda$ 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 $d \leq \lceil \mathbb{F} : K \rceil$. 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 choose 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 8 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.³ The intuition behind the optimizations is detailed in Sect. 6. We compare the performances of our optimized multiplication gadget with a few existing constructions in Fig. 1. We highlight that Algorithm 8 and ISW are the only multiplication gadgets that can be securely composed without extra refreshing. In terms of bilinear multiplication, Algorithm 8 is worse than GJR+ and Belaïd bil [BBP+17], but better than Belaïd rand [BBP+17] and ISW. In terms of randomness, Algorithm 8 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 the full

² 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} .

³ This multiplication gadget actually behaves as a KIOS refresh with regards to region-probing secure composition. It introduces d random elements to increment the number of regions when composed with other circuits.

version, including estimates of the probing ratio of the gadgets, where Algorithm 8 is also competitive.

	ISW	Belaïd bil	Belaïd rand	GJR+	Algorithm 8
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, Belaïd rand [BBP+17] Alg. 5, Belaïd bil [BBP+17] Alg. 4, and GJR+ [GPRV21]). The composable row answers the question: “Is naive composition of this multiplication gadget secure ?”

1.2 Limitations and Open Questions

Lack of Concreteness. Our contribution mostly stands on the theoretical side. While we give performance comparisons in the full version and make a toy implementation in sage available, the concrete evaluation of the algorithms developed in this paper 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 cyclotomic 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, and appears to be more resilient against glitches (due to the fact that the order in which a computation is made is irrelevant, as long as the wires are K -linear) thus it is also an interesting open question to construct a verification tool for implementations.

Remark 1. The proofs of Lemmas, 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

Algebra. 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 $\pi_K(\omega)$. The notation $[n]$ shall denote the set $\{1, \dots, n\}$.

Distributions. 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. For random variables X, Y , we write $X \perp Y$ when X is independent of Y . For a random variable X and a set A in the domain of X , we use the standard notation $X(A) = \sum_{a \in A} X(a)$. We write $(X|Y)$ the conditional probability of X given Y . To ease the notations, we write $(X|Y, Z) = (X|(Y, Z))$.

Circuits. 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 ρ 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 $P = (p_1, \dots, p_n)$ of a circuit with input χ and internal randomness ρ , we write $P(\chi, \rho) = (p_1(\chi, \rho), \dots, p_n(\chi, \rho))$. When ρ is not in the argument of P , we shall write $P(\chi)$ the random variable $P(\chi, \rho)$ for a uniformly random ρ . 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 usually write \mathcal{W} its set of wires, and we shall write $|\mathcal{W}|$ the number of intermediate variables of \mathcal{C} .

2.2 Masking

Encodings. For a vector $\mathbf{v} \in (\mathbb{F} \setminus \{0\})^d$, 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 $\boldsymbol{\omega}_d = (\omega^i)_{0 \leq i \leq d-1}$. We say that a vector $\mathbf{x} \in \mathbb{F}^d$ is an $\boldsymbol{\omega}_d$ -encoding of a field element $x \in \mathbb{F}$ when $\boldsymbol{\omega}_d^T \mathbf{x} = x$ (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 superscript \mathbf{v} when it is clear from context, and we notice that $H_x^{\boldsymbol{\omega}_d}$ can also be seen as the set of degree d polynomials \mathbf{x} such that $\mathbf{x}(\omega) = x$. We define $\mathcal{U}_\mathbf{v}(x)$ to be the uniform distribution over $H_x^\mathbf{v}$, and extend it coordinate-wise when applied on multiple entries. We say that $(\mathbf{x}_1, \dots, \mathbf{x}_n)$ are mutually independent $\boldsymbol{\omega}_d$ -encodings when for all x_1, \dots, x_n , the distributions $(\mathbf{x}_1 | \boldsymbol{\omega}_d^T \mathbf{x}_1 = x_1), \dots, (\mathbf{x}_n | \boldsymbol{\omega}_d^T \mathbf{x}_n = x_n)$ are mutually independent.

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

Security Properties. We define the threshold-probing security game, region-probing security game, the simulation-based Input-Output Separation property for refresh gadgets and the associated composition theorem.

Definition 1 (*t*-probing security game). *Let $n, t \geq 1$, \mathcal{C} be a circuit and \mathcal{W} be its set of intermediate variables. Let χ be the distribution of the input \mathbf{in} of \mathcal{C} and x_1, \dots, x_n be secret random variables following a distribution ϕ . A *t*-probing adversary \mathcal{A} on $(\mathcal{C}, \chi, \phi)$ plays the following game:*

1. *The challenger samples the input \mathbf{in} from χ*
2. *\mathcal{A} chooses a set of probes $P \subset \mathcal{W}$ with $|P| \leq t$*
3. *The challenger runs $\mathcal{C}(\mathbf{in})$ and sends $P(\mathbf{in})$ to \mathcal{A}*
4. *\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 distribution χ of \mathcal{C} contains uniform encodings of the secret inputs, and the decoding of these are the secrets of this circuit that the adversary attempts to guess after probing.

Definition 2 (r -region probing security game). Let $n \geq 1$, $0 < r < 1$, \mathcal{C} be a circuit with input random variable \mathbf{in} following a distribution χ and x_1, \dots, x_n be secret random variables following a distribution ϕ . Let $\mathcal{C}_1, \dots, \mathcal{C}_m$ be subcircuits of \mathcal{C} such that $(\mathcal{C}_1, \dots, \mathcal{C}_m)$ is a disjoint covering of \mathcal{C} , $\mathcal{W}_1, \dots, \mathcal{W}_m$ be the respective sets of intermediate variables of each subcircuit. A r -region probing adversary against $(\mathcal{C}, \chi, \phi)$ with regions $\mathcal{C}_1, \dots, \mathcal{C}_m$ plays the following game :

1. The challenger samples the input \mathbf{in} from χ
2. \mathcal{A} chooses m sets of probes $(P_i \subset \mathcal{W}_i)_{i \leq m}$ with $|P_i| \leq \lceil r|\mathcal{W}_i| \rceil$
3. The challenger runs $\mathcal{C}(\chi)$ and sends $(P_i(\chi))_{i \leq m}$ to \mathcal{A}
4. \mathcal{A} returns (y_1, \dots, y_n) . He wins if $(y_1, \dots, y_n) = (x_1, \dots, x_n)$.

With identical input distribution χ 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 $\lceil r|\mathcal{C}| \rceil$ -probing secure. When χ and ϕ are clear from context, we simply say that \mathcal{C} is t -probing secure, and similarly for region-probing security. For saving space and improving the readability, we omit the input of the probes when it is clear from context and write P instead of $P(\mathbf{in})$.

Definition 3 (t -input-output separation). Let $\mathbf{v} \in (\mathbb{F} \setminus \{0\})^d$. 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 $|\mathcal{W}| \leq t$, we have that there exists a two-stage simulator $\mathcal{S}_{G^R, \mathcal{W}} = (\mathcal{S}_{G^R, \mathcal{W}}^1, \mathcal{S}_{G^R, \mathcal{W}}^2)$ with the following properties.

1. The first one $\mathcal{S}_{G^R, \mathcal{W}}^1$, returns two sets of indices $\mathcal{I}, \mathcal{J} \subset [d]$ such that $|\mathcal{I}|, |\mathcal{J}| \leq |\mathcal{W}|$.
2. The second one $\mathcal{S}_{G^R, \mathcal{W}}^2$, 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 involving non-arithmetic gadgets.

Theorem 1 (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

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

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 $\mathbf{v} \in (\mathbb{F} \setminus \{0\})^d$).

3.1 Redefining Probing Security Through Sets of Probes and Distribution of Secrets

The t -probing security game, as defined in Definition 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 \leq 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 4 (Partial order of probe sets). *Let P, Q be two sets of probes on a circuit \mathcal{C} , taking as input a random variable \mathbf{in} following a distribution χ and manipulating secret random variables x_1, \dots, x_n following a distribution ϕ . We say that Q contains more information than P , and we write $P \leq Q$, when*

$$((x_1, \dots, x_n) | (P(\mathbf{in}), Q(\mathbf{in}))) = ((x_1, \dots, x_n) | Q(\mathbf{in})).$$

When $P \leq Q$, intuitively, all the sensitive information on the input \mathbf{in} of \mathcal{C} carried by P is also carried by Q . The binary relation \leq 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 chooses 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 $P_3 \geq P_2 \geq P_1$, thus $P_3 \geq P_1$. 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 chooses 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 $P_2 \geq P_1$. 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 $P_3 \geq P_2$. 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 $Q_2 \geq Q_1$. Examples of proofs that rely on an increasing sequence of sets of probes can be found in the proofs of Propositions 5 and 6 and Theorems 5 and 6.

We now proceed to define t -probing security and r -region probing security for masked circuit from a probabilistic perspective.

Definition 5 (t -probing security of linear-masked circuits, convenient version). Let $\mathbf{v} \in (\mathbb{F} \setminus \{0\})^d$, \mathcal{C} be a circuit taking as input \mathbf{v} -encodings $\mathbf{x}_1, \dots, \mathbf{x}_n$ 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 $|P| \leq t$, we have

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

Definition 6 (r -region-probing security of linear-masked circuits, convenient version). Let $\mathbf{v} \in (\mathbb{F} \setminus \{0\})^d$, $0 < r < 1$, \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 disjoint covering of \mathcal{C} , $\mathcal{W}_1, \dots, \mathcal{W}_m$ be the induced sets of intermediate variables of the subcircuits. We let $\mathbf{x}_1, \dots, \mathbf{x}_n$ be the input \mathbf{v} -encodings of \mathcal{C} . Then \mathcal{C} is r -region-probing secure when $\forall P = (P_1, \dots, P_m) \subset \mathcal{W}_1 \times \dots \times \mathcal{W}_m$, with $P_i \subset \mathcal{W}_i$ and $|P_i| \leq \lceil r|\mathcal{C}_i| \rceil$, we have

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

In both definitions, the information learnt by the adversary (i.e. $P(\mathbf{x}_1, \dots, \mathbf{x}_n)$) 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 \leq to region probing security.

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

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

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 $P \leq Q$ 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 5) and for exposing the close relation between KIOS Definition 11 and IOS Definition 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 Algorithms 2 and 4 from an RTC generator of encodings of 0.

Definition 7. (*Reducible-To-Coordinates*) *Let $\mathbf{v} \in (\mathbb{F} \setminus \{0\})^d$, 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. $|Q_1| \leq t$
2. $(Q_1, Q_2) \geq P$
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 \leq is taken with respect to the secret r_0, \dots, 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 2. *Algorithm 1 is RTC with $\mathbf{v} = (1, \dots, 1)$.*

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

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

```

1: if  $d = 1$  then
2:   return 0
3: end if
4: if  $d = 2$  then
5:    $r \leftarrow \mathbb{F}$ 
6:   return  $(-r, r)$ 
7: end if
8:  $(r_0, \dots, r_{\lfloor d/2 \rfloor - 1}) = \text{ArithGenZero}(\lfloor d/2 \rfloor)$ 
9:  $(r_{\lfloor d/2 \rfloor}, \dots, r_{d-1}) = \text{ArithGenZero}(\lceil d/2 \rceil)$ 
10: for  $i = 0$  to  $\lfloor d/2 \rfloor - 1$  do
11:    $s_i \leftarrow \mathbb{F}$ 
12:    $t_i = r_i + s_i$ 
13:    $t_{\lfloor d/2 \rfloor + i} = r_{\lfloor d/2 \rfloor + i} - s_i$ 
14: end for
15: if  $d$  is odd then
16:    $t_{d-1} = r_{d-1}$ 
17: end if
18: return  $\mathbf{t}$ 

```

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 $d \geq 1$ by considering the refresh gadget from Appendix C [BCPZ16].

Definition 8. (*Input-Output Separative*) Let $\mathbf{v} \in (\mathbb{F} \setminus \{0\})^d$, 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. $|Q_x| \leq t, |Q_y| \leq t$
2. $(Q_x, Q_y, Q_2) \supseteq P$
3. Every probe in Q_x is a coordinate of \mathbf{x} and every probe in Q_y is a coordinate of \mathbf{y}
4. The distributions Q_2 and $((\mathbf{x}, \mathbf{y}) | (Q_x, Q_y))$ are independent

Proposition 3. Let $\mathbf{v} \in (\mathbb{F} \setminus \{0\})^d$, 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 8, then it is also t -IOS according to Definition 3 and vice-versa.

Algorithm 2. RTC generator to IOS refresh template

Require: Masking order d , $\mathbf{v} \in (\mathbb{F} \setminus \{0\})^d$, RTC generator of arithmetic encodings of 0 R , \mathbf{v} -encoding \mathbf{x}

Ensure: $\mathbf{y} \in \mathbb{F}^d$ such that $\mathbf{v}^T \mathbf{y} = \mathbf{v}^T \mathbf{x}$

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

Proposition 4. *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 $\omega_d = (1, \omega, \omega^2, \dots, \omega^{d-1})$ 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 Sect. 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 connection 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 1 and 2 that are building blocks for proving t -probing security of ω_d -masked circuits.

From a high level, the first Lemma 1 claims that when $\deg_K(\omega) \geq d$, 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 connection 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 2.

To illustrate the correspondance between K -linear circuits and threshold-probing security, 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 $\deg_K(\omega) \geq d$ is satisfied, Lemma 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 2. This Lemma is central in our framework: every security notion introduced in the next subsection relates to it. The convenient form of Lemma 2 makes it likely to find other applications in constructing efficient masked gadgets.

Lemma 1. *Let \mathbb{F} be a finite field, K be a subfield of \mathbb{F} , $\mathbf{P} \in K^{t \times d}$ such that $\text{rank } \mathbf{P} = t$ and $\omega \in \mathbb{F}$. If $\deg_K(\omega) \geq d$ 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 $\text{rank} \begin{bmatrix} \mathbf{P} \\ \omega_d^T \end{bmatrix} = t$, i.e $\omega_d \in \text{Span } \mathbf{P}^T$.

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 $\mathbf{p}_{t+1}, \dots, \mathbf{p}_d$ 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

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

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

Taking the last row in the last equality, we get $\mathbf{q}^T \boldsymbol{\omega}_d = 0$, and due to the invertibility of \mathbf{Q} , $\mathbf{q} \neq \mathbf{0}$. In other words, the polynomial with coefficients \mathbf{q} cancels ω and has degree at most $d - 1$, which is a contradiction with $\deg_K(\omega) \geq d$, and the claim follows.

Lemma 2. *Let d be an order of masking, \mathcal{C} be a circuit taking as input a uniform $\boldsymbol{\omega}_d$ -encoding \mathbf{x} with $\omega \in \mathbb{F}$. 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 ϕ be the distribution of the secret input x , inducing by uniformity a distribution $\bar{\phi}(\mathbf{x}) = \frac{1}{|\mathbb{F}|^{d-1}} \phi(\omega^T \mathbf{x})$. 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, otherwise some rows of \mathbf{P} are redundant and the matrix \mathbf{P}' obtained by removing redundancy defines a set of probes $P' \geq P$, and is full-rank. For $x \in \mathbb{F}$, $\mathbf{v} \in \mathbb{F}^{d-1}$, we have

$$\mathbb{P}(\omega_d^T \mathbf{x} = x \cap P(\mathbf{x}) = \mathbf{v}) = \mathbb{P}(\omega_d^T \mathbf{x} = x \cap \mathbf{P}\mathbf{x} = \mathbf{v}) \tag{1}$$

$$= \bar{\phi} \left(\ker \begin{bmatrix} \mathbf{P} \\ \omega_d^T \end{bmatrix} + \mathbf{x}^* \right) \tag{2}$$

$$= \bar{\phi}(\mathbf{x}^*) = \frac{1}{|\mathbb{F}|^{d-1}} \phi(x) \tag{3}$$

$$= \mathbb{P}(P(\mathbf{x}) = \mathbf{v}) \cdot \mathbb{P}(\omega_d^T \mathbf{x} = x), \tag{4}$$

where Eq. (1) is the hypothesis of the Lemma, Eq. (2) holds for some solution \mathbf{x}^* to the equation $\begin{bmatrix} \mathbf{P} \\ \omega_d^T \end{bmatrix} \mathbf{x}^* = \begin{bmatrix} \mathbf{v} \\ x \end{bmatrix}$, Equation (3) follows from Lemma 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 Eq. (4) holds because $\mathbb{P}(P(\mathbf{x}) = \mathbf{v}) = \mathbb{P}(\mathbf{x} \in \ker \mathbf{P} + \mathbf{x}') = \frac{1}{|\mathbb{F}|^{d-1}} \sum_{y \in \mathbb{F}} \phi(y) = \frac{1}{|\mathbb{F}|^{d-1}}$, where \mathbf{x}' is an offset vector solution to $P(\mathbf{x}') = \mathbf{v}$. Since we have $\mathbb{P}(P(\mathbf{x}) = \mathbf{v}) = |\mathbb{F}|^{-(d-1)}$ and $\phi(x) = \mathbb{P}(\omega_d^T \mathbf{x} = x)$, we conclude independence.

4.2 Weaker Condition for Region-Probing Security in Extension Fields

In this section, we extend the results of the above subsection to circuits manipulating several $\boldsymbol{\omega}_d$ -encodings. Namely, we introduce the RTIK security notion and show that RTIK circuits are region-probing secure. Rephrasing (and simplifying) the RTIK property: an $\boldsymbol{\omega}_d$ -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 $\boldsymbol{\omega}_d$ -masked encoding.

Definition 9 (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} , \mathcal{W} be the set of wires of \mathcal{C} and $(\mathbf{x}_1, \dots, \mathbf{x}_n)$ be mutually independent ω_d -encodings. We say that \mathcal{C} is RTIK w.r.t $(\mathbf{x}_1, \dots, \mathbf{x}_n)$ when for all set of probes $P \subset \mathcal{W}$, there exists a set of probes $Q = (Q_1, \dots, Q_n) \subset \mathcal{W}$ such that the following holds:

1. $Q \geq P$
2. $\forall i \in [n], |Q_i| \leq |P|$
3. For all $i \in [n]$, every probe in Q_i is a linear function of \mathbf{x}_i over K .

Theorem 2 (Security of RTIK circuits.). Let n, d be integers, \mathcal{C} be a circuit over a finite field \mathbb{F} , K be a subfield of \mathbb{F} , \mathcal{W} be the set of wires of \mathcal{C} , $\omega \in \mathbb{F}$ be a field element such that $\deg_K(\omega) \geq d$ and $(\mathbf{x}_1, \dots, \mathbf{x}_n)$ be mutually independent ω_d -encodings.

If \mathcal{C} is RTIK with respect to $(\mathbf{x}_1, \dots, \mathbf{x}_n)$, then there exists a number $m \geq n$, a ratio r , and m regions $(\mathcal{C}_1, \dots, \mathcal{C}_m)$ such that \mathcal{C} is r -region-probing secure with respect to $(\mathcal{C}_1, \dots, \mathcal{C}_m)$. The probing ratio r is given by

$$\min_{i \in [n]} \left(\frac{d-1}{\sum_{\substack{I \subset [n] \\ s.t. i \in I}} |\mathcal{W}_I|} \right),$$

\mathcal{W}_I and the subcircuits $\mathcal{C}_1, \dots, \mathcal{C}_m$ are explicited in the proof.

Regions and Probing Ratio. Our proof of Theorem 2 is tight for two reasons. First, it is tight in the sense that any ratio r greater than the one defined in the proof leads to an attack in the region-probing model. Second, it is tight in the sense that there exists an RTIK circuit \mathcal{C} (wrt encodings $\mathbf{x}_1, \dots, \mathbf{x}_n$) such that the ratio r satisfies $r|C| = n(d-1)$, which is optimal. The latter justifies an improvement upon the direct reduction from the threshold probing model.

4.3 Composition Notions for RTIK Circuits

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

Theorem 3. 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 is RTIK with respect to encodings $(\mathbf{x}_1^1, \dots, \mathbf{x}_n^1)$
2. \mathcal{C}_2 is RTIK with respect to encodings $(\mathbf{x}_1^2, \dots, \mathbf{x}_m^2)$
3. The intersection of the input encodings of \mathcal{C}_2 and the output encodings of \mathcal{C}_1 is contained in both $(\mathbf{x}_1^1, \dots, \mathbf{x}_n^1)$ and $(\mathbf{x}_1^2, \dots, \mathbf{x}_m^2)$,

then \mathcal{C} is RTIK.

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 aforementioned 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 3 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 2. from Definition 9. While it is possible to construct gadgets that verify 1. 2. and 4. as well as $|Q_i| \leq \alpha|P|$ for some slack factor α (e.g. the NaiveFold algorithm defined in Sect. 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 over i of the individual $\frac{d-1}{\sum_{i \in I} |\mathcal{W}_I|}$, where I is a subset of indices containing i , and \mathcal{W}_I is the set of wires mapped to $|I|$ probes, each on a single encoding $\mathbf{x}_j, j \in I$. When one of the subcircuits

$$\bigcup_{\substack{I \subset [n] \\ i \in I}} \mathcal{W}_I,$$

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 10. (*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) $|Q_1| \leq \alpha t.$
- 2) $(Q_1, Q_2) \geq P$

- 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 $\mathbf{P}\mathbf{x}$ and $\mathbf{P}\mathbf{y}$, 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 11. ((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. $|Q_x| \leq \alpha t, |Q_y| \leq \alpha t$
2. $(Q_x, Q_y, Q_2) \leq P$
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 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 $(\mathbf{x}_1, \dots, \mathbf{x}_n)$ 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. (KIOS Composition Theorem). *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.*

5 Miscellaneous RTIK and KIOS Gadgets

This section contains two ω_d -encodings building-block algorithms for constructing a 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 $d \leq \deg_K(\omega)$ and for correctness of the gadgets presented in this section, we also need $d \geq \deg_K(\omega)$. 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 $d \geq \deg_K(\omega)$ 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 $d_1 \geq d_2$. 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 $d_1 \geq 2d - 1$. 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: $y_i = x_i + \omega^d x_{d+i}$
 - 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 $\deg_K(\omega) \geq d$, therefore, so we can also use Lemma 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 $0 \leq i \leq d-2$ is a polynomial in ω with coefficients in K and degree $\leq d-1$. Let us therefore write π_i the column vector of coefficients of the i -th polynomial, for example $\pi_0 = \pi$. One can check that the matrix

$$\mathbf{F} = [\mathbf{I}_d \ \pi_0 \ \pi_1 \ \dots \ \pi_{d-2}]$$

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 choose 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 choose ω to be a primitive $d+1$ -th root of unity. This way, the minimal polynomial of ω is $1 + X + \dots + X^d$, and $\omega^{d+1} = 1$. Then, for any $0 \leq i \leq d-3$, 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 $d \mid [\mathbb{F} : K]$.

2-RTK Algorithm. For the first construction, we require, on top of the condition $d \mid [\mathbb{F} : K]$, 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: $\mathbf{r} = \text{NaiveFold}(\mathbf{s})$
 - 4: return \mathbf{r}
-

Proposition 5. *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 8 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 the full version of the paper. 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}' .

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

Algorithm 5. KaratsubaRTK**Require:** Masking order d with $d = \deg_K(\omega)$ **Ensure:** $\mathbf{s} \in \mathbb{F}^d$ such that $\mathbf{s}^T \boldsymbol{\omega}_d = 0$

-
- 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: $\mathbf{s} = \mathbf{FM}_2(\mathbf{x}')$
 - 5: return \mathbf{s}
-

5.3 Square Gadget in Characteristic 2

In this subsection, we show that the usual square gadget in characteristic 2 is RTIK. The typical example of use of this gadget is to compute the inverse of an element of \mathbb{F}_{256} in the AES S-box as a subalgorithm of the square-and-multiply computation of the 255-th power. The RTIK security of this gadget falls into the wider class of coordinate-wise gadgets.

The algorithm works as follows: since we are working in characteristic 2, we have the classical identity that for any $x, y \in \mathbb{F}$, $(x + y)^2 = x^2 + y^2$. We apply this identity to the decryption of the encoding \mathbf{x} :

$$\left(\sum_{i=0}^{d-1} x_i \omega^i \right)^2 = \sum_{i=0}^{d-1} x_i^2 \omega^{2i}.$$

In other words, to compute an encoding \mathbf{y} of the square of $\mathbf{x}^T \boldsymbol{\omega}_d$, we can square each coordinate of \mathbf{x} , and multiply the result coordinate-wise with the vector $\mathbf{w} = (\omega^{-i})_{0 \leq i \leq d-1}$. Correctness follows from the latter identity, and since all the operations are coordinate-wise, this gadget is RTIK.

Algorithm 6. SquareGadget**Require:** Encoding $\mathbf{x} \in \mathbb{F}^d$ of length d **Ensure:** $\mathbf{y} \in \mathbb{F}^d$ such that $\mathbf{y}^T \boldsymbol{\omega}_d = (\mathbf{x}^T \boldsymbol{\omega}_d)^2$

-
- 1: $\mathbf{z} = \mathbf{x}^2$ ▷ Coordinate-wise operation
 - 2: $\mathbf{y} = \mathbf{z} \odot \mathbf{w}$
 - 3: return \mathbf{y}
-

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 the full version for detailed comparison with existing gadgets.) The multiplication gadgets presented in this section verify the extra composability condition from Theorem 3.

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 $\hat{\cdot}$ 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 12. (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 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 $d \leq \log |\mathbb{F}|$). This condition is $d \leq k$ where $k = \lceil \mathbb{F} : K \rceil$, in order to apply Lemma 1. To be specific, K is defined as the subfield such that \mathcal{M} is a K -IM, as defined in Definition 12. 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 $\mathbf{a}\mathbf{b}$ 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 $\mathbf{a}\mathbf{b}$. 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 $\mathbf{c} = \mathbf{F}\mathbf{z}$ 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 Sect. 5.1.⁴

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 7. 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}	

⁴ 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.

Theorem 5. *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 7 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 8 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 Sect. 5.1.

Choice of Karatsuba’s Multiplication. Choosing particularly Karatsuba’s multiplication benefits our algorithm in several ways. Firstly, Karatsuba’s algorithm offers a trade-off between the size of the circuit and the number of bilinear multiplications that is advantageous for degrees relevant to masking in practice (e.g. between 2 and a few dozens). Second, the subfield K associated to Karatsuba’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 Sect. 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 $\mathbf{r}\mathbf{u}(\omega) = 0$ from which we deduce the correctness of the gadget. The second consequence of this condition is that the reduction modulo (π) of the polynomial $\mathbf{r}\mathbf{u}$ 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 8, 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 8 whenever $d \geq 3$, for $d = 2$, this variant uses only one random element versus two for Algorithm 8.

Algorithm 8. Multiplication gadget $\text{karaopti}(\mathbf{a}, \mathbf{b})$

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:	$\mathbf{c} = \mathbf{Fz}$	▷ Folding
10:	return \mathbf{c}	

Theorem 6. 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. We assume that the two entries \mathbf{a}, \mathbf{b} are mutually independent encodings.

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 with respect to $\mathbf{a}, \mathbf{b}, \mathbf{c}$.

References

- [ADF16] Andrychowicz, M., Dziembowski, S., Faust, S.: Circuit compilers with $O(1/\log(n))$ leakage rate. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9666, pp. 586–615. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_21
- [AIS18] Ananth, P., Ishai, Y., Sahai, A.: Private circuits: a modular approach. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018. LNCS, vol. 10993, pp. 427–455. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96878-0_15

- [BBC+18] Barthe, G., Belaïd, S., Cassiers, G., Fouque, P.-A., Grégoire, B., Standaert, F.-X.: maskverif: Automated analysis of software and hardware higher-order masked implementations. *Cryptology ePrint Archive* (2018)
- [BBC+19] Barthe, G., Belaïd, S., Cassiers, G., Fouque, P.-A., Grégoire, B., Standaert, F.-X.: maskVerif: automated verification of higher-order masking in presence of physical defaults. In: Sako, K., Schneider, S., Ryan, P.Y.A. (eds.) *ESORICS 2019*. LNCS, vol. 11735, pp. 300–318. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29959-0_15
- [BBD+16] Barthe, G., et al.: Strong non-interference and type-directed higher-order masking. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 116–129 (2016)
- [BBP+16] Belaïd, S., Benhamouda, F., Passelègue, A., Prouff, E., Thillard, A., Vergnaud, D.: Randomness complexity of private circuits for multiplication. In: Fischlin, M., Coron, J.-S. (eds.) *EUROCRYPT 2016*. LNCS, vol. 9666, pp. 616–648. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_22
- [BBP+17] Belaïd, S., Benhamouda, F., Passelègue, A., Prouff, E., Thillard, A., Vergnaud, D.: Private multiplication over finite fields. In: Katz, J., Shacham, H. (eds.) *CRYPTO 2017*. LNCS, vol. 10403, pp. 397–426. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63697-9_14
- [BC22] Bronchain, O., Cassiers, G.: Bitslicing arithmetic/boolean masking conversions for fun and profit with application to lattice-based KEMs. *Cryptology ePrint Archive* (2022)
- [BCLV17] Bernstein, D.J., Chuengsatiansup, C., Lange, T., van Vredendaal, C.: NTRU prime: reducing attack surface at low cost. In: Adams, C., Camenisch, J. (eds.) *SAC 2017*. LNCS, vol. 10719, pp. 235–260. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-72565-9_12
- [BCP+20] Belaïd, S., Coron, J.-S., Prouff, E., Rivain, M., Taleb, A.R.: Random probing security: verification, composition, expansion and new constructions. In: Micciancio, D., Ristenpart, T. (eds.) *CRYPTO 2020*. LNCS, vol. 12170, pp. 339–368. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-56784-2_12
- [BCPZ16] Battistello, A., Coron, J.-S., Prouff, E., Zeitoun, R.: Horizontal side-channel attacks and countermeasures on the ISW masking scheme. In: Gierlichs, B., Poschmann, A.Y. (eds.) *CHES 2016*. LNCS, vol. 9813, pp. 23–39. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53140-2_2
- [BRT21] Belaïd, S., Rivain, M., Taleb, A.R.: On the power of expansion: more efficient constructions in the random probing model. In: Canteaut, A., Standaert, F.-X. (eds.) *EUROCRYPT 2021*. LNCS, vol. 12697, pp. 313–343. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77886-6_11
- [BRTV21] Belaïd, S., Rivain, M., Taleb, A.R., Vergnaud, D.: Dynamic random probing expansion with quasi linear asymptotic complexity. In: Tibouchi, M., Wang, H. (eds.) *ASIACRYPT 2021*. LNCS, vol. 13091, pp. 157–188. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-92075-3_6
- [CGMZ21] Coron, J.-S., Gérard, F., Montoya, S., Zeitoun, R.: High-order polynomial comparison and masking lattice-based encryption. *Cryptology ePrint Archive* (2021)
- [CPRR13] Coron, J.-S., Prouff, E., Rivain, M., Roche, T.: Higher-order side channel security and mask refreshing. In: Moriai, S. (ed.) *FSE 2013*. LNCS, vol. 8424, pp. 410–424. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43933-3_21

- [CS20] Cassiers, G., Standaert, F.-X.: Trivially and efficiently composing masked gadgets with probe isolating non-interference. *IEEE Trans. Inf. Forensics Secur.* **15**, 2542–2555 (2020)
- [DDF14] Duc, A., Dziembowski, S., Faust, S.: Unifying leakage models: from probing attacks to noisy leakage. In: Nguyen, P.Q., Oswald, E. (eds.) *EUROCRYPT 2014*. LNCS, vol. 8441, pp. 423–440. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55220-5_24
- [DFS15] Dziembowski, S., Faust, S., Skorski, M.: Noisy leakage revisited. In: Oswald, E., Fischlin, M. (eds.) *EUROCRYPT 2015*. LNCS, vol. 9057, pp. 159–188. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_6
- [DVBV22] D’Anvers, J.-P., Van Beirendonck, M., Verbauwhede., I.: Revisiting higher-order masked comparison for lattice-based cryptography: algorithms and bit-sliced implementations. *Cryptology ePrint Archive* (2022)
- [GJR18] Goudarzi, D., Joux, A., Rivain, M.: How to securely compute with noisy leakage in quasilinear complexity. In: Peyrin, T., Galbraith, S. (eds.) *ASIACRYPT 2018*. LNCS, vol. 11273, pp. 547–574. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03329-3_19
- [GPRV21] Goudarzi, D., Prest, T., Rivain, M., Vergnaud, D.: Probing security through input-output separation and revisited quasilinear masking. *IACR Trans. Cryptograph. Hardw. Embed. Syst.* 599–640 (2021)
- [ISW03] Ishai, Y., Sahai, A., Wagner, D.: Private circuits: securing hardware against probing attacks. In: Boneh, D. (ed.) *CRYPTO 2003*. LNCS, vol. 2729, pp. 463–481. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45146-4_27
- [KJJ99] Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) *CRYPTO 1999*. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1_25
- [Koc96] Kocher, P.C.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Kobitz, N. (ed.) *CRYPTO 1996*. LNCS, vol. 1109, pp. 104–113. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-68697-5_9
- [PR13] Prouff, E., Rivain, M.: Masking against side-channel attacks: a formal security proof. In: Johansson, T., Nguyen, P.Q. (eds.) *EUROCRYPT 2013*. LNCS, vol. 7881, pp. 142–159. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38348-9_9
- [RP10] Rivain, M., Prouff, E.: Provably secure higher-order masking of AES. In: Mangard, S., Standaert, F.-X. (eds.) *CHES 2010*. LNCS, vol. 6225, pp. 413–427. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15031-9_28