

Evaluating and Engineering Masking Security: Random Probing Tools and Deterministic Gadgets

Proefschrift ter verkrijging van de graad van doctor
aan de Radboud Universiteit Nijmegen
op gezag van de rector magnificus prof. dr. J.M. Sanders,
volgens besluit van het college voor promoties
in het openbaar te verdedigen op

woensdag 4 november 2026
om 14.30 uur precies

door

Vahid Jahandideh Khajeh

Promotoren:

prof. dr. Lejla Batina

prof. dr. Bart Mennink (Universiteit Maastricht)

Manuscriptcommissie:

prof. dr. Peter Schwabe

prof. dr. ing. Tim Güneysu
(Ruhr-Universität Bochum, Duitsland)

prof. dr. Sebastian Faust
(Technische Universität Darmstadt, Duitsland)

prof. dr. Stefan Dziembowski
(Uniwersytet Warszawski, Polen)

dr. Sonia Belaïd
(CryptoExperts, Frankrijk)

Acknowledgments

Many people have supported me on the long path that eventually led to this PhD, and I would like to thank them here.

My first and deepest academic gratitude goes to my master's supervisor, Mahmoud Salmasizadeh. I first met him seventeen years ago in his office, when he was advising me which courses to take during my master's. From that day on, he always listened patiently to my rough ideas and half-formed explanations—with a smile—and, more importantly, with trust. That trust became a lasting source of motivation for me to keep doing research alongside many other projects and responsibilities, from then until now. After I finished my master's, he gave me the opportunity to prepare and teach lectures on several cryptography topics, which was an enjoyable and formative experience. During the COVID lockdown, Dr. Daneshgar sometimes joined our meetings; his criticism and questions helped me to sharpen my early ideas.

I am also truly grateful to my close friend Amir Mortazavi. Our friendship began in high school twenty-four years ago and continued through our BSc and MSc studies and later work and projects. I have learned countless things from him, and I am thankful for his constant encouragement and support. Mahmoud, Amir, and others around me were among the people who encouraged me to apply for a PhD abroad.

When I first started exploring this life-changing idea, I came across a familiar name: Shahram Rasoolzadeh (then a postdoctoral researcher in the Digital Security Group at Radboud University). I knew Shahram from earlier lectures, when he was a master's student taking our course. I wrote to him immediately and asked for advice. He kindly introduced me to Lejla Batina and Joan Daemen, and only a few months later I began a new life as a PhD student in the Netherlands.

Moving to the Netherlands brought many cultural surprises. Meetings scheduled for 10:00 started at 10:00 (not 10:00 plus a random 1–15 minutes). The strict hierarchy I had known for many years was simply not there. I still

remember that my first email began with “Prof. Mennink,” and Bart replied kindly: “Just call me Bart.” It felt strange at first to address supervisors by their first names, and even not to stand up when they entered the room. On my first day, I was saying “thank you” after almost every point; Bart stopped me and said, with a smile, “No more ‘thank you’s today.” In those first months, cultural differences were at least as striking to me as the scientific challenges.

I have learned an enormous amount from Bart Mennink—more than he probably realizes. He was always available for questions, always willing to help, and his speed, talent, openness, and humility made working with him truly enjoyable.

I am deeply thankful to Lejla Batina for trusting me and giving me the freedom to explore my own ideas and research directions. That freedom played an essential role in shaping both my academic development and several core parts of this thesis. Throughout my PhD, she was consistently supportive, approachable, and generous with her time. I especially appreciated her kindness and encouragement during the more difficult stages of the research, when her support made a real difference. I am very grateful for her guidance, confidence, and constant willingness to help.

I am also grateful to Joan Daemen for several fruitful discussions. He was always open to new ideas: you could step into his office, and he would listen carefully and enthusiastically. In particular, Chapter 7 benefited greatly from our early discussions.

I would like to thank Jan for many discussions and for the great time as office mates. He repeatedly amazed me with his mathematical and logical talent. We collaborated on results that are presented in Chapter 7. In several of our brainstorming sessions, Koustabh was also actively involved, and I learned a lot from him as well. I also enjoyed our conversations during long train trips.

Daily life in the group was made much lighter by friends and colleagues. Seyyed was often my first go-to person for a coffee break—sometimes after a few hours of work, sometimes even before that—together with Mostafa and Zahra. During part of the thesis-writing period, we were also receiving sad news from Iran, which made meeting deadlines emotionally difficult at times.

I am thankful to my other office mates as well—Daniel, Bobby, Solane, and Bart—for the supportive atmosphere. I feel lucky to have worked with such kind people.

In my first year of the PhD, I had many questions and technical challenges, and Léo was often my first point of contact. He was always available and generous with his time, and we also collaborated on my first paper.

I also would like to express my gratitude to all current and former members of the CESCA and ESCADA groups with whom I exchanged ideas and had

inspiring discussions.

Last but not least, my most heartfelt thanks go to Mitra, my beloved wife. Throughout these years she has been my steady support. I borrowed many evenings, weekends, and holidays to meet deadlines, and she carried that burden with patience, kindness, and understanding. Her encouragement helped me through the difficult moments, and her presence made the good moments even better. I am also deeply grateful to our parents. Starting this journey meant leaving them behind and being far from home, which is never easy. Their love, support, and sacrifices made it possible for us to build a life in the Netherlands and to complete this PhD.

Contents

List of Abbreviations	xi
1 Introduction	1
1.1 Research Questions	3
1.2 Contributions	4
1.2.1 Line I: Noisy Leakage and Random Probing (Chapters 3–5)	4
1.2.2 Line II: Deterministic Masking (Chapters 6–7)	6
1.3 Author Contributions in the Accompanying Papers	7
1.4 Thesis Outline	8
2 Preliminaries	9
2.1 Notation and Basic Tools	9
2.2 Mathematical Background	10
2.3 Circuit Model and Masking	12
2.4 Leakage Models	13
2.5 Security Metrics and Security Notions	14
2.6 From Noisy Leakage to Random Probing	15
2.6.1 Statistical Distance and δ -Noisy Leakage	15
2.6.2 Doeblin Coefficient and Leakage Simulation via Erasures	16
2.6.3 Binary Tightness	16
2.6.4 Example: Leakage of One Bit	16
2.6.5 A Useful Bound in Terms of $\epsilon_{\min}$	16
2.6.6 $\epsilon_{\min}$ Is Not Always Tight	17
2.7 Direct Versus Indirect Security Valuation	17
2.8 Extended Probing Model	18
2.9 Threshold Implementation	18

3	Algebraic Approach for Random Probing Security	21
3.1	Introduction	22
3.1.1	Open Challenges in the RPM	22
3.1.2	Related Work	23
3.1.3	Our Contributions	23
3.1.4	Our Methodology	24
3.2	EEC for a Linear SC	25
3.2.1	Recovering Parity Relations and Estimating E_{SC}	28
3.2.2	Case Study on Linear Gadgets	32
3.3	Reduction in RPM Framework	35
3.3.1	Reduction with Leak-Free SR Gadget	36
3.3.2	A Describing System of Equations	36
3.3.3	Modeling the Refresh Gadget	38
3.4	A Multiplication Gadget with RPM Security	48
3.4.1	RPM Security of SAND-Rec	50
3.5	RPM Secure Masking of S-box in AES	53
3.5.1	RPM Security of the Proposed SS-box	53
3.6	RPM Secure Masking of AES	55
3.6.1	Estimating the RPM Security	56
3.6.2	Comparison with Previous Results	57
4	Decomposition Approach	59
4.1	Introduction	60
4.1.1	Related Work and Limitations	61
4.1.2	Our Contributions	61
4.2	Security of Mask Encoding	62
4.2.1	Mask Encoding	62
4.2.2	Decomposition into Binary Subfields	66
4.2.3	Tightness of the Decomposition	69
4.2.4	Application of Decomposition to Mask Encodings	71
4.2.5	Experimental Results and Determining Masking Order .	75
4.3	Masking in Odd Prime Fields	77
4.3.1	Adv_X Decays Faster in Prime Fields	77
4.3.2	When Does Adv_X Decrease with Masking?	78
4.3.3	Advantage in Terms of $\langle X, h \rangle$ Projections	80
4.4	General Requirement for Secure Masking	81
4.4.1	Comparison with Known Bounds	84
4.5	Security of Linear Gadgets	84
4.5.1	Problem Statement	85
4.5.2	Decomposition into Binary Systems	86

5	Leakage-Diagrams and Importance Sampling	89
5.1	Introduction	89
5.1.1	Our Contributions	90
5.2	Reduction of Noisy Hamming Weight to Random Probing . . .	91
5.2.1	Noisy Hamming-Weight Leakage	92
5.2.2	Why $\mathbb{F}_2$ -Linear Matters	94
5.3	From Local Security to Global Security	95
5.4	Random Probing Leakage in $\mathbb{F}_q$ -Linear Circuits	97
5.4.1	Monte Carlo Estimation	99
5.4.2	Importance Sampling	100
5.4.3	Leakage-Diagram View	102
5.5	Composition in the Random Probing Model	106
5.5.1	On the Role of Refresh Gadgets	107
5.5.2	Composition Rule in RPM	109
5.5.3	Leakage-Diagram View	110
5.6	LAPSE	112
5.6.1	Domain-Specific Language	112
5.6.2	Internal Representation	113
5.6.3	Security Parameters	114
5.6.4	Estimation Methods	114
5.6.5	Typical Usage	115
6	Deterministic Masking, Part I	117
6.1	Introduction	117
6.1.1	Our Contributions	120
6.2	Deterministic Masking	121
6.2.1	Composition Problem	122
6.3	Simulation-Based Security and Its Limitations	124
6.3.1	Composition Rules	125
6.3.2	Limitations of Simulation-Based Security Proof	126
6.3.3	Probe Elimination	127
6.4	First-Order Deterministic Masking	129
6.4.1	Independence of n -Sharings	130
6.4.2	Lemma 6.2 in Bricklayer Designs	132
6.5	Higher-Order Probing Security in Round-Based Compositions .	133
6.5.1	Properties of the Diffusion Layer	134
6.5.2	Joint Independence Across the Diffusion Layer	138
6.5.3	Our Higher-Order Composition Rule	140
6.6	Case Study on ASCON	145
6.6.1	ASCON's Structure	145

6.6.2	Protected Implementation of the Permutation	147
6.6.3	Application of Our Composition Rules	149
6.6.4	Direct Analysis of the $d = 2$ Case	150
6.6.5	Randomness Savings	152
6.6.6	Implementation Results	152
6.6.7	Power Analysis	154
7	Deterministic Masking, Part II	159
7.1	Introduction	159
7.1.1	Higher-Order Deterministic AND Gadgets	160
7.1.2	Our Contribution	160
7.2	Share Clustering	161
7.2.1	Maximum Number of SMNO Clusters	163
7.2.2	Finite Affine Geometry	167
7.2.3	Non-Completeness of Multi-Shares	169
7.2.4	Linear Independence of Clusters	169
7.3	Higher-Order Deterministic Multiplication	172
7.3.1	The SAND-DN Gadget	174
7.3.2	Properties of the Proposed SAND-DN Gadget	175
7.3.3	A Uniform Deterministic Multiplication Gadget	176
7.3.4	Properties of the Proposed SAND-DU Gadget	179
7.4	Higher-Order Deterministic SToffoli Gadget	181
7.4.1	Properties of the SToffoli Gadget	182
7.5	Case Study on χ_m	183
7.5.1	The S_{χ_m} Gadget	184
7.6	Application to ASCON	186
7.6.1	A Deterministic SASCON	187
7.6.2	Implementation Results	188
8	Conclusions and Outlook	191
	Bibliography	195
	Research Data Management	211
	Summary	213
	Samenvatting	217
	About the Author	221

List of Abbreviations

AES Advanced Encryption Standard.

ARM Arm processor architecture (embedded cores such as Cortex-M0/M4).

BSC Binary Symmetric Channel.

CHES Conference on Cryptographic Hardware and Embedded Systems.

CPA Correlation Power Analysis.

CW308 ChipWhisperer CW308 target board.

DAG Directed Acyclic Graph.

DSL Domain-Specific Language.

EEC Equivalent Erasure Channel.

GF Galois Field (finite field), e.g., $\text{GF}(2)$ or $\text{GF}(2^u)$.

HW Hamming weight.

ISA Instruction Set Architecture.

ISW Ishai–Sahai–Wagner (masking/probing-model framework).

IV Initialization Vector.

LAPSE Tool used in this thesis for describing linear circuits and evaluating/estimating their probing-security parameters.

LLN Law of Large Numbers.

LSB Least Significant Bit.

LUT Look-Up Table.

- MAP** Maximum a Posteriori (decision rule/adversary).
- MC** Monte Carlo (estimation/simulation).
- MI** Mutual Information.
- MNO** Maximally Non-Overlapping.
- NIST** National Institute of Standards and Technology.
- PDT** Probe Distribution Table.
- PINI** Probe-Isolating Non-Interference.
- ReLU** Rectified Linear Unit (activation function).
- RPM** Random Probing Model.
- S-box** Substitution box.
- SCA** Side-Channel Attack(s).
- SD** Statistical distance (total variation distance).
- SHA-3** Secure Hash Algorithm 3.
- SMNO** Simultaneous Maximally Non-Overlapping.
- SNI** Strong Non-Interference.
- SNR** Signal-to-Noise Ratio.
- SR** Success Rate.
- SSA** Static Single Assignment.
- STM32F303** STMicroelectronics STM32F303 microcontroller (target device).
- TCC** Theory of Cryptography Conference.
- TI** Threshold Implementation(s).
- TPM** Threshold Probing Model.
- TV** Total Variation.
- UFO** ChipWhisperer “UFO” target-board designation.
- XOR** Exclusive OR.

Chapter 1

Introduction

Cryptography is typically designed and proven secure in abstract, black-box models where an adversary only learns what a protocol intentionally reveals. Real implementations, however, execute on physical devices whose behavior can unintentionally reveal information through power consumption, electromagnetic emanations, timing behavior, or other micro-architectural effects. Such *implementation leakages* can correlate with intermediate values of a computation and enable attacks that recover long-term secrets even when the underlying cryptographic primitive is mathematically secure.

A *side channel* is any unintended information source that an adversary can observe while a device computes, beyond the algorithm’s specified input/output behavior. In the standard attack setting, the adversary repeatedly runs the target on chosen (or known) inputs, records noisy measurements (e.g., power or EM traces), and then exploits statistical dependencies between these observations and secret-dependent intermediates. Classic examples include timing attacks [Koc96] and differential power analysis [KJJ99]; similar techniques apply across a wide range of platforms and implementation styles [MOP07].

Providing rigorous guarantees in this setting is challenging: leakage is noisy and device-specific, yet countermeasures must remain secure when many components interact and are composed. This thesis adopts a combined *theory-and-practice* perspective. On the theory side, we use leakage models that enable proofs and composition (threshold probing, noisy leakage, and random probing) while paying particular attention to parameter regimes that occur in practice (e.g., low-noise/high-SNR measurements). On the practice side, we connect these models to measurable quantities and evaluate constructions on concrete ciphers, aiming for countermeasures that are both provably secure and implementable at scale.

Masking as a principled countermeasure. Among practical countermeasures, *masking* is particularly attractive because it can offer provable guarantees. A sensitive value X is encoded into n randomized *shares* $\mathbf{X} = (X_1, \dots, X_n)$ such that $X = X_1 + \dots + X_n$ (over a suitable field or additive group), and computations are performed on the shares using small circuits called *gadgets*. The number of shares n acts as a security parameter: increasing n raises the implementation cost and complexity, but is expected to yield stronger protection. Informally, masking aims to ensure that any limited observation of intermediate values is statistically (or nearly) independent of the secrets. Making this intuition precise requires specifying (i) a leakage model and (ii) a security notion that remains meaningful when many gadgets are composed.

Two recurring bottlenecks. Despite a large literature, this thesis highlights two persistent bottlenecks in provable side-channel security:

1. **Realism and scalability of leakage models.** Proof-oriented models such as the *threshold probing model* (TPM) [ISW03] enable clean compositional theorems, but may fail to capture practical leakage phenomena (e.g., *horizontal* and averaging attacks that exploit many correlated observations) [BCPZ16]. More realistic models, such as the *noisy leakage model* [PR13], are closer to measurement reality but are difficult to analyze directly. A major bridge is the reduction of Duc et al. [DDF19] from noisy leakage to the *random probing model* (RPM), where each wire leaks independently with some probability $\epsilon \in [0, 1]$. The parameter ϵ , called the *leakage rate*, quantifies how often a wire leaks; larger ϵ corresponds to more leakage (i.e., lower measurement noise, equivalently higher signal-to-noise ratio (SNR)). This reduction is conceptually clarifying. However, existing bounds in these models often become pessimistic (or even vacuous) in *low-noise/high-SNR* regimes, in large fields, or for large circuits, and they frequently induce undesirable dependencies such as “ ϵ must shrink with the sharing order n ” or “security degrades with circuit size”.
2. **Randomness cost and the desire for deterministic masking.** Standard masking methods consume substantial *online* (fresh) randomness, in particular for nonlinear gadgets and refresh operations needed for secure composition [BBD⁺16, BGR18]. High-quality randomness generation can itself be expensive and difficult to secure under side-channel leakage [CGZ20, CMM⁺24]. This motivates *deterministic mask-*

ing, where the circuit consumes no fresh randomness during execution and relies only on the *offline* randomness already embedded in the initial sharings. While deterministic gadgets exist (e.g., Threshold Implementations [NRR06]), composing them securely at the circuit level is notoriously difficult without refresh gadgets [CPRR14, Rep15].

Scope and structure of this thesis. This thesis addresses these bottlenecks through two complementary research lines:

- **Line I (Chapters 3–5):** evaluate and reason about masking security in the noisy leakage model and in the RPM, with a focus on *constant-noise/low-noise* regimes and on *composition without unrealistic assumptions* (e.g., without assuming leak-free refresh).
- **Line II (Chapters 6–7):** develop theory and constructions for *deterministic masking* that removes online randomness, and show how cryptographic structure (notably diffusion in bricklayer permutations) can act as an *implicit refresh* enabling secure composition across rounds, with applications to ASCON.

The preliminary chapter (Chapter 2) recalls the formal models and metrics used throughout this thesis, including TPM, noisy leakage, RPM, and success-rate/advantage metrics, as well as simulation-based security notions and the standard noisy-to-RPM reduction [DDF19].

1.1 Research Questions

The thesis is organized around the following questions.

Line I: noisy leakage and random probing.

- Q1 Concrete evaluation at constant leakage rates.** How can we evaluate (or tightly bound) the adversary’s success probability in the RPM for *large sharing orders* n , without requiring that the leakage rate ϵ shrinks with n ?
- Q2 Compositional reasoning in RPM without leak-free refresh.** Can one obtain circuit-level guarantees in the RPM from per-gadget analyses without assuming leak-free refreshing and without bounds that deteriorate linearly with circuit size?

- Q3 Tight noise thresholds in the noisy leakage model.** What is the minimal noise level under which masking *provably* improves security, especially in *borderline/low-noise* regimes and beyond binary settings? Can we obtain reductions to RPM that remain meaningful when classical bounds (e.g., scaling with field/ring size) become vacuous?
- Q4 From physical leakage parameters to model parameters.** For practical leakage functions such as noisy Hamming weight, how do physical parameters (e.g., SNR) translate into RPM leakage rates, at least for important subclasses of computations (notably $\mathbb{F}_2$ -linear components)? And when security events are extremely rare (below Monte Carlo resolution), how can we estimate them efficiently?

Line II: deterministic masking.

- Q5 Deterministic composition without refresh gadgets.** When on-line randomness is unavailable, what replaces the standard refresh-based composition methodology? In particular, can the diffusion layers of round-based *bricklayer* permutations act as implicit refresh mechanisms that enable multi-round probing security?
- Q6 Higher-order deterministic nonlinear gadgets.** Beyond first-order deterministic gadgets, do there exist systematic constructions of higher-order deterministic nonlinear gadgets (e.g., multiplication/AND) that preserve probing security, enforce *uniformity* of the output sharing, and remain resilient to glitch-extended probing effects [FGMDP⁺18]?

1.2 Contributions

We now summarize the main contributions of the thesis, grouped by the two lines.

1.2.1 Line I: Noisy Leakage and Random Probing (Chapters 3–5)

Algebraic evaluation and secure masking of AES at constant leakage rates. Chapter 3 develops an algebraic methodology for evaluating RPM security. The core idea is to model the adversary’s post-leakage knowledge via systems of (parity) relations among intermediates, and to characterize when leakage determines the underlying secret. For *linear* masked circuits, this yields an efficient procedure that scales to practically large sharing orders n and supports an *Equivalent Erasure Channel* viewpoint: leakage either fully

determines the secret or is completely uninformative. For *nonlinear* circuits, the chapter introduces linearization-based approximations in bounded leakage regimes.

A central outcome is a composition methodology in RPM *without assuming leak-free refresh*: refresh effects are captured as an effective leakage-rate inflation at gadget boundaries, enabling security claims that do not trivially scale with the total number of gadgets. On the construction side, the chapter refines a conjectured RPM-secure multiplication gadget by reducing its complexity to $\mathcal{O}(n^2)$ and substantiating security in the relevant parameter range. These ingredients lead to a practical $\mathcal{O}(n^2)$ masking scheme for the AES block cipher, whose concrete security evaluation combines analytical reductions with Monte Carlo estimates of constituent-gadget leakage probabilities. The resulting estimates support security at leakage rates around $\epsilon \approx 0.02$ (and slightly higher for the S-box component) over the tested parameter range.

Tight noise thresholds and optimal reductions via decomposition.

Chapter 4 addresses the fundamental question of *when* masking improves security in the noisy-leakage model. Motivated by low-noise/high-SNR practical settings, the chapter develops a decomposition framework that reduces leakage over extension fields to binary projections, enabling tight success-rate bounds in regimes where classical reductions become loose.

As a central theoretical result, the chapter proves a conjecture of Dziembowski et al. [DFS16]: for an additive group $\mathbb{G}$ with largest proper subgroup $\mathbb{H}$, masking strictly improves security whenever the single-variable leakage is δ -noisy with $\delta < 1 - \frac{|\mathbb{H}|}{|\mathbb{G}|}$. This yields a tight characterization of the noise threshold governing when masking helps, across both binary and non-binary settings. For masking over $\mathbb{F}_{2^u}$, the chapter also presents an improved noisy-to-RPM reduction for masked circuits that are $\mathbb{F}_2$ -linear. The resulting bound remains meaningful even in low-noise regimes where generic $2^u\delta$ -type reductions [DDF19] can be vacuous. The chapter further draws practical implications for leakage certification and for choosing sharing orders under realistic noise conditions.

Low-noise tooling: SNR-to- ϵ , composition bounds, and rare-event estimation. Chapter 5 builds on the previous chapters and develops additional tools for compositional RPM security in constant-noise regimes. It instantiates the decomposition approach for noisy Hamming-weight leakage and derives concrete RPM leakage-rate parameters as functions of SNR for $\mathbb{F}_2$ -linear computations, clarifying which SNR regimes correspond to meaning-

ful ϵ values for current RPM analyses and compilers.

The chapter further connects *direct* per-gadget metrics (success probability/advantage) to *global* simulation-based security in RPM, without relying on leak-free refresh, thereby linking proof-oriented notions to concrete evaluation. To handle extremely small failure probabilities, it adapts *importance sampling* to leakage-pattern sampling in RPM. Finally, it develops explicit *leakage-diagram* bounds for refresh gadgets and uses them to derive analytic composition parameters. This analysis isolates *bridges* created by refresh leakage as the key mechanism behind inter-round dependence, and explains why TPM-centric notions such as *Strong Non-Interference* (SNI) [BBD⁺16] do not directly yield RPM composition guarantees. The chapter also introduces LAPSE, a tool for constructing linear-algebraic representations of linear gadgets and computing/estimating their probing-security parameters (exactly when feasible, otherwise via Monte Carlo enhanced with importance sampling).

Limitations of the approach. The numerical security estimates are subject to sampling uncertainty and do not constitute formal proofs; statistical bounds apply at their stated confidence levels, while fitted expressions provide empirical evidence over the tested parameter range. When disclosure probabilities decrease exponentially with the sharing order n , maintaining fixed relative accuracy with ordinary Monte Carlo requires exponentially many trials. The importance-sampling approach developed in Chapter 5 mitigates this difficulty, but the resulting estimates remain subject to statistical uncertainty.

1.2.2 Line II: Deterministic Masking (Chapters 6–7)

Deterministic composition via diffusion in bricklayer permutations. Chapter 6 tackles the randomness bottleneck of masking by developing a deterministic-masking perspective at the circuit level. The key observation is that in round-based *bricklayer* permutations, the linear diffusion layer can act as an *implicit refresh* by mixing the offline randomness embedded in the sharings produced by deterministic gadgets. This yields a route to secure multi-round composition *without* online randomness and without inserting explicit refresh gadgets (which would contradict determinism).

The chapter provides composition theorems and conditions for first-order composability, including a definition of independence for n -shared values that does not rely on assumptions about the underlying secrets. For higher orders, it shows that probing inputs and outputs of a diffusion layer preserves independence up to the diffusion’s *linear branch number* [DR20], giving a provable composition result for two rounds and a heuristic extension to more rounds

based on diffusion properties. It also clarifies why standard simulation-based probe-propagation frameworks depend inherently on fresh randomness, and proposes refinements to enable simulation-flavored reasoning in deterministic contexts. On the practical side, the chapter instantiates the framework using known deterministic first- and second-order masked S-box constructions and provides a software case study for the protected permutation of the NIST-standardized cipher ASCON [TMC⁺24].

Higher-order deterministic nonlinear gadgets via share clustering.

Chapter 7 addresses a longstanding open problem in masking theory: constructing deterministic masked multiplication/AND gadgets with probing security beyond first order, while simultaneously ensuring output uniformity and resilience to glitch-extended probing effects.

The chapter introduces *share clustering*, inspired by finite affine geometry, as a systematic design principle that enforces security and uniformity constraints simultaneously. The resulting deterministic multiplication gadget uses no online randomness, requires only n field multiplications, and achieves $d = \sqrt{n} - 1$ (glitch-extended) probing security for values of n that are squares of prime powers. Building on this, the chapter proposes deterministic masking circuits for the χ_m maps [Dae95] that underpin the nonlinearity of several standardized designs, achieving up to $d = \sqrt{n} + 1$ probing security; in particular, the $n = 4$ instance yields third-order security and applies directly to the χ_5 layer of ASCON as well as to other widely used primitives. The chapter also reports implementations and an application to a third-order deterministically masked ASCON permutation, leveraging the deterministic composition principles of Chapter 6.

1.3 Author Contributions in the Accompanying Papers

Most chapters of this thesis are based on joint publications. Unless stated otherwise, I am the main contributor to the accompanying works: I formulated the research questions, derived the main technical results, carried out the experiments, implemented the supporting software artifacts (including LAPSE and the protected ASCON code), and prepared the initial manuscript drafts. In particular, Bart Mennink and Lejla Batina provided close supervision and detailed feedback that improved the proofs and the expositions. The coauthors further contributed through scientific guidance, discussions, and extensive feedback, improving both the technical development and the presentation.

Specific contribution in Chapter 7. For the paper underlying Chapter 7 [JSB25], Jan Schoone contributed the finite affine-geometry material and participated in refining several lemmas and proofs.

1.4 Thesis Outline

Chapter 2 introduces the leakage models, masking conventions, security metrics, and the reductions between noisy leakage and random probing used throughout the thesis.

Chapters 3–5 form **Part I** and develop theory and tools for evaluating and composing masking in the noisy leakage model and the RPM, with an emphasis on low-noise regimes, tight parameters, and scalable analysis.

Chapters 6–7 form **Part II** and develop deterministic masking theory and constructions, showing how cipher structure (diffusion layers) enables deterministic composition and how share-clustering yields higher-order deterministic nonlinear gadgets with uniformity and glitch resistance, resulting in deterministic masked implementations of ASCON.

Chapter 2

Preliminaries

This chapter recalls standard background on side-channel leakage, masking, and leakage models used throughout the thesis. Cryptographic implementations on physical devices may leak information through power consumption, electromagnetic emanations, timing, etc. *Masking* [CJRR99] mitigates such leakage by randomizing internal computations and representing sensitive intermediates as multiple randomized *shares*.

To reason formally about masking, one needs a *leakage model* describing how an adversary’s observations relate to internal circuit values. The classical *threshold probing model* (TPM) [ISW03] enables clean, compositional proofs but can miss practical phenomena (e.g., horizontal attacks [BCPZ16]). The *noisy leakage model* [PR13] is more faithful to measurement reality but is harder to analyze directly. Duc et al. [DDF19] showed a useful *reduction* from sufficiently noisy leakage to an auxiliary *random probing model* (RPM), which we use repeatedly.

2.1 Notation and Basic Tools

Notation. Lowercase letters denote parameters (e.g., n for the number of shares); uppercase letters denote random variables and secrets (e.g., X). We denote the domain (alphabet) of a random variable by the corresponding black-board letter (e.g., $X \in \mathbb{X}$). We write $\mathbb{E}[\cdot]$ and $\mathbb{V}\text{ar}[\cdot]$ for expectation and variance. Boldface (e.g., $\mathbf{X}$) denotes vectors/tuples; calligraphic letters (e.g., $\mathcal{L}$) denote leakage objects, leakage patterns, or wire families. For a finite set S , $|S|$ denotes its cardinality. Circuits, gadgets, and some functions are written in sans-serif (e.g., C). Masked objects are written with the prefix $\mathbb{S}\cdot$ (e.g., $\mathbb{S}\mathsf{C}$). Unless stated otherwise, variables and operations are over $\mathbb{F}_q$. For a matrix $\mathbf{G}$, $\mathbf{G}(i, j:k)$ denotes the row subvector formed by columns j through k of the

i th row. The notation $R \xleftarrow{\$} \mathbb{F}_q$ means that R is sampled uniformly at random from $\mathbb{F}_q$. We introduce less frequent notation at first use.

Statistical distance (total variation). For distributions P, Q on the same finite set,

$$\text{SD}(P; Q) \triangleq \frac{1}{2} \sum_x |P(x) - Q(x)|.$$

For random variables A, B , $\text{SD}(A; B)$ denotes the distance between their induced distributions. For random variables A, B, C , we define the conditional version

$$\text{SD}(A; (B \mid C)) \triangleq \mathbb{E}_{c \leftarrow C} [\text{SD}(A, (B \mid C=c))].$$

Mutual information. For finite random variables A, B with supports $\mathbb{A}, \mathbb{B}$, the mutual information $\text{MI}(A; B)$ is

$$\text{MI}(A; B) = \sum_{a \in \mathbb{A}} \sum_{b \in \mathbb{B}} \Pr[A = a, B = b] \log_2 \left(\frac{\Pr[A = a, B = b]}{\Pr[A = a] \Pr[B = b]} \right).$$

Throughout this thesis, mutual information is expressed in bits.

2.2 Mathematical Background

In this section, we review the mathematical foundations necessary for verifying probing security. We discuss properties of functions such as bijectivity and explore the linear independence of random variables.

Probability computation from LUT. Security evaluation of protected circuits often requires the computation of (conditional) probabilities. Here, we describe how such probabilities can be precisely obtained using a lookup table (LUT) representation.

Let random variables X and Y be connected through a deterministic function $Y = F(X, R)$, where R is an r -bit random variable, and X is uniformly distributed over $|\mathbb{X}|$ elements. A LUT with $2^r |\mathbb{X}|$ rows represents all possible values taken by X and Y . Additional columns can be appended to this LUT to compute values of a function $G(X, Y)$. Using the LUT, the probability $\Pr(G_1(X, Y) = g_1)$ is computed as:

$$\Pr(G_1(X, Y) = g_1) = \frac{\#\{\text{rows} \mid G_1(X, Y) = g_1\}}{2^r |\mathbb{X}|},$$

and the conditional probability $\Pr(G_1(X, Y) \mid G_2(X, Y))$ is determined by:

$$\begin{aligned} \Pr(G_1(X, Y) = g_1 \mid G_2(X, Y) = g_2) \\ = \frac{\#\{\text{rows} \mid G_1(X, Y) = g_1 \text{ and } G_2(X, Y) = g_2\}}{\#\{\text{rows} \mid G_2(X, Y) = g_2\}}. \end{aligned}$$

Definition 2.1 (Joint Independence). A set of random variables $\{X_1, \dots, X_l\}$ is said to be m -jointly independent ($m \leq l$) if, for any subset of size m , $\{X_{i_1}, \dots, X_{i_m}\}$, the joint probability satisfies:

$$\Pr(X_{i_1}, \dots, X_{i_m}) = \Pr(X_{i_1}) \cdots \Pr(X_{i_m}).$$

For example, given random binary variables R_1 and R_2 , the set $\{R_1, R_2, R_1 \oplus R_2\}$ is 2-jointly independent.

Bijjective mapping and collisions. A function (also referred to as a mapping) $F: \{0, 1\}^m \rightarrow \{0, 1\}^m$ is *bijective* if it establishes a *one-to-one correspondence* between elements of its *domain* and *codomain*. A bijective F is also known as a *permutation* and is invertible. Conversely, if F is not bijective, there exist distinct inputs x_1 and x_2 in its domain such that $F(x_1) = F(x_2)$. In this case, we say that F has a *collision*.

Polynomials and parity relations. Given a set of binary and independent random variables $\{X_1, \dots, X_l\}$, we define binary polynomials as:

$$F_j = \bigoplus_{i=1}^l c_i^j X_i,$$

where the constants c_i^j belong to $\{0, 1\}$. The Hamming weight (HW) of a polynomial F_j is defined as $\text{wt}(F_j) = \sum_i c_i^j \in \mathbb{N}$.

In a set of polynomials $\mathcal{F} = \{F_1, \dots, F_m\}$, we are interested in identifying *parity relations*, and, in particular, the one with the smallest HW. A parity relation is a subset of $\mathcal{F}$ that sums to zero:

$$\bigoplus_{j=1}^m d_j F_j = 0,$$

where its HW is given by $\sum_j d_j$ with d_j belonging to $\{0, 1\}$. If no such parity relation exists, the m polynomials are said to be *linearly independent*. Since all

relations among the polynomials are linear, we conclude that these polynomials (random variables) are m -jointly independent.

For example, let $l = 3$ and $\mathcal{F} = \{X_1 \oplus X_2, X_1 \oplus X_3, X_2 \oplus X_3, X_1 \oplus X_2 \oplus X_3, X_1\}$. In $\mathcal{F}$, there is no parity relation with Hamming weight less than 3; hence, the polynomials are 2-jointly independent.

2.3 Circuit Model and Masking

Circuits, wires, and native (unshared) variables. Let a circuit C take inputs and produce outputs over $\mathbb{F}_q$. We denote by Σ_C the set of all *wires* of C (including input wires and intermediate values), and we refer to their values as *wire values*. A *native* is an unshared sensitive value manipulated by the unmasked circuit (e.g., a key-dependent intermediate). A masked implementation SC_n replaces each native by an n -sharing and has wire set Σ_{SC_n} consisting of shares and additional gadget internals.

Wire-enumeration convention (SSA/DAG). Security measures in the probing-style models depend on how the circuit's *wires* (intermediates) are enumerated. We use a single-static-assignment (SSA) representation: inputs appear first, and each operation produces exactly one new wire whose value is never updated. To model fan-out explicitly, we include a `copy` primitive; we enforce bounded fan-out (e.g., out-degree ≤ 2) by inserting copies, and obtain larger fan-out by chaining them. This yields a directed acyclic graph (DAG), matching the standard representation used for masked circuits and gadgets.

Sharing convention. Masking is parameterized by the number of shares $n \geq 2$. In TPM terminology, this corresponds to (at most) security order $d = n - 1$.

Definition 2.2 (n -sharing). For $X \in \mathbb{F}_q$, an n -sharing of X is a tuple $\mathbf{X} = (X_1, \dots, X_n) \in \mathbb{F}_q^n$ such that: (i) $\sum_{i=1}^n X_i = X$, and (ii) any strict subset of the shares is statistically independent of X . A standard construction samples $X_1, \dots, X_{n-1} \xleftarrow{\$} \mathbb{F}_q$ and sets $X_n = X - \sum_{i=1}^{n-1} X_i$.

Masked circuits, gadgets, and refresh. The masked circuit SC_n is obtained by first converting each native input into an n -sharing and then performing all internal computations on sharings.

Let G range over basic gates (e.g., $+$, $\times$). The circuit SC_n is obtained by replacing each G with a corresponding gadget SG .

Definition 2.3 (Gadget). A gadget $\mathbb{SG}$ for a gate G takes n -shared inputs and outputs n -shared values that encode G applied to the corresponding natives.

Gadgets are built from primitives such as $\{+, \times, \text{copy}, \stackrel{s}{\leftarrow}\}$. For linear/additive gates, gadget design is typically straightforward. For nonlinear operations (e.g., multiplication), gadget construction is more intricate; many proposals exist (e.g., [ISW03, BDF⁺17, CS20, WJZY23, BBP⁺17]). In the RPM, proving such gadgets secure is substantially harder; one can compile gadgets to achieve RPM security using the expansion strategies of [BCP⁺20, BRT21], often at significant complexity cost.

To limit inter-gadget leakage correlations, one often inserts a refresh gadget $\mathbb{SR}$ that re-randomizes an n -sharing of a value X using fresh randomness, without changing the underlying native.

Size. Circuit size is the number of operations in the SSA/DAG representation. Typically, $|\mathbb{SG}| = \mathcal{O}(n)$ for linear operations and $|\mathbb{SG}| = \mathcal{O}(n^2)$ for masked multiplication; refresh gadgets usually fall between $\mathcal{O}(n)$ and $\mathcal{O}(n^2)$. Hence, for a circuit C containing nonlinear gates (e.g., multiplications),

$$|\mathbb{SC}_n| = \mathcal{O}(|C| n^2),$$

while linear-only circuits typically scale as $\mathcal{O}(|C| n)$.

2.4 Leakage Models

Erasure channel. We will use the erasure channel both as a leakage model (RPM) and as an auxiliary channel in reductions from noisy leakage.

Definition 2.4 (Erasure channel). The *erasure channel* with parameter $\epsilon \in [0, 1]$ is the randomized map

$$\phi_\epsilon : \mathbb{F}_q \rightarrow \{\perp\} \cup \mathbb{F}_q, \quad \phi_\epsilon(x) = \begin{cases} x & \text{with probability } \epsilon, \\ \perp & \text{with probability } 1 - \epsilon, \end{cases}$$

where $\perp \notin \mathbb{F}_q$ denotes an erasure.

Threshold probing model (TPM) [ISW03]. In one execution, the adversary adaptively selects up to d wires and learns their exact values (typically $d = n - 1$).

Random probing model (RPM) [BCP⁺20]. Each wire $X \in \Sigma_{\mathbb{S}C_n}$ leaks independently with probability ϵ : equivalently, the adversary observes $\phi_\epsilon(X)$ for every wire with independent channel randomness. We denote the full leakage in one execution by

$$\mathcal{L} = (\phi_\epsilon(X))_{X \in \Sigma_{\mathbb{S}C_n}}.$$

The parameter ϵ is often called the *leakage rate*. This model embodies an *independent leakage assumption* across wires.

Noisy leakage [DDF19]. For each wire $X \in \Sigma_{\mathbb{S}C_n}$, the adversary observes $L(X)$ for some (randomized) leakage function $L : \mathbb{F}_q \rightarrow \mathcal{Y}$, where $\mathcal{Y}$ is a finite alphabet (e.g., after discretization/quantization of a physical measurement). As in the RPM, the leakage channel uses independent randomness across wires. Following [DDF19], L is δ -noisy if, for uniform $X \xleftarrow{\$} \mathbb{F}_q$,

$$\text{SD}(X; X | L(X)) \leq \delta,$$

i.e., the expected total-variation distance between the prior distribution of X and the posterior distribution $X | L(X)$ is at most δ .

2.5 Security Metrics and Security Notions

MAP adversary, success rate, and advantage. Let X be a uniform native (e.g., a key-dependent intermediate) and let $\mathcal{L}$ denote the leakage observed in one execution. A *maximum a posteriori* (MAP) adversary outputs

$$\hat{X} \in \operatorname{argmax}_{\alpha \in \mathbb{F}_q} \Pr[X = \alpha | \mathcal{L}].$$

Define the conditional guessing advantage for a leakage realization ℓ as

$$\text{adv}_X(\ell) \triangleq \max_{\alpha} \Pr[X = \alpha | \mathcal{L} = \ell] - \frac{1}{q}.$$

The (average) success probability is

$$P_c \triangleq \Pr[\hat{X} = X] = \mathbb{E}_{\ell} \left[\max_{\alpha} \Pr[X = \alpha | \mathcal{L} = \ell] \right],$$

and the *advantage* over random guessing is

$$\text{Adv}_X(\mathcal{L}) \triangleq P_c - \frac{1}{q} = \mathbb{E}_{\ell} [\text{adv}_X(\ell)].$$

Security metrics. Several metrics quantify leakage informativeness. The *success rate* (SR) measures $\Pr[\hat{X} = X]$ (equivalently Adv_X). The *statistical distance* (often denoted δ) measures how far the posterior given leakage deviates from the prior. The *mutual information* (MI) measures the expected information about X contained in the leakage. These notions are related (e.g., via Pinsker-type inequalities [FHT03] and bounds connecting SR and SD ; see Lemma 2.6), but translations are often too loose for tight thresholds. In this thesis we frequently focus on SR /advantage because it directly captures attack effort (e.g., trace complexity) and aligns with divide-and-conquer attacks.

Global security: leakage simulation (composition-friendly). For proof-oriented composition results, we use simulation-based security.

Definition 2.5 (ε -close leakage simulation). Let W be a subset of wire indices in $\Sigma_{\mathbb{S}C_n}$, and let $\mathcal{L}_W$ denote the leakage restricted to W . We say $\mathcal{L}_W$ is ε -simulatable if there exists a (randomized) simulator Sim such that, for all secret inputs and for the public inputs P of C ,

$$SD(\mathcal{L}_W, \text{Sim}(P, W)) \leq \varepsilon.$$

The parameter ε depends on the leakage model:

- **TPM.** $\mathbb{S}C_n$ is secure if for all W with $|W| \leq t$, the leakage is simulatable with $\varepsilon = 0$.
- **RPM/Noisy.** One common goal is to simulate a *random* leakage instance with $\varepsilon \leq 2^{-\kappa}$ for a security parameter κ [BCP⁺20], or to obtain $\varepsilon = \varepsilon(n)$ negligible in n [JMB24].

2.6 From Noisy Leakage to Random Probing

This section collects standard facts on single-variable leakage, and the Duc et al. [DDF19] reduction from sufficiently noisy leakage to an erasure leakage (random probing) view.

2.6.1 Statistical Distance and δ -Noisy Leakage

Let X be uniformly distributed over $\mathbb{F}_q$ and let $L(X)$ be the leakage. The statistical distance $SD(X; X \mid L(X))$ measures leakage informativeness. Equivalently, writing $l \leftarrow L(X)$, one may express it via total variation distances of posteriors.

Lemma 2.6. *Let $X \in \mathbb{F}_q$ and let $L(X)$ be δ -noisy. Then:*

$$\frac{\delta}{q-1} \leq \text{Adv}_X \leq \delta.$$

In the binary case ($q = 2$), we have $\text{Adv}_X = \delta$.

2.6.2 Doeblin Coefficient and Leakage Simulation via Erasures

Duc et al. [DDF19] showed that for sufficiently noisy leakage $L(X)$ one can construct a function L' such that the distributions of $L(X)$ and $L'(\phi_\epsilon(X))$ match (for a suitable ϵ). A sufficient condition is $\epsilon \geq \epsilon_{\min}$, where

$$\epsilon_{\min} \triangleq 1 - \sum_{l \in \mathcal{Y}} \min_{\alpha \in \mathbb{F}_q} \Pr(L(X) = l \mid X = \alpha) \leq q\delta. \quad (2.1)$$

The value $\epsilon_{\min}$ is the *Doeblin coefficient* [BCGR24]. It lies in $[0, 1]$: $\epsilon_{\min} = 0$ means $L(X)$ is independent of X , while $\epsilon_{\min} = 1$ indicates that the reduction becomes vacuous.

2.6.3 Binary Tightness

Lemma 2.7. *Let $(X, L(X))$ be a joint distribution with $\text{SD}(X; X \mid L(X)) = \delta$ and X uniform over $\mathbb{F}_2$. Then:*

$$\epsilon_{\min} = 2\delta.$$

2.6.4 Example: Leakage of One Bit

Example 2.1. Let $X \in \mathbb{F}_{2^u}$ and consider $L(X) = X_1 \oplus E$, where X_1 is the least significant bit of X and E is a binary error random variable, independent of X , with $\Pr(E = 1) = \mathbf{e} \leq \frac{1}{2}$. Then $L(X)$ leaks noisy information about the LSB only, and one obtains

$$\text{Adv}_X = \frac{1 - 2\mathbf{e}}{2^u}, \quad \text{SD}(X; X \mid L(X)) = \frac{1}{2} - \mathbf{e}, \quad \epsilon_{\min} = 1 - 2\mathbf{e}.$$

2.6.5 A Useful Bound in Terms of $\epsilon_{\min}$

Since $(X, L(X))$ and $(X, L'(\phi_{\epsilon_{\min}}(X)))$ are identically distributed, the adversary's advantage remains unchanged:

$$\text{Adv}_X[L(X)] = \text{Adv}_X[L'(\phi_{\epsilon_{\min}}(X))].$$

Moreover, for any $\epsilon \geq \epsilon_{\min}$, the erasure view $\phi_\epsilon(X)$ is at least as informative as $\phi_{\epsilon_{\min}}(X)$ (one can obtain $\phi_{\epsilon_{\min}}(X)$ from $\phi_\epsilon(X)$ by additional random erasures). Since L' is a (randomized) post-processing of $\phi_{\epsilon_{\min}}(X)$, any reasonable leakage metric (e.g., success probability/advantage or statistical distance) can only decrease under the mapping $\phi_\epsilon(X) \mapsto \phi_{\epsilon_{\min}}(X) \mapsto L'(\phi_{\epsilon_{\min}}(X))$. Consequently,

$$\text{Adv}_X[L(X)] \leq \text{Adv}_X[\phi_\epsilon(X)], \quad \text{SD}(X; X \mid L(X)) \leq \text{SD}(X; X \mid \phi_\epsilon(X)). \quad (2.2)$$

Thus, to prove security under leakage $L(X)$, it suffices to prove it under $\phi_{\epsilon_{\min}}(X)$. This reduction from δ -noisy to ϵ -random probing leakage was introduced in [DDF19]. The following lemma is a concrete application.

Lemma 2.8 ([BCGR24], Proposition 1). *Let $X \in \mathbb{F}_q$, and let $L(X)$ be a leakage with Doeblin coefficient $\epsilon_{\min}$. Then:*

$$\text{Adv}_X \leq \frac{q-1}{q} \cdot \epsilon_{\min}.$$

2.6.6 $\epsilon_{\min}$ Is Not Always Tight

While the reduction from δ -noisy leakage to ϵ -random probing is powerful, it is not always tight. For instance, for some leakage functions (e.g., Hamming-weight leakage), Eq. (2.1) can yield $\epsilon_{\min} = 1$ even when $L(X)$ does not fully determine X . A similar issue can occur for “zero-value” style leakages. Later chapters develop refined approaches to obtain tighter bounds in such cases.

2.7 Direct Versus Indirect Security Valuation

Theoretical soundness of masking schemes can be supported by either *indirect* (simulation-based) or *direct* (metric-based) arguments (e.g., [PGMP19]).

Indirect (simulation-based) proofs. A simulator reproduces leakage from scratch without accessing secrets, implying the leakage carries no useful information about the secrets. This is the approach in [ISW03, DDF19, BCP⁺20]. Such proofs enable strong composition theorems, but may overestimate the adversary and can yield loose bounds. They also often require re-masking steps that enlarge the circuit.

Direct evaluations. One demonstrates that a concrete success metric—e.g., key rank in a key guessing attack, success rate/advantage, or mutual information—decreases with increasing number of shares n [CJRR99, PR13, MS23]. This aligns closely with practical side-channel methodology and can highlight which components dominate the leakage.

Limitation of single-target definitions. Some definitions (e.g., an advantage for a single native X) implicitly assume a single target secret. In full ciphers (e.g., AES) an adversary may attempt to recover functions of multiple secrets (e.g., several round keys). Simulation-based definitions avoid fixing the number of targets but may be less tight in practice.

2.8 Extended Probing Model

In the *standard* probing model, an adversary learns stabilized wire values. In the first three technical chapters (Chapters 3–5), we work in this standard model. In Chapters 6 and 7, we additionally consider *extended* probing models that capture transient hardware effects such as glitches.

Glitch-extended probing. In hardware implementations¹, transient effects known as *glitches* can leak additional information [MOP07]. To capture this phenomenon, Faust et al. [FGMDP⁺18] proposed *glitch-extended probing*, where a single probe of the form $P = F(X_i, Y_j, \dots)$ does not only reveal the probed signal, but effectively reveals *all* values feeding the combinational computation F within the same clock cycle. Here, $X_i, Y_j, \dots$ denote (masked or unmasked) *intermediate variables* carried by distinct wires/register outputs (indices i, j refer to different signals) that lie in the fan-in of F during that cycle. A standard mitigation is to insert *synchronization layers* that restrict how many inputs can interact simultaneously [FGMDP⁺18, CGLS21], thereby limiting glitch-induced leakage.

2.9 Threshold Implementation

Our second line of research explores the extent to which the use of online (fresh) randomness can be minimized. Threshold Implementations (TI) serve as a starting point for this direction. This approach for masking nonlinear functions relies on three core properties: *correctness*, *non-completeness*, and

¹Also shown to occur in software, see [GMPO20].

uniformity. TI is well-regarded for its effectiveness in mitigating information leakage due to hardware glitches; however, it typically requires more shares than other masking methods in the probing model.

Properties of TI [NRR06, BGN⁺14]. Let $F: \mathbb{F}_{2^u}^{m_{\text{in}}} \rightarrow \mathbb{F}_{2^u}^{m_{\text{out}}}$ be a function defined by

$$F(X^1, \dots, X^{m_{\text{in}}}) = (Y^1, \dots, Y^{m_{\text{out}}}),$$

and let $\mathbb{S}F: \mathbb{F}_{2^u}^{m_{\text{in}}n} \rightarrow \mathbb{F}_{2^u}^{m_{\text{out}}n}$ be an n -share masked implementation of F . Let the input sharings be denoted as $\mathbf{X}^i = \{X_1^i, \dots, X_n^i\}$ for $1 \leq i \leq m_{\text{in}}$, and the output sharings as $\mathbf{Y}^i = \{Y_1^i, \dots, Y_n^i\}$ for $1 \leq i \leq m_{\text{out}}$. The implementation $\mathbb{S}F$ is a Threshold Implementation (TI) of F at order d if it satisfies the following properties:

- *Correctness.* For all inputs, the recombination of the output shares matches the function output on the recombined inputs:

$$F(\oplus_{j=1}^n X_j^1, \dots, \oplus_{j=1}^n X_j^{m_{\text{in}}}) = (\oplus_{j=1}^n Y_j^1, \dots, \oplus_{j=1}^n Y_j^{m_{\text{out}}}).$$

- *Non-completeness.* Any set of up to d output shares Y_j^i depends on at most $n - 1$ shares from each input sharing $\mathbf{X}^i$.
- *Uniformity.* For any input $(X^1, \dots, X^{m_{\text{in}}})$, the output $\mathbb{S}F$ must produce a uniform n -sharing of $F(X^1, \dots, X^{m_{\text{in}}})$. That is, each output $\mathbf{Y}^i$ should form an n -sharing where any subset of $n - 1$ shares Y_j^i ($1 \leq j \leq n$) is uniformly distributed over $\mathbb{F}_{2^u}^{n-1}$, regardless of the distribution of the underlying secrets.

As a concrete example, Nikova et al. [NRR06] proposed a first-order (i.e., $d = 1$) TI gadget for the binary (i.e., $u = 1$) AND function. Given two secret variables A and B with 4-share encodings $\mathbf{A} = \{A_1, \dots, A_4\}$ and $\mathbf{B} = \{B_1, \dots, B_4\}$, the gadget computes $\mathbf{C} = \{C_1, \dots, C_4\}$ such that $C = A \cdot B$, with:

$$\begin{cases} C_1 = (1 \oplus A_3 \oplus A_4)(1 \oplus B_2 \oplus B_3) \oplus B_4 \oplus A_2, \\ C_2 = (1 \oplus A_1 \oplus A_3)(1 \oplus B_1 \oplus B_4) \oplus B_3 \oplus A_4, \\ C_3 = (A_2 \oplus A_4)(B_1 \oplus B_4) \oplus B_2 \oplus A_2, \\ C_4 = (A_1 \oplus A_2)(B_2 \oplus B_3) \oplus B_1 \oplus A_1. \end{cases}$$

This construction satisfies the TI conditions:

- **Correctness:** $\oplus_{i=1}^4 C_i = (\oplus_{i=1}^4 A_i) \cdot (\oplus_{i=1}^4 B_i)$ for all $2^4 \times 2^4$ input combinations.
- **Non-completeness:** Each output share C_i depends on at most $n - 1 = 3$ shares from both $\mathbf{A}$ and $\mathbf{B}$.
- **Uniformity:** For any secret values $A = a$, $B = b$, the output $\mathbf{C}$ is a uniform 4-sharing of ab , even though ab may be biased. That is, all 4-tuples (C_1, C_2, C_3, C_4) summing to ab are equally likely.

Limitations of circuit-level TI. Extending Threshold Implementations to the circuit level at higher orders ($d > 1$) remains a significant challenge. Existing approaches, such as “Consolidating Masking Schemes” [RBN⁺15], apply TI gadgets locally and rely on additional refresh gadgets to enable circuit-level composition. However, Moos et al. [MMSS19] demonstrated through several concrete attacks that ad hoc refresh strategies do not reliably ensure higher-order security. They advocate instead for the proper use of standard composition techniques and probe-propagation-based frameworks.

Unfortunately, both refreshing mechanisms and standard probe-propagation techniques inherently require online randomness. As a result, these methods are fundamentally incompatible with deterministic masking, where no fresh randomness is available during execution.

Glitch resistance via non-completeness. Recall that a glitch-extended probe reveals *all* intermediate values involved in its computation. Therefore, if the collected intermediates—across all probes—form a *non-complete* subset of the shares for each secret, then glitches do not compromise the underlying secrets.

We say that a deterministic gadget is *d-order glitch-resistant* if any set of d glitch-extended probes does not collectively involve all n shares of any secret. This ensures that no secret can be fully reconstructed from such probes.

Example 2.2. Let $\mathbf{X} = (X_1, X_2, X_3)$ and $\mathbf{Y} = (Y_1, Y_2, Y_3)$ be two independent 3-sharings. Define

$$P_0 = (X_1 \oplus X_2)(Y_1 \oplus Y_2) \quad \text{and} \quad P_1 = (X_2 \oplus X_3)(Y_2 \oplus Y_3).$$

Although the set $\{P_0, P_1\}$ is independent of the secrets—thus achieving probing security order 2—computing both probes requires access to all shares of X and Y . As a result, the glitch-resistance order is strictly less than 2. On the other hand, each individual probe (P_0 or P_1) depends on only two shares per secret (i.e., at most $n - 1$), so the gadget has glitch-resistance order 1. $\square$

In general, the glitch resistance order is upper bounded by the probing security order.

Chapter 3

An Algebraic Approach for Evaluating Random Probing Security with Application to AES

This chapter presents our results on random probing security, as published at CHES 2024 [JMB24]. The core technical development follows the paper, while the introductory material, notation, and exposition up to Section 3.2 have been adapted to better match the style and conventions used throughout this thesis.

We consider a side-channel adversary in the Random Probing Model (RPM), where each intermediate variable of a masked implementation leaks independently with probability ϵ . Our approach is algebraic: we model the adversary’s post-leakage knowledge through systems of parity relations and estimate the adversary’s success probability by quantifying when these relations determine the underlying secret. For masked *linear* circuits, this yields an efficient procedure that scales to practically large sharing orders n . For *nonlinear* circuits, we use a linearization technique to obtain tight approximations in bounded leakage rates.

To reason about complex structures such as S-boxes, we introduce a composition theorem that reduces the RPM security of a circuit to that of its constituent gadgets (without assuming leak-free refreshing). We additionally refine the multiplication gadget of Battistello et al. [BCPZ16]: we reduce its complexity from $\mathcal{O}(n^2 \log n)$ to $\mathcal{O}(n^2)$ and substantiate its conjectured RPM security in the relevant parameter range. These ingredients enable a practical $\mathcal{O}(n^2)$ masking scheme for AES that remains secure for leakage rates up to about $\epsilon \leq 0.02 \approx 2^{-5.6}$.

3.1 Introduction

The noisy leakage model—and, equivalently, the RPM abstraction—provides a realistic setting for analyzing side-channel leakage from masked implementations. A substantial body of work studies masking in these models [PR13, DDF19, DFŽ19, BCP⁺20, CFOS21, MS23, BFO23, BCGR24]. Nevertheless, two recurring obstacles remain when one aims for both *meaningful leakage rates* and *scalable guarantees*. This chapter addresses these obstacles via a direct (non-simulation-based) evaluation methodology.

3.1.1 Open Challenges in the RPM

Leakage rate versus masking order. A primary challenge is that, for many common masking schemes, existing security claims in RPM require the leakage rate ϵ to decrease as the sharing order n increases. Concretely, conditions of the form $\epsilon < \frac{1}{n}$ appear in several analyses [PR13, DDF19]. Intuitively, this is problematic: if the leakage rate is fixed by the device and measurement setup, increasing n should not *force* us to assume a smaller ϵ .

Battistello et al. [BCPZ16] showed that this dependence is not merely an artifact of proof techniques: for widely used gadgets such as the ISW multiplication [ISW03], the issue is structural. Multiplying two shared secrets X and Y at order n computes n^2 intermediates, and each input share participates in $\Theta(n)$ of them. Thus, the expected number of times a given share is revealed through leakage is about ϵn , which naturally pushes analyses toward regimes with $\epsilon < \frac{1}{n}$.

Circuit-size dependence and lack of compositionality. In the threshold probing model (TPM), composability is well understood: one can often preserve a security order independent of the number of composed gadgets [BBD⁺16]. In contrast, in the RPM, security claims frequently degrade with circuit size, which does not fully align with empirical observations. For example, the success rate of state-of-the-art side-channel attacks does not necessarily grow markedly with the number of cipher rounds. A key reason is the lack of native composition theorems in RPM.

In simulation-based approaches, RPM is typically reduced to TPM by limiting the *expected* number of leaked variables $\epsilon|C|$ (where $|C|$ is the circuit size) below a threshold t , making the dependence on $|C|$ explicit. In direct approaches, one often assumes *leak-free* refreshing to recover conditional independence properties, and then applies union bounds (e.g., over mutual information) across components—again producing bounds that scale with $|C|$.

3.1.2 Related Work

Ajtai [Ajt11] explored theoretical compilers based on expander graphs to amplify resistance to leakage in the RPM. Andrychowicz et al. [ADF16] developed an expander-based compiler for arbitrary circuits that tolerates a sufficiently small constant leakage rate in the RPM.

In a more practical direction, Ananth et al. [AIS18] introduced *expansion*, which uses iterative masking to target constant leakage. While initial parameters tolerated leakage rates around 2^{-25} , refinements by Belaïd et al. [BCP⁺20, BRT21, BRTV21] improved efficiency but still left significant overhead, and the practical margins remain limited.

Dziembowski et al. [DFŻ19], using the *leakage diagram* viewpoint [BFO23], demonstrated that certain existing gadgets can be secure at constant leakage rates without additional compilation. Their analysis—focusing primarily on linear refreshing—served as a key inspiration for our linear-algebraic treatment.

Cassiers et al. [CFOS21] introduced the Probe Distribution Table (PDT) technique to evaluate $\mathcal{O}(n^2)$ masking based on ISW [RP10], including for protecting the AES S-box. However, the ISW multiplication used there is not secure at constant leakage rates.

Finally, Prouff and Rivain [PR13] proposed mutual-information-based direct evaluations, further developed by Masure and Standaert [MS23] in the noisy leakage model. A key limitation is the typical reliance on leak-free refreshing assumptions to preserve the conditional-independence structure needed for the proofs [BCGR24].

3.1.3 Our Contributions

We address both RPM challenges highlighted in Section 3.1.1 via a direct evaluation framework that is effective for large n , and by instantiating a practical protection for AES. Concretely:

- For linear masked circuits, we show that leakage either fully determines the native secret V or remains completely uninformative about it, and that the probability of disclosure depends only on (n, ϵ) (not on the concrete leaked values). We provide tools to efficiently estimate this disclosure probability for practically large n ; see Section 3.2.
- To handle larger constructions, we analyze refreshing gadgets and model their impact as an increase of the effective leakage rate at gadget inter-

faces—*without* assuming leak-free refresh. This yields an RPM-native composition theorem; see Section 3.3.

- We modify the Battistello et al. [BCPZ16] multiplication gadget (conjectured RPM-secure) to reduce complexity from $\mathcal{O}(n^2 \log n)$ to $\mathcal{O}(n^2)$, and we substantiate its RPM security over the parameter range of interest using linearization and the linear-circuit machinery; see Section 3.4.
- Using the composition theorem and secure multiplication, we design an $\mathcal{O}(n^2)$ masking scheme for the AES S-box tolerating leakage rates up to about $\epsilon \leq 0.03$; see Section 3.5.
- We extend the evaluation to a full $\mathcal{O}(n^2)$ masked AES and obtain empirical support for security at leakage rates up to $\epsilon \leq 0.02$; see Section 3.6. Notably, the resulting bound does not directly scale with the total number of gadgets.

3.1.4 Our Methodology

Our evaluation follows a *direct* approach: we bound the adversary’s success probability for recovering the native secret V from the leaked variables in the RPM. We combine (i) analytic arguments to derive structural lemmas and reductions, with (ii) Monte Carlo estimation to obtain numerical bounds for specific gadgets when a closed form is unavailable.

Suppose an event **bad** (e.g., “the leakage determines V ”) occurs with probability $\Pr(\mathbf{bad}) = f(n, \epsilon)$ for some function f . If f is unknown or impractical to derive analytically, we estimate it empirically by running N trials at a fixed (n, ϵ) and recording the number of occurrences $N_{\mathbf{bad}}$. By the Law of Large Numbers (LLN) [PP02],

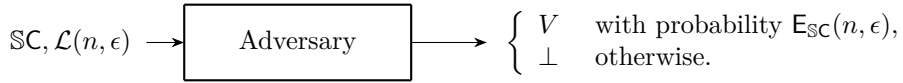
$$\Pr(\mathbf{bad}) = \lim_{N \rightarrow \infty} \frac{N_{\mathbf{bad}}}{N}. \quad (3.1)$$

Limitation and reliability of Monte Carlo. If $\Pr(\mathbf{bad}) < \frac{1}{N}$, then N trials cannot be expected to observe **bad**, and an observed estimate $P_{\mathbf{bad}} = \frac{N_{\mathbf{bad}}}{N} = 0$ only implies $\Pr(\mathbf{bad}) < \frac{1}{N}$. Moreover, $P_{\mathbf{bad}}$ is itself a random variable with mean $\Pr(\mathbf{bad})$ and variance σ_N^2 with $\lim_{N \rightarrow \infty} \sigma_N = 0$. By Chebyshev’s inequality [PP02], for any $k > 0$,

$$\Pr(|\Pr(\mathbf{bad}) - P_{\mathbf{bad}}| \geq k\sigma_N) \leq \frac{1}{k^2}. \quad (3.2)$$

Interpolating empirical bounds. Given estimates $f(n, \epsilon)$ on a grid of (n, ϵ) , we may fit an explicit upper bound $g(n, \epsilon)$ so that $\Pr(\text{bad}) \leq g(n, \epsilon)$ holds over the tested region. This makes it possible to propagate bounds through compositions of gadgets. When using such fitted bounds, one should keep in mind that validity outside the tested (n, ϵ) region is not guaranteed.

Equivalent erasure channel. For a linear masked circuit SC processing a single native secret V , we will show that the RPM leakage $\mathcal{L}(n, \epsilon)$ either determines V or provides no information about it. This can be summarized by an Equivalent Erasure Channel (EEC) with parameter $E_{\text{SC}}(n, \epsilon)$:



If the adversary does not learn V , it can still guess it. Writing $q = |\mathbb{F}_{2^u}|$, the advantage for recovering V becomes:

$$\text{Adv}_V(n, \epsilon) = E_{\text{SC}}(n, \epsilon) + \frac{1}{q}(1 - E_{\text{SC}}(n, \epsilon)) - \frac{1}{q} = \frac{q-1}{q} E_{\text{SC}}(n, \epsilon). \quad (3.3)$$

For nonlinear masked circuits, we will also obtain EEC-style bounds via linearization; these are approximations that hold in a bounded leakage regime.

3.2 EEC for a Linear SC

Let us consider a masked circuit SC processing a native random variable V . In each execution of SC, the leakage reveals to the adversary a random leakage set $\mathcal{L} \subseteq \Sigma_{\text{SC}}$. The native variable V is related to the variables in Σ_{SC} , and the adversary can exploit these dependencies—referred to as *parity relations*—to gain information about V .

Definition 3.1 (Linear SC). We call a masked circuit SC *linear* if all its native variables and intermediate variables take values in the same field $\mathbb{F}_{2^u}$, and all parity relations among $\{V\} \cup \Sigma_{\text{SC}}$ are $\mathbb{F}_{2^u}$ -linear.

For instance, the SR-Simple refresh gadget [RP10] in Algorithm 1 processes a single native variable and is linear.

Algorithm 1 SR-Simple

Input: $V^1 = (V_1^1, \dots, V_n^1)$
Output: $V^2 = (V_1^2, \dots, V_n^2)$
1: $R_{\text{sum}} \leftarrow 0$
2: **for** $i = 1$ **to** $n - 1$ **do**
3: $R \xleftarrow{\$} \mathbb{F}_{2^u}$
4: $V_i^2 \leftarrow V_i^1 \oplus R$
5: $R_{\text{sum}} \leftarrow R_{\text{sum}} \oplus R$
6: $V_n^2 \leftarrow V_n^1 \oplus R_{\text{sum}}$
7: **return** V^2

In this section, our primary objective is to identify linear dependencies within $\{V\} \cup \Sigma_{\text{SC}}$ and to evaluate the extent to which they empower the adversary.

Extracting parity relations. For a linear SC, the existence of $\mathbb{F}_{2^u}$ -linear relations over $[V, \Sigma_{\text{SC}}]$ implies that there exists a parity matrix as $\mathbf{P}_n \in \mathbb{F}_{2^u}^{p(n) \times (m(n)+1)}$ such that, in each execution, the realized values $[v, \sigma_{\text{SC}}]$ satisfy

$$\mathbf{P}_n \cdot [V, \Sigma_{\text{SC}}]^\top = \mathbf{0}_{[p(n) \times 1]}, \quad (3.4)$$

where $p(n)$ is the number of relations, $m(n)$ is size of Σ_{SC} at order n , and $^\top$ denotes transpose. The rows of $\mathbf{P}_n$ are *linearly independent*, and any other parity relation over $[V, \Sigma_{\text{SC}}]$ can be expressed as an $\mathbb{F}_{2^u}$ -linear combination of these rows.

Taking leakage into account. Upon observing a leakage instance L , the adversary learns the realized values of some variables in Σ_{SC} . After substituting these leaked values, the system in (3.4) becomes

$$\mathbf{P}_n^r \cdot [V, \Sigma_{\text{SC}}]^\top = \mathbf{b}_{[p(n) \times 1]}. \quad (3.5)$$

Here, $\mathbf{b}$ is a known vector (of size $p(n) \times 1$), and $\mathbf{P}_n^r$ is obtained from $\mathbf{P}_n$ by replacing every column corresponding to a leaked variable with the all-zero column. By characterizing the solution set of (3.5), the adversary obtains a posterior distribution on V .

Remark 3.2. Over a finite field, a linear system has either no solution, a unique solution, or finitely many solutions.

For (3.5), the realized values of $[v, \sigma_{\text{SC}}]$ always form a solution; hence the system has either a unique solution or finitely many solutions. Let $\mathcal{S}$ denote

the set of all solutions. Each $S_i \in \mathcal{S}$ is a vector of length $m(n) + 1$, whose first entry $S_i[1]$ corresponds to v ; moreover, S_i is consistent with the leakage on all leaked variables. The induced posterior distribution of $\tilde{V}$ is therefore

$$\Pr(\tilde{V} = \alpha \mid \mathcal{L} = L) = \frac{|\{S_i \in \mathcal{S} : S_i[1] = \alpha\}|}{|\mathcal{S}|}. \quad (3.6)$$

The adversary then outputs *Maximum A Posteriori* (MAP) estimate as $\operatorname{argmax}_{\alpha \in \mathbb{F}_{2^u}} \Pr(\tilde{V} = \alpha \mid \mathcal{L} = L)$. To gain further insight into the structure of $\Pr(\tilde{V} \mid \mathcal{L} = L)$, we compute the *row-echelon form* [LDP20] of $\mathbf{P}_n^r$.

Let $\mathbf{G}$ denote the row-echelon form of $\mathbf{P}_n^r$. Then $\mathbf{G}$ defines an equivalent system: for some constant vector $\mathbf{c}$ (computed from $\mathbf{b}$), we have

$$\mathbf{G} \cdot [V, \Sigma_{\text{SC}}]^\top = \mathbf{c}. \quad (3.7)$$

By construction, the variable V is a *pivot* variable and thus appears as the leading variable in exactly one relation (namely, the first row of $\mathbf{G}$). If this relation contains only V , then V is uniquely determined; otherwise, $\Pr(\tilde{V} \mid \mathcal{L} = L)$ is uniform.

Lemma 3.3. *Consider an $\mathbb{F}_{2^u}$ -linear circuit. If the first row of $\mathbf{G}$ contains no variable other than V , then the adversary can uniquely determine V . Otherwise, the presence of any additional (necessarily free) variable in the first row renders V perfectly hidden (i.e., $\Pr(\tilde{V} \mid \mathcal{L} = L)$ is uniform).*

Proof. Fix a leakage instance L , and consider the (equivalent) non-homogeneous system $\mathbf{G} \cdot [V, \Sigma_{\text{SC}}]^\top = \mathbf{c}$. Let $[v, \sigma_{\text{SC}}]_p$ be a particular solution (in realized values). Then the full set of solutions can be written as

$$[v, \sigma_{\text{SC}}] = [v, \sigma_{\text{SC}}]_p \oplus [v, \sigma_{\text{SC}}]_f, \quad (3.8)$$

where $[v, \sigma_{\text{SC}}]_f$ ranges over all solutions of the associated homogeneous system $\mathbf{G} \cdot [V, \Sigma_{\text{SC}}]^\top = \mathbf{0}$. The number of such tuples is determined by the number of free variables of $\mathbf{G}$: each assignment to the free variables yields a distinct $[v, \sigma_{\text{SC}}]_f$. Recall that variables revealed by the leakage are eliminated by zeroing their columns, and thus they are not treated as free variables.

Since V is not leaked and there exists at least one relation involving V , the elimination process makes V a pivot variable. Assume $\mathbf{G}$ has ℓ free variables, denoted $\{Z_1, \dots, Z_\ell\}$. Then v can be expressed as

$$v = v_p \oplus Z_1 a_1 \oplus \dots \oplus Z_\ell a_\ell, \quad (3.9)$$

where v_p is the first component of $[v, \sigma_{\text{SC}}]_p$, and a_i is the contribution to v from the homogeneous solution obtained by setting all free variables to 0 except $Z_i = 1$.

The free variables $Z_1, \dots, Z_\ell$ range uniformly over $\mathbb{F}_{2^u}$. Therefore, if at least one coefficient a_i is non-zero, then as $(Z_1, \dots, Z_\ell)$ varies, the value of v ranges uniformly over $\mathbb{F}_{2^u}$, independently of v_p ; hence $\Pr(\tilde{V} \mid \mathcal{L} = L)$ is uniform.

Conversely, v is uniquely determined if and only if all $a_i = 0$, i.e., $v = v_p$ for every homogeneous solution. In row-echelon form, these coefficients correspond to the non-leading entries of the first row: for $2 \leq i \leq m(n) + 1$, the coefficient of the variable in column i equals $-\mathbf{G}(1, i)$. Thus, V is uniquely determined exactly when $\mathbf{G}(1, 2 : m(n) + 1) = \mathbf{0}$. Any non-zero entry in $\mathbf{G}(1, 2 : m(n) + 1)$ together with its associated free variable masks the particular solution v_p . $\square$

Lemma 3.3 characterizes the adversary's post-leakage information about V via an EEC, whose parameter we define as

$$\mathbf{E}_{\text{SC}}(n, \epsilon) = \Pr_{L \leftarrow \mathcal{L}(n, \epsilon)} [\mathbf{G}(1, 2 : m(n) + 1) = \mathbf{0}_{[1 \times m(n)]}], \quad (3.10)$$

where each leakage instance L yields a derived matrix $\mathbf{G}$ depending only on which members of Σ_{SC} are revealed. Importantly, $\mathbf{G}$ (and hence $\mathbf{E}_{\text{SC}}$) is independent of the actual leakage values. This enables efficient estimation of $\mathbf{E}_{\text{SC}}$: for a fixed pair (n, ϵ) and a sufficient number of trials, we sample leakage instances, compute the corresponding $\mathbf{G}$, and count the frequency of the event $\mathbf{G}(1, 2 : m(n) + 1) = \mathbf{0}$. We provide the algorithms and implementation details in the following section.

3.2.1 Recovering Parity Relations and Estimating $\mathbf{E}_{\text{SC}}$

This section presents algorithms for extracting a parity matrix from a given $\mathbb{F}_{2^u}$ -linear circuit, computing the row-echelon form of this parity matrix for an instance of leakage, and approximating the corresponding $\mathbf{E}$ parameter.

Packing $[V, \Sigma_{\text{SC}}]$ realizations in $\mathbf{M}_n$. For a circuit family SC processing a native random variable V at order n , we construct a matrix $\mathbf{M}_n$ with $m(n) + 1$ columns and R rows. We populate $\mathbf{M}_n$ by executing SC a total of R times, each time with an independently sampled realization $v \xleftarrow{\$} \mathbb{F}_{2^u}$ and fresh randomness. In the i -th execution, the realized values assigned to V and to all variables in Σ_{SC} form the i -th row of $\mathbf{M}_n$, as shown in Algorithm 2. Recall

that Σ_{SC} denotes the list of random variables of SC, and that $m(n)$ is the length of this list.

Algorithm 2 Create-M

Input SC, n
Output M_n

- 1: **for** $i = 1$ **to** R **do**
- 2: $v \xleftarrow{\$} \mathbb{F}_{2^u}$
- 3: $M_n(i, 1) \leftarrow v$
- 4: Execute SC $\triangleright$ with fresh shares for the native inputs
- 5: $M_n(i, 2 : m(n) + 1) \leftarrow \sigma_{\text{SC}}$
- 6: **return** M_n

Example 3.1. Consider SR-Simple from Algorithm 1 at $n = 3$. One possible instantiation of the variable list is

$$\Sigma_{\text{SR-Simple}} = \{V_1^1, V_2^1, V_3^1, R_3, R_a, V_1^1 \oplus R_a, R_3 \oplus R_a, R_b, V_2^1 \oplus R_b, R_3 \oplus R_b, V_3^1 \oplus R_3\}.$$

Here, R_a and R_b denote the random variables corresponding to the randomness sampled in two different iterations of the for-loop, and R_3 denotes the running-sum random variable as it is updated during the algorithm. The exact composition of $\Sigma_{\text{SR-Simple}}$ may vary with implementation details; for instance, if the outputs are copied to additional wires, then $\{V_1^2, V_2^2, V_3^2\}$ may also appear in $\Sigma_{\text{SR-Simple}}$. Recall that elements of Σ_{SC} leak independently; hence, if a variable appears multiple times in the list, its realized value has a higher probability of being observed. $\square$

The resulting matrix M_n consists of $R \times (m(n) + 1)$ realizations in $\mathbb{F}_{2^u}$. In the following, we discuss how to use standard Gaussian elimination to extract a set of parity relations that describe the $\mathbb{F}_{2^u}$ -linear dependencies among the random variables in $[V, \Sigma_{\text{SC}}]$.¹

Extracting P_n from M_n . By computing the row-echelon form of M_n using Algorithm 3 (i.e., $D_n \leftarrow \text{Gaussian-Elim}(M_n)$), we can identify linear dependencies between the columns of M_n . Since M_n contains numerical realizations, the recovered dependencies may be incomplete or even incorrect; that is, we may miss some parity relations or include spurious ones. However, the

¹Alternatively, one may derive a set of parity equations directly over $[V, \Sigma_{\text{SC}}]$ from the defining relations of SC. While this may yield a different representation of the parity matrix, the resulting value of E_{SC} is independent of the chosen representation.

accuracy and completeness of the recovered relations improve as the number of rows R increases.

While the number of parity relations, denoted $p(n)$, is not known a priori, we have $p(n) = m(n) + 1 - \text{rank}(\mathbf{M}_n)$, and consequently $p(n) \leq m(n) + 1$. Therefore, we require $R > m(n)$ to extract $p(n)$ linearly independent parity relations.

On the other hand, to minimize the probability that a non-relation (i.e., an incorrect linear combination of columns) happens to evaluate to zero across all sampled rows, we choose R with additional slack. For example, if we set $R > m(n) + W$, then the probability that at least one of the $2^{m(n)+1} - p(n)$ candidate (incorrect) linear relations among the columns holds for all R rows is approximately upper bounded by 2^{-uW} , where W is a positive integer and u is the bit-width of the underlying field. Note that the rows of $\mathbf{M}_n$ are generated using fresh randomness.

Algorithm 3 Gaussian-Elim

Input $\mathbf{M}_n$
Output Row-echelon form of $\mathbf{M}_n$

```

1:  $\text{pivot\_row} \leftarrow 1$  ▷ Initialize the pivot row
2: for  $\text{col} = 1$  to  $m(n) + 1$  do
3:   for  $\text{row} = \text{pivot\_row}$  to  $R$  do
4:     if  $\mathbf{M}_n(\text{row}, \text{col}) \neq 0$  then
5:        $\mathbf{M}_n(\text{row}, :) \leftarrow \mathbf{M}_n(\text{row}, :) / \mathbf{M}_n(\text{row}, \text{col})$  ▷ Normalize pivot to 1
6:        $\mathbf{M}_n(\text{pivot\_row}, :) \leftrightarrow \mathbf{M}_n(\text{row}, :)$  ▷ Swap the rows
7:       for  $i = 1$  to  $R$ ,  $i \neq \text{pivot\_row}$  do
8:          $\mathbf{M}_n(i, :) \leftarrow \mathbf{M}_n(i, :) - \frac{\mathbf{M}_n(i, \text{col})}{\mathbf{M}_n(\text{pivot\_row}, \text{col})} \mathbf{M}_n(\text{pivot\_row}, :)$ 
9:        $\text{pivot\_row} \leftarrow \text{pivot\_row} + 1$  ▷ Increase the pivot row
10:    break
11:  $\mathbf{D}_n \leftarrow \mathbf{M}_n$ 
12: return  $\mathbf{D}_n$ 

```

Within $\mathbf{D}_n$, there are two types of columns: *pivot* and *free*. Each free column corresponds to a parity relation, namely a set of columns in $\mathbf{D}_n$ whose sum equals zero. To extract these parity relations, we employ Algorithm 4 (i.e., $\mathbf{P}_n \leftarrow \text{Extract-Linear}(\mathbf{D}_n)$).

Example 3.2. Consider the following, where $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{P}$ are matrices in a binary field:

$$M = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \xrightarrow{\text{Gaussian-Elim}} D = \begin{bmatrix} 1 & 0 & 0 & \textcolor{blue}{1} \\ 0 & 1 & 0 & \textcolor{blue}{0} \\ 0 & 0 & 1 & \textcolor{blue}{1} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \xrightarrow{\text{Extract-Linear}} P = \begin{bmatrix} 1 & 0 & 1 & 1 \end{bmatrix}. \quad (3.11)$$

The first three columns of D are pivot columns, but the fourth one is a free column. This column, specified in blue, is the sum of columns 1 and 3, as reflected in P . $\square$

Algorithm 4 Extract-Linear

Input D_n
Output Parity matrix P_n

```

1:  $row_D = 1, row_P = 1$ 
2: for  $col_D = 1$  to  $m(n) + 1$  do
3:   if  $D_n(row_D, col_D) = 0$  &  $D_n(:, col_D) \neq \mathbf{0}$  then                                 $\triangleright$  A free variable
4:      $P_n(row_P, col_D) = 1$                                                           $\triangleright$  Add a row to  $P_n$ 
5:     for  $j = 1$  to  $row_D - 1$  do
6:        $P_n(row_P, Pivots(j)) = D_n(j, col_D)$ 
7:      $row_P = row_P + 1$ 
8:   elseif  $D_n(:, col_D) \neq \mathbf{0}$  then
9:      $Pivots(row_D) = col_D$                                                           $\triangleright$  Record pivot column of each row
10:     $row_D = row_D + 1$ ;
11: return  $P_n$ 

```

Sampling P_n^r from P_n . P_n encompasses all the linear dependencies of SC, with its number of rows equaling the number of these relations. The next step is to sample leaking members of Σ_{SC} and derive the corresponding system of equations as in (3.5). To sample an instance of P_n^r , we proceed as follows: First, P_n^r is set to P_n . Then, each column (apart from column 1) of P_n^r is randomly and independently set to all-zero with probability ϵ , indicating that the intermediate associated with this column has leaked.

Remark 3.4. In sampling P_n^r , the decision for each column is made separately. Consequently, it is possible to consider non-identical values of the leakage rate ϵ for different members of Σ_{SC} . In practice, side-channel noise over different intermediates may not be equal, and hence, their corresponding ϵ values will be dissimilar [DFS19].

Estimating E_{SC} from sampled P_n^r s. Algorithm 5 outlines the steps to approximate E_{SC} using N number of trials. The quality of the approximation

improves as the value of N increases. For the numerical results presented, we used a step size of 10^{-2} for sweeping ϵ values.

Algorithm 5 Approximate E_{SC}

Input The family SC
Output $E_{\text{SC}}(n, \epsilon)$

```

1: for  $n = 2$  to  $n_{\max}$  do
2:    $M_n = \text{Create-M}(\text{SC}, n)$ 
3:    $D_n = \text{Gaussian-Elim}(M_n)$ 
4:    $P_n = \text{Extract-Linear}(D_n)$ 
5:   for  $\epsilon = \epsilon_{\min}$  to  $\epsilon_{\max}$  do
6:      $N_{\text{recovered}} = 0$ 
7:     for  $i = 0$  to  $N$  do
8:        $P_n^r \leftarrow \text{Sample}(P_n, \epsilon)$ 
9:        $G = \text{Gaussian-Elim}(P_n^r)$ 
10:      if  $G(1, 2 : m(n) + 1) = \mathbf{0}$  then                                 $\triangleright$  The secret is recovered
11:         $N_{\text{recovered}} = N_{\text{recovered}} + 1$ 
12:       $E_{\text{SC}}(n, \epsilon) = \frac{N_{\text{recovered}}}{N}$                                  $\triangleright$  Approximation with empirical mean
13: return  $E_{\text{SC}}$ 

```

After obtaining a table of empirical mean values for $E_{\text{SC}}(n, \epsilon)$, efforts are made to bound them, if possible, such that $E_{\text{SC}}(n, \epsilon) \leq \alpha(\beta\epsilon)^\gamma$, where α , β , and γ are constants with $\gamma < 1$. This expression facilitates the deployment of the results obtained for various gadgets in the security evaluation of more complex circuits, such as an S-box. Moreover, deriving a single expression that fits all the tested (n, ϵ) pairs enhances the credibility of estimations. However, we note that the derived bound may not hold out of the tested region.

3.2.2 Case Study on Linear Gadgets

The serial composition of k (linear) SR gadgets, as depicted in Figure 3.1, represents an example of a linear SC with a single native random variable V . As highlighted by Dziembowski et al. [DFZ19], this construction is not only theoretically significant but also holds practical relevance. We will also make use of the results for the case $k = 1$ later.

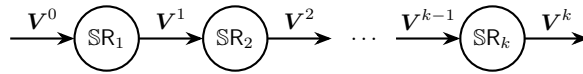


Figure 3.1: Multiple refresh gadgets working in tandem.

Instances of SR include SR-Simple [RP10], SR-SNI [BBD⁺16], and SR-

Rot [BDF⁺17], which are described in Algorithms 1, 6, and 7, respectively. The input to these algorithms is the share vector $\mathbf{V}^1 = (V_1^1, \dots, V_n^1)$ and the output is $\mathbf{V}^2 = (V_1^2, \dots, V_n^2)$.

Algorithm 6 SR-SNI

Input $\mathbf{V}^1 = (V_1^1, \dots, V_n^1)$
Output $\mathbf{V}^2 = (V_1^2, \dots, V_n^2)$

- 1: **for** $i = 1$ **to** n **do**
- 2: **for** $j = i + 1$ **to** n **do**
- 3: $R \xleftarrow{\$} \mathbb{F}_{2^u}$
- 4: $V_i^1 \leftarrow V_i^1 \oplus R$
- 5: $V_j^1 \leftarrow V_j^1 \oplus R$
- 6: $\mathbf{V}^2 \leftarrow \mathbf{V}^1$
- 7: **return** $\mathbf{V}^2$

Algorithm 7 SR-Rot

Input $\mathbf{V}^1 = (V_1^1, \dots, V_n^1)$
Output $\mathbf{V}^2 = (V_1^2, \dots, V_n^2)$

- 1: **for** $i = 1$ **to** n **do**
- 2: $R_i \xleftarrow{\$} \mathbb{F}_{2^u}$
- 3: **for** $i = 1$ **to** 2 **do**
- 4: **for** $j = 1$ **to** n **do**
- 5: $R \leftarrow R_{(i+j-2)\%n+1}$
- 6: $V_j^1 \leftarrow V_j^1 \oplus R$
- 7: $\mathbf{V}^2 \leftarrow \mathbf{V}^1$
- 8: **return** $\mathbf{V}^2$

SR-Simple requires the least amount of fresh randomness and has the least computational complexity. In contrast, SR-SNI involves more fresh randomness and is computationally more demanding. However, the additional randomness and increased processing complexity do not make SR-SNI more secure (based on the E_{SR} value) compared to SR-Simple and SR-Rot. For any tuple $(k, n > 2, \epsilon)$ studied, $E_{\text{SR} \rightarrow \dots \rightarrow \text{SR}}(n, \epsilon)$ is higher with SR-SNI than with the other two. Refer to the results in Figure 3.2. Nonetheless, SR-SNI shows good properties that help to prove the RPM security of combinations of gadgets. This point will be treated in Section 3.3.

Remark 3.5. Within our tested range ($n \leq 30$), we observed that for SR-Simple and SR-Rot, the function $E_{\text{SR}}(n, \epsilon)$ is non-increasing in n for all tested ϵ . In contrast, for SR-SNI this monotonicity breaks around $\epsilon \approx 0.45$: for larger leakage rates, $E_{\text{SR-SNI}}(n, \epsilon)$ increases with n .

Estimating $E_{\text{SR-SNI}}(n, \epsilon)$. The value of $E_{\text{SR-SNI}}(n, \epsilon)$ is used for the security evaluation of certain circuits in this chapter. Using Algorithm 5, we can empirically estimate $\frac{1}{n} \log_{10}(E_{\text{SR-SNI}}(n, \epsilon))$ for various orders and leakage rates. These estimations for the range $\epsilon < 0.15$ are plotted in Figure 3.3. From these results, it gets apparent that there exists a value γ such that $\frac{1}{n} \log_{10}(E_{\text{SR-SNI}}(n, \epsilon)) \leq \gamma \log_{10}(\epsilon)$. By setting $\gamma = 0.6$, we satisfy this bound for the range $\epsilon < 0.15$. Hence, we can derive the following expression for $E_{\text{SR-SNI}}(n, \epsilon)$, which is valid for $(n \geq 2, \epsilon < 0.15)$:

$$E_{\text{SR-SNI}}(n, \epsilon) \leq \epsilon^{0.6n}. \quad (3.12)$$

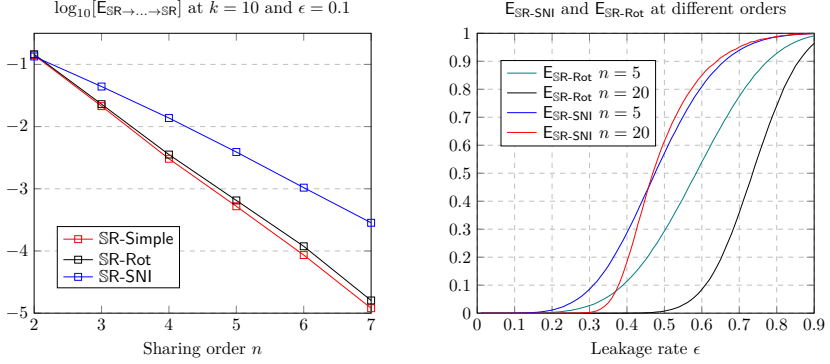


Figure 3.2: (Left) $\log_{10}(\mathbb{E}_{\text{SR} \rightarrow \dots \rightarrow \text{SR}}(n, 0.1))$ for a chain of length $k = 10$. (Right) For higher leakage rates (around $\epsilon > 0.45$), $\mathbb{E}_{\text{SR-SNI}}(n, \epsilon)$ increases with the order n .

Upon closer inspection of the results in Figure 3.3, it becomes evident that for lower leakage rates, we can derive tighter bounds for $\mathbb{E}_{\text{SR-SNI}}(n, \epsilon)$ by choosing larger values for γ . Additionally, we observe that as ϵ increases, γ should decrease. However, $\mathbb{E}_{\text{SR-SNI}}(n, \epsilon)$ in the range $\epsilon \leq 0.15$ is a decreasing function of n . We illustrate this with depictions of the estimated $\log_{10}(\mathbb{E}_{\text{SR-SNI}}(n, \epsilon))$ in Figure 3.3-(Right).

Remark 3.6. By deploying more complex functions of (n, ϵ) , one can derive tighter expressions for $\mathbb{E}_{\text{SR-SNI}}(n, \epsilon)$. However, for the illustration of our proposed methodology, the given upper bound in (3.12) suffices.

As the order of masking increases, empirical estimation of a probability that exponentially decays with n becomes more challenging. This difficulty arises not only because the size of gadgets such as SR-SNI grows in $\mathcal{O}(n^2)$, limiting the number of trials that can be performed in terms of computational power, but also because the events become rare and their probability falls below $\frac{1}{N}$. Employing Algorithm 5 with $N = 10^8$ trials, we conclude that for our tested range $n \leq 30$, no violation of the bound given in (3.12) was observed. We also note that there is no guarantee that the derived bound holds outside the tested region of n values. Specifically, for understanding the behavior of $\mathbb{E}_{\text{SR-SNI}}(n, \epsilon)$ at $n \rightarrow \infty$, we need rigorous analytical approaches, which we leave for future research.

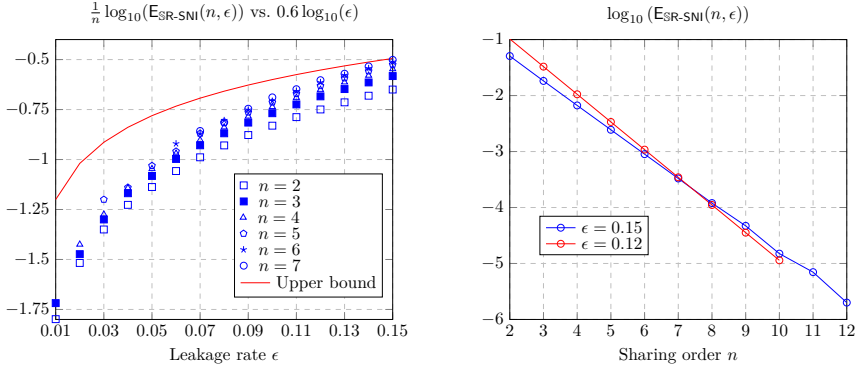


Figure 3.3: (Left) Estimation results for $E_{\text{SR-SNI}}$. (Right) The slope of decay of $\log_{10}(E_{\text{SR-SNI}}(n, \epsilon))$ is independent of order n , but depends on leakage rate ϵ .

3.3 Reduction in RPM Framework

Suppose $\mathbb{S}G_1$ and $\mathbb{S}G_2$, that both can be nonlinear, are processing the same native variable V sequentially. Instead of directly passing the output V^0 from $\mathbb{S}G_1$ to $\mathbb{S}G_2$, there is an intermediary (linear) SR gadget. See Figure 3.4. The adversary's objective is to exploit leakages of the three gadgets and estimate the value of V .

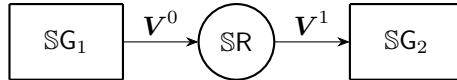


Figure 3.4: Composition using a refresh gadget.

We will transfer leakage of SR to its input/output interfaces and thereby *reduce* the security of this chain to the security of its composing gadgets. More precisely, we propose the following theorem.

Theorem 3.7. *For a bounded region of ϵ values, the gadgets, and hence the composition, behave as an erasure channel for which*

$$E_{\mathbb{S}G_1 \rightarrow \text{SR} \rightarrow \mathbb{S}G_2}(n, \epsilon) \leq E_{\mathbb{S}G_1}(n, \epsilon') + E_{\text{SR}}(n, \epsilon) + E_{\mathbb{S}G_2}(n, \epsilon') \quad (3.13)$$

holds, where $\epsilon' \geq \epsilon$ is a function of (n, ϵ) and the structure of SR.

This reduction is a significant step toward estimating the RPM security of more intricate compositions. The remainder of this section consists of technical steps to derive (3.13). Before delving into details, let us briefly develop some intuition about the impact of internal leakage of SR.

3.3.1 Reduction with Leak-Free SR Gadget

By assuming that the SR gadget is leak-free (i.e., an adversary cannot probe its internal variables), we can decompose the security of the construction $\mathbb{SG}_1 \rightarrow \mathbb{SR} \rightarrow \mathbb{SG}_2$ in the mutual-information metric into the security of $\mathbb{SG}_1$ and $\mathbb{SG}_2$. A leak-free SR gadget is typically required to preserve the conditional independence of $\mathbf{V}^0$ and $\mathbf{V}^1$, which is critical for the proofs in [PR13, MS23, BCGR24]. However, the SR gadget (like the other SGs) is a piece of code, and the adversary can access its internal variables just like any other variable of the overall masked algorithm. Therefore, the assumption of leak-free SR gadgets is not practically sound.

Effectively incorporating leakage from SR gadgets, rather than neglecting it, is a long-standing challenge in the RPM. As a contribution to this problem, we provide a relatively tight bound on the additional advantage that an adversary can gain from leakage of an SR gadget in our security metric.

Effects of leakage from SR gadgets. In the following, we describe two intuitive effects caused by leakage from SR:

1. **Direct effect.** Since the secret realization v depends on the internal variables in $\Sigma_{\mathbb{SR}}$, if the adversary learns a sufficiently large subset of $\Sigma_{\mathbb{SR}}$, it may recover v directly.
2. **Indirect effect.** Leakage from $\Sigma_{\mathbb{SR}}$ may induce additional parity relations between $\mathbf{V}^0$ and $\mathbf{V}^1$. These extra equations can couple the parity relations of $\mathbb{SG}_1$ and $\mathbb{SG}_2$, thereby reducing the RPM security of the overall composition.

3.3.2 A Describing System of Equations

Let the following set of equations define $\mathbb{SG}_1 \rightarrow \mathbb{SR} \rightarrow \mathbb{SG}_2$, where the boundary variables, $\mathbf{V}^0$ and $\mathbf{V}^1$, are separated from the list of internal variables, and NL (resp. L) denotes a nonlinear (resp. linear) set of equations:

$$\mathbf{A}_n = \begin{cases} \text{NL}_{\mathbb{S}\mathbf{G}_1}(\Sigma_{\mathbb{S}\mathbf{G}_1}, \mathbf{V}^0, V) = \mathbf{0}, \\ \text{L}_{\mathbb{S}\mathbf{R}}(\Sigma_{\mathbb{S}\mathbf{R}}, \mathbf{V}^0, \mathbf{V}^1, V) = \mathbf{0}, \\ \text{NL}_{\mathbb{S}\mathbf{G}_2}(\Sigma_{\mathbb{S}\mathbf{G}_2}, \mathbf{V}^1, V) = \mathbf{0}. \end{cases} \quad (3.14)$$

Our primary focus will be on the structure of the linear subsystem; thus, it is highlighted with a different color to assist following the operations. We denote the system with $\mathbf{A}_n$ to emphasize the dependency of the relations on n . We have deviated slightly from our notation: for the gadgets, we have excluded boundary variables, i.e., $\mathbf{V}^0$, $\mathbf{V}^1$ or both, from the corresponding Σ sets. Hence, $\Sigma_{\mathbb{S}\mathbf{G}_1}$, $\Sigma_{\mathbb{S}\mathbf{R}}$, and $\Sigma_{\mathbb{S}\mathbf{G}_2}$ contain only the internal variables. This way, we avoid accounting for the leakage of boundary variables twice.

Remark 3.8. Joint equation sets in $\mathbf{A}_n$ empower the adversary more than separated subsets. As an extreme case, the three sets combined may uniquely determine the secret V , but V appears uniformly random while considering each system alone.

For the subsequent discussions, we place two requirements on the gadgets: *symmetry* over the shares for all the gadgets and *maximal independence* for the $\mathbb{S}\mathbf{R}$.

Definition 3.9 (Symmetry). A gadget is symmetric if any reordering of the shares inside it does not change the parity description.

Definition 3.10 (Maximal Independence). In a linear $\mathbb{S}\mathbf{R}$ gadget, input $\mathbf{V}^0$ and output $\mathbf{V}^1$ are *maximally independent* if, apart from $\oplus_{i=1}^n V_i^0 = \oplus_{i=1}^n V_i^1$, no other relation exists among the input and output shares.

For instance, $\mathbb{S}\mathbf{R}\text{-SNI}$ and $\mathbb{S}\mathbf{R}\text{-Rot}$ are symmetric, but $\mathbb{S}\mathbf{R}\text{-Simple}$ is not because the equation defining the last share differs from that of the other shares. For these two $\mathbb{S}\mathbf{R}$ s, input and output shares are maximally independent.²

Example 3.3. To clarify these definitions, we provide a simple instance of a refresh gadget that is symmetric but does not have maximally independent input-output shares. For $n = 4$, consider a refresh gadget that operates as follows:

$$\begin{aligned} (V_1^1, V_2^1, V_3^1, V_4^1) &\leftarrow (V_1^0, V_2^0, V_3^0, V_4^0) \oplus (R_1, R_2, R_3, R_4), \\ (V_1^1, V_2^1, V_3^1, V_4^1) &\leftarrow (V_1^1, V_2^1, V_3^1, V_4^1) \oplus (R_2, R_1, R_4, R_3). \end{aligned} \quad (3.15)$$

²This can be verified empirically by sampling many input/output pairs and extracting all parity relations using the *Create-M*, *Gaussian-Elim*, and *Extract-Linear* pipeline, restricting $\mathbf{M}$ to the $2n$ columns corresponding to the input/output shares.

Here, R_1 to R_4 are randomness variables, and the $\oplus$ operation for two vectors is element-wise. While it is clear that $\oplus_{i=1}^4 V_i^0 = \oplus_{i=1}^4 V_i^1$, this is not the only relation. Indeed, we additionally have $V_1^0 \oplus V_2^0 = V_1^1 \oplus V_2^1$ and $V_3^0 \oplus V_4^0 = V_3^1 \oplus V_4^1$. $\square$

Definition 3.11 (Primary Solution). $\mathbf{A}_n$ in each run of the underlying circuit has a specific solution, which is the realized value of its variable. We call it the primary solution and denote it by S^* .

3.3.3 Modeling the Refresh Gadget

We first focus on a single instance of leakage and then extend the obtained results to the general case of random leakage. Let the adversary learn a chunk of the primary solution S^* via leakage. By substituting it into $\mathbf{A}_n$, a new system of equations as (3.16), denoted $\mathbf{A}_n^r$, will emerge:

$$\mathbf{A}_n^r = \begin{cases} \text{NL}_{\text{SG}_1}^r(\Sigma_{\text{SG}_1}^r, \mathbf{V}^{0,r}, V) = \mathbf{b}_1, \\ \textcolor{blue}{\text{L}}_{\text{SR}}^r(\Sigma_{\text{SR}}^r, \mathbf{V}^{0,r}, \mathbf{V}^{1,r}, V) = \mathbf{b}_2, \\ \text{NL}_{\text{SG}_2}^r(\Sigma_{\text{SG}_2}^r, \mathbf{V}^{1,r}, V) = \mathbf{b}_3. \end{cases} \quad (3.16)$$

The superscript r specifies the remaining unknowns and the relations among them. $\mathbf{b}_1$ to $\mathbf{b}_3$ are known vectors that appear after the substitution. Based on the leaked shares of $\mathbf{V}^0$ and $\mathbf{V}^1$, we have the following identities:

$$\begin{cases} V = (\oplus \mathbf{V}^{0,r}) \oplus b_4, \\ V = (\oplus \mathbf{V}^{1,r}) \oplus b_5, \end{cases} \quad (3.17)$$

where $\oplus$ before a set means XOR of all its entries, and scalar values b_4 and b_5 are respective XOR of the leaked shares of $\mathbf{V}^0$ and $\mathbf{V}^1$. The first row of (3.17) is a parity relation that is common to $\text{NL}_{\text{SG}_1}^r$ and $\textcolor{blue}{\text{L}}_{\text{SR}}^r$, and the second row is common to $\text{NL}_{\text{SG}_2}^r$ and $\textcolor{blue}{\text{L}}_{\text{SR}}^r$.

Extracting non-informative parity relations. Our aim is to simplify $\mathbf{A}_n^r$ by decomposing parity relations in $\textcolor{blue}{\text{L}}_{\text{SR}}^r$ into two groups of non-informative and informative parity relations.

Lemma 3.12. *Without changing the set of solutions of $\mathbf{A}_n^r$, $\textcolor{blue}{\text{L}}_{\text{SR}}^r$ can be grouped into the following two linearly independent sets of equations:*

$$\begin{cases} \text{L}_1(\Sigma_{\text{SR}}^r) = \text{L}_2(\mathbf{V}^{0,r}, \mathbf{V}^{1,r}, V) \oplus \mathbf{b}_{2,1}, \\ \text{L}_3(\mathbf{V}^{0,r}, \mathbf{V}^{1,r}, V) = \mathbf{b}_{2,2}, \end{cases} \quad (3.18)$$

where $\mathbf{b}_{2,1}$ and $\mathbf{b}_{2,2}$ are constant vectors, and equations in L_1 are linearly independent of each other.

Proof. A coefficients matrix $\mathbf{P}$ can describe $\mathbf{L}_{\text{SR}}^r(\Sigma_{\text{SR}}^r, \mathbf{V}^{0,r}, \mathbf{V}^{1,r}, V) = \mathbf{b}_2$ as

$$\mathbf{P} \cdot [\Sigma_{\text{SR}}^r, \mathbf{V}^{0,r}, \mathbf{V}^{1,r}, V]^\top = \mathbf{b}_2. \quad (3.19)$$

Using $\mathbf{P}$, we create an *augmented matrix* $\mathbf{P}'$ by appending $\mathbf{b}_2$ to it as $\mathbf{P}' = [\mathbf{P} | \mathbf{b}_2]$. Let the row-echelon form of $\mathbf{P}'$ be $\mathbf{G} = \text{Gaussian-Elim}(\mathbf{P}')$. From $\mathbf{G}$, we can derive L_1 , L_2 , and L_3 as defined in the lemma.

Inside $\mathbf{G}$, because it is a row-echelon matrix, if the i th row is all-zero, then all the rows beneath i are all-zero. Let I be the first row of $\mathbf{G}$ that is all-zero. In any row $i < I$, there exists a column J such that $\mathbf{G}(i, J) = 1$, and for every $j < J$, $\mathbf{G}(i, j) = 0$. Column J is called a *pivot* variable. Let function $\text{Pivots}(\cdot)$, for each row $i < I$, give its corresponding column J . $\text{Pivots}(\cdot)$ is a monotonically increasing function. We define $\text{Pivots}^{-1}(j)$ as the biggest number r such that $\text{Pivots}(r) \leq j$.

Inside $\mathbf{P}$, define the last column containing elements of Σ_{SR}^r with $k = |\Sigma_{\text{SR}}^r|$. Note that Σ_{SR}^r may contain an exact copy of $\mathbf{V}^0$ and $\mathbf{V}^1$. However, this will not alter our procedure or results.

We split $\mathbf{G}$'s columns into three parts as $\mathbf{G} = [\mathbf{G}_1 | \mathbf{G}_2 | \mathbf{G}_3]$. Where, $\mathbf{G}_1 = \mathbf{G}(:, 1 : k)$, $\mathbf{G}_2 = \mathbf{G}(:, k + 1 : \text{end} - 1)$, and $\mathbf{G}_3 = \mathbf{G}(:, \text{end})$. We also split the rows of $\mathbf{G}$ into the following two disjoint sets.

- Rows $i = 1$ to $T = \min[I - 1, \text{Pivots}^{-1}(k)]$. The parity equations corresponding to these rows have a pivot variable from Σ_{SR}^r . That is because, at least, $\mathbf{G}(i, \text{Pivots}(i)) = 1$, and since $\text{Pivots}(\cdot)$ is monotonically increasing, we have:

$$\text{Pivots}(i) \leq \text{Pivots}(T) = \min[\text{Pivots}(I - 1), k] \leq k.$$

For these parity equations, we can write:

$$\mathbf{G}_1(i, :) \cdot [\Sigma_{\text{SR}}^r]^\top \oplus \mathbf{G}_2(i, :) \cdot [\mathbf{V}^{0,r}, \mathbf{V}^{1,r}, V]^\top = \mathbf{G}_3(i). \quad (3.20)$$

Based on (3.20), we label the equations defined by $\mathbf{G}_1(1 : T, :)$ as L_1 , the equations defined by $\mathbf{G}_2(1 : T, :)$ as L_2 , and the constants at the column vector $\mathbf{G}_3(1 : T)$ as $\mathbf{b}_{2,1}$. Each row of $\mathbf{G}_1(1 : T, :)$ has a pivot variables. Therefore, these rows (that are also denoted by L_1) are linearly independent.

- Rows $i = T + 1$ to $I - 1$. These rows exist only if $\text{Pivots}^{-1}(k) < I - 1$. For these rows, we have $\text{Pivots}(i) > k$. Consequently, $\mathbf{G}_1(i, :) = \mathbf{0}$. Hence,

no variable from Σ_{SR}^r will exist in the equations defined by these rows, for rows $T < i < I$, we can write:

$$\mathbf{G}_2(i, :) \times [\mathbf{V}^{0,r}, \mathbf{V}^{1,r}, V]^\top = \mathbf{G}_3(i). \quad (3.21)$$

Based on (3.21), we label the equations defined by $\mathbf{G}_2(T+1 : I-1, :)$ as $\mathbf{L}_3$, and the constants at the column vector $\mathbf{G}_3(T+1 : I-1)$ as $\mathbf{b}_{2,2}$.

Finally, the equations in $\mathbf{G}$ will be of the following structure, and this completes proof of the lemma.

$$\begin{cases} \mathbf{L}_1(\Sigma_{\text{SR}}^r) = \mathbf{L}_2(\mathbf{V}^{0,r}, \mathbf{V}^{1,r}, V) \oplus \mathbf{b}_{2,1}, \\ \mathbf{L}_3(\mathbf{V}^{0,r}, \mathbf{V}^{1,r}, V) = \mathbf{b}_{2,2}. \end{cases} \quad (3.22)$$

□

Example 3.4. As a toy example, to illustrate the proof of the lemma, consider the following instance of matrix $\mathbf{G}$:

$$\mathbf{G} = \left[\begin{array}{cccc|cccc|c} 1 & 0 & 0 & 0 & \alpha_0 & 0 & \beta_0 & 0 & \gamma_0 & \delta_0 \\ 0 & 1 & 0 & 0 & \alpha_1 & 0 & \beta_1 & 0 & \gamma_1 & \delta_1 \\ 0 & 0 & 0 & 1 & \alpha_2 & 0 & \beta_2 & 0 & \gamma_2 & \delta_2 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 & \beta_3 & 0 & \gamma_3 & \delta_3 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right]. \quad (3.23)$$

In this example, we have set $k = 5$, so the first 5 columns correspond to the variables of Σ^r . We also consider the next four columns as $[\mathbf{V}^{0,r}, \mathbf{V}^{1,r}, V]$. The last column is the augmentation with the $\mathbf{b}_2$ vector, see (3.19). Columns 1, 2, 4, and 6 are pivots. Column 3 and 8 are part of leakage, and column 5, 7, and 9 are free. The respective pivot columns of the rows (1, 2, 3, 4) are (1, 2, 4, 6). The 5th row of $\mathbf{G}$ is the first all-zero row. Hence, we have $I = 5$, and $T = \min[I - 1, \text{Pivots}^{-1}(k)] = \min[4, 3] = 3$. Therefore, the first 3 rows identify $\mathbf{L}_1$, $\mathbf{L}_2$, and $\mathbf{b}_{2,1}$. The 4th row is the only relation in $\mathbf{L}_3$. □

Next, we show that the subsystem given in the first row of (3.18) will not change the probability distribution of $\tilde{V}$ (the adversary's output). Therefore, we label them as non-informative parity relations. Lemma 3.13 gives the necessary technical conditions.

Lemma 3.13. *Consider the following system of equations over $\mathbb{F}_{2^u}$:*

$$\begin{cases} \mathbf{L}_4(\Sigma_1) = \mathbf{L}_5(V, \Sigma_2), \\ \text{NL}(V, \Sigma_2, \Sigma_3) = \mathbf{0}. \end{cases} \quad (3.24)$$

If Σ_1 , Σ_2 , and Σ_3 are pairwise disjoint, and the equations in L_4 are linearly independent, then the posterior distribution of $\tilde{V}$ (conditioned on the full system) depends only on the nonlinear subsystem $NL(V, \Sigma_2, \Sigma_3) = \mathbf{0}$. Consequently, for the purpose of estimating V , the linear subsystem $L_4(\Sigma_1) = L_5(V, \Sigma_2)$ can be ignored in (3.24).

Proof. Since we work over a finite field, each subsystem admits either no solution or finitely many solutions. Let $\mathcal{S} = \{S^1, \dots, S^N\}$ denote the set of all solutions of the nonlinear subsystem $NL(V, \Sigma_2, \Sigma_3) = \mathbf{0}$, where each S^i is a tuple of the form $S^i = (v^i, \sigma_2^i, \sigma_3^i)$.

Fix any $S^i \in \mathcal{S}$. Then $L_5(V, \Sigma_2)$ evaluates to a known constant right-hand side, which we denote by $\mathbf{d}^i := L_5(v^i, \sigma_2^i)$. We now consider the linear system

$$L_4(\Sigma_1) = \mathbf{d}^i.$$

Because the equations in L_4 are linearly independent, this system is consistent and has exactly $|\ker(L_4)|$ solutions, namely

$$\Sigma_1 = \sigma_{1,p}^i \oplus \sigma_{1,f}, \quad (3.25)$$

where $\sigma_{1,p}^i$ is any particular solution and $\sigma_{1,f}$ ranges over the homogeneous solution space $\ker(L_4) = \{\sigma_1 : L_4(\sigma_1) = \mathbf{0}\}$. Importantly, $|\ker(L_4)|$ depends only on L_4 and is therefore independent of i (equivalently, independent of $\mathbf{d}^i$).

Since Σ_1 is disjoint from (V, Σ_2, Σ_3) , every solution S^i of the nonlinear subsystem can be extended to exactly $|\ker(L_4)|$ solutions of the full system (3.24), and this multiplicative factor is the same for all i . Thus, when forming the posterior distribution of $\tilde{V}$ via counting solutions (as in (3.6)), this common factor cancels out in the numerator and denominator, and the resulting distribution depends only on the nonlinear subsystem. $\square$

Example 3.5. To illustrate how the decomposition of parity relations of SR into two groups of informative and non-informative works, we provide a simple example of a refresh gadget for $n = 3$ as follows:

$$(V_1^1, V_2^1, V_3^1) \leftarrow (V_1^0, V_2^0, V_3^0) \oplus (R_1, R_2, R_1 \oplus R_2). \quad (3.26)$$

Assume the leakage has only revealed the value of R_1 to the adversary as $R_1 = b$. We can arrange the parity equations describing the given refresh instance as follows:

$$\begin{cases} V_1^0 \oplus V_2^0 \oplus V_3^0 \oplus V = 0, \\ V_1^0 \oplus V_1^1 = b, \\ R_2 \oplus V_2^0 \oplus V_2^1 = 0, \\ R_2 \oplus V_3^0 \oplus V_3^1 = b. \end{cases} \quad (3.27)$$

Using the approach of (the proof of) Lemma 3.12, this parity system can be reordered to yield:

$$\begin{cases} R_2 \oplus V_2^0 \oplus V_2^1 = b, \\ V_1^0 \oplus V_2^1 \oplus V_3^1 \oplus V = b, \\ V_2^0 \oplus V_3^0 \oplus V_2^1 \oplus V_3^1 = b, \\ V_1^1 \oplus V_2^1 \oplus V_3^1 \oplus V = 0. \end{cases} \quad (3.28)$$

The first equation, in the format of the first row of (3.24), is non-informative (per Lemma 3.13). Intuitively, this equation increases the number of unknowns by one, and this unknown (R_2) is not related to the rest of the system. Note that if such dummy equations were helpful, the adversary could add many of them, each introducing a new unknown, in the hope of increasing the chance of gaining more knowledge about V . However, we know that these artificial parities are not helpful. $\square$

Processing the informative parity relations. By application of Lemmas 3.12 and 3.13, we can decompose A_n^r as follows:

$$A_n^r = \begin{cases} \text{NL}_{\text{SG}_1}^r(\Sigma_{\text{SG}_1}^r, \mathbf{V}^{0,r}, V) = \mathbf{b}_1, \\ \text{L}_{\text{SR},3}^r(\mathbf{V}^{0,r}, \mathbf{V}^{1,r}, V) = \mathbf{b}_{2,2}, \\ \text{NL}_{\text{SG}_2}^r(\Sigma_{\text{SG}_2}^r, \mathbf{V}^{1,r}, V) = \mathbf{b}_3. \end{cases} \quad (3.29)$$

The linear equation set $\text{L}_{\text{SR},3}^r$, which only depends on the boundary shares and V , is obtained via the process in Lemma 3.12, and they are the only informative parities in L_{SR}^r .

From Lemma 3.3, we know that a linear system as $\text{L}_{\text{SR},3}^r(\mathbf{V}^{0,r}, \mathbf{V}^{1,r}, V) = \mathbf{b}_{2,2}$ either recovers V or gives nothing about it. Let e_{SR} be the event that $\text{L}_{\text{SR},3}^r$ alone determines the secret V . If e_{SR} occurs, V is recovered. Otherwise, we proceed with narrowing down the equations in $\text{L}_{\text{SR},3}^r$: for each equation containing V , we add either of the identities in (3.17) and, consequently, remove V from the resulting equation. At this point, the trivial relation

$$(\oplus \mathbf{V}^{0,r}) \oplus b_4 = (\oplus \mathbf{V}^{1,r}) \oplus b_5 \quad (3.30)$$

will always exist in this system. Based on the maximally independence requirement, there will be no other *omnipresent equation*³ in this system.

Relation (3.30) is a dependent equation: it is XOR of the first line of (3.17), present in $\text{NL}_{\text{SG}_1}^r$, and the second line of (3.17), present in $\text{NL}_{\text{SG}_2}^r$. Therefore, we

³Omnipresent in the sense that it exists irrespective of the realization of leakage.

ignore (3.30) and denote the final system with $\mathbf{L}_{\text{SR},4}^r(\mathbf{V}^{0,r}, \mathbf{V}^{1,r}) = \mathbf{b}_{2,3}$.⁴ By substituting this into (3.29), we get

$$\mathbf{A}_n^r = \begin{cases} \text{NL}_{\text{SG}_1}^r(\Sigma_{\text{SG}_1}^r, \mathbf{V}^{0,r}, V) = \mathbf{b}_1, \\ \mathbf{L}_{\text{SR},4}^r(\mathbf{V}^{0,r}, \mathbf{V}^{1,r}) = \mathbf{b}_{2,3}, \\ \text{NL}_{\text{SG}_2}^r(\Sigma_{\text{SG}_2}^r, \mathbf{V}^{1,r}, V) = \mathbf{b}_3. \end{cases} \quad (3.31)$$

In this phase, the subsystem defined by $\mathbf{L}_{\text{SR},4}^r$ is without V . Our next goal is to separate $\mathbf{A}_n^r$ into systems governed by $\text{NL}_{\text{SG}_1}^r$ and $\text{NL}_{\text{SG}_2}^r$.

Example 3.6. Building on Example 3.5, we illustrate how the three informative equations in (3.28) can be narrowed down to derive $\mathbf{L}_{\text{SR},4}^r$. The subsystem consisting of the three last equations of (3.28) corresponds to $\mathbf{L}_{\text{SR},3}^r$. We aim to derive $\mathbf{L}_{\text{SR},4}^r$ in this example. Here, the boundary parity relations are:

$$\begin{cases} V = V_1^0 \oplus V_2^0 \oplus V_3^0, \\ V = V_1^1 \oplus V_2^1 \oplus V_3^1. \end{cases} \quad (3.32)$$

Based on the discussion around (3.30), these two boundary relations are already present outside of SR (in SG_1 and SG_2 , respectively). With the help of boundary parities, we eliminate V from $\mathbf{L}_{\text{SR},3}^r$ and obtain:

$$\begin{cases} V_1^0 \oplus V_1^1 = b, \\ V_2^0 \oplus V_3^0 \oplus V_2^1 \oplus V_3^1 = b. \end{cases} \quad (3.33)$$

Now, any parity relation in $\mathbf{L}_{\text{SR},3}^r$ that is linearly dependent in the presence of these boundary relations can be eliminated. For instance, if we select the parity relation $V_1^0 \oplus V_1^1 = b$, the other one will be dependent and can be ignored. Consequently, $\mathbf{L}_{\text{SR},4}^r$ will be $V_1^0 \oplus V_1^1 = b$. Note that we could have selected the other parity, namely $V_2^0 \oplus V_3^0 \oplus V_2^1 \oplus V_3^1 = b$, as $\mathbf{L}_{\text{SR},4}^r$. The first choice has only two unknowns. Hence, it is *sparser* than the second one, which contains four unknowns. However, both can be considered as $\mathbf{L}_{\text{SR},4}^r$. $\square$

Disclosing unknowns of residual parity equations. To facilitate the separation of $\mathbf{A}_n^r$ into its constituent subsystems, we allow the adversary to learn the *remaining unknowns* of $\mathbf{L}_{\text{SR},4}^r$. This additional leakage increases the success probability of the adversary and consequently loosens the tightness

⁴Typically at this point, with small ϵ , there remains no equations in $\mathbf{L}_{\text{SR},4}^r$.

of the final security bound. Nevertheless, this intervention is crucial to our proposed security reduction.

We denote the unknowns of $\mathbf{L}_{\text{SR},4}^r$ that are members of $\mathbf{V}^0$ and $\mathbf{V}^1$ with $\mathbf{V}^{0,\dagger}$ and $\mathbf{V}^{1,\dagger}$, respectively. The disclosure of this extra leakage enables us to split $\mathbf{A}_n^r$ into two (almost) disjoint systems $\mathbf{A}_n^1$ and $\mathbf{A}_n^2$, with the only common variable being V :

$$\mathbf{A}_n^1 = \begin{cases} \text{NL}_{\text{SG}_1}^r(\Sigma_{\text{SG}_1}^r, \mathbf{V}^{0,r}, V) = \mathbf{b}_1, \\ \mathbf{V}^{0,\dagger} = \mathbf{b}_6, \end{cases} \quad (3.34)$$

$$\mathbf{A}_n^2 = \begin{cases} \text{NL}_{\text{SG}_2}^r(\Sigma_{\text{SG}_2}^r, \mathbf{V}^{1,r}, V) = \mathbf{b}_3, \\ \mathbf{V}^{1,\dagger} = \mathbf{b}_7. \end{cases} \quad (3.35)$$

In the context of Example 3.6, with $\mathbf{L}_{\text{SR},4}^r$ being $V_1^0 \oplus V_1^1 = b$, $\mathbf{V}^{0,\dagger}$ and $\mathbf{V}^{1,\dagger}$ are V_1^0 and V_1^1 , and their values are given to the adversary in accordance with the primary solution (See Definition 3.11). Choosing a sparser representation for $\mathbf{L}_{\text{SR},4}^r$ helps to derive a tighter security bound. However, finding a sparse representation is not a trivial problem. We leave optimizations of the results based on this direction for future work.

Counting the number of informative parity relations. The discussion up to here was based on a single instance of leakage. To proceed the reduction, we zoom out and make use of the probabilistic nature of leakage. In the random leakage setting, $\mathbf{V}^{0,\dagger}$ and $\mathbf{V}^{1,\dagger}$ are not fixed. We utilize the symmetry (Definition 3.9), and model the extra leakage provided through $\mathbf{V}^{0,\dagger}$ and $\mathbf{V}^{1,\dagger}$ as an *equivalent leakage rate* ϵ' on the interface vectors $\mathbf{V}^0$ and $\mathbf{V}^1$.

For each interface, we define $\mathbf{T}_{\text{SR}}(n, \epsilon)$ as:

$$\mathbf{T}_{\text{SR}}(n, \epsilon) = \mathbb{E}[|\mathbf{V}^{i,\dagger}|]. \quad (3.36)$$

Here, the expectation is over instances of leakage and i is in $\{0, 1\}$. For the particular case of SR-SNI, due to the present symmetry, we have $\mathbb{E}[|\mathbf{V}^{0,\dagger}|] = \mathbb{E}[|\mathbf{V}^{1,\dagger}|]$.

For a target SR at a desired pair (n, ϵ) , we estimate $\mathbf{T}_{\text{SR}}(n, \epsilon)$ using a Monte Carlo approach with N trials, where in the k th trial, we sample a random leakage and compute $h(k) = |\mathbf{V}^{i,\dagger}|$. For a sufficiently large N , the sample average, $\frac{1}{N} \sum_{k=1}^N h(k)$, approaches $\mathbf{T}_{\text{SR}}(n, \epsilon)$. We give a detailed algorithmic approach in the following.

Algorithm 8 Approximate T_{SR}

Input The family SR
Output $T_{\text{SR}}(n, \epsilon)$

```

1: for  $n = 2$  to  $n_{\max}$  do
2:    $M_n = \text{Create-M}(\text{SR}, n)$   $\triangleright$  Order of columns:  $[\Sigma_{\text{SR}} \mid V^0 \mid V^1 \mid V]$ ,  $m(n) + 1$  columns
3:    $D_n = \text{Gaussian-Elim}(M_n)$ 
4:    $P_n = \text{Extract-Linear}(D_n)$   $\triangleright L_{\text{SR}}(\Sigma_{\text{SR}}, V^0, V^1, V)$  as in (3.14)
5:   for  $\epsilon = \epsilon_{\min}$  to  $\epsilon_{\max}$  do
6:      $N_{\text{recovered}} = 0, h = 0$ 
7:     for  $i = 1$  to  $N$  do
8:        $P_n^r \leftarrow \text{Sample}(P_n, \epsilon)$   $\triangleright$  Sampling  $L_{\text{SR}}^r$  as in (3.16)
9:        $G = \text{Gaussian-Elim}(P_n^r)$   $\triangleright$  Approach of Lemma 3.12
10:      if  $G(\text{end}, 1 : m(n)) = 0$  then  $\triangleright e_{\text{SR}}$  event.  $V$  is in the last column of  $M$ 
11:         $N_{\text{recovered}} = N_{\text{recovered}} + 1$ 
12:        Continue
13:       $G_2 = \text{Extract}(G)$   $\triangleright$  In accordance with (3.21)
14:       $h = h + \text{Count-Unknowns}(G_2)$ 
15:       $T_{\text{SR}}(n, \epsilon) = \frac{h}{N - N_{\text{recovered}}}$   $\triangleright$  Approximation with empirical mean
16: return  $T_{\text{SR}}$ 

```

Algorithm for estimating $T_{\text{SR}}(n, \epsilon)$. Algorithm 8 outlines the procedure described in this section to estimate $T_{\text{SR}}(n, \epsilon)$ for a given SR gadget.

The function $\text{Count-Unknowns}(G_2)$ performs three tasks: first, it adds either $V = \oplus V^{0,r} \oplus b_4$ or $V = \oplus V^{1,r} \oplus b_5$ (as in (3.17)) to any row that has a non-zero coefficient for V . Then, it removes $\oplus V^{0,r} \oplus b_4 = \oplus V^{1,r} \oplus b_5$ from the set of rows of G_2 (see discussion around (3.30)). Finally, it counts the number of shares of V^0 or V^1 that have at least one non-zero coefficient in G_2 , as in (3.36).

Estimation of $T_{\text{SR-SNI}}(n, \epsilon)$. For later reference, we have estimated $T_{\text{SR-SNI}}$ using Algorithm 8 at various pairs of (n, ϵ) . The results are depicted in Figure 3.5, and they fit in the following bound:

$$T_{\text{SR-SNI}}(n, \epsilon) \leq \frac{1}{3}n\epsilon, \quad \text{valid for } (n \geq 3, \epsilon \leq 0.1). \quad (3.37)$$

With $n = 3$, the given bound is only valid for $\epsilon < 0.8$. One can find a better matching bound with $n \geq 4$. However, for our illustration purpose, the given bound suffices.

Remark 3.14. SR-Rot satisfies Definitions 3.9 and 3.10. However, $T_{\text{SR-Rot}}(n, \epsilon)/n$ grows with n for any tested region of ϵ . See Figure 3.5-(Right).

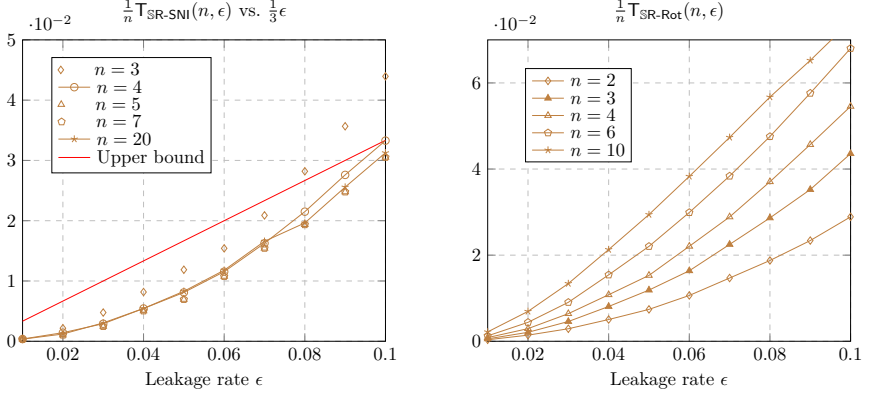


Figure 3.5: (Left) Estimation results for $\frac{1}{n} T_{\text{SR-SNI}}(n, \epsilon)$. The given upper bound holds with acceptable accuracy. Note that $\frac{1}{n} T_{\text{SR-SNI}}(n, \epsilon)$ decreases with n . (Right) The estimated value for $\frac{1}{n} T_{\text{SR-Rot}}(n, \epsilon)$ increases with n . Hence, we cannot devise a similar upper bound.

Devising a union bound. In the following lemma, we use $T_{\text{SR}}(n, \epsilon)$ to compute an equivalent leakage rate ϵ' .

Lemma 3.15. *In the RPM setting, at pair (n, ϵ) , the distribution of $\tilde{V}$ conditioned on*

$$\begin{cases} \text{NL}_{\text{SG}}(\Sigma_{\text{SG}}, \mathbf{V}, V) = \mathbf{0}, \\ \mathbf{V}^\dagger = \mathbf{b}, \end{cases} \quad (3.38)$$

with $\mathbf{T} = \mathbb{E}[|\mathbf{V}^\dagger|]$ is statistically equivalent to the distribution of $\tilde{V}$ conditioned on

$$\text{NL}_{\text{SG}}(\Sigma_{\text{SG}}, \mathbf{V}, V) = \mathbf{0}, \quad (3.39)$$

where the shares in $\mathbf{V}$ leak at rate:

$$p' = p + \frac{\mathbf{T}}{n} - p \frac{\mathbf{T}}{n} \approx p + \frac{\mathbf{T}}{n}. \quad (3.40)$$

Proof. The set of equations defined by NL_{SG} is deterministic, with only the collection of known variables being probabilistic. Our goal is to demonstrate

that the distribution of the known elements in the input $\mathbf{V}$ is identical in both systems.

In the first system, members of $\mathbf{V}$ are known to the adversary from two sources: the usual ϵ leakage from the $\mathbb{S}\mathbb{G}$ side and $\mathbf{V}^\dagger$ resulting from the $\mathbb{S}\mathbb{R}$ side. As the parities corresponding to the $\mathbb{S}\mathbb{R}$ gadget are symmetric for the members of V , each member of $\mathbf{V}$ is in $\mathbf{V}^\dagger$ independently with a probability of $\frac{T}{n}$. Hence, the combined probability of leakage for each member of $\mathbf{V}$ is:

$$\epsilon' = \epsilon + \frac{T}{n} - \epsilon \frac{T}{n}. \quad (3.41)$$

In the second system, members of $\mathbf{V}$ are directly leaking with a probability of ϵ' . Therefore, the inputs to both systems are statistically equivalent, and they should produce statistically equivalent results. $\square$

Remark 3.16. If $\mathbb{S}\mathbb{R}$ is leak-free, due to the maximal independence requirements on the input-output shares $\mathbf{V}^0$ and $\mathbf{V}^1$, there will be no other relation than trivial $\oplus_{i=1}^n V_i^0 = \oplus_{i=1}^n V_i^1$. Therefore, the sets $\mathbf{V}^{0,\dagger}$ and $\mathbf{V}^{1,\dagger}$ will be empty, and consequently, $T(n, \epsilon)$ will be zero, which results in $\epsilon' = \epsilon$.

To proceed our approach to prove (3.13), assume that there exists an EEC for (non-)linear $\mathbb{S}\mathbb{G}_1$ and $\mathbb{S}\mathbb{G}_2$ gadgets. Developing an EEC for a multiplication gadget will be treated in Section 3.4.

The following lemma establishes how to merge information from multiple EECs.

Lemma 3.17. *Learning V through a set of independent EECs with parameters $\{\mathbf{E}_1, \dots, \mathbf{E}_k\}$ is equivalent to learning it via an EEC with parameter $\mathbf{E}' \leq \sum_{i=1}^k \mathbf{E}_i$.*

Proof. The lemma is obtained by applying the union bound on the probabilities. $\square$

Remark 3.18. For any gadget $\mathbb{S}\mathbb{G}$, at any fixed order n , $\mathbf{E}_{\mathbb{S}\mathbb{G}}(n, \epsilon)$ is a monotonically increasing function of ϵ .

We are now able to compute an EEC for the chain $\mathbb{S}\mathbb{G}_1 \rightarrow \mathbb{S}\mathbb{R} \rightarrow \mathbb{S}\mathbb{G}_2$:

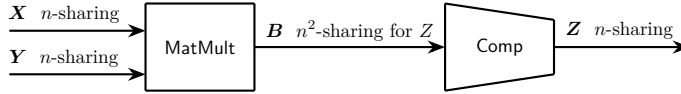
$$\begin{aligned} \mathbf{E}_{\mathbb{S}\mathbb{G}_1 \rightarrow \mathbb{S}\mathbb{R} \rightarrow \mathbb{S}\mathbb{G}_2}(n, \epsilon) &= \Pr(\tilde{V} = v) \\ &= \Pr(\tilde{V} = v \mid e_{\mathbb{S}\mathbb{R}}) \Pr(e_{\mathbb{S}\mathbb{R}}) + \Pr(\tilde{V} = v \mid \overline{e_{\mathbb{S}\mathbb{R}}}) \Pr(\overline{e_{\mathbb{S}\mathbb{R}}}) \\ &\leq \Pr(e_{\mathbb{S}\mathbb{R}}) + \Pr(\tilde{V} = v \mid \overline{e_{\mathbb{S}\mathbb{R}}}) \\ &= \mathbf{E}_{\mathbb{S}\mathbb{R}}(n, \epsilon) + \Pr(\tilde{V} = v \mid \overline{e_{\mathbb{S}\mathbb{R}}}) \\ &\leq \mathbf{E}_{\mathbb{S}\mathbb{R}}(n, \epsilon) + \mathbf{E}_{\mathbb{S}\mathbb{G}_1}(n, \epsilon') + \mathbf{E}_{\mathbb{S}\mathbb{G}_2}(n, \epsilon'), \end{aligned} \quad (3.42)$$

where ϵ' is as given in (3.40), and ϵ should be bounded such that EECs for $\mathbb{S}\mathbb{G}_1$ and $\mathbb{S}\mathbb{G}_2$ remain in their valid region.

3.4 A Multiplication Gadget with RPM Security

A multiplication gadget, denoted as $\mathbb{S}\mathbb{A}\mathbb{N}\mathbb{D}$, takes two n -sharings $\mathbf{X}$ and $\mathbf{Y}$ and computes an n -sharing $\mathbf{Z}$ in a way that preserves the security of the three native values while ensuring that $Z = XY$. This section considers the $\mathbb{S}\mathbb{A}\mathbb{N}\mathbb{D}$ candidate introduced by Battistello et al. in [BCPZ16], labeled as $\mathbb{S}\mathbb{A}\mathbb{N}\mathbb{D}$ -Rec. With certain modifications, we demonstrate the conjectured RPM security of this gadget. Our primary technique is inclusion or exclusion of specific leakage information, which enables linearization of the describing parity relations.

Structure of $\mathbb{S}\mathbb{A}\mathbb{N}\mathbb{D}$ -Rec. $\mathbb{S}\mathbb{A}\mathbb{N}\mathbb{D}$ -Rec consists of two algorithms: **MatMult** and **Comp**. **MatMult** takes two n -sharings $\mathbf{X}$ and $\mathbf{Y}$ as input and generates an n^2 -sharing for $Z = XY$. These n^2 values are organized into an $n \times n$ matrix, denoted as $\mathbf{B}$. **Comp** then operates on $\mathbf{B}$ using fresh randomness to compress it into an n -sharing, whose secret is Z .



MatMult algorithm. **MatMult**, outlined in Algorithm 9, is a recursive process. At the initial invocation, it takes an n_X -sharing $\mathbf{X}$ and an n_Y -sharing $\mathbf{Y}$ with $n_X = n_Y = n$. In the subsequent calls, if n_X and n_Y are both 1, the algorithm sets $\mathbf{B}$ to X_1Y_1 and the computation terminates. Otherwise, the input vectors $\mathbf{X}$ and $\mathbf{Y}$ are divided into left and right subvectors. We denote the left subvectors as $\mathbf{X}_L$ and $\mathbf{Y}_L$ and the right subvectors as $\mathbf{X}_R$ and $\mathbf{Y}_R$ as defined in lines 4 and 5 of the algorithm.

With $n_X = 1$ and $n_Y > 1$ (resp. $n_Y = 1$ and $n_X > 1$), $\mathbf{X}_L$ (resp. $\mathbf{Y}_L$) will be an empty vector. Empty vectors are represented by a $\emptyset$ symbol. For length-one inputs, the refreshing gadget is the identity function. When $\mathbf{X}_L$ is empty, both $\mathbf{B}_{LL}$ and $\mathbf{B}_{LR}$ are empty. Similarly, if $\mathbf{Y}_L$ is empty, then $\mathbf{B}_{LL}$ and $\mathbf{B}_{RL}$ will be empty as well.

Algorithm 9 MatMult

Input X, Y
Output An n^2 -sharing for $Z = XY$ saved in B

```

1: if  $n_X = 1$  and  $n_Y = 1$  then
2:    $B = X_1 Y_1$ 
3: else
4:    $X_L = (X_1, \dots, X_{\lfloor n_X/2 \rfloor})$ ,  $X_R = (X_{\lfloor n_X/2 \rfloor + 1}, \dots, X_{n_X})$ 
5:    $Y_L = (Y_1, \dots, Y_{\lfloor n_Y/2 \rfloor})$ ,  $Y_R = (Y_{\lfloor n_Y/2 \rfloor + 1}, \dots, Y_{n_Y})$ 
6:   if  $X_L \neq \emptyset$  &  $Y_L \neq \emptyset$  then
7:      $B_{LL} = \text{MatMult}(\text{SR}(X_L), \text{SR}(Y_L))$ 
8:   if  $X_L \neq \emptyset$  then
9:      $B_{LR} = \text{MatMult}(\text{SR}(X_L), \text{SR}(Y_R))$ 
10:  if  $Y_L \neq \emptyset$  then
11:     $B_{RL} = \text{MatMult}(\text{SR}(X_R), \text{SR}(Y_L))$ 
12:   $B_{RR} = \text{MatMult}(\text{SR}(X_R), \text{SR}(Y_R))$ 
13:   $B = \begin{bmatrix} B_{LL} & B_{LR} \\ B_{RL} & B_{RR} \end{bmatrix}$ 
14: return  $B$ 

```

Algorithm 10 Comp

Input The n^2 -sharing in $B_{[n \times n]}$
Output An n -sharing Z for z

```

1: for  $i = 1$  to  $n$  do
2:   for  $j = i + 1$  to  $n$  do
3:      $R_{i,j} \xleftarrow{\$} \mathbb{F}_{2^u}$ 
4:      $R_{j,i} = (R_{i,j} \oplus B_{i,j}) \oplus B_{j,i}$ 
5: for  $i = 1$  to  $n$  do
6:    $Z_i = B_{i,i}$ 
7:   for  $j = 1$  to  $n$ ,  $j \neq i$  do
8:      $Z_i = Z_i \oplus R_{i,j}$ 
9: return  $Z = (Z_1, Z_2, \dots, Z_n)$ 

```

Comp algorithm. Comp, described in Algorithm 10 [BCPZ16], squeezes the n^2 -sharing in B into an n -sharing in Z . Comp is linear, and its input and output are sharings for the same secret. It is symmetric, and, except for $\oplus B = \oplus Z$, there is no relation between its input and output shares. These similarities with SR gadgets enable the application of techniques developed in Section 3.3 to estimate the RPM security of $\text{MatMult} \rightarrow \text{Comp}$ composition as:

$$\mathsf{E}_{\text{MatMult} \rightarrow \text{Comp}}(n, \epsilon) \leq \mathsf{E}_{\text{MatMult}}(n, \epsilon') + \mathsf{E}_{\text{Comp}}(n, \epsilon). \quad (3.43)$$

Here, p' exceeds p and depends on the structure of **Comp**.

3.4.1 RPM Security of $\mathbb{S}\text{AND-Rec}$

For **MatMult**, we exhibit an EEC, with parameter $\mathbf{E}^+$, that serves as an upper bound on the adversary's post-leakage knowledge. To illustrate the tightness of this upper bound, we put forward a lower bound modeled via another EEC with parameter $\mathbf{E}^-$. The crucial observation is that the output of the multiplication (Line 2 in Algorithm 9) is not involved in any further computation except storing for later processing in **Comp**. This observation is visually represented in Figure 3.6. Moreover, according to Algorithm 9, the two linear sets of operations before multiplications, which are processing $\mathbf{X}$ and $\mathbf{Y}$, are independent and identical.

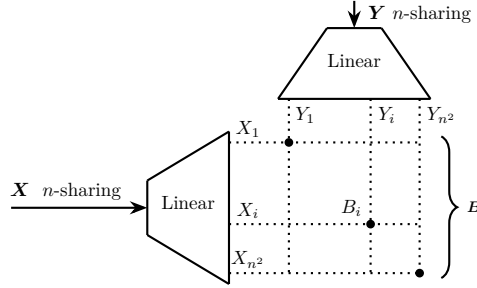


Figure 3.6: Interpretation of **MatMult**. The n^2 X_i (and Y_i) values represent the output of the respective linear operations. Black-filled circles denote multiplication.

Devising $\mathbf{E}^-$. Nonlinear relations among the variables in Σ_{MatMult} stem from operands stored in $\mathbf{B}$. By excluding $\mathbf{B}$ from Σ_{MatMult} , we ensure that all relations are linear. Therefore, if we assume that the adversary does not receive any leakage from $\mathbf{B}$, the parity description for the native X and Y becomes linear. As a result, we can represent the adversary's post-leakage information about X (resp. Y) via an EEC with parameter $\mathbf{E}^-$.

Devising $\mathbf{E}^+$. As depicted in Figure 3.6, for each entry B_i in $\mathbf{B}$, there exist corresponding X_i and Y_i in Σ_{MatMult} such that $B_i - X_i Y_i = 0$. Other than this equation, B_i does not participate in any other relation.

If B_i is not leaked to the adversary, this parity relation does not help the adversary, and we can safely ignore it.⁵

If B_i leaks, it creates a nonlinear relation that cannot be ignored. In this scenario, we let the adversary know both X_i and Y_i . This intervention increases the adversary's success probability in learning information about the natives and eliminates the nonlinear parity equation. This trick allows us to attribute **MatMult** with a linear (and more informative) alternative parity description. The resulting linear system, as per Lemma 3.3, either reveals X (resp. Y) or provides no information. Hence, we can model the post-leakage knowledge with an EEC, denoted by E^+ .

Choosing an appropriate secret. As shown in Figure 3.6, computations on $\mathbf{X}$ and $\mathbf{Y}$ vectors are carried out separately until the final multiplications. Consequently, in both E^+ and E^- , the parity equations for $\mathbf{X}$ and $\mathbf{Y}$ form two separate, identical, and independent subsystems. This implies that the EEC for each of the native inputs will be independent and similar. Later for deriving the upper bound in (3.43), we assume that recovery of each native variable will disclose all the native variables.

Computing E^+ and E^- . In Figure 3.7, we present estimations for $E^+(n, \epsilon)$ and $E^-(n, \epsilon)$ at various orders. These results are based on the instantiation of **SR** with **SR-Simple**.

Based on experiments for $(n \leq 30, \epsilon \leq 0.15)$ using Algorithm 5, we have the following approximations:

$$E^+(n, \epsilon) \lesssim \epsilon^{0.3n}, \quad \text{and} \quad E^-(n, \epsilon) \gtrsim \epsilon^{0.8n}. \quad (3.44)$$

Remark 3.19. From the results in Figure 3.7, it is evident that for leakage rates $\epsilon > 0.22$, as the masking order increases, E^+ also increases. Consequently, **MatMult** with **SR-Simple** may be insecure at these rates.

Remark 3.20. **SR-SNI** is utilized as the **SR** component of **MatMult** in [BCPZ16], resulting in an overall complexity of $\mathcal{O}(n^2 \log(n))$. However, in our numerical experiments, we observed that for any tested pair (n, ϵ) , using **SR-Simple** as the instantiation for **MatMult** led to a lower success probability for the adversary compared to when **MatMult** was instantiated with **SR-SNI**. Additionally, using **SR-Simple**, complexity of the gadget drops to $\mathcal{O}(n^2)$.

⁵A similar proof to that of Lemma 3.13 can demonstrate that this parity relation does not affect the posterior distribution of the native values.

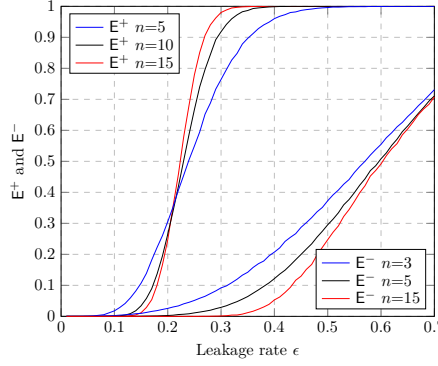


Figure 3.7: E^+ and E^- for MatMult at different orders.

Including leakage of Comp. Both MatMult and Comp are symmetric (Definition 3.9). Comp is linear, and its input and output shares are maximally independent (Definition 3.10), and they represent sharings of the same secret. This allows us to apply a similar procedure used in Section 3.3 to bound $E_{\text{SAND-Rec}}(n, \epsilon)$. The main difference is that Comp's input vector consists of n^2 variables. Consequently, the computation of ϵ' is altered as follows:

$$\epsilon' \approx \epsilon + \frac{T_{\text{Comp}}(n, \epsilon)}{n^2}. \quad (3.45)$$

Here, $T_{\text{Comp}}(n, \epsilon)$ represents the expected number of extra variables that will be revealed to the adversary (for the derivation of the upper bound (3.43)). In the notation of Section 3.3, we have:

$$T_{\text{Comp}}(n, \epsilon) = \mathbb{E}[|\mathbf{B}^\dagger|]. \quad (3.46)$$

Using Algorithm 8, for $n \leq 30$, we obtained the following estimations:

$$\begin{aligned} T_{\text{Comp}}(n, \epsilon) &\leq \epsilon n^2, & \text{valid for } (n \geq 4, \epsilon \leq 0.07), \\ E_{\text{Comp}}(n, \epsilon) &\leq \epsilon^{0.6n}, & \text{valid for } (n \geq 2, \epsilon \leq 0.15). \end{aligned} \quad (3.47)$$

By substituting the estimations in (3.47) and (3.44) into the $E_{\text{SAND-Rec}}(n, \epsilon)$ relation given in (3.43), for $(n \geq 4, \epsilon \leq 0.07)$, we have:

$$\begin{aligned}
E_{\text{SAND-Rec}}(n, \epsilon) &\leq E_{\text{MatMult}}(n, \epsilon') + E_{\text{Comp}}(n, \epsilon) \\
&\leq_{(a)} \left(\epsilon + \frac{\epsilon n^2}{n^2} \right)^{0.3n} + \epsilon^{0.6n} \approx (2\epsilon)^{0.3n}.
\end{aligned} \tag{3.48}$$

This bound demonstrates the RPM security of the proposed SAND-Rec structure in the tested region of the (n, ϵ) pairs. Note that the left side of (a) is analytically proved, but the right side is based on our probability estimation approach and might not hold outside the tested region.

3.5 RPM Secure Masking of S-box in AES

The SAND-Rec and SR-SNI gadgets enable a masked instantiation of the AES's S-box based on the work of Rivain and Prouff [RP10]. We denote this masked circuit as SS-box, and in addition to demonstrating RPM security at $\epsilon \leq 0.03$ rates, we outline an EEC for it.

For an n -sharing V corresponding to a native V in a specified $\mathbb{F}_{2^8}$, SS-box needs to compute an n -sharing of V^{-1} . Rivain and Prouff's approach for computing this inverse is based on the following relation:

$$V^{-1} = V^{254} = [(V^2V) (V^2V)^4]^{16} (V^2V)^4 V^2. \tag{3.49}$$

Since $(.)^{2^i}$ is an additive operation in $\mathbb{F}_{2^8}$, only four non-affine multiplications are required to compute (3.49). These multiplications are performed using SAND gadgets. Thus, SS-box operates as visualized in Figure 3.8. To prepare for RPM security reduction, we have included more refresh gadgets compared to [RP10].

3.5.1 RPM Security of the Proposed SS-box

In the structure of SS-box, SR gadgets enable us to apply the reduction method developed in Section 3.3 to translate the RPM security of the composition into the RPM security of its constituent gadgets. Furthermore, since we have already developed an EEC for the individual gadgets, by applying Lemma 3.17, we can derive an EEC for the entire SS-box.

Unifying the secret. Secrets of the gadgets are different powers of V . To unify the secrets, at the cost of increasing the upper bound, we assume that knowledge of V^j reveals V to the adversary. This assumption also allows us

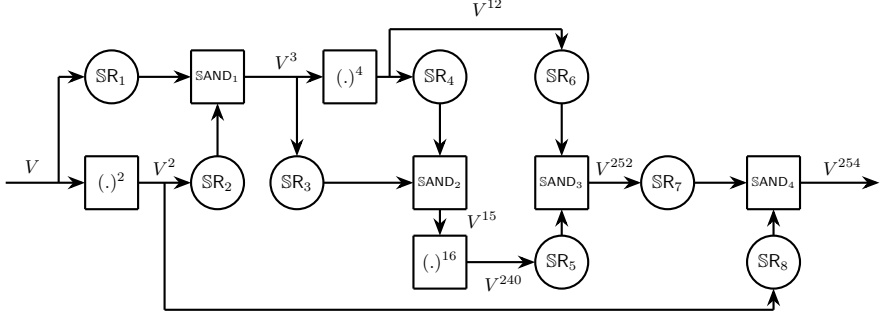


Figure 3.8: Structure of $\mathbb{S}\mathbb{S}$ -box. Powers of V represent the corresponding native values.

to disregard leakage from additive $(\cdot)^{2^i}$ gadgets, as they do not introduce new intermediates and only modify the exponent of V for the next gadget.

The case of multiple SR gadgets. When feeding multiple, say k , SR gadgets with the same n -sharing, we can obtain the corresponding ϵ' as $\epsilon' \approx \epsilon + \frac{kT_{\mathbb{S}\mathbb{R}}}{n}$.

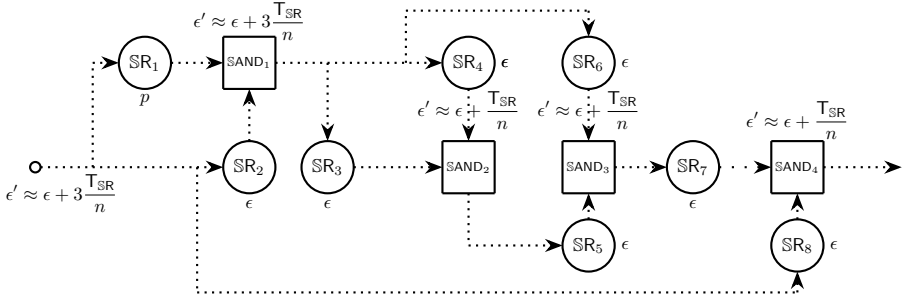


Figure 3.9: Reduction of RPM security of $\mathbb{S}\mathbb{S}$ -box to RPM security of its composing gadgets.

As illustrated in Figure 3.9 and with the application of the reduction method in Section 3.3 and Lemma 3.17, for $(n \geq 4, \epsilon \leq 0.03)$, we can use

(3.12), (3.37), and (3.48) to obtain:

$$\begin{aligned}
E_{\text{SS-box}}(n, \epsilon) &\leq 8E_{\text{SR}}(n, \epsilon) \\
&\quad + 3E_{\text{SAND}}\left(n, \epsilon + \frac{T_{\text{SR}}(n, \epsilon)}{n}\right) + E_{\text{SAND}}\left(n, \epsilon + 3\frac{T_{\text{SR}}(n, \epsilon)}{n}\right) \\
&\leq_{(a)} 8\epsilon^{0.6n} + 3\left(\frac{8}{3}\epsilon\right)^{0.3n} + (4\epsilon)^{0.3n} \approx 4(4\epsilon)^{0.3n}.
\end{aligned} \tag{3.50}$$

Note that at $\epsilon \leq 0.03$, the derived ϵ' and EECs are valid. Also, note that the left side of (a) is based on analytical approach. However, the right side of it is based on probability estimation.

3.6 RPM Secure Masking of AES

Having established an RPM secure SS-box, we now aim to explore RPM security in a more extensive composition: a masked AES cipher. In this cipher, operations other than S-box are affine and, consequently, straightforward to mask (see [RP10] for details). This section reviews the structure of a masked AES, denoted as SAES, and approximates the RPM security for a specific target: the first sub-key of the first round.

Overview of SAES. AES [oST01] is a block cipher supporting various key sizes. In this chapter, we focus on the 128-bit key variant, consisting of 10 rounds. Each round updates the *state*, which is a 16-byte vector initially filled with the plaintext. The state is transformed using a 16-byte round key derived from the input key. Our goal is to approximate the adversary's post-leakage information on K^1 , the first byte of the key for the first round. The operations of a SAES in the initial two rounds are outlined in Figure 3.10.

SShift-Rows and SMix-Columns are affine and operate on the n -shared counterpart of the state. The S-box, aside from computing the inverse, also applies a linear operation at its output, which is considered part of the large affine block in Figure 3.10. The refresh gadget in our experiments is SR-SNI.

Remark 3.21. Since round keys only depend on the input key, they can be computed once. In a masked domain, this translates to saving n -sharings of the sub-keys and refreshing them before each invocation.

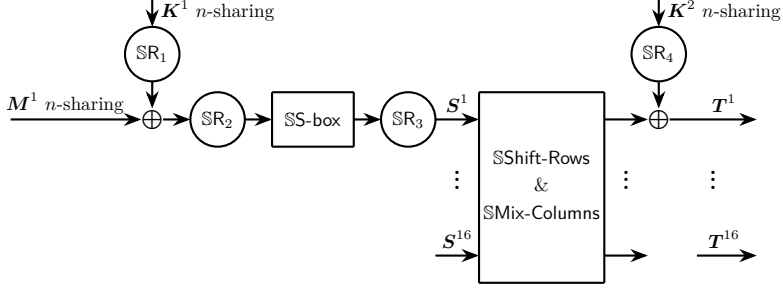


Figure 3.10: Targeting K^1 (secret of K^1) in SAES. M^1 is an n -sharing for the first byte of plaintext, and K^2 is an n -sharing for the first sub-key of the second round. S^1 to S^{16} and T^1 to T^{16} are n -shared bytes of the state.

3.6.1 Estimating the RPM Security

To assess the adversary's post-leakage information about K^1 , that is the secret of K^1 , we employ the reduction framework detailed in Section 3.3, focusing on SR_3 . We calculate EEC for both the left and right components of this SR and combine the results using Lemma 3.17. For the derivation of results in this section, we assume that K^2 and K^1 are independent of each other.

On the left side of SR_3 , secret of the gadgets are either $M^1 \oplus K^1$ or K^1 . However, since M^1 (the secret of M^1) is known to the adversary, these two native values are equivalent. Hence, using the reduction, we have:

$$E_{\text{left}}(n, \epsilon) = E_{SS\text{-box}}\left(n, \epsilon + \frac{T_{SR}}{n}\right) + 2E_{SR}(n, \epsilon). \quad (3.51)$$

On the right side of SR_3 , we have an $\mathbb{F}_{2^8}$ -affine block, for which the approach developed in Section 3.2 is sufficient to sketch an EEC. To account for the leakage of the rounds (3 to 10), at the cost of slightly increasing the gap of the upper bound, we assume that shares of T^1 to T^{16} are known to the adversary. We also include secrets of the other 15 SS -boxes as auxiliary intermediates inside the affine block and assume that each of these secrets is leaking with a probability $E_{\text{left}}(n, \epsilon)$. More specifically, we make the following tweaks to Algorithm 5.

Estimating E_{right} . In the Create-M sub-algorithm, the secrets s^i (secrets of SS -boxes) are placed in the first 16 columns of M_n . The remaining columns are

assigned to intermediates inside the affine block. No adjustments are necessary for the **Extract-Linear** sub-algorithm. However, the $\mathbf{P}_n^r = \text{Sample}(\mathbf{P}_n, \epsilon)$ function should apply different probabilities of leakage to its columns. Columns 2 to 16 will leak with probability $E_{\text{left}}(n, \epsilon)$. Shares in $\mathbf{T}^1$ to $\mathbf{T}^{16}$ will leak with probability one. The remaining columns will leak independently with probability ϵ . The rest of Algorithm 5 is computed as usual.

Estimation results. By running the algorithm with sufficiently many trials, for $\epsilon \leq 0.02$, we obtain the following upper bound:

$$E_{\text{right}}(n, \epsilon) \leq 3\epsilon^{n-1}. \quad (3.52)$$

Adding the leakage of SR_3 to the combination (3.51) and (3.52), for $(n \geq 4, \epsilon \leq 0.02)$, we derive the following estimation for the targeted K^1 , which uses (3.12) and (3.50):

$$\begin{aligned} E(n, \epsilon) &\leq E_{\text{right}}(n, \epsilon) + E_{\text{SR}}(n, \epsilon) + E_{\text{left}}(n, \epsilon) \\ &\leq E_{\text{SS-box}}\left(n, \epsilon + \frac{T_{\text{SR}}}{n}\right) + 3E_{\text{SR}}(n, \epsilon) + E_{\text{left}}(n, \epsilon) \\ &\leq_{(a)} E_{\text{SS-box}}\left(n, \frac{4}{3}\epsilon\right) + 3\epsilon^{0.6n} + 3\epsilon^{n-1} \approx 4(5.3\epsilon)^{0.3n}. \end{aligned} \quad (3.53)$$

The left side of (a) is analytically obtained. However, the right side of it is based on our probability estimation approach and, hence, is only valid in the tested region of (n, ϵ) pairs. The bound based on these estimations is exponentially decaying. Thus, it shows the RPM security of the scheme. Note that the threshold $\epsilon \leq 0.02$ is chosen such that the maximum leakage values for the constituent gadgets are all in their tested region.

3.6.2 Comparison with Previous Results

The expansion-based compiler technique, initially introduced by Ananth et al. [AIS18] and subsequently refined by Belaïd et al. in a series of works [BCP⁺20, BRT21, BRTV21], offers RPM security through an indirect (simulation-based) proof. In contrast, our approach involves direct security evaluation, as discussed in the Introduction. While both research directions provide quantitative claims based on their respective security metrics within the same leakage model, a comparison of their complexity and achieved security level can provide valuable insights. Below, we provide an overview of the expansion technique.

Expansion approach. The expansion approach aims to boost the security of a circuit C through iterative masking. Initially, C undergoes masking, resulting in SC where the gates of C are substituted with gadgets. This process repeats recursively, with SC itself undergoing masking, replacing its gates with gadgets. This iteration persists until the desired security level is achieved. Although this method exponentially inflates the size of the circuit, it can enhance security under specific conditions.

Comparison of security claims. In our approach, the size of the final circuit is $\mathcal{O}(|C|n^2)$, with the security bound being $4(5.3\epsilon)^{0.3n}$ as given in (3.53), specifically for the case of AES. The factor $\mathcal{O}(|C|n^2)$ represents the standard size expansion in TPM. Our work achieves this size expansion by utilizing only SAND-Rec and SR-SNI gadgets, which are of size $\mathcal{O}(n^2)$.

In contrast, the expansion approach for achieving a security bound of $2^{-\kappa}$ increases the size of the circuit to $\mathcal{O}(|C|\kappa^e)$, where e is determined by the structure of the basic gadgets used. The value of e ranges from 7.5 in early works to 3.2 in more recent refinements [BCP⁺20, BRT21]. Therefore, to achieve the same security level as our work, the expansion approach results in a final circuit size of $\mathcal{O}(|C||2 + 0.3n \log_2(5.3\epsilon)|^e)$, exhibiting an asymptotic n^e increase in circuit size, that is higher than our approach.

Furthermore, our approach withstands $\epsilon < 0.02$, while the expansion approach withstands a smaller rate $\epsilon < 0.007$. However, our work's final security bound expression as a function of (n, ϵ) is based on probability estimations of constituent gadgets. Hence, this expression is empirically supported over the tested (n, ϵ) values.

Chapter 4

A Decomposition Approach for Evaluating Security of Masking

This chapter is based on our Asiacrypt 2025 paper [JMB25a]. Compared to the conference version, we harmonized notation with the rest of the thesis and moved the general definitions of the noisy leakage model and the security metrics to the preliminary chapter. We still recall the essentials locally for readability.

Masking is a widely deployed countermeasure against side-channel attacks: it encodes a sensitive value into multiple shares so that each share, viewed in isolation, reveals only partial information. A central question is under which noise conditions masking *provably* improves security, and how the resulting security scales with the number of shares. While several sufficient conditions were known (often restricted to binary settings or high-noise regimes), the borderline and low-noise cases have remained poorly understood.

In this work, we resolve this gap via a decomposition-based analysis. Our approach reduces leakage in extended fields to binary projections, enabling tight bounds on the adversary’s success rate and yielding an optimal reduction from the noisy leakage model to the random probing model—even in regimes where classical reductions become vacuous. As a central theoretical result, we prove a conjecture of Dziembowski et al. [DFS16] for the cyclic case $\mathbb{G} = \mathbb{Z}_N$: masking strictly improves security whenever the leakage is δ -noisy with

$$\delta < 1 - \frac{|\mathbb{H}|}{|\mathbb{G}|},$$

where $\mathbb{H}$ is a largest proper subgroup of $\mathbb{Z}_N$.

We further demonstrate practical implications for leakage certification and for choosing the required masking order under realistic low-noise conditions.

4.1 Introduction

Masking in the noisy leakage model. Side-channel information refers to unintended physical leakages—such as power consumption, electromagnetic emanations, or timing behavior—that can reveal information about intermediate values in cryptographic implementations. To reason about such leakages formally, the *noisy leakage model* was introduced by Prouff and Rivain [PR13] and subsequently studied in a series of works [DDF19, DFS19, DFS15, DFS16, PGMP19]. In this model, each sensitive intermediate random variable $X \in \mathbb{F}_q$ produces a leakage observation $L(X)$, where L is a randomized channel (e.g., noisy Hamming weight). We refer to Chapter 2 for the formal definitions; here we only recall the notation needed in this chapter.

Masking encodes X into n random shares

$$\mathbf{X} := (X_1, \dots, X_n) \in \mathbb{F}_q^n \quad \text{such that} \quad X = X_1 + \dots + X_n.$$

A masked implementation then manipulates shares rather than X itself, so that the adversary observes leakages of the form $L(X_i)$ (and, in full circuits, leakages of additional intermediate values computed from shares). The effectiveness of masking is quantified by how a chosen *security metric* (e.g., the adversary’s success probability) evolves as the sharing order n increases.

Open challenges. If the leakage channel reveals X essentially noiselessly, masking offers no meaningful benefit. Thus, a key question is: *what is the minimal noise level that suffices for masking to improve security, and how does the gain scale with n ?* Existing analyses provide sufficient conditions, but a complete characterization has been missing—especially in *borderline regimes* where noise is low and reductions to auxiliary models become loose or meaningless. Our work targets precisely these regimes.

Practical relevance. Low-noise (high-SNR) leakages arise naturally in practice: (i) embedded processors such as the ARM Cortex-M0 can exhibit highly repeatable power patterns [BS21]; (ii) static power analysis aggregates stable signals over long time windows, yielding cleaner measurements than dynamic power analysis [Moo19]; and (iii) horizontal/averaging attacks combine samples of the same intermediate to reduce measurement noise [BCPZ16]. In these settings, earlier noise thresholds can be too conservative or even inapplicable, motivating tighter and more accurate criteria.

4.1.1 Related Work and Limitations

Single encodings. Duc et al. [DDF19, DFS19] provided an influential reduction showing that the condition $q\delta < 1$ suffices for masking security in their setting. Dziembowski et al. [DFS16] sharpened the analysis in binary extension fields, establishing that $\delta < 1/2$ suffices. However, the regime $\delta \geq 1/2$ —which is critical for low-noise leakages—remained unresolved.

Later works analyzed leakage using mutual information (MI): Ito et al. [IUH22] showed that $\text{MI}(X; L(X)) < 0.72$ suffices under their assumptions, while Béguinot et al. [BCG⁺23] relaxed uniformity requirements by allowing ≤ 1 for some shares. Despite these advances, existing criteria did not yield a tight and general characterization that applies uniformly across leakage models and algebraic structures.

Protected circuits. For full masked circuits, Prouff and Rivain [PR13] initiated a line of work later refined by Masure and Standaert [MS23]. A prominent approach is to reduce the noisy leakage model to the *random probing model* (RPM), as proposed by Duc et al. [DDF19]. This transfer gives circuit-level guarantees, but the resulting probing probability bound scales as $q\delta$, which becomes vacuous for large q or for δ close to 1. Dziembowski et al. [DFS15] introduced the *average probing model*, and Brian et al. [BDF24] subsequently developed a generic compiler from random probing to average probing. However, Béguinot and Masure [BM25] identified a mismatch between the definitions of average probing used in these works, showing that their combination does not establish the claimed general field-size-independent reduction from noisy leakage to random probing.

4.1.2 Our Contributions

This chapter develops a *decomposition-based framework* for analyzing masking security, yields tight reductions, and closes several longstanding gaps:

- **Tight characterization of noise thresholds.** We prove a conjecture of Dziembowski et al. [DFS16]: for an additive group $\mathbb{G}$ with largest proper subgroup $\mathbb{H}$, any δ -noisy leakage satisfying $\delta < 1 - \frac{|\mathbb{H}|}{|\mathbb{G}|}$ guarantees that masking *strictly* improves security. This gives, for the first time, a necessary-and-sufficient threshold governing when masking helps across both binary and non-binary settings.
- **Decomposition technique for binary extensions.** We introduce a bitwise projection method reducing the analysis of leakage in $\mathbb{F}_{2^u}$ to

binary subproblems. This yields tight lower and upper bounds on the success rate of masked encodings and leads to an optimal reduction from noisy leakage to RPM, including the low-noise regimes where classical bounds collapse.

- **Generality and practical relevance.** We extend the framework beyond binary settings, covering prime fields and general additive groups. Practically, we show how the resulting criteria improve leakage certification and guide the selection of masking order to meet concrete security targets; our device-level experiments confirms that the decomposition-based criteria provide accurate guidance under realistic low-noise conditions.

4.2 Security of Mask Encoding

4.2.1 Mask Encoding

A standard method for protecting a sensitive variable X is *masking*, where X is split into a random tuple of n shares $\mathbf{X} = (X_1, \dots, X_n)$ such that

$$X = \sum_{i=1}^n X_i.$$

The adversary then observes the leakage vector

$$\mathbf{L}(\mathbf{X}) = [\mathsf{L}_1(X_1), \dots, \mathsf{L}_n(X_n)].$$

We assume for simplicity that all leakage functions are identical, i.e., $\mathsf{L}_i = \mathsf{L}$, and their randomness are independent. A key question is how the adversary's advantage $\text{Adv}_X[\mathbf{l} \leftarrow \mathbf{L}(\mathbf{X})]$ depends on the number of shares n and the structure of the field $\mathbb{F}_q$.

Intractability of exact metrics. The space of possible leakage vectors $\mathbf{L}(\mathbf{X})$ grows exponentially with n , rendering the exact evaluation of security metrics computationally infeasible. A common workaround—adopted also in this thesis—is to approximate these metrics based on the distribution of $\mathsf{L}(X)$, albeit at the cost of potential inaccuracies.

A loose bound. Duc et al. [DFS19] applied the δ -noisy to ϵ -random probing reduction, in which each share is revealed with probability $\epsilon \leq q\delta$. In the

random probing model, the adversary learns each share independently with probability ϵ , so the probability of learning all n shares is at most $(q\delta)^n$. The corresponding leakage vector in this model is:

$$\phi_\epsilon(\mathbf{X}) = [\phi_\epsilon(X_1), \dots, \phi_\epsilon(X_n)].$$

Generalizing from (2.2), the statistical distance satisfies:

$$\Delta = \text{SD}(X; X \mid \mathbf{L}(\mathbf{X})) \leq \text{SD}(X; X \mid \phi_\epsilon(\mathbf{X})).$$

If at least one share is not revealed, the posterior distribution of X is uniform, yielding zero statistical distance. Only when all shares are leaked does the distance reach its maximum value of $1 - \frac{1}{q}$. Therefore, we obtain the bound:

$$\Delta \leq \left(1 - \frac{1}{q}\right) q^n \delta^n. \quad (4.1)$$

A tighter bound. The bound in (4.1) increases rapidly with q , yet empirical results in [DFS19] showed no such dependency. This led to the conjecture that the q -factor may be a proof artifact. A tighter bound, removing this dependency, was later proven by Masure et al. [MRS22]:

Lemma 4.1 [MRS22], Proposition 4). *Let $X = (X_1, \dots, X_n)$ be a masking of $X \in \mathbb{F}_q$, and suppose $\text{SD}(X; X \mid \mathbf{L}(X)) = \delta$. Then*

$$\Delta = \text{SD}(X; X \mid \mathbf{L}(\mathbf{X})) \leq 2^{n-1} \delta^n.$$

This result confirms an observation of Dziembowski et al. [DFS16]: when $\delta < \frac{1}{2}$, the posterior distribution of X becomes increasingly uniform as n increases. In this setting, the adversary's best guess converges to $\frac{1}{q}$, and applying Lemma 2.6 yields:

$$\text{Adv}_X[\mathbf{l} \leftarrow \mathbf{L}(\mathbf{X})] \leq 2^{n-1} \delta^n.$$

Case of $q = 2$. We prove that for binary fields the given bound is tight.

Lemma 4.2. *In the setting of Lemma 4.1, if the underlying field is $\mathbb{F}_2$, then*

$$\text{Adv}_X[\mathbf{l} \leftarrow \mathbf{L}(\mathbf{X})] = 2^{n-1} \delta^n.$$

Proof. We extend a technique by Wyner [Wyn75], originally developed for wiretap channels.

To recover $X = X_1 \oplus \dots \oplus X_n$ from the leakage vector $\mathbf{L}(\mathbf{X})$, the adversary estimates each X_i individually. Let $\hat{X}_i$ denote the estimate of X_i based on

$\mathcal{L}(X_i)$, and let $\mathbf{e}_i = \Pr(\hat{X}_i \neq X_i)$ be the average error probability, taken over both the uniform choice of $X_i \in \{0, 1\}$ and the leakage randomness:

$$\mathbf{e}_i = \mathbb{E}_{X_i, l \leftarrow \mathcal{L}(X_i)} \left[\Pr(\hat{X}_i \neq X_i \mid l) \right].$$

Assume $\mathbf{e}_i \leq \frac{1}{2}$ (the case $\mathbf{e}_i > \frac{1}{2}$ is similar). The corresponding advantage is then

$$\text{Adv}_{X_i} = (1 - \mathbf{e}_i) - \frac{1}{2} = \frac{1}{2} - \mathbf{e}_i.$$

By Lemma 2.6, we have $\text{Adv}_{X_i} = \delta$, implying $\mathbf{e}_i = \frac{1}{2} - \delta$. Let $\mathbf{e} = \frac{1}{2} - \delta$ for simplicity.

The adversary computes $\hat{X} = \hat{X}_1 \oplus \cdots \oplus \hat{X}_n$. This estimate equals the true value X if an even number of errors occur. Thus, the success probability is:

$$\begin{aligned} \Pr(\hat{X} = X) &= \sum_{j=0}^{\lfloor n/2 \rfloor} \binom{n}{2j} \mathbf{e}^{2j} (1 - \mathbf{e})^{n-2j} \\ &= \frac{1}{2} \left[\sum_{i=0}^n \binom{n}{i} \mathbf{e}^i (1 - \mathbf{e})^{n-i} + \sum_{i=0}^n \binom{n}{i} (-\mathbf{e})^i (1 - \mathbf{e})^{n-i} \right] \\ &=_{(I)} \frac{1}{2} [(\mathbf{e} + 1 - \mathbf{e})^n + (-\mathbf{e} + 1 - \mathbf{e})^n] \\ &= \frac{1}{2} [1^n + (1 - 2\mathbf{e})^n] = \frac{1}{2} + 2^{n-1} \delta^n, \end{aligned}$$

where step (I) uses the binomial expansion:

$$(\pm \mathbf{e} + (1 - \mathbf{e}))^n = \sum_{i=0}^n \binom{n}{i} (\pm \mathbf{e})^i (1 - \mathbf{e})^{n-i}.$$

Subtracting the baseline guessing probability $\frac{1}{2}$, we get:

$$\text{Adv}_X[\mathbf{l} \leftarrow \mathcal{L}(\mathbf{X})] = \Pr(\hat{X} = X) - \frac{1}{2} = 2^{n-1} \delta^n. \quad \square$$

The binary case forms the basis for our reasoning over extended fields. Before proceeding, we highlight an important observation.

Optimality of the reduction at $q = 2$. Lemma 2.7 establishes that the δ -noisy to ϵ -random reduction is tight for $q = 2$, with $\epsilon_{\min} = 2\delta$. Substituting this into (4.1), we obtain:

$$\Delta \leq \text{SD}(X; X \mid \phi_{\epsilon_{\min}}(\mathbf{X})) = \left(1 - \frac{1}{2}\right) \cdot 2^n \delta^n = 2^{n-1} \delta^n.$$

On the other hand, Lemma 4.2 gives $\Delta = 2^{n-1}\delta^n$, implying that

$$\Delta = \text{SD}(X; X \mid \phi_{\epsilon_{\min}}(\mathbf{X})).$$

This equality shows that the reduction is tight not only for a single variable, but also in the context of mask encoding.¹

Furthermore, when the leakage functions differ across shares—each with corresponding parameter δ_i —the bound generalizes to:

$$\Delta = 2^{n-1} \prod_{i=1}^n \delta_i.$$

For $q > 2$, the bound in Lemma 4.1 is loose. We now present a concrete example for $q = 4$, illustrating that the bound in Lemma 4.1 is not tight in general.

Example 4.1. Let $X \in \mathbb{F}_{2^2}$, and define the leakage function as

$$\mathbf{L}(X) = (x_1 \oplus e_1) \parallel (x_0 \oplus e_0),$$

where x_0, x_1 are the bits of X , and e_0, e_1 are independent Bernoulli variables with $\Pr(e_i = 1) = \mathbf{e} < \frac{1}{2}$. A direct computation yields:

$$\delta = \text{SD}(X; X \mid \mathbf{L}(X)) = \left(\frac{1}{2} - \mathbf{e}\right) \left(\frac{3}{2} - \mathbf{e}\right). \quad (4.2)$$

This leakage is effectively the concatenation of two independent binary leakages. From previous results, we know that under such a leakage model, a masked bit is recovered with error probability:

$$\mathbf{e}_n = \frac{1}{2} [1 - (1 - 2\mathbf{e})^n].$$

Using this, we define an equivalent leakage function:

$$\mathbf{L}'(X) = (x_1 \oplus e'_1) \parallel (x_0 \oplus e'_0),$$

where $e'_i \sim \text{Ber}(\mathbf{e}_n)$. Then, by applying the same structure as in (4.2), we get:

$$\begin{aligned} \Delta &= \text{SD}(X; X \mid \mathbf{L}(\mathbf{X})) = \text{SD}(X; X \mid \mathbf{L}'(X)) \\ &= \left(\frac{1}{2} - \mathbf{e}_n\right) \left(\frac{3}{2} - \mathbf{e}_n\right) \\ &= 2^{n-1} \left(\frac{1}{2} - \mathbf{e}\right)^n \left(1 + \frac{1}{2}(1 - 2\mathbf{e})^n\right). \end{aligned}$$

¹That is, for the metrics Δ and Adv_X . The tightness may not extend to other metrics, such as the mutual information between $\mathbf{L}(\mathbf{X})$ and X .

In contrast, Lemma 4.1 gives the bound:

$$\Delta \leq 2^{n-1} \delta^n,$$

which is looser than our exact computation:

$$\begin{aligned} \Delta &= 2^{n-1} \left(\frac{1}{2} - \mathbf{e}\right)^n \left(1 + \frac{1}{2}(1 - 2\mathbf{e})^n\right) \\ &<_{(I)} 2^{n-1} \left(\frac{1}{2} - \mathbf{e}\right)^n \left(1 + \frac{1}{2}(1 - 2\mathbf{e})\right)^n = 2^{n-1} \delta^{n-1}, \end{aligned}$$

where step (I) uses the inequality:

$$\left(1 + \frac{1}{2}t^n\right) < \left(1 + \frac{1}{2}t\right)^n \quad \text{for } 0 < t < 1, n > 1. \quad \square$$

Need for more fine-tuned analysis. Our findings thus far indicate that for $q = 2^u$ with $u > 1$, the standard δ -noisy to ϵ -random probing reduction—such as in the case of $\mathbf{L}(X) = \mathbf{ZV}(X)$ —and indirect metric estimates (as illustrated in Example 4.1) introduce a non-negligible gap. The central contribution of this chapter is to close this gap via a new *decomposition-based* approach.

4.2.2 Decomposition into Binary Subfields

The observation that exact metrics are tractable and the reduction is tight in binary fields motivates a decomposition strategy: we reduce computations in $\mathbb{F}_{2^u}$ to binary relations, where metrics can be efficiently analyzed, and then lift the results back. We begin by outlining the foundational concepts.

Consider two u -bit integers, A and B , and define their bitwise *inner product* as:

$$\langle A, B \rangle = \bigoplus_{i=1}^u (a_i \cdot b_i),$$

where a_i and b_i are the i th bits of A and B , respectively.

Let $X \in \mathbb{F}_{2^u}$ be a random variable with an n -shared masked encoding $\mathbf{X} = (X_1, \dots, X_n)$. For any integer $h \in \{1, \dots, 2^u - 1\}$ —interpreted via its u -bit binary representation—we can project the equality $X = X_1 \oplus \dots \oplus X_n$ into the binary domain as:

$$\langle X, h \rangle = \langle X_1, h \rangle \oplus \dots \oplus \langle X_n, h \rangle.$$

We will later formalize the validity of this mapping in the context of Boolean systems. For now, to illustrate its practical utility, Lemma 4.3 will show that if the adversary fails to recover any of the $2^u - 1$ binary projections $\langle X, h \rangle$ from the leakage $\mathbf{L}(X)$, then they cannot infer X with meaningful advantage.

Lemma 4.3. *Given the leakage $L(X)$ for $X \in \mathbb{F}_{2^u}$, the adversary's advantage in recovering X satisfies:*

$$\frac{1}{2^{u-1}} \max_h \text{Adv}_{\langle X, h \rangle} \leq \text{Adv}_X \leq \frac{1}{2^{u-1}} \sum_{h=1}^{2^u-1} \text{Adv}_{\langle X, h \rangle} < 2 \max_h \text{Adv}_{\langle X, h \rangle}.$$

Here, $\text{Adv}_{\langle X, h \rangle}$ denotes the adversary's advantage in recovering the binary inner product $\langle X, h \rangle$.

Proof. Let us denote $\mu_h = \text{Adv}_{\langle X, h \rangle}$. Given the set $\{\mu_1, \dots, \mu_{2^u-1}\}$, we aim to bound Adv_X from above and below.

Let $\{p_0, p_1, \dots, p_{2^u-1}\}$ denote the posterior distribution of X given the leakage realization l . For any h , the advantage μ_h can be written as:

$$\begin{aligned} \mu_h &= \mathbb{E}_l \left[\max_{\alpha \in \{0,1\}} \Pr(\langle X, h \rangle = \alpha \mid l) \right] - \frac{1}{2} \\ &= \mathbb{E}_l \left[\max_{\alpha \in \{0,1\}} \sum_{i \in [0, 2^u-1], \langle i, h \rangle = \alpha} p_i \right] - \frac{1}{2} \\ &= \mathbb{E}_l \left[\max_{\alpha \in \{0,1\}} \left(\sum_{\langle i, h \rangle = \alpha} p_i - \frac{1}{2} \right) \right] \\ &\stackrel{(I)}{=} \frac{1}{2} \mathbb{E}_l \left[\left| \sum_{\langle i, h \rangle = 0} p_i - \sum_{\langle i, h \rangle = 1} p_i \right| \right] \\ &= \frac{1}{2} \mathbb{E}_l \left[\left| \sum_{i=0}^{2^u-1} p_i (-1)^{\langle i, h \rangle} \right| \right]. \end{aligned}$$

In step (I), we used the fact that $a + b = 1 \Rightarrow |a - b| = 2(\max\{a, b\} - \frac{1}{2})$. Let us now define the following quantity:

$$\theta_h \triangleq \sum_{i=0}^{2^u-1} p_i (-1)^{\langle i, h \rangle},$$

so that $\mathbb{E}_l[|\theta_h|] = 2\mu_h$.

Now, by the inverse Walsh-Hadamard transform, we can express p_i as:

$$p_i = \frac{1}{2^u} \sum_{h=0}^{2^u-1} \theta_h (-1)^{\langle i, h \rangle} = \frac{1}{2^u} \left(\theta_0 + \sum_{h=1}^{2^u-1} \theta_h (-1)^{\langle i, h \rangle} \right),$$

and since $\theta_0 = \sum_i p_i = 1$, we have:

$$p_i - \frac{1}{2^u} = \frac{1}{2^u} \sum_{h=1}^{2^u-1} \theta_h (-1)^{\langle i, h \rangle}. \quad (4.3)$$

For derivation of upper bounds in the lemma, we can write:

$$\begin{aligned} \text{Adv}_X &= \mathbb{E}_l \left[\max_i p_i - \frac{1}{2^u} \right] = \frac{1}{2^u} \mathbb{E}_l \left[\max_i \left[\sum_{h=1}^{2^u-1} \theta_h (-1)^{\langle i, h \rangle} \right] \right] \\ &\leq \frac{1}{2^u} \mathbb{E}_l \left[\max_i \left| \sum_{h=1}^{2^u-1} \theta_h (-1)^{\langle i, h \rangle} \right| \right] \leq \frac{1}{2^u} \mathbb{E}_l \left[\sum_{h=1}^{2^u-1} |\theta_h| \right] \\ &= \frac{1}{2^u} \sum_{h=1}^{2^u-1} 2\mu_h = \frac{1}{2^{u-1}} \sum_{h=1}^{2^u-1} \mu_h < 2 \max_h \mu_h. \end{aligned}$$

To derive the lower bound, let $h^* = \text{argmax}_h \mu_h$, and define the random index $J \in \{0, \dots, 2^u - 1\}$ uniformly sampled from:

$$\{J \in [0, 2^u - 1] \mid (-1)^{\langle J, h^* \rangle} = \text{sign}(\theta_{h^*})\}.$$

We have:

$$\begin{aligned} \text{Adv}_X &= \frac{1}{2^u} \mathbb{E}_l \left[\max_i \sum_{h=1}^{2^u-1} \theta_h (-1)^{\langle i, h \rangle} \right] \\ &\geq \frac{1}{2^u} \mathbb{E}_l \left[\mathbb{E}_J \left[\sum_{h=1}^{2^u-1} \theta_h (-1)^{\langle J, h \rangle} \right] \right] \\ &= \frac{1}{2^u} \mathbb{E}_l \left[\sum_{h=1}^{2^u-1} \theta_h \mathbb{E}_J [(-1)^{\langle J, h \rangle}] \right]. \end{aligned}$$

The inner expectation simplifies because:

- For $h = h^*$, we have $\mathbb{E}_J [(-1)^{\langle J, h^* \rangle}] = \text{sign}(\theta_{h^*})$;
 - For $h \neq h^*$, since $\langle J, h \rangle$ is independent of $\langle J, h^* \rangle$, we get $\mathbb{E}_J [(-1)^{\langle J, h \rangle}] = 0$.
- Therefore:

$$\text{Adv}_X \geq \frac{1}{2^u} \mathbb{E}_l [\theta_{h^*} \cdot \text{sign}(\theta_{h^*})] = \frac{1}{2^u} \mathbb{E}_l [|\theta_{h^*}|] = \frac{1}{2^u} \cdot 2\mu_{h^*} = \frac{\mu_{h^*}}{2^{u-1}}.$$

This completes the proof. $\square$

4.2.3 Tightness of the Decomposition

We continue with the convention from the previous proof and denote $\text{Adv}_{\langle X, h \rangle}$ by μ_h . The decomposition-based approach bounds the adversary's advantage by:

$$\text{Adv}_X \leq \frac{1}{2^{u-1}} \sum_{h \neq 0} \mu_h.$$

Our goal in this subsection is twofold: first, we show that this upper bound is tight (i.e., achievable); second, we demonstrate that for any δ -noisy leakage function, the quantity $\frac{1}{2^{u-1}} \sum_{h \neq 0} \mu_h$ remains strictly below δ .

Achievability. Consider the trivial case where the leakage function fully reveals the variable, i.e., $L(X) = X$. Then, for every nonzero h , the adversary perfectly learns the binary variable $\langle X, h \rangle$, and thus $\mu_h = \frac{1}{2}$.

Substituting into the decomposition bound gives:

$$\text{Adv}_X \leq \frac{1}{2^{u-1}} \cdot \left((2^u - 1) \cdot \frac{1}{2} \right) = 1 - \frac{1}{2^u}.$$

On the other hand, since the adversary recovers X exactly, we have:

$$\text{Adv}_X = \Pr(\hat{X} = X) - \frac{1}{2^u} = 1 - \frac{1}{2^u}.$$

Hence, the bound is tight in this case.

Lemma 4.4. *Let $X \in \mathbb{F}_{2^u}$ be a uniform variable with leakage function $L(X)$ such that $\text{SD}(X; X | L(X)) = \delta$. Then:*

$$\frac{1}{2^{u-1}} \sum_{h=1}^{2^u-1} \mu_h < \delta.$$

Proof. Let $\{p_0, p_1, \dots, p_{2^u-1}\}$ denote the posterior distribution of X given a leakage instance l . Since $\langle X, h \rangle$ is a binary random variable, we can express μ_h as:

$$\mu_h = \frac{1}{2} \mathbb{E}_l \left[\left| \sum_{\langle i, h \rangle=0} p_i - \frac{1}{2} \right| + \left| \sum_{\langle i, h \rangle=1} p_i - \frac{1}{2} \right| \right]. \quad (4.4)$$

Let us define:

$$q_i \triangleq p_i - \frac{1}{2^u}, \quad \text{so that} \quad \sum_i q_i = 0.$$

Then, the definitions of δ and μ_h become:

$$\delta = \frac{1}{2} \mathbb{E}_l \left[\sum_i |q_i| \right], \quad \mu_h = \frac{1}{2} \mathbb{E}_l \left[\left| \sum_{\langle i, h \rangle=0} q_i \right| + \left| \sum_{\langle i, h \rangle=1} q_i \right| \right]. \quad (4.5)$$

While the triangle inequality immediately gives $\mu_h \leq \delta$, we aim to show that the average over all nonzero h is smaller than $\frac{1}{2}\delta$. This requires a refined argument.

Partitioning strategy. We begin by partitioning each $q_\beta > 0$ into non-negative components:

$$q_\beta = \sum_{\alpha=0}^{2^u-1} a_{\beta,\alpha}, \quad \text{where } a_{\beta,\alpha} \geq 0.$$

These components are then redistributed to offset the negative entries, by defining:

$$q_\alpha = - \sum_{\beta} a_{\beta,\alpha}, \quad \text{for all } q_\alpha \leq 0.$$

We illustrate this partitioning strategy with a simple example. Let $u = 3$, and suppose that q_0, q_1, q_2 are positive while q_3 through q_7 are negative. The partitioning can be visualized as the following table:

	q_3	q_4	q_5	q_6	q_7
q_0	$a_{0,3}$	$a_{0,4}$	$a_{0,5}$	$a_{0,6}$	$a_{0,7}$
q_1	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$	$a_{1,6}$	$a_{1,7}$
q_2	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$	$a_{2,6}$	$a_{2,7}$

For brevity, we omit the details of how this partitioning is constructed.

Now consider the effect on absolute value terms such as $|q_\beta + q_\alpha + c|$, where $q_\beta > 0$, $q_\alpha \leq 0$, and c is the sum of the remaining terms. We can show that $|q_\beta + q_\alpha| \leq |q_\beta| + |q_\alpha| - 2a_{\beta,\alpha}$, and we derive:

$$|q_\beta + q_\alpha + c| \leq |q_\beta + q_\alpha| + |c| \leq |q_\beta| + |q_\alpha| - 2a_{\beta,\alpha} + |c|.$$

Loss factor from pairwise cancellation. Applying this to all sums in μ_h , we observe that each pair ($q_\beta > 0, q_\alpha < 0$) appears in the same half of the

partition (either 0 or 1 side) for half of the nonzero h 's. Hence, each such pair contributes a “loss” of at least:

$$\frac{2^u - 1}{2} \cdot 2a_{\beta,\alpha}.$$

Summing over all such pairs yields a total loss of:

$$(2^u - 1) \sum_{\beta,\alpha} a_{\beta,\alpha} = (2^u - 1) \sum_{\alpha, q_\alpha < 0} (-q_\alpha),$$

where we used $\sum_{\beta} a_{\beta,\alpha} = -q_\alpha$. Since $\sum_i q_i = 0$, the total positive mass equals the total negative mass:

$$\sum_{\alpha, q_\alpha < 0} (-q_\alpha) = \sum_{\beta, q_\beta > 0} q_\beta = \frac{1}{2} \sum_i |q_i|.$$

Final bound. Combining everything:

$$\begin{aligned} \sum_{h=1}^{2^u-1} \mu_h &\leq \frac{2^u - 1}{2} \mathbb{E}_l \left[\sum_i |q_i| \right] - \frac{2^u - 1}{2} \cdot \frac{1}{2} \mathbb{E}_l \left[\sum_i |q_i| \right] \\ &= \frac{2^u - 1}{4} \mathbb{E}_l \left[\sum_i |q_i| \right] = \frac{2^u - 1}{2} \delta. \end{aligned}$$

Dividing both sides by 2^{u-1} , we get:

$$\frac{1}{2^{u-1}} \sum_{h=1}^{2^u-1} \mu_h \leq \frac{2^u - 1}{2^u} \delta < \delta,$$

which concludes the proof. $\square$

4.2.4 Application of Decomposition to Mask Encodings

Lemma 4.3 expresses the side-channel security of a variable $X \in \mathbb{F}_{2^u}$ in terms of the security of its binary projections $\langle X, h \rangle$. This decomposition reduces the complex task of estimating $\text{Adv}_X[l \leftarrow \mathbf{L}(X)]$ to the simpler computation of the binary advantages $\mu_h = \text{Adv}_{\langle X, h \rangle}[l \leftarrow \mathbf{L}(X)]$. The strength of this approach becomes especially apparent when applied to masked encodings.

Consider the setting of masked encoding, where a secret X is shared as $\mathbf{X} = (X_1, \dots, X_n)$, and the adversary observes $\mathbf{L}(\mathbf{X}) = (\mathbf{L}(X_1), \dots, \mathbf{L}(X_n))$.

For each nonzero $h \in \{1, \dots, 2^u - 1\}$, the inner product satisfies the binary relation:

$$\langle X, h \rangle = \langle X_1, h \rangle \oplus \dots \oplus \langle X_n, h \rangle.$$

That is, $\langle X, h \rangle$ is itself masked by the shares $\langle X_i, h \rangle$, making it a binary secret with a standard masked encoding.

Applying Lemma 4.2, the advantage of the adversary in recovering $\langle X, h \rangle$ from the masked leakage is given by:

$$\text{Adv}_{\langle X, h \rangle}[l \leftarrow \mathbf{L}(\mathbf{X})] = 2^{n-1} (\text{Adv}_{\langle X, h \rangle}[l \leftarrow \mathbf{L}(X)])^n = 2^{n-1} (\mu_h)^n,$$

where $\mu_h = \text{Adv}_{\langle X, h \rangle}[l \leftarrow \mathbf{L}(X)]$ is the advantage from a single share.

Combining this with the upper bound from Lemma 4.3, we obtain:

$$\text{Adv}_X[l \leftarrow \mathbf{L}(\mathbf{X})] \leq \frac{1}{2^{u-1}} \sum_{h=1}^{2^u-1} 2^{n-1} \mu_h^n = \frac{1}{2^u} \sum_{h=1}^{2^u-1} (2\mu_h)^n <_{(I)} \Delta \leq_{(II)} 2^{n-1} \delta^n,$$

where

- $\Delta = \text{SD}(X; X | \mathbf{L}(\mathbf{X}))$;
- (I) follows from Lemma 4.4, applied to the masked case;
- (II) follows from Lemma 4.1.

For clarity, we restate the result as a theorem.

Theorem 4.5. *Let μ_h denote the adversary's advantage in predicting $\langle X, h \rangle$ from the leakage $\mathbf{L}(X)$, for $h \in \{1, \dots, 2^u - 1\}$. Then, in a masked encoding with share leakage $\mathbf{L}(\mathbf{X})$, the adversary's advantage satisfies:*

$$\frac{1}{2^u} \max_h (2\mu_h)^n \leq \text{Adv}_X[l \leftarrow \mathbf{L}(\mathbf{X})] \leq \frac{1}{2^u} \sum_{h=1}^{2^u-1} (2\mu_h)^n < \max_h (2\mu_h)^n.$$

Interpretation in terms of mutual information. Theorem 4.5 shows that if $2\mu_h < 1$ for all h , then the adversary's advantage Adv_X decays exponentially with the number of shares n . In this subsection, we connect the condition $\mu_h < \frac{1}{2}$ to the mutual information between the binary projection $\langle X, h \rangle$ and the leakage $\mathbf{L}(X)$. Specifically, we show that:

$$\text{MI}(\langle X, h \rangle; \mathbf{L}(X)) < 1 \implies \mu_h < \frac{1}{2}.$$

By definition of mutual information for a binary variable [CT06], we have:

$$\text{MI}(\langle X, h \rangle; \mathbf{L}(X)) = 1 - \mathbf{H}\left(\frac{1}{2} \pm \mu_h\right), \quad (4.6)$$

where $\mathbf{H}(\cdot)$ denotes the binary entropy function.²

The mutual information reaches its maximum value of 1 only when $\mathbf{H}(\frac{1}{2} \pm \mu_h) = 0$, i.e., when $\mu_h = \frac{1}{2}$. This implies that the leakage $\mathbf{L}(X)$ fully determines $\langle X, h \rangle$. Therefore, whenever

$$\text{MI}(\langle X, h \rangle; \mathbf{L}(X)) < 1,$$

the corresponding advantage μ_h must satisfy $\mu_h < \frac{1}{2}$, ensuring that masking offers meaningful security for the binary component $\langle X, h \rangle$.

Revisiting previous examples.

Example 4.2. We revisit Example 4.1 to highlight the strength of the decomposition approach. For $X \in \mathbb{F}_{2^2}$ with leakage $\mathbf{L}(X) = (x_1 \oplus e_1) \parallel (x_0 \oplus e_0)$, where $\Pr(e_0 = 1) = \Pr(e_1 = 1) = \mathbf{e}$, we had:

$$\delta = \text{SD}(X; X \mid \mathbf{L}(X)) = \left(\frac{1}{2} - \mathbf{e}\right) \left(\frac{3}{2} - \mathbf{e}\right).$$

Under mask encoding, we previously derived:

$$\Delta = \text{SD}(X; X \mid \mathbf{L}(\mathbf{X})) = 2^{n-1} \left(\frac{1}{2} - \mathbf{e}\right)^n \left(1 + \frac{1}{2}(1 - 2\mathbf{e})^n\right).$$

For this leakage, Example 2.1 (with $u = 1$) yields:

$$\mu_1 = \mu_2 = \frac{1}{2} - \mathbf{e},$$

corresponding to the adversary's advantage in predicting x_0 and x_1 , respectively. Similarly, for μ_3 , which corresponds to predicting $x_0 \oplus x_1$, we compute:

$$\mu_3 = \frac{1}{2} - 2\mathbf{e}(1 - \mathbf{e}).$$

By Lemma 4.3, the decomposition-based bound for unmasked X becomes:

$$\text{Adv}_X[l \leftarrow \mathbf{L}(X)] \leq \frac{1}{2}(\mu_1 + \mu_2 + \mu_3) = \left(\frac{1}{2} - \mathbf{e}\right) \left(\frac{3}{2} - \mathbf{e}\right) = \delta.$$

²The mutual information of a Binary Symmetric Channel (BSC) between A and B is

$$\text{MI}(A; B) = \mathbf{H}(A) - \mathbf{H}(A \mid B) = 1 - \mathbf{H}(P_e),$$

where P_e is the probability of incorrectly estimating A given B . In our setting, $\mu_h = |P_e - \frac{1}{2}|$, so $P_e = \frac{1}{2} \pm \mu_h$.

For masked encoding, Theorem 4.5 gives:

$$\text{Adv}_X[\mathbf{l} \leftarrow \mathbf{L}(\mathbf{X})] \leq 2^{n-2} (\mu_1^n + \mu_2^n + \mu_3^n) = \Delta.$$

This confirms the tightness of our bound. For instance, setting $\mathbf{e} = 0.1$, we get $\delta = 0.56$. Since $\delta > \frac{1}{2}$, Lemma 4.1 cannot guarantee masking security. Meanwhile, mutual information each share is $\text{MI}(X_i; \mathbf{L}(X_i)) = 1.06$, which exceeds the 0.72 threshold required by Ito et al. [IUH22] for masking to be secure. Thus, neither criterion provides a conclusive answer—while our method confirms that masking is indeed secure in this case. $\square$

Example 4.3. Recall the leakage function $\text{ZV}(X)$, previously introduced as:

$$\text{ZV}(X) = \begin{cases} \nu_a & \text{if } X = 0, \\ \nu_b \neq \nu_a & \text{otherwise.} \end{cases}$$

For this model, (2.1) yields $\epsilon_{\min} = 1$, rendering the δ -noisy to ϵ -random probing reduction inapplicable for analyzing masking security.

In contrast, applying our decomposition approach, we compute each $\mu_h = \frac{1}{2^u}$ using relation (4.4). Then, by Theorem 4.5, the adversary's advantage under masking satisfies:

$$\text{Adv}_X[\mathbf{l} \leftarrow \mathbf{L}(\mathbf{X})] \leq 2^{n-u} \sum_{h=1}^{2^u-1} \mu_h^n = (2^u - 1) 2^{n-u-nu},$$

which decreases exponentially with n , provided $u > 1$. Thus, even though the standard reduction fails, our approach confirms that masking remains effective in this setting. $\square$

Application to leakage certification. Leakage certification laboratories evaluate cryptographic implementations on physical devices to assess their resistance against side-channel attacks (see [DSV14, ABB⁺20]). These evaluations often rely on estimating metrics such as $\text{MI}(X; \mathbf{L}(X))$ and $\text{SD}(X; X | \mathbf{L}(X))$. Estimating these quantities requires knowledge of the joint distribution $(X, \mathbf{L}(X))$, which can be derived either through parametric models (e.g., Gaussian with estimated parameters) or non-parametric methods (e.g., histogram-based) [BGP⁺11, GBTP08].

Our work introduces an additional criterion for leakage assessment. Specifically, for a u -bit variable X , we propose verifying the condition:

$$\text{MI}(\langle X, h \rangle; \mathbf{L}(X)) < 1 \quad \text{for all } h \in [1, 2^u - 1].$$

Masking provides meaningful side-channel protection if and only if this condition holds for every nontrivial bitwise projection $\langle X, h \rangle$.

Using Equation (4.6), we can express the bounds from Theorem 4.5 in terms of mutual information:

$$\frac{1}{2^u} \left(2 \max_h \left| \frac{1}{2} - H^{-1}(I_h) \right| \right)^n \leq \text{Adv}_X[l \leftarrow L(X)] < \left(2 \max_h \left| \frac{1}{2} - H^{-1}(I_h) \right| \right)^n,$$

where $I_h = 1 - \text{MI}(\langle X, h \rangle; L(X))$, and H^{-1} is the inverse of the binary entropy function.

4.2.5 Experimental Results and Determining Masking Order

We perform our experiments on a ChipWhisperer CW308 board hosting an STM32F303 UFO processor,³ running an unprotected 8-bit software implementation of AES (included in the ChipWhisperer package). Power traces are captured using a ChipWhisperer-Lite, synchronized at a sampling rate four times the target’s clock frequency.

The targeted intermediate value is the output of the first S-box in the first round:

$$X = \text{S-box}(P[0] \oplus K[0]).$$

A correlation power analysis (CPA) using only 20 traces successfully recovers the secret byte, indicating a low noise level and motivating the need to assess the feasibility and necessary order of masking.

To this end, we estimate the bias parameters μ_h through profiling. Specifically, we train a deep neural network to predict X from power traces. The network processes one-dimensional input traces and has the following architecture:

- A 1D convolutional layer with 32 filters of size 3, followed by ReLU activation.
- A max-pooling layer with pool size 2.
- A second 1D convolutional layer with 64 filters of size 3, followed by ReLU activation.
- Another max-pooling layer with pool size 2.
- A flattening layer, followed by a dense layer with 128 ReLU-activated neurons.

³<https://github.com/newaetech/chipwhisperer>

- A final dense output layer with 256 neurons (corresponding to all possible S-box output values), followed by softmax activation.

We trained this model on 10,000 labeled traces with ground truth $X = \text{S-box}(P[0] \oplus K[0])$. In the attack phase, the model's output corresponds to the estimated posterior distribution of X given the leakage in a trace.

Using this posterior, we estimate the binary error probability P_e for computing the bitwise projection $\langle X, h \rangle = \bigoplus_{i=0}^7 x_i h_i$ for each $h \in \{1, \dots, 255\}$. The neural network demonstrates high accuracy in estimating these projections.

Estimating P_e . For a given attack trace, let the predicted value be $\hat{X}$ and the true value be X . For each h , we compute the frequency with which $\langle \hat{X}, h \rangle = \langle X, h \rangle$ happens. The error is defined as:

$$P_e = 1 - \Pr[\langle \hat{X}, h \rangle = \langle X, h \rangle].$$

Figure 4.1 shows the estimated P_e for all values of h , based on 1,000 attack traces. The minimum error occurs at $h = 255$, which corresponds to the parity of Hamming weight leakage computed by $\text{mod}(\text{HW}(X), 2)$. The corresponding (maximum) bias is:

$$\mu_h = (1 - P_e) - \frac{1}{2} = 0.32.$$

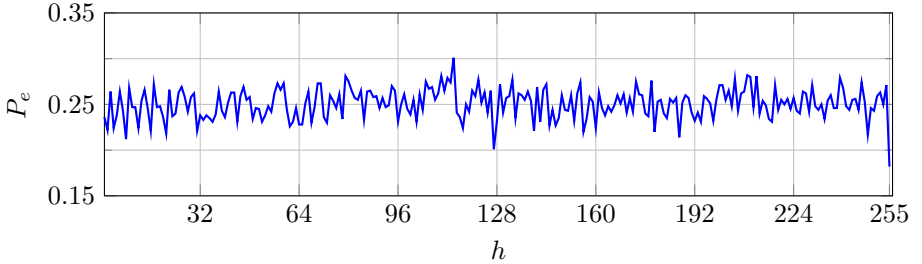


Figure 4.1: Binary error probability P_e in estimating $\langle X, h \rangle$.

Substituting this value into the bound from Theorem 4.5, we obtain the following security bounds for masking in this setting:

$$\frac{1}{256} \cdot (0.64)^n \leq \text{Adv}_X[\mathbf{l} \leftarrow \mathbf{L}(\mathbf{X})] < (0.64)^n.$$

These bounds illustrate that while the adversary retains a noticeable advantage for small n , increasing the masking order rapidly improves security.

4.3 Masking in Odd Prime Fields

Grassi et al. [GMM⁺24], building on results by Dziembowski et al. [DFS16], showed that masking becomes ineffective in the presence of highly informative leakage functions—such as $L(X) = \text{HW}(X)$ —unless the masking is performed over an odd prime field. This insight has sparked further investigation into the security properties of masking in prime fields [FMM⁺24].

In this section, we contribute to this line of work with the following findings:

- In Subsection 4.3.1, we observe that the adversary’s advantage Adv_X decays more rapidly in prime fields.
- In Subsection 4.3.2, we establish a general condition under which masking guarantees a strict reduction in Adv_X .
- In Subsection 4.3.3, we show that in prime fields, Adv_X decreases exponentially with n as long as $\min_h \text{MI}(\langle X, h \rangle; L(X)) < 1$. This condition implies that there exists at least one projection not fully leaked to the adversary—a substantially more relaxed requirement compared to the case of binary extension fields.

4.3.1 Adv_X Decays Faster in Prime Fields

We define a leakage class as *symmetric* if, for a uniform $X \in \mathbb{F}_q$, observing an instance l of the leakage transforms the posterior distribution $X \mid l$ into $(p_{e_0}, p_{e_1}, \dots, p_{e_{q-1}})$, where $\sum_{i=0}^{q-1} p_{e_i} = 1$ and p_{e_i} denotes the probability mass on the value $X + i$.⁴ In this notation, p_{e_0} represents the probability of a correct estimation, and the adversary’s advantage is:

$$\text{Adv}_X = p_{e_0} - \frac{1}{q}.$$

Consider a masked encoding with $n = 2$, where X_1 and X_2 are the shares of X such that $X = X_1 + X_2$. Upon observing the leakages, the adversary forms estimates $\hat{X}_1$ and $\hat{X}_2$, and reconstructs X via $\hat{X}_1 + \hat{X}_2$. The reconstruction is correct if either both estimates are correct or the errors in $\hat{X}_1$ and $\hat{X}_2$ cancel out (i.e., are i and $q - i$). Hence, the updated success probability becomes:

$$p'_{e_0} = (p_{e_0})^2 + \sum_{i=1}^{q-1} p_{e_i} p_{e_{q-i}}. \quad (4.7)$$

⁴We refer to this as a symmetric leakage class because it generalizes the binary symmetric channel.

Lemma 4.6. *For the defined symmetric leakage class, when the field order q is prime, Adv_X decays faster with increasing n .*

Proof. In a prime field, the elements i and $q-i$ are distinct for all $1 \leq i \leq \frac{q-1}{2}$, so we can rewrite Equation (4.7) as:

$$p'_{e_0} = (p_{e_0})^2 + 2 \sum_{i=1}^{\frac{q-1}{2}} p_{e_i} p_{e_{q-i}}.$$

In contrast, when $q = 2^u$, we have $q-i = i$, so the equation simplifies to:

$$p'_{e_0} = (p_{e_0})^2 + \sum_{i=1}^{q-1} p_{e_i}^2.$$

By applying the inequality $2ab \leq a^2 + b^2$, we obtain:

$$(p_{e_0})^2 + 2 \sum_{i=1}^{\frac{q-1}{2}} p_{e_i} p_{e_{q-i}} \leq (p_{e_0})^2 + \sum_{i=1}^{q-1} p_{e_i}^2.$$

This shows that p'_{e_0} , and hence the adversary's success probability and advantage, are lower in prime fields than in binary extension fields. $\square$

4.3.2 When Does Adv_X Decrease with Masking?

For a masked encoding of the secret X , we show that if there is no *hole* in the posterior distribution of the shares after observing the leakage vector, then Adv_X strictly decreases.

Definitions. For later reference, we define a leakage instance l as *dummy* if it induces no change in the distribution of $X \mid l$.⁵ Under dummy leakage, the peak point of the posterior distribution is $\frac{1}{q}$. A *hole* in a distribution is an element of its domain with zero probability mass. The *support* of a random variable refers to the number of domain elements with non-zero probability mass, and we denote the support size of X by $|X|$.

⁵We refer to it as dummy because any random variable that is independent of X has a similar effect.

Problem statement. Let X_1 and X_2 be shares of X in $\mathbb{F}_q$, and suppose the adversary observes leakage instances $l_1 \leftarrow \mathbf{L}(X_1)$ and $l_2 \leftarrow \mathbf{L}(X_2)$. The resulting posterior distributions are denoted by $\mathcal{P}^1 = (p_0^1, \dots, p_{q-1}^1)$ and $\mathcal{P}^2 = (p_0^2, \dots, p_{q-1}^2)$, respectively. Let $p_{i^*}^1$ and $p_{j^*}^2$ denote the peak probabilities in these distributions.

To estimate the value of X , a maximum a posteriori (MAP) adversary computes the distribution of the sum $(X_1 \mid l_1) + (X_2 \mid l_2)$, denoted by $\mathcal{P} = (p_0, \dots, p_{q-1})$. The adversary outputs the index of the peak point of $\mathcal{P}$ as $\hat{X}$, and the corresponding advantage is denoted $\text{Adv}_X[l_1, l_2 \leftarrow \mathbf{L}(X_1), \mathbf{L}(X_2)]$. Our goal is to identify conditions ensuring:

$$\text{Adv}_X[l_1, l_2 \leftarrow \mathbf{L}(X_1), \mathbf{L}(X_2)] < \text{Adv}_{X_i}[l_i \leftarrow \mathbf{L}(X_i)],$$

which implies that masking strictly improves the side-channel security of X .

Lemma 4.7. *If, for at least one non-dummy leakage instance, the posterior distributions $\mathcal{P}^1$ and $\mathcal{P}^2$ have no holes—that is, $\min \mathcal{P}^1 > 0$ and $\min \mathcal{P}^2 > 0$ —then the adversary’s advantage strictly decreases.*

Proof. For $0 \leq i < q$, define $\zeta_i = \frac{p_i^1}{p_{i^*}^1}$ and $\xi_i = \frac{p_i^2}{p_{j^*}^2}$. Since $p_{i^*}^1$ and $p_{j^*}^2$ are the maximum values, we have $\zeta_i \leq 1$ and $\xi_i \leq 1$. From the normalization $\sum_i p_i^1 = 1$, we obtain:

$$p_{i^*}^1 = \frac{1}{\sum_i \zeta_i}, \quad p_{j^*}^2 = \frac{1}{\sum_i \xi_i}.$$

Let k^* be the index of the peak of $\mathcal{P}$. We show that $p_{k^*} \leq \min\{p_{i^*}^1, p_{j^*}^2\}$. Without loss of generality, assume $\min = p_{i^*}^1$. Then:

$$p_{k^*} = \sum_i p_{k^*-i}^1 p_i^2 = p_{i^*}^1 p_{j^*}^2 \sum_i \zeta_{k^*-i} \xi_i \leq p_{i^*}^1,$$

because $p_{i^*}^1 > 0$ and each $\zeta_{k^*-i} \leq 1$. The inequality holds since:

$$\sum_i \zeta_{k^*-i} \xi_i \leq \sum_i \xi_i.$$

Equality would require:

$$\forall i, \quad \zeta_{k^*-i} = 1, \tag{4.8}$$

which implies $p_i^1 = p_{i^*}^1$ for all i , hence a uniform distribution: $p_{i^*}^1 = \frac{1}{q}$. This contradicts the assumption that the leakage is non-dummy. Thus, the strict inequality $p_{k^*} < \min\{p_{i^*}^1, p_{j^*}^2\}$ must hold for at least one instance.

Therefore, taking expectations over all leakage instances, we obtain:

$$\mathbb{E}_l[p_{k^*}] < \mathbb{E}_l[p_{i^*}^1] = \mathbb{E}_l[p_{j^*}^2] \quad \longrightarrow \quad \text{Adv}_X[l \leftarrow \mathbf{L}(\mathbf{X})] < \text{Adv}_X[l \leftarrow \mathbf{L}(X)]. \quad \square$$

Reaching a hole-free posterior distribution. When the leakage function is less noisy—e.g., $L(X) = HW(X)$ —the posterior distribution of $X_i \mid L(X_i)$ typically contains holes, preventing the application of Lemma 4.7. However, increasing the number of shares can eliminate such holes, as we now explain.

Although Lemma 4.7 applies directly to the case $n = 2$, the result generalizes naturally to higher n . Let $X_1, \dots, X_{2n}$ be shares of X , with corresponding leakages $L(X_1), \dots, L(X_{2n})$. The task of estimating X from its leakage vector can be decomposed into two stages: first, estimating $S_1 = X_1 + \dots + X_n$ and $S_2 = X_{n+1} + \dots + X_{2n}$, and then estimating $X = S_1 + S_2$.

If, beyond a certain threshold n_0 , the distribution of $S_1 \mid L(X_1), \dots, L(X_n)$ has no holes, then Lemma 4.7 implies that Adv_X will decrease with n , indicating security improvement through masking.

Because the shares and leakages are independent, the distribution of $S_1 \mid (L(X_1), \dots, L(X_n))$ equals the convolution $X_1 \mid l_1 + \dots + X_n \mid l_n$. At any leakage instance, each $X_i \mid l_i$ is a distribution, and our goal is to determine when the support of the sum reaches the full domain $\mathbb{F}_q$, which ensures a hole-free distribution.

Lemma 4.8 (Generalized Cauchy-Davenport Theorem). *Let $Z_1, \dots, Z_t$ be independent random variables with supports $|Z_1|, \dots, |Z_t|$ over a prime field $\mathbb{F}_q$. Then:*

$$|Z_1 + \dots + Z_t| \geq \min \{|Z_1| + \dots + |Z_t| - t + 1, q\}.$$

Proof. This is a direct generalization of the classical Cauchy-Davenport theorem, which states the result for $t = 2$. The case $t = 2$ was previously used in the side-channel literature by Dziembowski et al. [DFS16]. $\square$

Using this lemma, we conclude that if each $X_i \mid l_i$ has support size greater than 1 (i.e., $|X_i \mid l_i| > 1$), then there exists some $n \geq n_0$ for which the convolution $X_1 \mid l_1 + \dots + X_n \mid l_n$ is hole-free. Consequently, Adv_X decreases with increasing n .

4.3.3 Advantage in Terms of $\langle X, h \rangle$ Projections

In the case of fields with characteristic two ($q = 2^u$), we previously reduced the security of masking to the security of binary projections $\langle X, h \rangle$. In this subsection, we explore the implications of these projections for the security of masking in prime fields. To this end, we rely on the following result derived from a bound by Dziembowski et al. [DFS16].

Lemma 4.9 ([DFS16], Theorem 1). *Let $\mathbb{F}_q$ be a prime field, and suppose the leakage is δ -noisy. That is, $\text{SD}(X; X \mid \mathbf{L}(X)) = \delta < 1 - \frac{1}{q}$. Then:*

$$\Delta \leq \frac{1}{q} \cdot 2^{-\frac{n\sigma^4}{2}},$$

where $\Delta = \text{SD}(X; X \mid \mathbf{L}(\mathbf{X}))$ and $\sigma = (1 - \frac{1}{q}) - \delta$.

A straightforward corollary (using Lemma 2.6) yields:

$$\text{Adv}_X[\mathbf{l} \leftarrow \mathbf{L}(\mathbf{X})] \leq \frac{1}{q} \cdot 2^{-\frac{n\sigma^4}{2}}. \quad (4.9)$$

Now, suppose there exists h such that the leakage $\mathbf{L}(X)$ does not fully determine $\langle X, h \rangle$. That is, the bias μ_h of this bit is bounded by $0 \leq \mu_h < \frac{1}{2}$. In the borderline case where $\mu_h \rightarrow \frac{1}{2}$, we can approximate the statistical distance as follows:⁶

$$\text{SD}(X; X \mid \mathbf{L}(X)) \lesssim \frac{1}{2} + \mu_h - \frac{1}{q}.$$

Substituting into the bound (4.9), we obtain:

$$\text{Adv}_X[\mathbf{l} \leftarrow \mathbf{L}(\mathbf{X})] \leq \frac{1}{q} \cdot 2^{-\frac{n(1/2 - \mu_h)^4}{2}}, \quad (4.10)$$

which indicates that the adversary's advantage decreases exponentially in n as long as $\mu_h < \frac{1}{2}$.

We emphasize that the restriction to a single projection is made solely for illustrative purposes; the bound provided by Lemma 4.9 remains tighter and more general.

4.4 General Requirement for Secure Masking

For a finite abelian group $\mathbb{G}$, Dziembowski et al. [DFS16] showed that there exists a leakage with noise parameter $\delta_0 = 1 - \frac{|\mathbb{H}|}{|\mathbb{G}|}$ that renders masking ineffective, where $\mathbb{H}$ denotes a largest proper (nontrivial) subgroup of $\mathbb{G}$. They further conjectured that for any leakage with $\delta < \delta_0$, masking should be effective. In this subsection we prove this conjecture for the practically relevant cyclic case $\mathbb{G} = \mathbb{Z}_N$ (with arbitrary $N \in \mathbb{N}$) and derive an explicit, exponentially decaying advantage bound.

⁶This approximation assumes a worst-case scenario where the only uncertainty remaining about X after observing $\mathbf{L}(X)$ lies in the value of $\langle X, h \rangle$, which takes values in $\{0, 1\}$ with probabilities $\frac{1}{2} \pm \mu_h$.

Theorem 4.10. *Let X be uniform on $\mathbb{Z}_N$ and let $\mathbb{H} \leq \mathbb{Z}_N$ be a largest proper subgroup (so $[\mathbb{Z}_N : \mathbb{H}] = p$, the smallest prime dividing N). If, for some $\sigma > 0$, the leakage satisfies*

$$\delta \leq 1 - \frac{|\mathbb{H}|}{|\mathbb{Z}_N|} - \sigma = 1 - \frac{1}{p} - \sigma,$$

then for every leakage outcome $\mathbf{l}$,

$$\text{Adv}_X[\mathbf{l} \leftarrow \mathbf{L}(X)] \leq \frac{N-1}{N} \rho(\sigma, p)^n,$$

where

$$\rho(\sigma, p) \leq \left| (1 - \sigma) + \sigma e^{-j \frac{2\pi}{p}} \right| = \sqrt{1 - 2\sigma \left(1 - \cos \frac{2\pi}{p}\right) + O(\sigma^2)}.$$

In particular, for small σ ,

$$\rho(\sigma, p) \leq 1 - \left(1 - \cos \frac{2\pi}{p}\right) \sigma + O(\sigma^2).$$

Proof. Fix a leakage vector $\mathbf{l} = (l_1, \dots, l_n)$. For each i , let $\mathcal{P}^i = \{p_t^i\}_{t=0}^{N-1}$ be the posterior distribution of X_i given l_i , and set

$$\theta_k^i \triangleq \sum_{t=0}^{N-1} p_t^i e^{-j \frac{2\pi t k}{N}}, \quad k = 0, \dots, N-1.$$

Then $\theta_0^i = 1$ and $|\theta_k^i| \leq 1$. Writing $\mathbf{P} = \mathcal{P}^1 * \dots * \mathcal{P}^n$, convolution (denoted by “ $*$ ” operation) becomes pointwise multiplication in the Fourier domain:

$$\Theta_k \triangleq \sum_{t=0}^{N-1} \mathbf{P}(t) e^{-j \frac{2\pi t k}{N}} = \prod_{i=1}^n \theta_k^i.$$

Inverting the transform, we have:

$$\mathbf{P}(t) = \frac{1}{N} \sum_{k=0}^{N-1} \Theta_k e^{j \frac{2\pi t k}{N}} = \frac{1}{N} + \frac{1}{N} \sum_{k=1}^{N-1} \Theta_k e^{j \frac{2\pi t k}{N}}.$$

Applying the triangle inequality gives

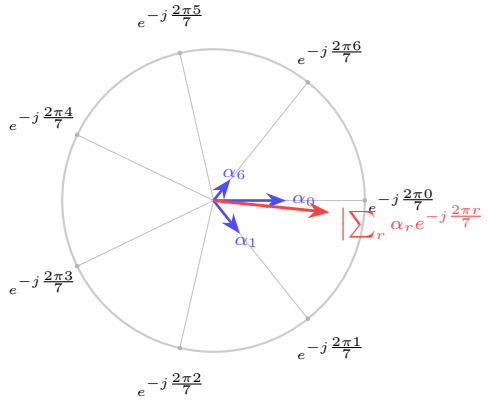
$$\text{Adv}_X[\mathbf{l} \leftarrow \mathbf{L}(X)] = \max_t \left| \mathbf{P}(t) - \frac{1}{N} \right| \leq \frac{1}{N} \sum_{k=1}^{N-1} \left| \prod_{i=1}^n \theta_k^i \right| \leq \frac{N-1}{N} \left(\max_{i, k \neq 0} |\theta_k^i| \right)^n.$$

It remains to upper bound $\max_{i,k \neq 0} |\theta_k^i|$ in terms of δ .

Fix i and $k \neq 0$, and consider the map $t \mapsto kt \bmod N$, which partitions $\mathbb{Z}_N$ into $m = N/\gcd(N, k)$ residue classes. Pushing $\mathcal{P}^i$ forward along this map does not increase the total variation distance, so on the m -point quotient we obtain a distribution $\alpha = (\alpha_0, \dots, \alpha_{m-1})$ with $\text{SD}(\alpha, U_m) \leq \delta$, and

$$|\theta_k^i| = \left| \sum_{t=0}^{N-1} p_t^i e^{-j \frac{2\pi t k}{N}} \right| = \left| \sum_{r=0}^{m-1} \alpha_r e^{-j \frac{2\pi r}{m}} \right|.$$

For instance, with $m = 7$, the following diagram shows how the maximization problem looks like.



Among nontrivial quotients, the worst case (largest possible modulus under the same TV budget) occurs for the minimal size $m = p = [\mathbb{Z}_N : \mathbb{H}]$. On this p -gon, moving the available total variation σ from the “main” direction to its nearest neighbor maximizes the modulus to first order. Concretely, take

$$\alpha_0 = 1 - \sigma, \quad \alpha_1 = \sigma, \quad \alpha_r = 0 \quad (r \geq 2).$$

A direct calculation shows that for $\sigma \leq 1/p$ its total variation from U_p is

$$\text{SD}(\alpha, U_p) = \frac{1}{2} \left(\left| (1 - \sigma) - \frac{1}{p} \right| + \left| \sigma - \frac{1}{p} \right| + (p - 2) \cdot \frac{1}{p} \right) = 1 - \frac{1}{p} - \sigma \geq \delta,$$

so the construction is admissible under the hypothesis. Its Fourier modulus equals

$$\left| \sum_{r=0}^{p-1} \alpha_r e^{-j \frac{2\pi r}{p}} \right| = \left| (1 - \sigma) + \sigma e^{-j \frac{2\pi}{p}} \right| = \sqrt{1 - 2\sigma \left(1 - \cos \frac{2\pi}{p} \right)} + O(\sigma^2).$$

This yields the asserted bound on $\rho(\sigma, p)$, with tightness for $\sigma \leq 1/p$. The Taylor expansion at $\sigma = 0$ gives the linearized form $\rho(\sigma, p) \leq 1 - (1 - \cos \frac{2\pi}{p})\sigma + O(\sigma^2)$.

With larger σ , more α_i components will be involved and this will decrease the overall absolute sum, as those α_i values are in other directions. Hence, we have $\max_{i, k \neq 0} |\theta_k^i| \leq \rho(\sigma, p)$. Plugging this into the derived bound for $\text{Adv}_X[\mathbf{l} \leftarrow \mathbf{L}(X)]$ completes the proof. $\square$

4.4.1 Comparison with Known Bounds

Case $N = 2^u$. In this setting, the index of the largest proper subgroup is $p = [\mathbb{Z}_N : \mathbb{H}] = 2$. The corresponding quotient group therefore has only two frequencies, namely 1 and $e^{-j\pi} = -1$. Substituting $\delta = 1 - \frac{1}{2} - \sigma$ into Theorem 4.10, the bound simplifies to

$$\text{Adv}_X[\mathbf{l} \leftarrow \mathbf{L}(X)] \leq \frac{2^u - 1}{2^u} (2\delta)^n.$$

This expression coincides with the bounds established earlier in Lemma 4.2 and Lemma 4.1, confirming the tightness of our general result in the binary case.

Case $N = q$ with q prime. When N is prime, we have $p = q$. In this case, the dedicated bound from Lemma 4.9 is tighter than the general bound derived in Theorem 4.10. The reason is that, in our general derivation, an additional multiplicative factor of $(N - 1)$ appears in the advantage bound. While negligible for small q , this factor makes our bound looser compared to the prime-field bound, particularly when q is large.

4.5 Security of Linear Gadgets

Our analysis so far has focused on standalone secrets and their masked encodings. We now extend this to more complex structures, specifically linear *gadgets*. Our goal is to demonstrate the applicability of the proposed decomposition technique for evaluating security in such settings.

Gates and gadgets. Recall that a gadget is a *family* of circuits (one per order n) that securely implements a masked version of a gate. Let $\mathbf{G}: (\mathbb{F}_{2^u})^b \rightarrow \mathbb{F}_{2^u}$ be a gate with fan-in b . The corresponding gadget, denoted $\mathbf{SG}: (\mathbb{F}_{2^u}^n)^b \rightarrow \mathbb{F}_{2^u}^n$, takes masked inputs and returns masked outputs.

Also recall that a *refresh gadget* has fan-in and fan-out 1, and it re-randomizes the encoding of a value $\mathbf{X}$ while preserving its underlying secret, i.e., $\bigoplus_i X_i = \bigoplus_i X'_i$. Here, we look at SR-SNI gadget [BBD⁺16], repeated here for ease of referencing in Algorithm 11.

Algorithm 11 SR-SNI

Input $\mathbf{X} = (X_1, \dots, X_n)$
Output $\mathbf{X}' = (X'_1, \dots, X'_n)$

- 1: **for** $i = 1$ **to** n **do**
- 2: **for** $j = i + 1$ **to** n **do**
- 3: $R \xleftarrow{\$} \mathbb{F}_{2^u}$
- 4: $X_i = X_i \oplus R$
- 5: $X_j = X_j \oplus R$
- 6: **return** $\mathbf{X}' = \mathbf{X}$

4.5.1 Problem Statement

Let $\Sigma_n = \{V_1, \dots, V_{m(n)}\}$ be the set of $m(n)$ intermediate variables in an $\mathbb{F}_2$ -linear gadget processing secret $X \in \mathbb{F}_{2^u}$. Besides the shares of X , Σ_n may include random values whose leakage could help an adversary estimate X .

As we already discussed in Section 3.2, the linearity assumption implies the existence of a matrix $\mathbf{P}_n \in \mathbb{F}_2^{P(n) \times (m(n)+1)}$ such that:

$$\mathbf{P}_n \cdot [X, V_1, \dots, V_{m(n)}]^\top = \mathbf{0}. \quad (4.11)$$

This system describes all parity constraints among the variables, and any additional relations are linear combinations of these rows. The adversary is assumed to know $\mathbf{P}_n$ (or any equivalent form).

Side-channel measurements yield:

$$\mathbf{L}_n = [\mathbf{L}(V_1), \dots, \mathbf{L}(V_{m(n)})],$$

and the adversary's goal is to estimate X from $\mathbf{L}_n$ and $\mathbf{P}_n$. We denote their advantage as $\text{Adv}_X[\mathbf{l}_n \leftarrow \mathbf{L}_n]$.

MAP adversary and exact advantage. Let $\mathcal{S}_n$ be the set of all solutions to (4.11). Each $\mathbf{S} \in \mathcal{S}_n$ is a vector of length $m(n) + 1$, with $\mathbf{S}[1]$ denoting the value of X . The MAP adversary outputs:

$$\tilde{X} = \underset{\alpha \in \mathbb{F}_{2^u}}{\operatorname{argmax}} \sum_{\substack{\mathbf{S} \in \mathcal{S}_n \\ \mathbf{S}[1] = \alpha}} \Pr(\mathbf{S} \mid \mathbf{l}_n).$$

The corresponding advantage is:

$$\text{Adv}_X[\mathbf{l}_n \leftarrow \mathbf{L}_n] = \mathbb{E}_{\mathbf{l}_n} \left[\Pr(\tilde{X} = x \mid \mathbf{l}_n) \right] - \frac{1}{2^u}.$$

A non-tight upper bound. To obtain a computable upper bound, we apply the reduction from δ -noisy leakage to ϵ -random probing. We replace each leakage $\mathbf{L}(V_i)$ with its erasure version $\phi_\epsilon(V_i)$, where $\epsilon \geq \epsilon_{\min}$ and $\epsilon_{\min}$ is derived from $\mathbf{L}$ (cf. (2.1)). This yields:

$$\text{Adv}_X[\mathbf{l}_n \leftarrow \mathbf{L}_n] \leq \text{Adv}_X[l_1, \dots, l_{m(n)} \leftarrow \phi_\epsilon(V_1), \dots, \phi_\epsilon(V_{m(n)})].$$

We denote this bound by $\text{Adv}(q, n, \epsilon)$, where $q = 2^u$. Since smaller ϵ yields less information, we have:

$$\text{Adv}_X[\mathbf{l}_n \leftarrow \mathbf{L}_n] \leq \text{Adv}(q, n, \epsilon_{\min}) \leq \text{Adv}(q, n, \epsilon). \quad (4.12)$$

In chapter 3, we estimated $\text{Adv}(q, n, \epsilon)$ for various gadgets. For the SR-SNI gadget with $n < 30$ and $\epsilon < 0.15$, we showed:

$$\text{Adv}(q, n, \epsilon) \leq \frac{q-1}{q} \cdot \epsilon^{0.6n}. \quad (4.13)$$

However, for some leakage functions—e.g., $\mathbf{L}(X) = \mathbf{ZV}(X)$ —the reduction gives $\epsilon_{\min} = 1$, leading to a trivial bound:

$$\text{Adv}_X[\mathbf{l}_n \leftarrow \mathbf{L}_n] \leq 1 - \frac{1}{q}.$$

In the next subsection, we show how our decomposition approach avoids this issue and yields tighter bounds in $\mathbb{F}_{2^u}$ settings.

4.5.2 Decomposition into Binary Systems

In an $\mathbb{F}_2$ -linear system such as

$$\mathbf{P}_n \cdot [X, V_1, \dots, V_{m(n)}]^\top = \mathbf{0},$$

the projections $\langle X, h \rangle$ and $\langle V_i, h \rangle$ also satisfy the same linear structure. More precisely:

Lemma 4.11. *Let $h \in \{1, \dots, 2^u - 1\}$. Then the system $\mathbf{P}_n \cdot [X, V_1, \dots, V_{m(n)}]^\top = \mathbf{0}$ implies:*

$$\mathbf{P}_n \cdot [\langle X, h \rangle, \langle V_1, h \rangle, \dots, \langle V_{m(n)}, h \rangle]^\top = \mathbf{0}.$$

Proof. Using the linearity of the binary inner product:

$$\langle V_1 \oplus V_2, h \rangle = \langle V_1, h \rangle \oplus \langle V_2, h \rangle, \quad \langle bV, h \rangle = b \cdot \langle V, h \rangle$$

for $b \in \mathbb{F}_2$, the result follows by applying these identities to each row of $\mathbf{P}_n$. $\square$

A tighter upper bound for $\text{Adv}_X[l_n \leftarrow L_n]$. The adversary also obtains side-channel leakage from each intermediate variable. From Lemma 4.3, we derive the following bound for the secret X :

$$\text{Adv}_X[l_n \leftarrow L_n] \leq \frac{1}{2^{u-1}} \sum_{h=1}^{2^u-1} \text{Adv}_{\langle X, h \rangle}[l_n \leftarrow L_n], \quad (4.14)$$

where each term $\text{Adv}_{\langle X, h \rangle}[l_n \leftarrow L_n]$ corresponds to the adversary's advantage in estimating a single binary projection $\langle X, h \rangle$ under δ -noisy leakage.

Applying the reduction from δ -noisy leakage to ϵ -random probing, we obtain:

$$\text{Adv}_{\langle X, h \rangle}[l_n \leftarrow L_n] \leq \text{Adv}(2, n, \epsilon_{\min}^h),$$

where $\epsilon_{\min}^h$ depends on both the projection index h and the leakage function $L(V)$. Importantly, while the systems for X and $\langle X, h \rangle$ are structurally the same, the field size is reduced to $\mathbb{F}_2$.

For binary variables, we previously established that:

$$\epsilon_{\min}^h = 2\mu_h, \quad (4.15)$$

where μ_h denotes the bias in estimating $\langle V, h \rangle$. One way to compute μ_h is via mutual information. As discussed in Section 4.2.4, it can be expressed as:

$$\mu_h = \left| \frac{1}{2} - H^{-1} [1 - \text{MI}(\langle V, h \rangle; L(V))] \right|,$$

where H^{-1} is the inverse of the binary entropy function.

Putting all of this together, we obtain:

$$\text{Adv}_X[l_n \leftarrow L_n] \leq \frac{1}{2^{u-1}} \sum_{h=1}^{2^u-1} \text{Adv}(2, n, \epsilon_{\min}^h) \leq 2 \cdot \max_h \text{Adv}(2, n, \epsilon_{\min}^h). \quad (4.16)$$

This shows that the decomposition approach enables a tighter reduction from the δ -noisy to the ϵ -random probing model for $\mathbb{F}_{2^u}$ -valued linear circuits.

Example 4.4. To illustrate the usefulness of the upper bound in (4.16), consider the leakage function $\text{ZV}(V)$ and the SR-SNI gadget. As discussed in Example 4.3, for $\text{ZV}(V)$, we have $\mu_h = \frac{1}{2^u}$, implying:

$$\epsilon_{\min}^h = 2\mu_h = \frac{1}{2^{u-1}}.$$

Applying the bound from (4.13), we obtain:

$$\text{Adv}_X[\mathbf{l}_n \leftarrow \mathbf{L}_n] \leq \text{Adv}(2, n, 2\mu_h) \leq \frac{1}{2} \left(\frac{1}{2^{u-2}} \right)^{0.6n},$$

which holds as long as $4\mu_h < 0.15$, i.e., $u \geq 5$.

Without the decomposition approach introduced in this chapter, the minimum erasure rate would be $\epsilon_{\min} = 1$, yielding a trivial upper bound of $\text{Adv}_X[\mathbf{l}_n \leftarrow \mathbf{L}_n] \leq 1 - \frac{1}{2^u}$. $\square$

Chapter 5

Leakage-Diagrams, Importance Sampling, and Composition in the Random Probing Model

This chapter presents follow-up results building on Chapters 3 and 4, and corresponds to work that is currently under submission [JMB26]. We follow the technical structure of the accompanying paper, while shortening overlapping background discussions.

Security evaluation of masking in *low-noise* regimes remains poorly understood: increasing the masking order does not automatically translate into higher concrete resistance once many correlated intermediates arise in a full implementation. This chapter addresses this gap from complementary angles and develops tools and bounds for compositional security in the random probing model (RPM), with an emphasis on constant-noise regimes.

5.1 Introduction

We already discussed the challenges of masking in *low-noise* settings. In such regimes, leakage can be sufficiently clean that classical first- and second-order masking no longer provides a comfortable security margin, motivating higher-order masking. For an isolated variable, increasing the masking order provably improves resistance in standard leakage models. However, the situation in fully masked cryptographic computations is less clear: many intermediate values interact across time and across gadgets, and the security gained by increasing the masking order depends delicately on the leakage rate, the circuit structure, and how randomness is consumed and refreshed. This chapter revisits these questions and provides additional results and estimation techniques tailored to the RPM.

5.1.1 Our Contributions

This work develops tools and bounds for compositional security in the random probing model, with an emphasis on constant-noise regimes:

- **Concrete noisy-to-RPM parameters for HW leakage (linear case).** We apply the decomposition technique to noisy Hamming-weight leakage and derive the resulting probing rate ϵ as a function of SNR for $\mathbb{F}_2$ -linear computations. This complements the general (nonlinear / arbitrary-field) treatment in [BCGR24] and provides a more tangible understanding of leakage-rate values (Section 5.2).
- **From local (advantage) to global (simulation) security in RPM without leak-free refresh.** We show how circuit-level simulatability can be reduced to per-gadget security at a leakage rate ϵ that does not need to decrease with circuit size or masking order (while the overall distinguishing probability still accumulates across blocks in the usual way).

This links simulation-based metrics used e.g., in [BRR25] to more direct per-gadget metrics, in the spirit of the methodology developed in Chapter 3 (Section 5.3).

- **Rare-event estimation via importance sampling.** We adapt importance sampling to RPM leakage-pattern sampling, enabling efficient estimation when the success probability is far below the Monte Carlo resolution, e.g., in low- ϵ and higher-order masking regimes (Section 5.4.2).
- **Leakage-diagram bounds for refresh gadgets.** Using the leakage-diagram technique of [DFŽ19], we give a complete and explicit derivation of provable bounds for a class of refresh gadgets, filling in technical steps that are only sketched at a high level in prior work (Section 5.4.3).
- **Composition via leakage-diagrams; clarifying the role of SNI in RPM.** We derive analytic bounds on the composition parameters (leakage-rate inflation at refresh boundaries) using leakage-diagrams, replacing Monte Carlo-only estimation. Our analysis identifies *bridges*—boundary-only relations created by refresh leakage—as the key mechanism governing dependence across refreshes. This “bridge view” explains why *Strong Non-Interference* (SNI) [BBD⁺16], although central for threshold probing model (TPM) composition, does not directly yield RPM composition guarantees, refining intuition put forward in [BC25] (Section 5.5).

- **Tool support: LAPSE.** We introduce LAPSE (Linear Algebraic Probing Security Estimator), which takes a Python-like description of linear gadgets, builds a vector-space (linear-algebraic) representation, and computes/estimates security parameters (exact when feasible; otherwise Monte Carlo enhanced with importance sampling).

5.2 Reduction of Noisy Hamming Weight to Random Probing

We derive the RPM leakage-rate parameter associated with *noisy Hamming-weight* leakage, following the decomposition technique introduced in Chapter 4. As discussed in the Preliminaries, side-channel leakage is naturally modeled as *noisy leakage*, but proving security directly in this model is often difficult. Duc et al. [DDF19] showed that security against δ -noisy leakage can be reduced to security in the random probing model (RPM) with a suitable leakage rate ϵ . Unfortunately, this generic reduction can be overly pessimistic: a circuit may remain secure under the original noisy leakage, while the derived ϵ is so large that known RPM guarantees become vacuous.

In this section, we instantiate the reduction for the practically important noisy Hamming-weight leakage. The Duc et al. parameter for this leakage was computed in [BCGR24]; here we instead derive the parameter induced by our decomposition approach.

Background: Duc et al. reduction. Let $X \in \mathbb{F}_q$ be uniform and let $L = \mathsf{L}(X)$ denote its leakage. Duc et al. [DDF19] show that the joint distribution (X, L) can be simulated from an *erasure view* $\phi_\epsilon(X)$ (i.e., X is revealed with probability ϵ and otherwise erased), for any $\epsilon \geq \epsilon_{\min}$, where

$$\epsilon_{\min} = 1 - q \sum_{\beta} \min_{\alpha \in \mathbb{F}_q} \Pr(L = \beta, X = \alpha) \leq q \delta. \quad (5.1)$$

In terms of mutual information (in bits) $\mathsf{I}_{\text{bits}}(X; L)$, one can bound [DFS19, BCGR24]

$$\frac{\mathsf{I}_{\text{bits}}(X; L)}{\log_2 q} \leq \epsilon_{\min} \leq q \sqrt{\frac{\ln 2}{2}} \mathsf{I}_{\text{bits}}(X; L).$$

This reduction is fully general (it does not depend on the structure of $\mathbb{F}_q$ or the circuit), but it can be loose, especially for large q .

Tighter reduction for $\mathbb{F}_2$ -linear circuits. For $q = 2^u$ and $\mathbb{F}_2$ -linear circuits, we proposed to reduce the leakage not to $\phi_\epsilon(X)$, but to erasures of binary linear forms $\phi_\epsilon(\langle H, X \rangle)$, where $H \in \{0, 1\}^u \setminus \{0\}$ and

$$\langle H, X \rangle = \bigoplus_{i=1}^u (h_i \cdot x_i),$$

with multiplication and addition in $\mathbb{F}_2$. In the binary case, the optimal parameter for a fixed H satisfies

$$\epsilon_{\min, H} = 2P_{\text{succ}}^H - 1,$$

where $P_{\text{succ}}^H \in [1/2, 1]$ is the optimal success probability of guessing $\langle H, X \rangle$ from the leakage L . The bottleneck parameter is

$$\epsilon_{\min}^b \triangleq \max_{H \neq 0} \epsilon_{\min, H}.$$

The reduction guarantees that if an $\mathbb{F}_2$ -linear circuit is secure in the RPM at leakage rate $\epsilon_{\min}^b$, then it is secure against the original leakage $L(X)$. Moreover, $\epsilon_{\min}^b \leq \epsilon_{\min}$, with strict improvement in regimes where $\epsilon_{\min}^b < \epsilon_{\min}$.

5.2.1 Noisy Hamming-Weight Leakage

Let $X \in \mathbb{F}_{2^u}$ be uniform and consider the noisy Hamming-weight leakage

$$L = L(X) = \text{HW}(X) + N, \quad N \sim \mathcal{N}(0, \sigma^2), \quad \text{HW}(X) \in \{0, \dots, u\}.$$

For this leakage, we have signal to noise ratio as $\text{SNR} = \frac{u}{4\sigma^2}$ [BCPZ16]. A direct computation of (5.1) yields $\epsilon_{\min} = 1 - 2Q(\frac{u}{2\sigma})$ [BCGR24, Prop. 2], where Q is the Gaussian tail function.

Computing $\epsilon_{\min}^b$. Fix $H \in \{0, 1\}^u \setminus \{0\}$, let $s = \text{HW}(H)$, and define the target bit $B = \langle H, X \rangle \in \{0, 1\}$. Let $A = \text{HW}(X)$. Writing

$$K_a(s) = \sum_{k=0}^a (-1)^k \binom{s}{k} \binom{u-s}{a-k},$$

one has the identities

$$\begin{aligned} \Pr(A = a, B = 0) &= \frac{1}{2^{u+1}} \left(\binom{u}{a} + K_a(s) \right), \\ \Pr(A = a, B = 1) &= \frac{1}{2^{u+1}} \left(\binom{u}{a} - K_a(s) \right). \end{aligned}$$

Therefore, the conditional density of L given $B = b$ is a Gaussian mixture

$$f_b(l) = \sum_{a=0}^u \alpha_a^{(b)} \varphi_\sigma(l - a), \quad \varphi_\sigma(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{-t^2/(2\sigma^2)},$$

with mixture weights

$$\begin{aligned} \alpha_a^{(0)} &= \Pr(A = a \mid B = 0) = \frac{\binom{u}{a} + K_a(s)}{2^u}, \\ \alpha_a^{(1)} &= \Pr(A = a \mid B = 1) = \frac{\binom{u}{a} - K_a(s)}{2^u}. \end{aligned}$$

The Bayes-optimal decision rule is the likelihood-ratio test

$$\hat{B}(l) = \arg \max_{b \in \{0,1\}} f_b(l),$$

and the optimal success probability satisfies

$$\begin{aligned} P_{\text{succ}}(u, \sigma, s) &= \frac{1}{2} + \frac{1}{4} \int_{-\infty}^{\infty} |f_0(l) - f_1(l)| \, dl \\ &= \frac{1}{2} + \frac{1}{2^{u+1}} \int_{-\infty}^{\infty} \left| \sum_{a=0}^u K_a(s) \varphi_\sigma(l - a) \right| \, dl. \end{aligned}$$

Hence

$$\epsilon_{\min, H} = 2P_{\text{succ}}^H - 1 = \frac{1}{2^u} \int_{-\infty}^{\infty} \left| \sum_{a=0}^u K_a(s) \varphi_\sigma(l - a) \right| \, dl.$$

Moreover, for uniform X the quantity above depends on H only through $s = \text{HW}(H)$, so $\epsilon_{\min}^b = \max_{1 \leq s \leq u} \epsilon_{\min}(u, \sigma, s)$. We numerically evaluate this integral for (u, σ, s) and report $\epsilon_{\min}$ and $\epsilon_{\min}^b$ for $u = 8$ across SNR values in Figure 5.1.

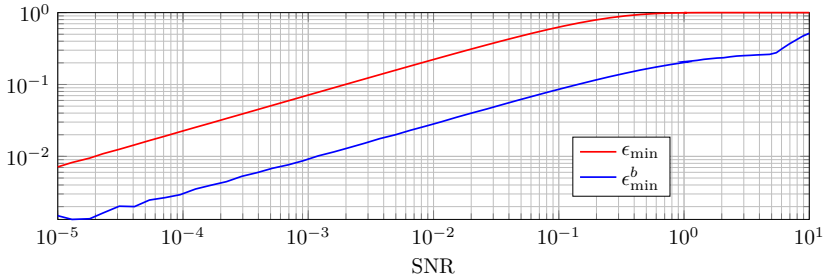


Figure 5.1: $\epsilon_{\min}$ and $\epsilon_{\min}^b$ at various SNR levels.

Interpretation of the results. State-of-the-art RPM-secure compilers and analyses typically operate in the regime of small probing rates, roughly $\epsilon \lesssim 0.02$ [JMB24, BCP⁺20, BRT21]. From Figure 5.1, the general Duc et al. reduction for noisy Hamming-weight leakage maps this regime to $\text{SNR} \lesssim 10^{-4}$ (for $u = 8$), whereas the $\mathbb{F}_2$ -linear reduction based on binary inner products permits substantially *lower* noise requirements (i.e., higher SNR), around $\text{SNR} \approx 5 \times 10^{-3}$. This improvement occurs because $\epsilon_{\min}^b \ll \epsilon_{\min}$ in this range.

At high SNR, the contrast becomes stark: for instance, at $\text{SNR} = 1$, the general reduction yields $\epsilon_{\min} \approx 1$ (and thus becomes essentially vacuous), while the binary-inner-product reduction still gives $\epsilon_{\min}^b \approx 0.2$. Although $\epsilon \approx 0.2$ is beyond what current full compilers can handle, it is already within reach for some standalone $\mathbb{F}_2$ -linear components such as linear refreshes considered in Chapter 3.

5.2.2 Why $\mathbb{F}_2$ -Linear Matters

$\mathbb{F}_2$ -linear subcircuits are ubiquitous in masked implementations. All refresh gadgets considered in this work are $\mathbb{F}_2$ -linear, and multiplication by a *public* scalar (applied sharewise) is also $\mathbb{F}_2$ -linear. In contrast, multiplying two n -sharings is not $\mathbb{F}_2$ -linear; nevertheless, for certain designs in Chapter 3, we showed how RPM security can be upper-bounded by the security of associated $\mathbb{F}_2$ -linear subcircuits.

Remaining gap. While reducing to binary inner products improves the parameter ($\epsilon_{\min}^b \leq \epsilon_{\min}$), the reduction can still be pessimistic at circuit level: even in the binary setting, an adversary with the original noisy leakage can be strictly weaker than an adversary given the corresponding RPM (erasure) leakage. We illustrate this with a toy example.

Example 5.1. Let the leakage be a binary symmetric channel (BSC) $L(X) = X \oplus E$ with $\Pr[E = 1] = P_e \leq \frac{1}{2}$. Let X, R be uniform bits and define $Y = X \oplus R$.

Noisy leakage. Given $\{L(X), L(R), L(Y)\}$, the adversary always has the option to output $L(X)$ as its guess for X , which succeeds with probability $1 - P_e$. Moreover, the other two samples do not improve this: indeed,

$$L(Y) \oplus L(R) = (Y \oplus E_3) \oplus (R \oplus E_2) = X \oplus (E_2 \oplus E_3),$$

and the effective error probability of $E_2 \oplus E_3$ equals $2P_e(1 - P_e) \geq P_e$ for all $P_e \in [0, \frac{1}{2}]$. Hence, $L(Y) \oplus L(R)$ is a (weakly) noisier observation of X than $L(X)$, and the optimal strategy is to rely on $L(X)$; thus $P_{\text{noisy}} = 1 - P_e$.

RPM (erasure) view. In the corresponding RPM view, the adversary obtains $\{\phi_\epsilon(X), \phi_\epsilon(R), \phi_\epsilon(Y)\}$, where each $\phi_\epsilon(Z)$ reveals Z with probability ϵ and otherwise returns $\perp$. For the BSC, the optimal success probability of guessing X from $L(X)$ is $1 - P_e$, and the induced binary parameter is

$$\epsilon = 2(1 - P_e) - 1 = 1 - 2P_e.$$

The optimal decision rule in the RPM view is: if $\phi_\epsilon(X) \neq \perp$, output it; else if $\phi_\epsilon(R), \phi_\epsilon(Y) \neq \perp$, output $\phi_\epsilon(Y) \oplus \phi_\epsilon(R)$; otherwise guess uniformly at random. Therefore,

$$P_{\text{RPM}} = \Pr[X \text{ revealed}] \cdot 1 + \Pr[X \text{ erased}] \cdot \left(\Pr[R, Y \text{ revealed}] \cdot 1 + \Pr[\text{otherwise}] \cdot \frac{1}{2} \right),$$

which simplifies to

$$P_{\text{RPM}} = \frac{1}{2}(1 + \epsilon + \epsilon^2 - \epsilon^3).$$

Substituting $\epsilon = 1 - 2P_e$ yields

$$P_{\text{RPM}} - P_{\text{noisy}} = P_e(1 - 2P_e)^2 > 0 \quad \text{for all } P_e \in (0, \frac{1}{2}),$$

exhibiting a strict gap. □

5.3 From Local Security to Global Security

This section relates two common ways of quantifying masking security. The *advantage* Adv is a *local* metric: it measures an adversary's optimal guessing advantage for a single target secret (e.g., a share-combination such as an S-box output). Following Prest et al. [PGMP19], we refer to such measures as *direct*.

In contrast, simulation-based definitions are *indirect*: security holds if there exists a simulator that can reproduce the leakage distribution (up to statistical distance) without access to the secrets. Because simulation ranges over leakage from (potentially) many intermediate values, it constitutes a *global* notion. We refer to Section 2.5 in the Preliminaries for the formal definitions of these metrics.

Why local security need not imply global security. In general, local indistinguishability does *not* yield global simulatability. Two typical failure modes are: (i) *non-independent leakage*, e.g., a leakage sample depending jointly on multiple shares, and (ii) *cross-gadget accumulation*, where leakage collected across many gadgets reveals enough shares to reconstruct a secret unless refresh/randomness prevents accumulation.

Assumptions (RPM with refreshed blocks). We consider a masked circuit SC_n composed of gadgets $\text{SG}_1, \dots, \text{SG}_m$, with corresponding natives $X_1, \dots, X_m$ (not necessarily independent). We assume:

1. **Independent leakage:** each leakage sample is a randomized function of a single wire, and conditioned on that wire, it is independent of all other wires and secrets.
2. **Block separation:** refresh and randomness allow the leakage to be partitioned as $\mathcal{L} = \{\mathcal{L}_1, \dots, \mathcal{L}_m\}$ such that, conditioned on public inputs P and the corresponding native X_i , the blocks $\mathcal{L}_i$ are mutually independent and independent of the other natives.
3. **Erasure-like view:** for each block, there exists a coupling to an erasure-channel view of the native, i.e., $\mathcal{L}_i$ can be replaced by $\phi_{\epsilon_i}(X_i)$ for some parameter ϵ_i .

Feasibility of the assumptions. Assumption (1) (independent leakage) is common in theoretical masking analyses, but its validity for software depends on the target processor and implementation. Register transitions and other microarchitectural effects can produce leakage depending jointly on multiple shares [GHP⁺21]. Applying our results to a concrete implementation therefore requires assessing whether this assumption is justified. Assumption (3) (erasure-like view) holds for $\mathbb{F}_{2^u}$ -linear gadgets in the RPM: a leakage instance either makes the native recoverable or reveals no information about it; see Lemma 3.3.

The only nonlinear gadget used by our compiler is multiplication. For the specific design considered in Chapter 3, we upper bound its RPM security by that of associated $\mathbb{F}_2$ -linear subcircuits.

Assumption (2) (block separation) does not hold a priori when refresh gadgets leak: refresh leakage can create cross-block dependencies (*bridges*) between variables on the two sides of the refresh boundary. We restore block separation via the bridge reduction: we replace each leaky refresh with an ideal leak-free refresh at the cost of (i) revealing the refreshed native with small probability and (ii) inflating the effective probing rate from ϵ to $\epsilon' > \epsilon$. We first discussed this transformation in Section 3.3, and we present a refined statement in Theorem 5.8.

Why not mutual information. Under block separation one can upper bound mutual information additively across gadgets. However, mutual infor-

mation is not a cryptographic security notion and can be vacuous in the presence of unrestricted post-processing using public inputs/outputs. We therefore work directly with statistical distance and simulation.

Lemma 5.1 (Erasure blocks imply global simulatability). *Let $X_1, \dots, X_m$ be arbitrary $\mathbb{F}_q$ -valued random variables and let $\mathcal{L}' = (\phi_{\epsilon_1}(X_1), \dots, \phi_{\epsilon_m}(X_m))$. There exists a simulator Sim (given only public inputs and wire indices) such that*

$$\text{SD}(\mathcal{L}', \text{Sim}) \leq 1 - \prod_{i=1}^m (1 - \epsilon_i) \leq \sum_{i=1}^m \epsilon_i.$$

Proof. Let Sim output $(\perp, \dots, \perp)$ deterministically. Then $\mathcal{L}' \neq (\perp, \dots, \perp)$ iff at least one coordinate is not erased, which happens with probability $1 - \prod_{i=1}^m (1 - \epsilon_i)$. Since $\text{SD}(P, Q) \leq \Pr[P \neq Q]$ under the natural coupling, the first bound follows, and the second is the union bound. $\square$

5.4 Random Probing Leakage in $\mathbb{F}_q$ -Linear Circuits

We revisit $\mathbb{F}_q$ -linear circuits and develop tools that go beyond the exploration in Section 3.2. As a running example, we consider the refresh construction SR-multi [DFZ19] (Algorithm 12), which repeats the SR-Simple refresh of [RP10] for $k \geq 1$ consecutive rounds. We start by explaining how a vector-space representation is derived for a linear algorithm and how it enables computation of $\mathbf{E}_X(n, \epsilon)$.

Algorithm 12 SR-multi

Input: $X = \{X_1, \dots, X_n\}$, k
Output: $Y = \{Y_1, \dots, Y_n\}$ with $X = \sum_{i=1}^n X_i = \sum_{i=1}^n Y_i$

```

1: for  $j = 1$  to  $k$  do
2:    $R_n \leftarrow 0$ 
3:   for  $i = 1$  to  $n - 1$  do
4:      $R_i \xleftarrow{\$} \mathbb{F}_q$ 
5:      $R_n \leftarrow R_n - R_i$ 
6:   for  $i = 1$  to  $n$  do
7:      $X_i \leftarrow X_i + R_i$ 
8:  $Y \leftarrow X$  ▷ Copy to outputs
9: return  $Y$ 

```

Vector-space representation. In an $\mathbb{F}_q$ -linear circuit, every SSA wire is an $\mathbb{F}_q$ -linear combination of the input shares and the fresh random variables. Fix an SSA enumeration of the wires, and let the basis consist of the n input-share

symbols together with the r randomness symbols introduced by $\stackrel{\$}{\leftarrow}$. Each wire W admits a representation vector $\vec{w} \in \mathbb{F}_q^{n+r}$ such that

$$W = \langle \vec{w}, (X_1, \dots, X_n, R_1, \dots, R_r) \rangle.$$

The set of all wire-vectors spans a subspace $\mathcal{V}_{\text{SC}_n} \subseteq \mathbb{F}_q^{n+r}$.

Our LAPSE tool constructs these vectors on the fly: each instruction produces a new vector derived from its operands; when sampling $R \stackrel{\$}{\leftarrow} \mathbb{F}_q$, a new basis element is added and the sampled wire is represented by the corresponding unit vector.

Example 5.2 (Vector space and SSA wire enumeration). For $(n = 2, k = 1)$, Algorithm 12 introduces one fresh random variable R_1 and sets $R_2 = -R_1$. An SSA enumeration of the wires is

$$\Sigma_{\text{SR-multi}, n=2} = \{X_1, X_2, R_1, R_2, X_1 + R_1, X_2 + R_2, Y_1, Y_2\}.$$

Using the basis (X_1, X_2, R_1) , the corresponding vectors (in that order) are

$$(1, 0, 0), (0, 1, 0), (0, 0, 1), (0, 0, -1), (1, 0, 1), (0, 1, -1), (1, 0, 1), (0, 1, -1).$$

The (unshared) native secret is $X \triangleq X_1 + X_2$, which corresponds to the vector $(1, 1, 0)$. $\square$

Erasure-channel viewpoint. For $\mathbb{F}_q$ -linear circuits handling a single uniform secret X , we proved in Chapter 3 that any set of leaked wires $\mathcal{L} \subseteq \Sigma_{\text{SC}_n}$ either determines X uniquely or is independent of X .

Concretely, let $\mathcal{L} \subseteq \Sigma_{\text{SC}_n}$ be the set of leaked SSA wires. If the vector of X lies in the $\mathbb{F}_q$ -span of the vectors corresponding to $\mathcal{L}$, then X is recovered uniquely; we write $\text{is-in-span}(X, \mathcal{L}) = 1$. Otherwise $\text{is-in-span}(X, \mathcal{L}) = 0$, in which case $\mathcal{L}$ reveals no information about X . LAPSE checks membership in $\text{span}(\mathcal{L})$ via Gaussian elimination.

Let $E_X(n, \epsilon)$ be the probability (over the random RPM leakage pattern) that X is revealed. We define

$$E_X(n, \epsilon) \triangleq \Pr_{\mathcal{L} \leftarrow \text{RPM}(\epsilon)} [\text{is-in-span}(X, \mathcal{L}) = 1].$$

Exact method. Using the vector-space representation, we can compute $E_X(n, \epsilon)$ *exactly* as

$$E_X(n, \epsilon) = \sum_{\mathcal{L} \subseteq \Sigma_{\text{SC}_n}} \text{is-in-span}(X, \mathcal{L}) \cdot \Pr[\mathcal{L}] = \mathbb{E}[\text{is-in-span}(X, \mathcal{L})].$$

Recall that $m(n) := |\Sigma_{\mathbb{S}C_n}|$ is the number of SSA wires. Under RPM, wires leak independently with rate ϵ , so

$$\Pr[\mathcal{L}] = \epsilon^{|\mathcal{L}|} (1 - \epsilon)^{m(n) - |\mathcal{L}|}.$$

This approach enumerates all $2^{m(n)}$ leakage patterns and is feasible only for small $m(n)$.

5.4.1 Monte Carlo Estimation

We used Monte Carlo (MC) estimation in Chapter 3; we briefly recall it here to motivate how *importance sampling* can improve naive MC in rare-event regimes.

When $m(n)$ is large, exact enumeration is infeasible. A standard alternative is the MC estimator in Algorithm 13. For a target (n, ϵ) , we estimate $E_X(n, \epsilon) = \Pr[X \in \text{span}(\mathcal{L})]$, where $\mathcal{L} \subseteq \Sigma_{\mathbb{S}C_n}$ is the random set of leaked SSA wires (each wire included independently with probability ϵ).

The estimator $\hat{E}_X$ is *unbiased*, i.e., $\mathbb{E}[\hat{E}_X] = E_X(n, \epsilon)$. Its cost is $\mathcal{O}(N \cdot C_{\text{span}})$, where C_{span} is the cost of one span-membership test (via Gaussian elimination; with bitset implementations this is near-quadratic in practice). In typical regimes where $N \ll 2^{m(n)}$, this is vastly more efficient than exact enumeration.

Algorithm 13 Monte Carlo estimation of $E_X(n, \epsilon)$

Input: $\mathbb{S}C, (n, \epsilon)$

Output: $\hat{E}_X \approx E_X(n, \epsilon)$

```

1:  $b \leftarrow 0$ 
2: for  $i = 1$  to  $N$  do
3:    $\mathcal{L} \leftarrow \emptyset$ 
4:   for each  $W \in \Sigma_{\mathbb{S}C_n}$  do
5:     Sample  $B \leftarrow \text{Ber}(\epsilon)$ 
6:     if  $B = 1$  then
7:        $\mathcal{L} \leftarrow \mathcal{L} \cup \{W\}$ 
8:    $b \leftarrow b + \text{is-in-span}(X, \mathcal{L})$ 
9: return  $\hat{E}_X \leftarrow \frac{b}{N}$ 
```

Accuracy of Monte Carlo. Let $Z_i \in \{0, 1\}$ be i.i.d. with $\Pr(Z_i = 1) = E_X(n, \epsilon)$ and define $B := \sum_{i=1}^N Z_i$ and $\hat{E}_X := B/N$. Then

$$\mathbb{E}[\hat{E}_X] = E_X(n, \epsilon), \quad \mathbb{V}\text{ar}[\hat{E}_X] = \frac{E_X(n, \epsilon)(1 - E_X(n, \epsilon))}{N}.$$

Hoeffding's inequality [Hoe63] gives, for any $\zeta > 0$,

$$\Pr\left(|\hat{E}_X - E_X(n, \epsilon)| \geq \zeta\right) \leq 2e^{-2N\zeta^2}.$$

Thus, for absolute accuracy $\zeta = 10^{-3}$ and confidence 95%, it suffices to take $N \geq \ln(40)/(2\zeta^2) \approx 1.9 \times 10^6$.

For small probabilities ($E_X(n, \epsilon) \ll 1$), the *relative* standard error satisfies

$$\frac{\sqrt{\mathbb{V}\text{ar}(\hat{E}_X)}}{E_X(n, \epsilon)} \approx \frac{1}{\sqrt{E_X(n, \epsilon)N}},$$

so achieving 10% relative error requires $N \approx 100/E_X(n, \epsilon)$ (and 1% requires $N \approx 10,000/E_X(n, \epsilon)$).

Limitation: rare events. MC can estimate an event only if it occurs in the sample. If $\theta := \Pr(\mathbf{e}) \ll 1/N$, then with high probability no trial exhibits $\mathbf{e}$ and the estimator returns 0. Indeed,

$$\Pr(\text{no occurrences in } N \text{ trials}) = (1 - \theta)^N \leq e^{-\theta N}.$$

To see at least one occurrence with confidence $1 - \alpha$, it suffices that

$$N \geq \frac{\ln(1/\alpha)}{\theta}.$$

For instance, if $\epsilon = 10^{-3}$ and $n = 3$, a crude proxy such as $\theta \approx \epsilon^n = 10^{-9}$ suggests that observing even a single hit with 90% confidence requires $N \gtrsim 2.3 \times 10^9$ trials, which is beyond practical budgets. Consequently, naive MC becomes ineffective precisely in the low- ϵ , higher- n regimes of interest.

5.4.2 Importance Sampling

Importance sampling (IS) estimates probabilities beyond the reach of naive Monte Carlo [Kv78]. To estimate the probability of a rare event $\mathbf{e}$, we sample leakage patterns from a *tilted* distribution under which $\mathbf{e}$ is no longer rare, and reweight samples to recover the target probability.

Change of measure for RPM. Let $m(n) := |\Sigma_{\mathbb{S}C_n}|$ and let $\mathcal{L} \subseteq \Sigma_{\mathbb{S}C_n}$ denote the (random) set of leaked wires. Under RPM with leakage rate ϵ , the probability of a leakage pattern $\mathcal{L}$ is

$$\Pr_{\epsilon}(\mathcal{L}) = \epsilon^{|\mathcal{L}|} (1 - \epsilon)^{m(n) - |\mathcal{L}|}.$$

Fix a *tilted* rate $\tilde{\epsilon} \in (0, 1)$ and sample $\mathcal{L}$ instead from $\Pr_{\tilde{\epsilon}}$ (typically $\tilde{\epsilon} > \epsilon$ so that patterns contain more leaks on average, making the target event less rare). Since $\epsilon, \tilde{\epsilon} \in (0, 1)$, $\Pr_{\tilde{\epsilon}}(\mathcal{L}) > 0$ whenever $\Pr_{\epsilon}(\mathcal{L}) > 0$, and for any function $f(\mathcal{L})$,

$$\mathbb{E}_{\epsilon}[f(\mathcal{L})] = \sum_{\mathcal{L}} f(\mathcal{L}) \Pr_{\epsilon}(\mathcal{L}) = \sum_{\mathcal{L}} f(\mathcal{L}) \underbrace{\frac{\Pr_{\epsilon}(\mathcal{L})}{\Pr_{\tilde{\epsilon}}(\mathcal{L})}}_{A(\mathcal{L})} \Pr_{\tilde{\epsilon}}(\mathcal{L}) = \mathbb{E}_{\tilde{\epsilon}}[f(\mathcal{L}) A(\mathcal{L})],$$

where the likelihood-ratio weight is

$$A(\mathcal{L}) = \left(\frac{\epsilon}{\tilde{\epsilon}}\right)^{|\mathcal{L}|} \left(\frac{1 - \epsilon}{1 - \tilde{\epsilon}}\right)^{m(n) - |\mathcal{L}|}.$$

Our target event is $f(\mathcal{L}) = \text{is-in-span}(X, \mathcal{L})$, hence

$$\mathbb{E}_X(n, \epsilon) = \mathbb{E}_{\epsilon}[f(\mathcal{L})] = \mathbb{E}_{\tilde{\epsilon}}[f(\mathcal{L}) A(\mathcal{L})].$$

IS estimator (unbiased). Algorithm 14 implements this identity.

Algorithm 14 Importance sampling estimation of $\mathbb{E}_X(n, \epsilon)$

Input: $\mathbb{S}C$, (n, ϵ) , $\tilde{\epsilon}$

Output: $\hat{\mathbb{E}}_X^{\text{IS}} \approx \mathbb{E}_X(n, \epsilon)$

```

1:  $b \leftarrow 0$ 
2: for  $i = 1$  to  $N$  do
3:    $\mathcal{L}_i \leftarrow \emptyset$ 
4:   for each  $W \in \Sigma_{\mathbb{S}C_n}$  do
5:     Sample  $B \leftarrow \text{Ber}(\tilde{\epsilon})$ 
6:     if  $B = 1$  then
7:        $\mathcal{L}_i \leftarrow \mathcal{L}_i \cup \{W\}$ 
8:    $A_i \leftarrow \left(\frac{\epsilon}{\tilde{\epsilon}}\right)^{|\mathcal{L}_i|} \left(\frac{1 - \epsilon}{1 - \tilde{\epsilon}}\right)^{m(n) - |\mathcal{L}_i|}$ 
9:    $b \leftarrow b + \text{is-in-span}(X, \mathcal{L}_i) \cdot A_i$ 
10: return  $\hat{\mathbb{E}}_X^{\text{IS}} \leftarrow \frac{b}{N}$ 

```

Its unbiasedness follows immediately from the change-of-measure identity: with $\mathcal{L}_i \sim \Pr_{\tilde{\epsilon}}$,

$$\mathbb{E}[\hat{\mathbf{E}}_X^{\text{IS}}] = \mathbf{E}_X(n, \epsilon).$$

Variance and estimation. Let $I_i := \text{is-in-span}(X, \mathcal{L}_i)$ and $Y_i := I_i A_i$. A consistent variance estimator is

$$\widehat{\text{Var}}(\hat{\mathbf{E}}_X^{\text{IS}}) = \frac{1}{N(N-1)} \sum_{i=1}^N (Y_i - \hat{\mathbf{E}}_X^{\text{IS}})^2.$$

If $\tilde{\epsilon}$ is chosen too far from ϵ , the weights A_i can become very large, which increases variance. We choose $\tilde{\epsilon}$ so that the success probability under the tilted measure, $\Pr_{\tilde{\epsilon}}[I = 1]$, lies in a comfortable range (e.g., 5%–30%). A short pilot run can tune $\tilde{\epsilon}$ by bisection. For simplicity, we apply a *single* tilt parameter $\tilde{\epsilon}$ to all wires.

Example 5.3. We evaluate $\mathbf{E}_X(n, \epsilon)$ for SR-multi (Alg. 12) using LAPSE. Given a threshold on the SSA wire count $m(n)$ below which exact enumeration is feasible, LAPSE proceeds as follows for each (n, ϵ) : it computes $\mathbf{E}_X(n, \epsilon)$ exactly when possible; otherwise it runs Monte Carlo, and switches to importance sampling when Monte Carlo falls into the rare-event (no-hit) regime. Figure 5.2 reports the resulting estimates. $\square$

5.4.3 Leakage-Diagram View

Our next contribution is an analytic estimation of $\mathbf{E}_X(n, \epsilon)$ using *leakage diagram* technique. Dziembowski et al. [DFŽ19] introduced this technique, which turns the event $\text{is-in-span}(X, \mathcal{L}) = 1$ for refresh gadgets into a graph-connectivity problem. For SR-multi with parameters (n, k) , the diagram is an $(n+1) \times (k+1)$ grid of vertices. Each leaked wire is represented by adding an edge between two neighboring vertices, see Figure 5.3. They show that the native $X = \sum_{i=1}^n X_i$ is recoverable from $\mathcal{L}$ if and only if the resulting diagram $G(\mathcal{L})$ contains a path connecting the leftmost column to the rightmost column. For completeness, we restate the construction and derive an explicit upper bound on $\mathbf{E}_X(n, \epsilon)$.

Leakage-diagram for SR-multi. Write the input shares to the j -th refresh as $\mathbf{X}^j = \{X_1^j, \dots, X_n^j\}$ for $j \in \{1, \dots, k\}$, where the refresh maps $\mathbf{X}^j \mapsto \mathbf{X}^{j+1}$.

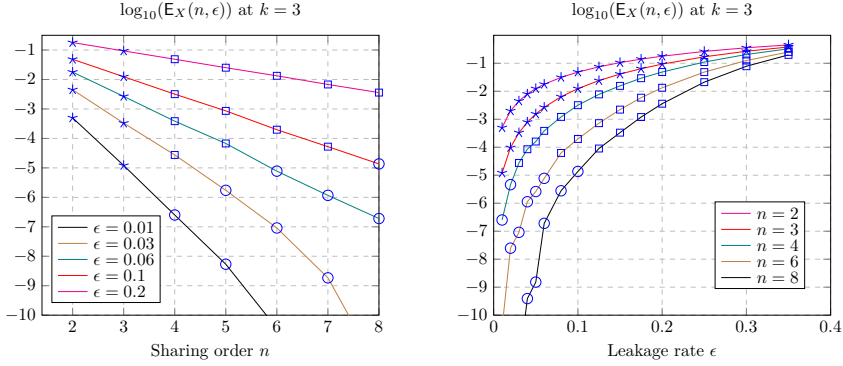


Figure 5.2: Evaluation of $E_X(n, \epsilon)$ over a grid of (n, ϵ) for SR-multi with $k = 3$. Stars: exact enumeration (Section 5.4); squares: Monte Carlo (Section 5.4.1); circles: importance sampling (Section 5.4.2). Each estimate uses $N = 10^5$ trials. Importance sampling allows reliable estimates well below the naive Monte Carlo resolution $1/N$ (here $\log_{10}(1/N) = -5$).

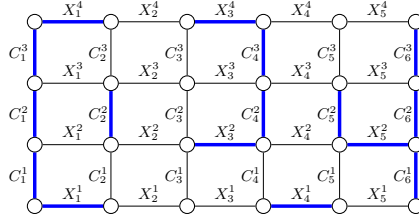


Figure 5.3: An example of leakage-diagram with $(n = 5, k = 3)$. An instance of leaking wires are specified in blue. Edges at leftmost and rightmost columns are also blue since they are set to 0. This example contains no path from the leftmost to the rightmost column.

In refresh round j , sample $R_1^j, \dots, R_{n-1}^j \xleftarrow{\$} \mathbb{F}_q$ and define

$$C_0^j \leftarrow 0, \quad C_i^j \leftarrow C_{i-1}^j + R_i^j \quad (1 \leq i < n),$$

and

$$R_n^j \leftarrow -C_{n-1}^j, \quad C_n^j \leftarrow 0.$$

The shares update as

$$X_i^{j+1} \leftarrow X_i^j + R_i^j \quad (1 \leq i \leq n).$$

Thus the relevant wire families are $\{X_i^j\}$, $\{C_i^j\}$, and the random wires $\{R_i^j\}$ used to build them.

Lemma 5.2 (Removing R -wires). *Let $\mathcal{L}$ be an $\text{RPM}(\epsilon)$ leakage instance on wires $\{X_i^j, C_i^j, R_i^j\}$ of SR-multi . Define a mapping τ from wire sets to $\{X_i^j, C_i^j\}$ -only wires by*

$$\tau(\mathcal{L}) = (\mathcal{L} \cap \{X_i^j, C_i^j\}) \cup \bigcup_{R_i^j \in \mathcal{L}} \{C_{i-1}^j, C_i^j\},$$

Let $\mathcal{L}'$ be $\text{RPM}(\epsilon_{\text{LD}})$ leakage instance on wires $\{X_i^j, C_i^j\}$ with $\epsilon_{\text{LD}} = \sqrt{\epsilon}$. Then for every fixed wire set $W \subseteq \{X_i^j, C_i^j, R_i^j\}$,

$$\Pr_{\epsilon}[W \subseteq \mathcal{L}] = \epsilon^{|W|} \leq (\sqrt{\epsilon})^{|\tau(W)|} = \Pr_{\epsilon_{\text{LD}}}[\tau(W) \subseteq \mathcal{L}'].$$

Consequently, since $\text{is-in-span}(X, \cdot)$ is monotone,

$$\Pr_{L \leftarrow \text{RPM}(\epsilon)}[\text{is-in-span}(X, L) = 1] \leq \Pr_{L' \leftarrow \text{RPM}(\epsilon_{\text{LD}})}[\text{is-in-span}(X, L') = 1].$$

Proof. Under $\text{RPM}(\epsilon)$, all wire-leak indicators are independent, hence $\Pr_{\epsilon}[W \subseteq \mathcal{L}] = \epsilon^{|W|}$. The set $\tau(W)$ replaces each R -wire by *two* C -wires, so

$$|\tau(W)| \leq |W \cap \{X_i^j, C_i^j\}| + 2|W \cap \{R_i^j\}| \leq 2|W|.$$

Therefore

$$(\sqrt{\epsilon})^{|\tau(W)|} \geq (\sqrt{\epsilon})^{2|W|} = \epsilon^{|W|}.$$

The right-hand side equals $\Pr_{\epsilon_{\text{LD}}}[\tau(W) \subseteq \mathcal{L}']$ because $\mathcal{L}'$ is i.i.d. $\text{RPM}(\epsilon_{\text{LD}})$ on $\{X_i^j, C_i^j\}$. $\square$

Lemma 5.3 (Leakage-diagram characterization [DFŽ19]). *Consider SR-multi with parameters (n, k) , and assume leakage is restricted to wires $\{X_i^j, C_i^j\}$. Define a graph $G(\mathcal{L})$ with vertex set $\{v_{i,j} : 0 \leq i \leq n, 0 \leq j \leq k\}$, see Figure 5.3. Add a horizontal edge $(v_{i-1,j}, v_{i,j})$ whenever X_i^{j+1} leaks, and a vertical edge $(v_{i,j-1}, v_{i,j})$ whenever C_i^j leaks. (Additionally, boundary vertices $v_{0,j}$ and $v_{n,j}$ are treated as known constants because $C_0^j = C_n^j = 0$.) Then*

$$\text{is-in-span}(X, \mathcal{L}) = 1 \iff G(\mathcal{L}) \text{ left and right columns are connected.}$$

If no such path exists, then $\mathcal{L}$ is statistically independent of X .

Lemma 5.4 (Path probability bound for the leakage-diagram). *Consider the leakage-diagram graph for SR-multi with (n, k) . Assume that vertical edges in the leftmost/rightmost columns are always present, and the other edges are present independently with probability ϵ_{LD} .*

Let $\text{Conn}_{n,k}$ be the event that the diagram contains a path connecting the leftmost and the rightmost columns. Then, for $3\epsilon_{\text{LD}} < 1$,

$$\Pr[\text{Conn}_{n,k}] \leq (k+1) \sum_{\ell \geq n} (3\epsilon_{\text{LD}})^{\ell-1} \epsilon_{\text{LD}} = (k+1) \frac{(3\epsilon_{\text{LD}})^n}{1 - 3\epsilon_{\text{LD}}}.$$

Proof. Let s denote the leftmost column and t the rightmost column. These are always-present vertical edges. Any s - t path must advance from column 0 to column n , hence must contain at least n horizontal (rightward) edges. In particular its total length ℓ satisfies $\ell \geq n$.

Fix an s - t walk of length ℓ with no immediate backtracking. All ℓ edges on this walk must be present for it to exist, which happens with probability $\epsilon_{\text{LD}}^\ell$.

We now bound the number of such s - t walks of length ℓ . From s , there are at most $(k+1)$ choices for the first step (one per row). After the first step, at each subsequent step there are at most 3 choices in the grid when immediate backtracking is forbidden. Thus the number of candidate walks of length ℓ is at most $(k+1) \cdot 3^{\ell-1}$.

By a union bound over all $\ell \geq n$,

$$\begin{aligned} \Pr[\text{Conn}_{n,k}] &\leq \sum_{\ell \geq n} (k+1) \cdot 3^{\ell-1} \cdot \epsilon_{\text{LD}}^\ell = (k+1) \sum_{\ell \geq n} \epsilon_{\text{LD}} (3\epsilon_{\text{LD}})^{\ell-1} \\ &= (k+1) \frac{(3\epsilon_{\text{LD}})^n}{1 - 3\epsilon_{\text{LD}}}, \end{aligned}$$

which holds for $3\epsilon_{\text{LD}} < 1$. □

Corollary 5.5. *Under the assumptions of Lemma 5.4 and Lemma 5.3,*

$$\mathbb{E}_X(n, \epsilon) \leq \Pr_{\epsilon_{\text{LD}}}[\text{Conn}_{n,k}] \leq (k+1) \frac{(3\epsilon_{\text{LD}})^n}{1 - 3\epsilon_{\text{LD}}} = (k+1) \frac{(3\sqrt{\epsilon})^n}{1 - 3\sqrt{\epsilon}} \quad (\epsilon < \frac{1}{9}).$$

Hence, $\mathbb{E}_X(n, \epsilon)$ for SR-multi is exponentially decreasing at any constant rate $\epsilon \leq 0.1$.

5.5 Composition in the Random Probing Model

We already discussed composition in the random probing model (RPM) in Chapter 3. Here, we revisit the composition problem to introduce a *bridging* viewpoint, which yields a tighter bound on the effective leakage-rate inflation induced by refresh boundaries (in particular, around SR). We also explain why *strong non-interference* (SNI), while a standard sufficient condition for composition in the threshold probing model (TPM), does not by itself lead to tight composition bounds in RPM.

In the TPM, SNI (introduced by Barthe et al. [BBD⁺16]) provides a convenient way to reason about gadget composition.

Definition 5.6 (SNI [BBD⁺16]). Let $\mathbb{S}\mathbb{G}$ be a gadget that takes an input sharing $\mathbf{X} = \{X_1, \dots, X_n\}$ and produces an output sharing $\mathbf{Y} = \{Y_1, \dots, Y_n\}$, and let $\Sigma_{\mathbb{S}\mathbb{G}}$ denote its internal wires. The gadget is *t-SNI* if for every choice of sets $S_Y \subseteq \{Y_1, \dots, Y_n\}$ and $S_I \subseteq \Sigma_{\mathbb{S}\mathbb{G}}$ with $|S_Y| + |S_I| \leq t$, the joint leakage of (S_Y, S_I) is simulatable from the public inputs and at most $|S_I|$ input shares.

If each gadget in a circuit is *t-SNI*, then any adversarial choice of at most t probes across the whole circuit can be *propagated* back to the circuit inputs without exceeding the probing threshold (typically $t = n - 1$). In the RPM, however, the leakage is a *random* set of wires and may contain far more than t probes in a single execution. As a result, TPM-style probe propagation does not directly yield tight RPM composition bounds, and several alternatives have been proposed.

Propagation approach. Belaïd et al. [BCP⁺20] proposed an RPM composition rule based on per-gadget simulation with a small failure probability: roughly, if for each gadget $\mathbb{S}\mathbb{G}$ one can simulate its RPM leakage together with up to $t_{\text{out}} = t$ additional output probes using only $t_{\text{in}} = t$ input shares, except with probability at most ϵ , then for a circuit $\mathbb{S}\mathbb{C}$ composed of such gadgets the overall failure probability is at most $\min\{1, |\mathbb{C}|\epsilon\}$. Cassiers et al. [CFOS21] refined this idea by tracking *which* input-share subsets are needed to simulate *which* output-share subsets, via a lookup table over all combinations; this becomes costly as n grows. Belaïd et al. [BRR25] instead track only the *number* of required input shares by computing an *envelope* $\mathcal{E}(n, \epsilon, t_{\text{out}})$ describing the distribution of t_{in} under $\text{RPM}(\epsilon)$. These envelopes are propagated backward from the circuit outputs toward the inputs and combined according to the circuit topology. While powerful (with further refinements in [BNR25]), such envelopes are typically obtained numerically, which makes closed-form and asymptotic analyses challenging.

Localization approach. Rather than propagating leakage to the circuit inputs, we proposed to *localize* leakage by inserting refresh gadgets between computational gadgets. This allows us to replace RPM leakage $\mathcal{L} = \{\mathcal{L}_1, \dots, \mathcal{L}_m\}$ sampled at rate ϵ by leakage $\mathcal{L}' = \{\mathcal{L}'_1, \dots, \mathcal{L}'_m\}$ sampled at a larger rate $\epsilon' > \epsilon$, such that the blocks $\mathcal{L}'_i$ become conditionally independent given their corresponding natives. In this section, we revisit this approach in order (i) to refine the resulting inflation bound via the bridging viewpoint, and (ii) to derive analytic bounds for the resulting composition parameters using the leakage-diagram technique, which previously we estimated only via Monte Carlo.

5.5.1 On the Role of Refresh Gadgets

Let $\mathbf{X}$ and $\mathbf{Y}$ denote the input and output n -sharings of a (linear) refresh gadget $\mathbb{SR}$. If $\mathbb{SR}$ is *leak-free*, i.e., $\mathcal{L}_{\mathbb{SR}} = \emptyset$, then the refresh randomness makes the output sharing fresh: conditioned on the underlying native $X = \sum_{i=1}^n X_i = \sum_{i=1}^n Y_i$, the input and output sharings are independent. Concretely, for any realizations $\mathbf{x}, \mathbf{y}$,

$$\Pr(\mathbf{Y} = \mathbf{y} \mid X, \mathbf{X} = \mathbf{x}) = \Pr(\mathbf{Y} = \mathbf{y} \mid X).$$

Consequently, in a circuit where gadgets are separated by leak-free refreshes, the circuit-wide leakage partitions into independent per-gadget blocks, as exploited in [PR13].

In the presence of leakage $\mathcal{L}_{\mathbb{SR}} \neq \emptyset$, $\mathbf{X}$ and $\mathbf{Y}$ may become dependent. Intuitively, leaked internal wires allow the adversary to eliminate internal variables and derive additional linear relations linking boundary shares. We discussed in Chapter 3 that for each leakage instance one can describe the induced linear constraints as a full-rank system

$$\begin{cases} \mathbf{P}_{\text{Inf}}(\mathbf{X}, \mathbf{Y}) = \mathbf{0}, \\ \mathbf{P}_{\text{Non-Inf}}(\mathbf{X}, \mathbf{Y}, \Sigma_{\mathbb{SR}}) = \mathbf{0}, \end{cases}$$

where $\mathbf{P}_{\text{Non-Inf}}$ is independent of the native X and does not affect the joint distribution of $(\mathbf{X}, \mathbf{Y})$ beyond correctness, while $\mathbf{P}_{\text{Inf}}$ appears only due to leakage and can introduce additional constraints involving boundary shares. Whenever such a constraint involves both $\mathbf{X}$ and $\mathbf{Y}$ (and no internal intermediates), it creates a dependency between input and output sharings.

Definition 5.7 (Bridge). In a refresh gadget SR with boundary sharings $\mathbf{X} = \{X_1, \dots, X_n\}$ and $\mathbf{Y} = \{Y_1, \dots, Y_n\}$, a *bridge* is a nontrivial $\mathbb{F}_q$ -linear relation

$$\sum_{i=1}^n a_i X_i + \sum_{i=1}^n b_i Y_i = c$$

that involves no internal intermediates, and is *essentially cross-boundary* in the following sense: there is no $\lambda \in \mathbb{F}_q$ such that, after adding $\lambda(\sum_i X_i - \sum_i Y_i) = 0$, the relation becomes input-only or output-only.

Without leakage, we do not expect any bridge connecting $\mathbf{X}$ and $\mathbf{Y}$. Leakage can *eliminate* intermediates and thereby *create* bridges. Bridges are precisely what enables leakage to propagate across refresh boundaries: from the output of SR to its input (and vice versa).

In comparison to our approach in Section 3.3, we have made a small refinement: we define bridges such that they should contain shares from *both* sides of the refresh gadget. These bridges are the source of dependency between input/output shares.

Example 5.4 (How leakage creates a bridge). Suppose $Y_i \leftarrow X_i + R_0$ for some internal intermediate R_0 . If $R_0 \in \mathcal{L}$, then the relation $Y_i - X_i = R_0$ becomes a boundary-only linear relation over (X_i, Y_i) with a *known* right-hand side—hence a bridge—occurring with probability ϵ . If instead $Y_i \leftarrow (X_i + R_0) + R_1$, then the same bridge requires $R_0, R_1 \in \mathcal{L}$ and occurs with probability ϵ^2 . $\square$

SNI refresh and bridges. Given the central role of SNI for TPM composition, one might expect a similar relevance in the RPM [BC25]. We refute this intuition with two refresh gadgets.

The SR-Random gadget (Alg. 15) [BRR25] is $(n-1)$ -SNI for sufficiently large randomness parameter λ , yet it admits bridges with high probability. Indeed, it contains boundary assignments of the form

$$Y_i = X_i + M_i,$$

where M_i is an internal intermediate. If M_i leaks, then the relation $Y_i - X_i = M_i$ becomes a bridge. For a fixed i this occurs with probability ϵ , and for some $i \in \{1, \dots, n\}$ it occurs with probability $1 - (1 - \epsilon)^n \approx n\epsilon$.

Algorithm 15 SR-Random

Input: $\mathbf{X} = \{X_1, \dots, X_n\}, \lambda$
Output: $\mathbf{Y} = \{Y_1, \dots, Y_n\}$

```

1: for  $i = 1$  to  $n$  do
2:    $M_i = 0$ 
3: for  $t = 1$  to  $\lambda$  do
4:    $R \xleftarrow{\$} \mathbb{F}_q, i, j \xleftarrow{\$} \{1, 2, \dots, n\}$ 
5:    $M_i \leftarrow M_i + R$ 
6:    $M_j \leftarrow M_j - R$ 
7: for  $i = 1$  to  $n$  do
8:    $Y_i = X_i + M_i$ 
9: return  $\mathbf{Y}$ 

```

In contrast, SR-multi with $k = n$ is also $(n-1)$ -SNI [CGPZ16], but we will prove that its bridge probability is negligible in n for the leakage rates of interest.

5.5.2 Composition Rule in RPM

A refresh boundary breaks the dependence between adjacent gadgets only if leakage from the refresh does not create *bridges*. When bridges do occur, a convenient reduction is to *expose* the boundary shares that participate in bridges. This removes bridges at the cost of giving additional information to the adversary.

Bridge-share rate. Let $\mathcal{L}_{\text{SR}}$ be the $\text{RPM}(\epsilon)$ leakage from SR. We define $\text{BridgeShares}(\mathcal{L}_{\text{SR}}) \subseteq \mathbf{X} \cup \mathbf{Y}$ as the set of boundary shares that appear in at least one essential bridge implied by $\mathcal{L}_{\text{SR}}$. If SR is *symmetric*, each boundary share has the same inclusion probability; we define

$$\beta(n, \epsilon) \triangleq \Pr[U \in \text{BridgeShares}(\mathcal{L}_{\text{SR}})],$$

for any fixed boundary share $U \in \mathbf{X} \cup \mathbf{Y}$.

Comparison with Section 3.3. In our proof approach in Section 3.3, we let the adversary to learn all the variables of $\mathbf{P}_{\text{Inf}}(\mathbf{X}, \mathbf{Y})$ subsystem. However, a closer look reveals that only essentially cross boundary equations (i.e., bridges) are making dependence. These are subset of all equations in $\mathbf{P}_{\text{Inf}}(\mathbf{X}, \mathbf{Y})$. This explains difference between $\beta(n, \epsilon)$ here as contrasted to $\frac{\mathbf{T}(n, \epsilon)}{n}$ in Section 3.3.

Theorem 5.8 (Refinement of Theorem 3.7). *Consider a linear refresh gadget SR carrying the native X with boundary sharings $\mathbf{X}$ and $\mathbf{Y}$, under $\text{RPM}(\epsilon)$.*

There is a reduction to an ideal leak-free refresh such that the original leakage is dominated (for monotone distinguishers) by the following leakage:

1. an erasure view of the native, i.e. $\phi_{E_X(n,\epsilon)}(X)$, where $E_X(n,\epsilon)$ is as in Section 5.4;
2. independent leakage of each boundary share in $\mathbf{X} \cup \mathbf{Y}$ with inflated rate

$$\epsilon' = 1 - (1 - \epsilon)(1 - \beta(n, \epsilon)) = \epsilon + \beta(n, \epsilon) - \epsilon\beta(n, \epsilon).$$

We estimate $\beta(n, \epsilon)$ by sampling leakage instances and identifying which boundary shares are implicated by *essential* bridges. Our LAPSE tool combines Monte Carlo with importance sampling for this task.

For the SR-multi gadget we additionally derive an analytic upper bound on $\beta(n, \epsilon)$. In particular, for the setting $k = n$ the bound is negligible as a function of n for $\epsilon \leq 1/9$.

Rule-of-thumb formulas. For $k = 1$, the dominant bridge pattern is $Y_i = X_i + R_i$, hence a boundary share participates only when R_i leaks while the two boundary shares themselves do not; this gives

$$\beta(n, \epsilon) = \epsilon(1 - \epsilon)^2 \leq \epsilon.$$

For $k = 2$, the dominant pattern is $Y_i = X_i + R_i + R'_i$, suggesting the rough scaling $\beta(n, \epsilon) \approx \epsilon^2$. For larger k there are more bridge patterns, but empirically small k (often $k = 2$) already suffices in practical designs.

5.5.3 Leakage-Diagram View

In this section, we show that bridge event also can be expressed as a connectivity problem of the associated leakage diagram of the refresh gadget.

Lemma 5.9 (Essential bridges $\Leftrightarrow$ top–bottom interior connectivity). *Consider SR-multi with parameters (n, k) and its leakage-diagram. Let $v_{i,j}$ denote the vertex in column $i \in \{0, \dots, n\}$ and row $j \in \{0, \dots, k\}$, and define the potential at this vertex as*

$$V_{i,j} \triangleq \sum_{t=1}^i X_t^{j+1}.$$

(So $V_{0,j} = 0$ and $V_{n,j} = X$ for all j .)

Let $G(\mathcal{L})$ be the subgraph containing exactly the interior edges whose corresponding wires leak (X -horizontal or C -vertical edges), together with the always-present boundary-column edges in columns 0 and n .

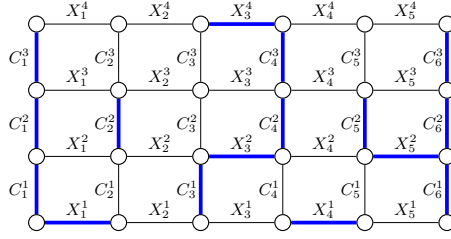


Figure 5.4: An example of leakage-diagram with $(n = 5, k = 3)$, with leaking and known wires specified in blue. There is a path connecting bottom and top rows, making a bridge. The parity equation of this bridge is $(X_2^1) \oplus (X_1^4 \oplus X_2^4) = c$, where c is known by the given leakage

Then a leakage instance $\mathcal{L}$ induces an essential bridge between the bottom sharing $\mathbf{X}^1$ and the top sharing $\mathbf{X}^{k+1}$ iff there exist interior columns $a, b \in \{1, \dots, n-1\}$ such that

$v_{a,0}$ is connected to $v_{b,k}$ by a path in $G(\mathcal{L})$ that never visits columns 0 or n .

Proof. Each leaked edge $e = (u, v)$ reveals a linear equation of the form $V(v) - V(u) = c_e$ for a known constant c_e (coming from the leaked X - or C -wire label). Therefore, within any connected component of $G(\mathcal{L})$, all pairwise differences $V(v) - V(u)$ are determined (as the signed sum of edge labels along any path), whereas for vertices in different components no such difference is determined.

If there is an interior path from some $v_{a,0}$ to some $v_{b,k}$, then $\mathcal{L}$ determines $V_{b,k} - V_{a,0}$, i.e.,

$$\sum_{t \leq b} X_t^{k+1} - \sum_{t \leq a} X_t^1 = c,$$

which is a boundary-only cross-round relation, hence a bridge; since $a, b \in \{1, \dots, n-1\}$ and the path avoids columns 0, n , this relation cannot be reduced to input-only or output-only by adding a multiple of correctness, hence it is essential.

Conversely, any essential bridge yields a boundary-only relation that (after fixing a representative modulo correctness) has the form $V_{b,k} - V_{a,0} = c$ with $a, b \in \{1, \dots, n-1\}$. Thus $v_{a,0}$ and $v_{b,k}$ must lie in the same connected component of $G(\mathcal{L})$, so there is a connecting interior path. $\square$

Lemma 5.10 (Vertical interior connectivity bound). *In the leakage-diagram of SR-multi with parameters (n, k) , assume all interior edges are present independently with probability ϵ_{LD} (boundary-column edges are irrelevant here). Let $\text{VConn}_{n,k}$ be the event that there exist $a, b \in \{1, \dots, n-1\}$ such that $v_{a,0}$ is connected to $v_{b,k}$ by a path that stays in interior columns $\{1, \dots, n-1\}$. Then, for $3\epsilon_{\text{LD}} < 1$,*

$$\Pr[\text{VConn}_{n,k}] \leq (n-1) \frac{(3\epsilon_{\text{LD}})^k}{1 - 3\epsilon_{\text{LD}}} = (n-1) \frac{(3\sqrt{\epsilon})^k}{1 - 3\sqrt{\epsilon}}.$$

Proof. Same union-bound over self-avoiding walks as Lemma 5.4.

Corollary 5.11. *Under the coupling of Lemma 5.2 (with $\epsilon_{\text{LD}} = \sqrt{\epsilon}$), bridges imply $\text{VConn}_{n,k}$ (Lemma 5.9), hence $\beta(n, \epsilon) \leq \Pr[\text{VConn}_{n,k}]$.*

5.6 LAPSE

We develop LAPSE (Linear Algebraic Probing Security Estimator), an automated analysis tool for evaluating $\mathbf{E}_X(n, \epsilon)$ and $\beta(n, \epsilon)$ for *linear* masked circuits over arbitrary finite fields. Depending on the circuit size and the probability regime, LAPSE provides both exact evaluation (by enumeration) and approximate estimation (via Monte Carlo and importance sampling).

5.6.1 Domain-Specific Language

LAPSE provides a domain-specific language (DSL) for describing linear circuits over finite fields. It supports the following field families:

- $\text{GF}(2)$: the binary field, typical for Boolean masking;
- $\text{GF}(p)$: prime fields used in arithmetic masking (e.g., $\text{GF}(257)$);
- $\text{GF}(2^u)$: extension fields specified by an irreducible polynomial (e.g., the AES field $\text{GF}(2^8)$ with modulus $x^8 + x^4 + x^3 + x + 1$).

The DSL supports circuit parameters, array declarations with role annotations (`input`, `output`, `rand`, `temp`, `const`), and simple control flow via parameterized loops. Only linear operations are permitted: field addition/subtraction, `xor` in characteristic 2, and scalar multiplication where one operand is a compile-time constant.

Example. The snippet below specifies the SR-SNI refresh gadget from Algorithm 6.

```

circuit SR_SNI(n);
input  bit A[n];
rand   bit R;
output bit B[n];

for i in [0 .. n) {
  for j in [i+1 .. n) {
    R = randbit();
    A[i] = A[i] xor R;
    A[j] = A[j] xor R;
  }
}

for i in [0 .. n) {
  B[i] = A[i];
}

const bit X;
X = sum_reduce(A, 0, n);

```

5.6.2 Internal Representation

LAPSE compiles a DSL program into a single-static-assignment (SSA) intermediate representation. Each assignment introduces a fresh SSA variable, making every wire value immutable and explicitly indexed. Since the DSL is linear, every SSA wire can be represented as a linear combination of *sources*, namely the circuit inputs and the fresh randomness sampled during execution.

Concretely, let T denote the number of sources (input-share variables and randomness variables), and let $m(n)$ denote the number of *leakable* wires in the SSA/DAG representation (typically including inputs, intermediates, and outputs; the exact convention is configurable and follows the leakage model). LAPSE constructs a matrix

$$\mathbf{A} \in \mathbb{F}^{m(n) \times T},$$

where $\mathbb{F}$ is the specified finite field, such that the i th row $\mathbf{A}_i$ encodes the linear expression for the i th leakable wire as a combination of the T sources. The target secret (or target linear expression) X is represented as a vector

$$\mathbf{x} \in \mathbb{F}^{1 \times T},$$

expressed in the same source basis. This representation reduces leakage questions to linear-algebraic span tests over $\mathbb{F}$.

5.6.3 Security Parameters

Evaluating $E_X(n, \epsilon)$. Given a leakage rate $\epsilon \in [0, 1]$, the random probing model selects each leakable wire independently with probability ϵ . Let $\mathcal{L} \subseteq \{1, \dots, m(n)\}$ be the random index set of leaked wires. LAPSE evaluates

$$E_X(n, \epsilon) = \Pr_{\mathcal{L} \leftarrow \text{RPM}(\epsilon)} [\mathbf{x} \in \text{span}(\{\mathbf{A}_i : i \in \mathcal{L}\})]. \quad (5.2)$$

For small circuits (by default $m(n) \leq M_{\max}$ with $M_{\max} = 26$), the tool computes (5.2) exactly by enumerating all $2^{m(n)}$ leakage patterns. For larger $m(n)$, it estimates $E_X(n, \epsilon)$ using Monte Carlo (MC) sampling or importance sampling (IS), and reports variance estimates and confidence intervals.

Expected essential bridges $\beta(n, \epsilon)$. In addition to the single-target event of (5.2), the tool can estimate the expected number of *essential bridges*: additional linear relations between the input sharing and output sharing that become derivable from leakage, excluding the trivial conservation relation (e.g., $\sum_i X_i = \sum_i Y_i$ for refresh gadgets). Formally, $\beta(n, \epsilon)$ counts (in expectation over $\mathcal{L}$) the dimension gain in the space of input–output relations induced by the leaked rows.

5.6.4 Estimation Methods

LAPSE implements multiple strategies and selects among them based on $m(n)$ and the observed rarity of the event.

Exact enumeration. If $m(n) \leq M_{\max}$, the tool enumerates all leakage patterns $\mathcal{L} \subseteq \{1, \dots, m(n)\}$ and computes $E_X(n, \epsilon)$ (or $\beta(n, \epsilon)$) exactly via a weighted sum under the RPM distribution.

Monte Carlo and importance sampling. For larger $m(n)$, the tool starts with MC sampling. If the number of “hits” (samples with $\mathbf{x} \in \text{span}(\{\mathbf{A}_i : i \in \mathcal{L}\})$) is too small (e.g., fewer than 10 hits in the initial budget), the tool treats the event as *rare* and switches to importance sampling. The tilted sampling rate $\tilde{\epsilon}$ is chosen by a short pilot run so that the hit rate under the tilted measure lies in a stable range, thereby controlling the variance of the likelihood-ratio weights.

5.6.5 Typical Usage

A typical workflow is as follows. One specifies a circuit with parameter n (the sharing order), chooses a set of n values and a grid of leakage rates ϵ , and requests $E_X(n, \epsilon)$ and/or $\beta(n, \epsilon)$. The tool returns an exact value (when enumeration is feasible) or an estimate with confidence information for each (n, ϵ) pair.

Chapter 6

Probing Secure Composability Without Online Randomness: Theory and Application to Ascon

This chapter presents our results on *deterministic masking* and on probing-secure composition *without* online (fresh) randomness, as published at CHES 2025 [JMB25b]. The technical core follows the paper, while the introduction, motivation, and exposition have been expanded to better match the style of this thesis and to connect to Chapter 7, which continues the study of deterministic masking from a complementary perspective.

Masking is the standard countermeasure against side-channel attacks, but higher-order protections are often bottlenecked by the cost of generating high-quality online randomness during execution. We investigate the extent to which this online randomness can be removed: the protected circuit should consume only the *offline* randomness used to create the initial n -sharing of the sensitive inputs.

Our main observation is structural. For round-based permutations with a bricklayer design, a linear diffusion layer can act as an *implicit refresh*: it mixes the shares across many parallel S-boxes and can restore the independence properties needed for secure composition up to a threshold number of probes. Building on this observation, we derive composability results in the probing model and instantiate them for ASCON using known deterministic first- and second-order masked S-boxes. In Chapter 7, we complement this instantiation by designing a third-order deterministic masked S-box for ASCON.

6.1 Introduction

Side-channel attacks and masking. Side-channel leakage can undermine an otherwise secure cryptographic design by revealing information about inter-

mediate values during execution. Masking counters this threat by representing each sensitive variable as an n -sharing and performing the entire computation on shares. Concretely, sensitive inputs (e.g., the key- and nonce-derived state) are first encoded using *offline* randomness. During computation, the unprotected operations (e.g., AND and XOR) are replaced by small *gadgets* that map n -shared inputs to n -shared outputs while ensuring probing security in the presence of a bounded number of observations.

Randomness as a bottleneck. Most practical higher-order gadgets rely on additional *online* randomness, consumed during execution, to prevent statistical dependencies from accumulating when gadgets are composed. Producing this randomness securely and cheaply is non-trivial in practice—especially in the presence of physical leakage—and can dominate the cost of a masked implementation [CGZ20, CMM⁺24]. This has motivated a long line of work on reducing randomness complexity [BBP⁺16, FPS17, SM21]. The problem is particularly pronounced for lightweight ciphers such as the NIST-standardized ASCON [TMC⁺24], whose target platforms are highly resource constrained.

Deterministic masking at circuit level. In this chapter we ask whether online randomness can be eliminated entirely. Deterministic probing-secure gadgets are known [NRR06, PAB⁺23, DDE⁺20, SM21], but composing them securely at the circuit level is subtle: even if each gadget is secure in isolation, the overall design can lose security without explicit refreshing [Rep15, CPRR14]. Standard composition methodologies therefore insert refresh gadgets [BBD⁺16], which inherently consume online randomness.

We show that for an important class of designs—round-based permutations with a bricklayer structure [DR20]—the linear diffusion layer can play the role of a refresh subcircuit. Intuitively, diffusion mixes the shares produced by many parallel S-boxes and thereby redistributes the offline randomness across the state. This observation underpins the composability theorems developed in this chapter and enables deterministic masked implementations of ASCON.

Bricklayer design of permutations [DR20]. A *public, unkeyed* (i.e., fixed and known) cryptographic permutation is the core of many modern primitives, including Keccak-f used in SHA-3 [BDPV13, oST15], ASCON-p used in ASCON and ISAP [TMC⁺24, DEM⁺20], and XODD00 used in XOODYAK [DHP⁺20]. Such permutations typically iterate a round function consisting of a parallel S-box (confusion) layer followed by a linear diffusion layer that mixes bits across the state [DR20].

Figure 6.1 sketches the deterministic masking strategy we study: each S-box is implemented by a deterministic masked gadget $\mathbb{S}\mathbb{S}$ -box, and the masked diffusion layer $\mathbb{S}\mathbb{D}$ iffusion linearly recombines sharings before the next round.

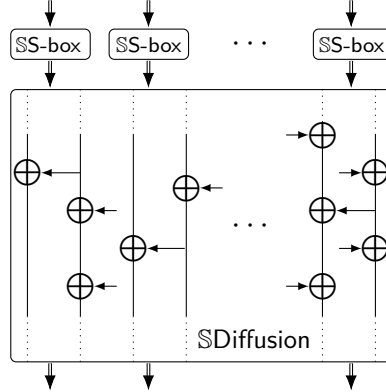


Figure 6.1: Deterministic masking in a bricklayer architecture. Without explicit refresh gadgets, the masked diffusion layer linearly mixes the output sharings of the S-box layer, providing an implicit refresh before the next round.

Related works. A commonly referenced software-oriented masking scheme for ASCON¹ is deterministic. However, its security has only been verified for a single round [GMSP23, DDE⁺20], leaving it unclear whether this scheme can maintain its security order when applied to multiple rounds.

At the gadget level, Nikova et al. [NRR06] introduced the Threshold Implementation (TI) approach, which requires more shares but operates deterministically. Their work also presented a first-order probing-secure deterministic multiplication gadget with $n = 4$ shares. Building on this foundation, several deterministic masking schemes for S-boxes have been proposed [BNN⁺12, MPL⁺11, PAB⁺23, SM21]. Bilgin et al. [BGN⁺14] extended the TI approach to higher-order security; however, at higher orders, TI-based designs typically require some randomness [DBR⁺15].

At the circuit level, Reparaz [Rep15] demonstrated that gate-level TI does not necessarily guarantee circuit-level security. For higher-order TI, Reparaz et al. [RBN⁺15] proposed adding a refresh layer to separate rounds, although

¹<https://github.com/ascon/simpleserial-ascon>

Moos et al. [MMSS19] showed that this can be problematic without a supporting security proof. Moreover, inserting a refresh layer inherently introduces randomness, rendering the masking approach nondeterministic.

In pursuit of higher-order, circuit-level deterministic masking, Beyne et al. [BDZ20] proposed using linear cryptanalysis—specifically, correlation matrices—to bound the correlation between two probes in different rounds. A low correlation implies that probes are (approximately) independent and uniformly distributed. This approach was further developed in [BDMS22]. In contrast, our work also leverages cryptographic structure, but focuses more narrowly on the role of the diffusion layer. Our method is conceptually simpler and avoids correlation analysis. Additionally, the approach in [BDZ20, BDMS22] is limited to bounding leakage from two probes and relies on assumptions regarding independence of round transformations due to key additions. In ASCON, which we study in this work, these transformations are public and do not involve round keys.

6.1.1 Our Contributions

This work advances both the theoretical foundations and practical deployment of deterministic masking schemes. Our main contributions are as follows:

- **Diffusion as implicit refresh.** We show that in bricklayer cipher structures, the diffusion layer can effectively act as a refresh mechanism by combining the offline randomness embedded within each gadget. This hinders adversaries from exploiting probes across different rounds, thus enabling secure composition without explicit refresh gadgets.
- **Limitations of simulation-based proof techniques.** We show that simulation-based frameworks and probe-propagation techniques [ISW03, BBD⁺16, CS20] inherently rely on fresh randomness, making them inadequate for verifying deterministic masking. To address this, we propose concrete refinements that allow applying simulation-based reasoning in a deterministic context.
- **Composability at first order.** We formally characterize the conditions under which first-order secure gadgets compose securely. This includes a precise definition of independence for n -shared values, without requiring assumptions about the underlying secrets. We apply this result to round-based cipher constructions.
- **Higher-order security via diffusion.** We show that probing the inputs and outputs of a diffusion layer preserves independence up to the

layer’s *linear branch number* [DR20]. This enables a provable composition of two rounds at higher orders. For more than two rounds, we provide a heuristic argument based on cryptographic properties of diffusion.

- **Implementation and case study on Ascon.** We implement our approach² on the NIST-standardized cipher ASCON [TMC⁺24], demonstrating both first- and second-order probing security under deterministic masking. The S-box gadgets used are based on [SM21]. For the second-order case, we observe a structural separation: while the native secrets evolve nonlinearly across rounds, the shares undergo only linear transformations. We use this to provide a probabilistic argument for second-order probing security across multiple rounds.

Scope and limitations. Our results are proved in the *standard probing model*. While this abstraction is widely used to establish feasibility and guide constructions, it does not directly capture certain physical effects that arise in practice. Several extensions of the probing model address, for instance, *glitch* and *transition* leakage [FGMDP⁺18, CS21]. Glitch leakage can often be mitigated by inserting *synchronization* layers, and transition leakage can be reduced by resetting or re-initializing *registers*; importantly, these countermeasures do not inherently require fresh randomness. Additional concerns include *micro-architectural* effects [DNR19] and *coupling* leakages [DBG⁺17], which are typically addressed by combining masking with higher security orders and device-specific protections.

In this chapter, our objective is to establish the *feasibility* of higher-order deterministic masking and to develop a composability theory under a clean and standard abstraction. Accordingly, following Ishai et al. [ISW03], we assume that the adversary does not probe during the initial encoding of secrets into shares, nor during the final decoding step. We return to the above practical considerations—and to what is required to obtain an implementation-level security claim in the case study of ASCON—in Section 6.6.

6.2 Deterministic Masking

We use the term *deterministic masking* to refer to any gadget or protected circuit (SC) that operates *without* consuming online (fresh) randomness. Early

²<https://github.com/vahid-jahandideh/deterministic-masking-ascon.git>

work by Nikova et al. [NRR06] identified a first-order probing-secure deterministic multiplication gadget with $n = 4$ shares, establishing the concept of Threshold Implementation (TI) for gadget-level protection. Many subsequent works have adopted TI-based techniques, which often yield deterministic masking.

However, extending from a deterministic gadget to a fully deterministic masked circuit is more challenging. Primarily because the *probe propagation* framework (see Section 6.3), which is a key approach for proving the security of compositions, relies on the use of online randomness, particularly through refresh gadgets. The main contribution of this chapter is to lay the groundwork for an alternative composition strategy that achieves probing security *without* the need for online randomness.

6.2.1 Composition Problem

Protecting a circuit requires more consideration than simply designing protected gates and combining them. It is a well-known issue that trivially composing d -probing secure gadgets can leave them vulnerable to fewer than d probes. This concern has been addressed in several works [CPRR14, Rep15, BBD⁺16]. However, since the composition challenge is central to our discussion, we pause to clarify it further with an example.

Example 6.1. Suppose a 3-bit S-box is formed by combining χ_3 and a linear layer. Given input bits $\{A, B, C\}$, χ_3 computes the output bits as:

$$\begin{cases} A' = A \oplus (1 \oplus B)C, \\ B' = B \oplus (1 \oplus C)A, \\ C' = C \oplus (1 \oplus A)B. \end{cases}$$

The subsequent linear layer computes $A'' = A' \oplus B'$, $B'' = B'$, and $C'' = 1 \oplus C'$. For this S-box, we propose the following SS-box at $n = 2$, where $\mathbb{S}\chi_3$ operates as:

$$\begin{cases} A'_1 = [A_1 \oplus (1 \oplus B_1)C_1] \oplus B_2C_1, & A'_2 = [A_2 \oplus (1 \oplus B_1)C_2] \oplus B_2C_2, \\ B'_1 = [B_1 \oplus (1 \oplus C_1)A_1] \oplus C_2A_1, & B'_2 = [B_2 \oplus (1 \oplus C_1)A_2] \oplus C_2A_2, \\ C'_1 = [C_1 \oplus (1 \oplus A_1)B_1] \oplus A_2B_1, & C'_2 = [C_2 \oplus (1 \oplus A_1)B_2] \oplus A_2B_2. \end{cases} \quad (6.1)$$

The masked linear layer works as:

$$\begin{aligned} A''_1 &= A'_1 \oplus B'_1, & A''_2 &= A'_2 \oplus B'_2, & B''_1 &= B'_1, \\ B''_2 &= B'_2, & C''_1 &= 1 \oplus C'_1, & C''_2 &= C'_2. \end{aligned}$$

This **SS-box** is first-order probing secure. The security claim can be directly proved by showing that each intermediate V is independent of the secrets (A, B, C) . That is, $\Pr(V \mid A, B, C) = \Pr(V)$. Verifying this involves enumerating all possible inputs, yielding $2^{3n} = 64$ inputs in this example.³

Now, consider a 3-bit permutation $f^{(r)}$, constructed by applying **S-box** r times, i.e.,

$$f^{(r)}(A, B, C) = \text{S-box}(\dots(\text{S-box}(A, B, C))\dots).$$

We show that trivially composing the given **SS-box** does not result in a secure computation of $\text{Sf}^{(r)}(A_1, A_2, B_1, B_2, C_1, C_2)$ for $r > 1$. For example, by enumerating all input shares, for A_1'' at the output of $\text{Sf}^{(16)}$, we find:

$$\Pr(A_1'' = 0 \mid A = 0, B = 0, C = 1) = 0, \quad \text{whereas} \quad \Pr(A_1'' = 0) = \frac{1}{2}.$$

This shows that A_1'' is *dependent* on native variables, violating first-order probing security.

Although this **S-box** is bijective, the given **SS-box** is *not* a bijective gadget and has *collisions*. For instance, the inputs:

$$\begin{cases} (A_1 = 1, A_2 = 1, B_1 = 1, B_2 = 1, C_1 = 0, C_2 = 0), \\ (A_1 = 0, A_2 = 0, B_1 = 0, B_2 = 0, C_1 = 1, C_2 = 1), \end{cases}$$

map to the same output. Collisions reduce the output space, consume initial randomness, and make the intermediates of subsequent rounds secret-dependent. For $f^{(r)}$, collisions increase with r , as shown in Figure 6.2.

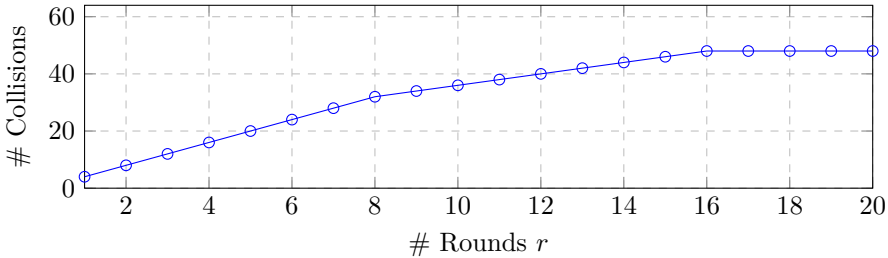


Figure 6.2: More collisions occur with increasing number of rounds.

³An intuitive explanation for the security is that each intermediate is *blinded* by separate shares of the inputs, as elaborated in the next section.

In the following, we briefly explain the fundamental cause of the security loss in this example:

The probing security of a gadget, such as a **SS-box**, is typically proven under the assumption that its inputs are both independent and uniformly distributed. However, if a collision occurs, the gadget's output distribution will no longer remain uniform. In this circuit, the output of one gadget is fed into the next. As soon as the inputs to any gadget deviate from uniformity, its probing security can no longer be guaranteed. $\square$

Role of online randomness. A generic solution to composition issues involves incorporating online randomness, particularly through the insertion of refresh gadgets. Online randomness compensates for entropy loss caused by collisions and blocks the *propagation* of adversary probes (detailed in the following section).

6.3 Simulation-Based Security and Its Limitations

Direct verification of d -probing security, which involves checking the independence of any d -tuple of intermediates from native variables, becomes impractical as the sharing order n increases. The *simulation* approach by Ishai et al. [ISW03] addresses this challenge. To prove d -probing security, one shows that the distribution of up to d probes (on intermediates) can be *simulated*, i.e., reproduced in an indistinguishable manner, using fewer than n shares of each input.

The standard simulation approach follows basic rules. Let V_1 and V_2 be intermediates (not necessarily shares of a secret), and let R represent an online randomness variable:

- If a probe P is of the form $P = R \oplus V_1$, and R is not involved in the computation of any other probe, then R *hides* V_1 . In this case, the distribution of P can be simulated using an independent random variable R' as $P = R'$, without requiring V_1 for the simulation.
- If a probe P is of the form $P = V_1 \oplus V_2$ or $P = V_1 V_2$, both V_1 and V_2 are required for the simulation.

Definition 6.1 (Probe Propagation). If simulating a probe P requires knowledge of some intermediate V , we say that P propagates to V , and V is called a *propagated probe*.

Probes always propagate toward inputs. The propagation rules are depicted in Figure 6.3. Notably, the addition of online randomness, as stated above, halts further propagation.

6.3.1 Composition Rules

The simulation rules are useful for proving the probing security of standalone gadgets. However, they are insufficient for verifying the security of compositions. Barthe et al. [BBD⁺16] introduced the concept of *strong non-interference* (SNI) to address this issue. A gadget is SNI if simulating up to a threshold number of output shares does not require any input shares. Thus, an SNI gadget acts as a barrier, stopping further propagation of probes. Cassiers and Standaert [CS20] later refined this concept by introducing *probe isolating non-interference* (PINI) gadgets. PINI gadgets are trivially composable and do not require additional refresh gadgets at interfaces for secure composition.

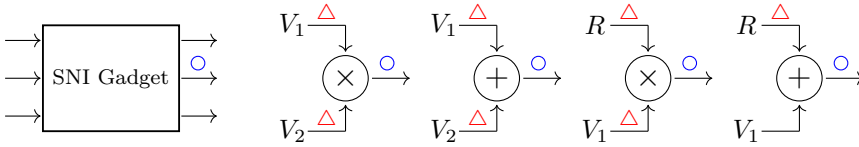


Figure 6.3: Probe propagation rules. Blue circles represent probes, and red triangles represent propagated probes. V_1 and V_2 are intermediates, and R is a randomness variable.

Simulation needs online randomness. The probe elimination technique, as well as the design of SNI and PINI gadgets, inherently requires randomness.⁴ This reliance on randomness limits their deployment for proving the probing security of deterministic masking schemes, where the use of online randomness is avoided. this chapter explores tweaks to handle the absence of online randomness. As a first step, we demonstrate that simulation-based security is more restrictive than the core definition of probing security.

⁴If no online randomness is involved, input shares *uniquely* determine each output share. Consequently, simulating (reproducing) output shares without using any of the input shares becomes impossible.

6.3.2 Limitations of Simulation-Based Security Proof

Given a set of probes $\mathbf{P}$ and a set of natives $\mathbf{V}$ (with any distribution), *probing security* requires that $\mathbf{P}$ conveys no information about $\mathbf{V}$, i.e.,

$$\Pr(\mathbf{P} \mid \mathbf{V}) = \Pr(\mathbf{P}).$$

Probing security addresses the equivocation about natives and, in some cases, is a broader concept than simulation of the probes. The following examples clarify this point.

A 2-input case. At $n = 2$, let $\{A_1, A_2\}$ and $\{B_1, B_2\}$ be shares of secrets A and B . For simulating a probe P of the form $P = A_1 \oplus B_1 B_2$, the introduced simulation rules require A_1 , B_1 , and B_2 . Specifically, since both shares of B are required, the approach suggests that P depends on B . However, if the shares of A and B are independent, direct enumeration over all $2^{2n} = 16$ possible inputs reveals that P is independent of A and B . In this case, A_1 hides the secret-dependent term $B_1 B_2$. $\square$

A 3-input case. Another example involves probing intermediates of $\mathbb{S}\chi_3$ as defined in Equation (6.1). For instance, a probe on A'_1 results in:

$$P = [A_1 \oplus (1 \oplus B_1)C_1] \oplus B_2 C_1.$$

Simulating P (based on the standard rules) requires the shares B_1 , B_2 , C_1 , and A_1 . Specifically, both shares of B are required, indicating that the standard simulation approach cannot prove probing security. However, the randomness brought in by A_1 hides the secret-dependent term:

$$(1 \oplus B_1)C_1 \oplus B_2 C_1 = (1 \oplus B)C_1. \quad \square$$

These examples illustrate that probing security is a broader notion than the standard simulation-based security. They also demonstrate how randomness introduced by the shares can be used to blind probes. However, this approach has an important prerequisite:

the shares of inputs must be independent.

Efficiency vs. generality. Barthe et al. [BBD⁺15b, Appendix E] also highlight the significant limitation of (standard) simulation-based approaches for probing security. On the one hand, simulation-based security enables more efficient verification by providing compositional rules that reduce probing security

of the entire circuit to manageable subproblems concerning its constituent gadgets. On the other hand, as our preceding examples illustrate, this approach may fail to prove probing security for certain gadgets.

These limitations of simulation-based probing security are particularly relevant when selecting tools for the formal verification of masking. For instance, the current version of `maskVerif` [BBC⁺19], which is simulation-based, produces false positives for the examples in this subsection. Meanwhile, `SILVER` [KSM20], which directly checks the statistical independence of probes from native variables, does confirm the probing security of the same examples.

To address this issue, Meyer et al. [MBR19] proposed using mutual information. In this chapter, we propose an alternative: we extend simulation rules by using the blinding effect of independent input shares. Our approach generalizes Coron's probe elimination technique [Cor18] to work in deterministic masking settings.

6.3.3 Probe Elimination

In a set of probes $\mathbf{P}$, let a probe P_i be of the form $P_i = R \oplus P'$, where P' is a function of intermediates other than R . If the randomness R is not involved in any other probe, then P_i can be *eliminated* from $\mathbf{P}$, and its distribution can be simulated using an independent randomness R' . In this case, R *hides* the intermediates in P' . We also say, R *blinds* the probe P_i . The probe elimination process can iteratively continue by updating $\mathbf{P} \leftarrow \mathbf{P} - P_i$. The remaining intermediates required to compute the final $\mathbf{P}$ are needed for the simulation. We clarify this approach with an example.

Example 6.2. Let the sharing order be $n = 2$, and let $\mathbf{A} = \{A_1, A_2\}$ be the shares of the native A . Assume the adversary has the following probes:

$$P_1 = R_1 \oplus A_1 A_2 \oplus R_2, \quad P_2 = R_2 \oplus A_1 \oplus A_2, \quad P_3 = A_1,$$

where R_1 and R_2 are randomness variables. Applying the probe elimination approach:

- P_1 is of the form $P_1 = R_1 \oplus P'$, and since R_1 is not involved in P_2 and P_3 , we eliminate P_1 , updating $\mathbf{P} = \{P_2, P_3\}$.
- In the second iteration, $P_2 = R_2 \oplus P'$, and R_2 is not involved in the computation of the current probes. We eliminate P_2 , updating $\mathbf{P} = \{P_3\}$.

The remaining probe $P_3 = A_1$ requires only one input share (A_1) for simulation. Thus, the adversary's probes can be simulated with knowledge of only one input share. $\square$

Proving probing security using probe elimination. To prove probing security of a circuit SC , we can use the probe elimination technique. Consider any set of d probes on intermediate values, each having the following form:

$$\mathbf{P} = \{P_1 = R_1 \oplus P'_1, \dots, P_d = R_d \oplus P'_d\},$$

where R_i , for $1 \leq i \leq d$, are randomness variables used only in the specific (explicitly shown) locations associated with each probe. Under these conditions, probe elimination ensures that all d probes can be “eliminated” after d iterations, thus proving security.

A slight modification generalizes the above scenario:

$$\mathbf{P} = \left\{ P_1 = \left(\bigoplus_{i \in \mathcal{I}_1} R_i \right) \oplus P'_1, \dots, P_d = \left(\bigoplus_{i \in \mathcal{I}_d} R_i \right) \oplus P'_d \right\},$$

with the assumption that each probe has a unique randomness variable used exclusively by that probe. Concretely, in each set $\mathcal{I}_{j^*}$, for $1 \leq j^* \leq d$, there is exactly one element that does not appear in

$$\bigcup_{j \neq j^*} \mathcal{I}_j.$$

Probe elimination applies in this setting and results in the full elimination of all probes.

Input shares as the source of randomness. We extend this principle to deterministic masking. In the absence of online randomness, we treat the input shares—generated using offline randomness—as the source of blinding.

Suppose the circuit C processes a state $S = \{X^1, X^2, \dots, X^b\}$. Each secret X^i is encoded into n shares $\mathbf{X}^i = \{X_{(n-1)i+1}, \dots, X_{ni}\}$, consuming a total of $(n-1)b$ independent randomness variables. Let $\{X_1, X_2, \dots, X_{nb}\}$ denote the full set of shares.

Now consider a probe of the form:

$$P = \left(\bigoplus_{i \in \mathcal{I}} X_i \right) \oplus F(S, \{X_i \mid i \in \mathcal{J}\}),$$

where $\mathcal{I} \cap \mathcal{J} = \emptyset$, and F is an arbitrary (possibly nonlinear) function of the inputs.

If there exists an index $i \in \mathcal{I}$ such that not all n shares of the corresponding secret $X^{\lceil i/n \rceil}$ appear in $\mathcal{I} \cup \mathcal{J}$, then X_i blinds the probe P . In that case, the

distribution of P becomes independent of the native secrets, and P can be safely eliminated.

This observation forms the basis of our analysis in Sections 6.5 and 6.6 for the case of $d > 1$, where we show that structural properties of the cipher ensure that such blinding holds across multiple compositions.

6.4 First-Order Deterministic Masking

Blinding using input shares and the non-completeness property (discussed in Section 2.9) are effective techniques for achieving probing security at the level of individual gadgets. However, these techniques alone do not guarantee the security of *deterministic masking* when composed across an entire circuit.

In this section, we focus on the simpler case of *first-order* probing security. In the context of Threshold Implementations (TI), Dhooghe et al. [DNR19] demonstrated that if all layers in a circuit are non-complete and uniform, then the composition remains secure at first order.

Related ideas have also been considered in the context of fault attacks. Notably, Daemen et al. [DDE⁺20] argue that when each gadget (e.g., masked S-box) is a permutation over the shares and is applied to uniformly shared inputs, the uniformity is preserved across rounds—ensuring first-order security without fresh randomness. While this aligns with the spirit of our approach, we revisit the argument more critically. In particular, we will provide a counterexample showing that the permutation property alone is not sufficient for guaranteeing probing security.

Building on these observations, we now formalize the conditions under which a deterministic masked circuit preserves first-order probing security:

Lemma 6.2. *Let n be the sharing order. A masked circuit composed entirely of first-order probing-secure gadgets is itself first-order probing secure if each gadget with m input variables receives m -jointly independent n -sharings at its inputs.*

Proof. The adversary is allowed to place a single probe, which necessarily targets an intermediate value within one of the gadgets. Since each gadget is assumed to be first-order probing secure, the probed value is independent of the secrets—*provided* that the gadget’s input sharings are m -jointly independent. Because this joint independence holds across the circuit, each gadget behaves as it would in isolation. Consequently, any single probe—regardless of placement—yields no information about the native secrets, ensuring first-order probing security of the entire circuit. $\square$

6.4.1 Independence of n -Sharings

A key element of Lemma 6.2 is the (joint) independence of the input n -sharings that feed into a gadget. In this subsection, we formalize this notion of independence. Recall that an n -sharing of a native variable X is denoted by $\mathbf{X} = \{X_1, \dots, X_n\}$, where $\mathbf{X}$ is uniformly distributed over all assignments satisfying $\bigoplus_{i=1}^n X_i = X$. As a result, any subset of $(n-1)$ shares in $\mathbf{X}$ also follows a uniform distribution. Note that this definition places no constraints on the distribution of X : it might be public (e.g., a nonce), fixed (e.g., an initialization vector), or secret. Similarly, the definition of independence for n -sharings does not depend on whether their underlying natives are secret or public.

More concretely, for an n -sharing $\mathbf{X}$,

$$\Pr(\mathbf{X} = \mathbf{x} \mid X = x) = \begin{cases} 0, & \text{if } \bigoplus_{i=1}^n x_i \neq x, \\ \frac{1}{2^{(n-1)u}}, & \text{otherwise,} \end{cases}$$

where u is the bit-length of the underlying field $\mathbb{F}_{2^u}$.

Two n -sharings $\mathbf{X}$ and $\mathbf{Y}$, corresponding to natives X and Y , are said to be *independent* if

$$\Pr((\mathbf{X}, \mathbf{Y}) \mid (X, Y)) = \Pr(\mathbf{X} \mid X) \Pr(\mathbf{Y} \mid Y).$$

This notion naturally extends to sets of more than two n -sharings:

Definition 6.3 (Independence of n -sharings). A collection of n -sharings as $\{\mathbf{X}_1, \dots, \mathbf{X}_l\}$ is *independent* if

$$\Pr((\mathbf{X}_1, \dots, \mathbf{X}_l) \mid (X_1, \dots, X_l)) = \prod_{i=1}^l \Pr(\mathbf{X}_i \mid X_i).$$

Equivalently, for any valid assignment of these n -sharings, i.e., $\{\mathbf{x}_1, \dots, \mathbf{x}_l\}$ summing to $\{x_1, \dots, x_l\}$ respectively, the probability is $\frac{1}{2^{(n-1)ul}}$, and 0 otherwise.

For example, the input and output of a refresh gadget form two independent n -sharings.⁵ Moreover, cascading a refresh gadget $l-1$ times yields l independent n -sharings.

⁵In the linear refresh gadget case, the only connection is the trivial relation $\bigoplus_{i=1}^n X_i = \bigoplus_{i=1}^n Y_i$.

It may happen that a set of l n -sharings is not fully independent, yet every subset of size $m \leq l$ is jointly independent. In this scenario, we say the n -sharings are *m -jointly independent*.

Definition 6.4 (*m -Joint Independence of n -sharings*). A collection of n -sharings $\{\mathbf{X}_1, \dots, \mathbf{X}_l\}$ is *m -jointly independent* if for any subset of indices $i_1, \dots, i_m$,

$$\Pr\left((\mathbf{X}_{i_1}, \dots, \mathbf{X}_{i_m}) \mid (X_{i_1}, \dots, X_{i_m})\right) = \prod_{j=1}^m \Pr(\mathbf{X}_{i_j} \mid X_{i_j}).$$

As an illustration, suppose $\mathbf{X}_1$ and $\mathbf{X}_2$ are independently sampled n -sharings, and let $\mathbf{X}_3 = \mathbf{X}_1 \oplus \mathbf{X}_2$, where $\oplus$ is applied share-wise. Then $\{\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3\}$ is 2-jointly independent.

The connection between a set of (m -jointly) independent n -sharings as $\{\mathbf{X}_1, \dots, \mathbf{X}_l\}$ and their respective natives can be “broken” by discarding one share. Specifically, if $\mathbf{X}^o \subset \mathbf{X}$ contains only $(n-1)$ shares, then $\Pr(\mathbf{X}^o \mid X) = \Pr(\mathbf{X}^o)$. From this observation, one can equivalently define a set of n -sharings $\{\mathbf{X}_1, \dots, \mathbf{X}_l\}$ as independent if

$$\Pr(\mathbf{X}_1^o, \dots, \mathbf{X}_l^o) = \prod_{i=1}^l \Pr(\mathbf{X}_i^o),$$

where each $\mathbf{X}_i^o \subset \mathbf{X}_i$ contains $(n-1)$ of the n shares in $\mathbf{X}_i$.

We next show that a masked version of any linear, bijective function preserves the independence of its input n -sharings:

Lemma 6.5. *Let F be a bijective function, and let $\mathbb{S}F$ be its masked counterpart. If the input to $\mathbb{S}F$ consists of l independent n -sharings $\{\mathbf{X}_1, \dots, \mathbf{X}_l\}$, then the output $(\mathbf{Y}_1, \dots, \mathbf{Y}_l) = \mathbb{S}F(\mathbf{X}_1, \dots, \mathbf{X}_l)$ is also a set of independent n -sharings.*

Proof. For any assignment of output shares $\{\mathbf{y}_1, \dots, \mathbf{y}_l\}$ with corresponding natives $\{y_1, \dots, y_l\}$, there is a unique preimage $\{\mathbf{x}_1, \dots, \mathbf{x}_l\} = \mathbb{S}F^{-1}(\mathbf{y}_1, \dots, \mathbf{y}_l)$ with natives $\{x_1, \dots, x_l\} = F^{-1}(y_1, \dots, y_l)$. Hence,

$$\begin{aligned} \Pr((\mathbf{Y}_1 = \mathbf{y}_1, \dots, \mathbf{Y}_l = \mathbf{y}_l) \mid (Y_1 = y_1, \dots, Y_l = y_l)) \\ &= \Pr((\mathbf{X}_1 = \mathbf{x}_1, \dots, \mathbf{X}_l = \mathbf{x}_l) \mid (X_1 = x_1, \dots, X_l = x_l)) \\ &= \frac{1}{2^{(n-1)ul}}, \end{aligned}$$

which, by Definition 6.3, confirms the independence of $\{\mathbf{Y}_1, \dots, \mathbf{Y}_l\}$. $\square$

Bijectivity vs. n -sharing. In our deterministic masking setup, a gadget's bijectivity is crucial for preserving the randomness of its inputs. However, bijectivity alone does not ensure that an n -shared input will yield an n -shared output. The following example illustrates why these two notions differ.

Example 6.3. Consider a gadget $\mathbb{S}G_i$ operating on an n -shared input $\mathbf{X}$. Suppose it produces the output

$$\mathbf{Y} = \{X_1 \oplus X_i, X_2, \dots, X_n\},$$

i.e., it modifies share X_1 by adding share X_i . Although this gadget is bijective—one can uniquely recover all input shares from the output shares—it does not preserve the n -sharing property. Hence, it compromises compositional security. Specifically, observe that

$$\mathbf{Y} = \mathbb{S}G_2\left(\mathbb{S}G_3\left(\dots\left(\mathbb{S}G_n(\mathbf{X})\right)\right)\right), \quad Y_1 = \bigoplus_{i=1}^n X_i = X,$$

which shows that applying $\mathbb{S}G_n, \mathbb{S}G_{n-1}, \dots, \mathbb{S}G_2$ in sequence removes all masks and reveals the secret. The flaw arises because the gadget's output is effectively an $(n-1)$ -sharing. $\square$

6.4.2 Lemma 6.2 in Bricklayer Designs

In this subsection, we investigate how Lemma 6.2 applies in the practical setting of a bricklayer design, where a b -bit state $S = S^1 \parallel \dots \parallel S^t$ is processed in t chunks, each of length m bits (so $b = t \times m$). A typical round-based operation on S proceeds as follows:

```

 $r = 1$ 
For ( $r \leq$  number of rounds)
   $S \leftarrow \text{S-box}(S^1) \parallel \dots \parallel \text{S-box}(S^t),$ 
   $S \leftarrow \text{Diffusion}(S),$ 
   $r \leftarrow r + 1.$ 

```

In the protected (masked) version, the b -bit state S is replaced by an (nb) -bit n -shared state $\mathbf{S}$, where each bit of S is independently shared. The **S-box** layer is replaced by **SS-box** gadgets, and the diffusion layer, being additive-

affine, is straightforward to mask. Hence, the protected circuit follows:

```

 $r = 1$ 
For ( $r \leq \text{number of rounds}$ )
   $\mathbf{S} \leftarrow \text{SS-box}(\mathbf{S}^1) \parallel \cdots \parallel \text{SS-box}(\mathbf{S}^t),$ 
   $\mathbf{S} \leftarrow \text{SDiffusion}(\mathbf{S}),$ 
   $r \leftarrow r + 1.$ 

```

To apply Lemma 6.2 in this context, we must verify that in each round of the bricklayer design, the inputs to the gadgets remain jointly independent n -sharings. Once this condition is satisfied, the entire circuit inherits first-order probing security by direct application of the lemma.

Lemma 6.6. *If SS-box is first-order probing secure, and on m independent n -shared inputs produces m independent n -shared outputs—and if the diffusion layer is bijective—then the resulting bricklayer design is first-order probing secure.*

Proof. By construction, the initial shared state forms b independent n -sharings, which feed into t separate SS-box gadgets. By the lemma’s assumption, these SS-box gadgets produce m independent n -sharings each, so overall they yield $b = t \times m$ independent n -sharings for the input to SDiffusion . Because Diffusion is invertible, Lemma 6.5 ensures it preserves the independence of its input n -sharings. Hence, at each gadget in the design, the input n -sharings meet the same security conditions as in the standalone scenario, and by Lemma 6.2, the entire design is first-order probing secure. $\square$

6.5 Higher-Order Probing Security in Round-Based Compositions

Achieving higher-order probing security (i.e., $d > 1$) for a round-based cipher with deterministic masking requires more than just the bijectivity of each round. To illustrate this, consider the serial composition of gadgets in Figure 6.4, where each gadget is second-order ($d = 2$) probing secure in isolation.

When instantiated with a second-order probing-secure SS-box , we observe that placing one probe at the input of the first gadget and a second at the output of the r th gadget (for large r) reveals statistical dependence on the underlying secrets.⁶ This demonstrates that simple serial composition—without

⁶In our specific experiment, we used the $\text{SM2-S}\chi_5$ gadget with $n = 3$ shares for the χ_5 component of ASCON. See Section 6.6.2.

an intervening diffusion layer—does *not* preserve second-order probing security.

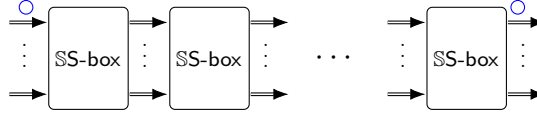


Figure 6.4: Serial composition of r gadgets. The blue circles indicate the probe locations in our numerical experiment.

Nevertheless, if each round transformation follows a bricklayer architecture, one can overcome this composition challenge by showing that the diffusion layer effectively behaves like a refresh gadget—up to a certain number of probes—and prevents an adversary from gaining any advantage by placing probes across different rounds. This observation calls for a closer examination of the diffusion layer’s structure.

6.5.1 Properties of the Diffusion Layer

A diffusion layer is an invertible linear transformation that operates on a b -bit state S . This transformation can be represented using matrix multiplication as $S_{1 \times b} \leftarrow S_{1 \times b} \mathbf{M}_{b \times b}$. The primary design goal of the diffusion layer is to combine state bits through the XOR operation. The cryptographic effectiveness of the diffusion layer is typically evaluated using two metrics: the linear and differential *branch numbers*, denoted respectively by $\mathcal{B}_l$ and $\mathcal{B}_d$.

Definition 6.7 (Branch Numbers [DR20]). The branch numbers $\mathcal{B}_l$ and $\mathcal{B}_d$ are defined as:

$$\mathcal{B}_l = \min_{S \neq 0^b} [\text{wt}(S) + \text{wt}(S\mathbf{M}^\top)], \quad \mathcal{B}_d = \min_{S \neq 0^b} [\text{wt}(S) + \text{wt}(S\mathbf{M})], \quad (6.2)$$

where $\mathbf{M}^\top$ is the transpose of $\mathbf{M}$, and wt computes the number of non-zero (active) elements in its input vector.

The metric $\mathcal{B}_d$ identifies the minimum Hamming weight of input/output pairs and specifies the least number of active output bits resulting from a single active input bit. Conversely, $\mathcal{B}_l$ determines the minimum Hamming weight of parity relations between inputs and outputs. For this work, the $\mathcal{B}_l$ metric is of greater relevance.

Cryptographic view. The metrics $\mathcal{B}_l$ and $\mathcal{B}_d$ are directly related to the strength of a cipher against linear and differential cryptanalysis, respectively. Higher values for these branch numbers indicate increased resistance to such attacks. However, achieving higher branch numbers typically comes at the cost of increased computational complexity.

Definition 6.8 (Complexity of Diffusion Layer [JPST17]). The average number of XOR operations required to compute each output bit is defined as the complexity of the diffusion layer.

For ASCON, the linear and differential branch numbers of the diffusion layer are both 4 [DEMS21]. Specifically, each input bit affects three output bits, and no parity relation with fewer than four bits exists. This property can be verified by inspecting Equation (6.5). It is noteworthy that, for ASCON, the branch number is relatively low.⁷ With the relatively low branch number, the diffusion layer's computational complexity is also very low: each output bit is computed using just 2 XOR operations.

Polynomial view. By representing the b binary random variables of the input state as $\{X_1, \dots, X_b\}$, we can express the output bits of the diffusion layer as polynomials:

$$f_j = \bigoplus_{i=1}^b m_i^j X_i,$$

for $1 \leq j \leq b$. Here, the (binary) coefficients m_i^j are elements of the matrix $\mathbf{M}$, with $m_i^j = \mathbf{M}[i, j]$. Identifying $\mathcal{B}_l$ then reduces to finding a parity relation in the set

$$\mathcal{F} = \{X_1, \dots, X_b, f_1, \dots, f_b\}$$

with the smallest Hamming weight. A *parity relation* is a subset of $\mathcal{F}$ whose elements (bits) sum to zero (see Section 2.2). Formally, we have:

Lemma 6.9. *With f_j polynomials defined as above, and their collection given in $\mathcal{F}$, one can compute $\mathcal{B}_l$ as*

$$\mathcal{B}_l = \min_{|\mathcal{C}|} \left[\bigoplus_{l \in \mathcal{C}} \mathcal{F}[l] = 0 \right], \quad (6.3)$$

where $\mathcal{F}[l]$ denotes the l -th element of $\mathcal{F}$, and $\mathcal{C}$ is a non-empty set of indices.

⁷Since a single active input can create at most b active outputs, branch numbers are upper-bounded by $b + 1$. In practice, this upper bound (or values close to it) is achievable for a special class of $\mathbf{M}$ matrices described by maximum distance separable (MDS) codes [DR20].

Proof. To prove the claim, note first:

$$\begin{aligned}\mathcal{B}_l &= \min_{S \neq 0^b} [\text{wt}(S) + \text{wt}(SM^\top)] \\ &= \min_{S \neq 0^b} [\text{wt}(S \parallel SM^\top)] \\ &= \min_{S \neq 0^b} \text{wt}\left(S \times [\mathbf{I}_{b \times b} \parallel \mathbf{M}^\top]\right).\end{aligned}$$

In coding theory terms, the vector $S \times [\mathbf{I} \parallel \mathbf{M}^\top]$ is a *codeword* for the generator matrix $\mathbf{G} = [\mathbf{I}_{b \times b} \parallel \mathbf{M}^\top]$. Correspondingly, there is a *parity-check* matrix $\mathbf{H} = [\mathbf{M} \parallel \mathbf{I}_{b \times b}]$ such that any codeword $C \in \{0, 1\}^{2b}$ satisfies

$$\mathbf{H}^\top C = \mathbf{0}_{b \times 1}.$$

Moreover, any $2b$ -bit vector C with $\mathbf{H}^\top C = \mathbf{0}$ is itself a codeword. Hence,

$$\min_{S \neq 0^b} \text{wt}(SG) = \min_{\{C \neq 0^{2b}, \mathbf{H}^\top C = \mathbf{0}\}} \text{wt}(C).$$

After some rearrangement:

$$\min_{\{C \neq 0^{2b}, \mathbf{H}^\top C = \mathbf{0}\}} \text{wt}(C) = \min_{|C|} \left[\bigoplus_{\text{col} \in C} \mathbf{H}[:, \text{col}] = \mathbf{0} \right],$$

where $\mathbf{H}[:, \text{col}]$ is the column vector of $\mathbf{H}$. Because $\mathbf{H} = [\mathbf{M} \parallel \mathbf{I}_{b \times b}]$, one can show that for $1 \leq l \leq 2b$:

$$\mathcal{F}[2b + 1 - l] = [X_1, \dots, X_b] \times \mathbf{H}[:, l].$$

Thus, for the sum of columns $\mathbf{H}[:, l]$ in C :

$$\bigoplus_{l \in C} \mathcal{F}[2b + 1 - l] = [X_1, \dots, X_b] \times \left(\bigoplus_{\text{col} \in C} \mathbf{H}[:, \text{col}] \right).$$

Putting it all together gives

$$\begin{aligned}\mathcal{B}_l &= \min_{|C|} \left[\bigoplus_{\text{col} \in C} \mathbf{H}[:, \text{col}] = \mathbf{0} \right] \\ &= \min_{|C|} \left[[X_1, \dots, X_b] \times \left(\bigoplus_{\text{col} \in C} \mathbf{H}[:, \text{col}] \right) = \mathbf{0} \right] \\ &= \min_{|C|} \left[\bigoplus_{\text{col} \in C} \mathcal{F}[2b + 1 - \text{col}] = \mathbf{0} \right] \\ &= \min_{|C'|} \left[\bigoplus_{l \in C'} \mathcal{F}[l] = \mathbf{0} \right],\end{aligned} \tag{6.4}$$

thus completing the proof. $\square$

Remark 6.10. In coding theory, the value of $\mathcal{B}_l$, derived as the least number of columns of the parity check matrix that sum to zero, is an important metric. It represents the *minimum distance* of the code created with $\mathbf{G}$, which indicates the error correction capability of the code.

Self-composition of the diffusion layer. The evaluation of the diffusion mapping is just one part of the computations in a single cipher round. A cipher typically operates over multiple rounds, with nonlinear operations interleaved between successive diffusion layers. To gain insight into how parity relations evolve with an increasing number of rounds, we temporarily disregard the presence of nonlinear layers and study the composition of the diffusion transformation itself. The composition is computed as $(\mathbf{SM})\mathbf{M} = \mathbf{SM}^2$. In general, the composition over r rounds is directly represented by the matrix $\mathbf{M}^r$.

Definition 6.11 (Order of the Diffusion Layer). The *order* of the diffusion layer is defined as the number of rounds r for which the layer composes with itself to produce the identity mapping. Formally, it is the smallest integer r such that $\mathbf{M}^r = \mathbf{I}$.

Composing the diffusion layer introduces new parity relations between the input S and the outputs $\mathbf{SM}^r$ (for varying r). Even before reaching the order, these new relations may have weights smaller than $\mathcal{B}_l$. More specifically, these relations arise in the $(r+1)b$ -element set of polynomials:

$$\mathcal{F}^r = \{X_1, \dots, X_b, f_1, \dots, f_b, f_{b+1}, \dots, f_{2b}, \dots, f_{(r-1)b+1}, \dots, f_{rb}\},$$

where, for $0 \leq \alpha < r$ and $1 \leq \beta \leq b$, $f_{\alpha b + \beta}$ is defined as:

$$f_{\alpha b + \beta} = \bigoplus_{i=1}^b \mathbf{M}^{\alpha+1}[\beta, i] X_i.$$

We denote the minimum weight of parity relations in the set $\mathcal{F}^r$ as $\mathcal{B}_l^r$. Note that $\mathcal{B}_l^r$ is a decreasing function of the number of rounds r , and at the order of the layer, it reaches the minimum value $\mathcal{B}_l^r = 2$.

Example 6.4. Let $b = 15$ and the diffusion be defined by the following relation:

$$\mathbf{X} \leftarrow \mathbf{X} \oplus (\mathbf{X} \lll 1) \oplus (\mathbf{X} \lll 3),$$

where $\mathbf{X}$ is a vector of 15 bits. Using a computer program, we verified that this map is invertible, has an order of 15, and its $\mathcal{B}_l$ is 4.⁸

For this map, we constructed the set $\mathcal{F}^r$ and computed the corresponding $\mathcal{B}_l^r$ by exhaustively testing all subsets of $\mathcal{F}^r$ with fewer than $\mathcal{B}_l$ entries. We observed that, up to the order, the minimum weight of parity relations does not fall below the initial branch number. $\square$

We note that for diffusion layers used in ciphers, the order is expected to be practically high. Moreover, $\mathcal{B}_l^r = \mathcal{B}_l$ is typically maintained for sufficiently high values of r in practical applications.

6.5.2 Joint Independence Across the Diffusion Layer

Consider a toy 3-bit linear transformation with the following input-output relation:

$$(X, Y, Z) \rightarrow (X \oplus Y, Y \oplus Z, X \oplus Y \oplus Z).$$

For this map, we have $\mathcal{B}_l = 3$. In this example, we can directly verify that the knowledge of a single input bit does not determine any of the output bits. More precisely, any two variables chosen from the inputs and outputs are jointly independent. The following lemma generalizes this observation.

Lemma 6.12. *Let $\mathcal{B}_l$ denote the branch number of a diffusion layer. Then, $\mathcal{B}_l - 1$ probes placed on the inputs and outputs of Diffusion are jointly independent. Alternatively, the collection of all probes on the inputs and outputs is $(\mathcal{B}_l - 1)$ -jointly independent.*

Proof. Assume these probes (variables) are dependent. Then, there must exist a parity relation between the inputs and outputs with fewer than $\mathcal{B}_l$ terms. This contradicts the definition of $\mathcal{B}_l$. $\square$

Lemma 6.12 can be extended to the composition of diffusion layers: any $\mathcal{B}_l^r - 1$ probes placed on the inputs and outputs of the diffusion layer in r rounds are jointly independent.

Joint independence of n -sharings across the diffusion layer. We have thus far shown that the diffusion layer provides a measure of joint independence among the *output* natives, assuming the *inputs* are unknown, uniform, and independent. However, this result, in its current form, is not necessarily useful in practice, because we cannot guarantee that every input meets

⁸This map can be expressed with *circulant matrices*, for which there is a straightforward approach to compute the order and inverse [LS16].

these assumptions (e.g., some natives like IVs or nonces, or their initial-round combinations, may be fixed or public). By shifting to a masked setting and considering n -shared inputs instead of natives, we arrive at the following more practical statement:

Lemma 6.13. *Any collection of $(\mathcal{B}_l - 1)$ n -sharings at the input and output of $\mathsf{SDiffusion}$ is jointly independent. Equivalently, all input and output n -sharings together form a set of $(\mathcal{B}_l - 1)$ -jointly independent n -sharings.*

Proof. Let $\mathcal{W} = \{\mathbf{X}^1, \dots, \mathbf{X}^{\mathcal{B}_l - 1}\}$ be a collection of n -sharings. Because each share index is processed by an identical, separate circuit, any dependency in $\mathcal{W}$ would imply that, for each share index i , the set $\mathcal{W}_i = \{X_i^1, \dots, X_i^{\mathcal{B}_l - 1}\}$ must also be dependent. Therefore, for some subset $\mathcal{V} \subseteq \{1, \dots, \mathcal{B}_l - 1\}$, we would have

$$\bigoplus_{j \in \mathcal{V}} X_i^j = 0,$$

which, when summed over all share indices $1 \leq i \leq n$, implies

$$\bigoplus_{j \in \mathcal{V}} X^j = 0,$$

where this relation holds on the *native* random variables. However, by definition of $\mathcal{B}_l$, no parity relation among the native variables can have fewer than $\mathcal{B}_l$ terms. Hence, such a dependency in $\mathcal{W}$ cannot occur.

Finally, note that n -sharing independence (Definition 6.3) places no constraints on the values of the underlying natives. In particular, this lemma still holds if all input natives are zero. $\square$

Joint independence of n -sharings at the SS -box level. The input and output wires of $\mathsf{SDiffusion}$ are grouped according to the adjacent SS -box layers. For our analysis, it is important to determine how many of these groups—each corresponding to one SS -box—remain jointly independent. We refer to this metric as $(\mathcal{B}_{l\text{-box}} - 1)$, where $\mathcal{B}_{l\text{-box}}$ is the *box linear branch number*.

Previously, we showed that up to $(\mathcal{B}_l - 1)$ individual n -sharings across the diffusion layer are jointly independent. Given that each S -box operates on m bits, the number of jointly independent SS -box inputs/outputs is bounded as follows:

$$\frac{\mathcal{B}_l - 1}{m} \leq (\mathcal{B}_{l\text{-box}} - 1) \leq (\mathcal{B}_l - 1).$$

The exact value of $\mathcal{B}_{l\text{-box}}$ can be computed as follows:

Definition 6.14 (Box Linear Branch Number). Let S be a b -bit input to the diffusion layer, and let $M^\top$ denote the transpose of the matrix defining this layer (as described in Section 6.5.1). The *box linear branch number* $\mathcal{B}_{l\text{-box}}$ is defined as

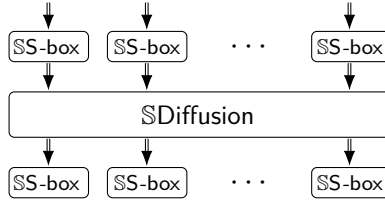
$$\mathcal{B}_{l\text{-box}} = \min_{S \neq 0^b} \left[\# \text{active S-boxes in } S + \# \text{active S-boxes in } SM^\top \right],$$

where an S-box is considered active if any of its m bits is active.

6.5.3 Our Higher-Order Composition Rule

While our motivating case involves deterministic masking, the result itself holds more broadly. We now present our main composition result, which underlies deterministic masking schemes but applies more generally to any probing-secure gadgets.

Theorem 6.15 (Composition Theorem). *Consider two consecutive $\mathbb{S}\mathbb{S}$ -box layers separated by a single $\mathbb{S}\text{Diffusion}$ layer:*



Assume the following conditions hold:

- Each $\mathbb{S}\mathbb{S}$ -box is $d_{\mathbb{S}\mathbb{S}\text{-box}}$ -probing secure.
- The $\mathbb{S}\text{Diffusion}$ layer is bijective.
- Given independent n -sharings as input, both $\mathbb{S}\mathbb{S}$ -box and $\mathbb{S}\text{Diffusion}$ produce independent n -sharings at their output.
- Let $\mathcal{B}_{l\text{-box}}$ denote the box linear branch number of the diffusion layer.

Then, the probing security order d_{comb} of the combined structure satisfies:

$$\min \{d_{\mathbb{S}\mathbb{S}\text{-box}}, (\mathcal{B}_{l\text{-box}} - 1)\} \leq d_{comb} \leq d_{\mathbb{S}\mathbb{S}\text{-box}}.$$

Proof. We prove the theorem in four steps.

Step 1: probes on $\mathbb{S}\mathbb{S}$ -box gadgets only. Assume all probes are placed on the $\mathbb{S}\mathbb{S}$ -box gadgets, and none target wires inside the $\mathbb{S}\text{Diffusion}$ layer. Let the probes be distributed across a set $\mathcal{T}$ of $\mathbb{S}\mathbb{S}$ -box gadgets. If $|\mathcal{T}| \leq (\mathcal{B}_{l\text{-box}} - 1)$, then the inputs and outputs of these gadgets remain jointly independent. Therefore, each gadget behaves as in the standalone case, preserving its local $d_{\mathbb{S}\mathbb{S}\text{-box}}$ -order security.

Step 2: total number of probes. We now determine how many probes these $|\mathcal{T}|$ gadgets can collectively tolerate. There are two cases:

- If $(\mathcal{B}_{l\text{-box}} - 1) \geq d_{\mathbb{S}\mathbb{S}\text{-box}}$, then the limiting factor is the local security of the gadgets. Thus, $d_{\text{comb}} = d_{\mathbb{S}\mathbb{S}\text{-box}}$.
- If $(\mathcal{B}_{l\text{-box}} - 1) < d_{\mathbb{S}\mathbb{S}\text{-box}}$, then even though each gadget can tolerate more probes, joint independence across too many gadgets may not hold. Therefore, we can guarantee:

$$(\mathcal{B}_{l\text{-box}} - 1) \leq d_{\text{comb}} \leq d_{\mathbb{S}\mathbb{S}\text{-box}}.$$

Hence, in all cases,

$$\min\{d_{\mathbb{S}\mathbb{S}\text{-box}}, (\mathcal{B}_{l\text{-box}} - 1)\} \leq d_{\text{comb}} \leq d_{\mathbb{S}\mathbb{S}\text{-box}}.$$

Step 3: probes on $\mathbb{S}\text{Diffusion}$. Now consider probes placed on the internal wires of $\mathbb{S}\text{Diffusion}$. We use the probe elimination strategy (see Section 6.3.3): any such probe can be eliminated using the independent offline randomness from the output of one $\mathbb{S}\mathbb{S}$ -box.

For example, if a wire carries $P = X_i \oplus P'$, where X_i is a share from the output of $\mathbb{S}\mathbb{S}\text{-box}_j$ (see Figure 6.5), then P can be eliminated if:

1. X_i does not appear in P' .
2. X_i is uniformly distributed.
3. X_i is independent of all other probes.
4. Not all shares of the native variable corresponding to X_i are involved in P' .

Thanks to the linearity of $\mathbb{S}\text{Diffusion}$ (no cross-share operations), Conditions (1) and (4) are automatically satisfied. Condition (2) follows because output of gadgets are n -sharing, which means that any $n - 1$ collection of shares are uniformly distributed. To ensure Condition (3), we add $\mathbb{S}\mathbb{S}\text{-box}_j$ to $\mathcal{T}$. If $|\mathcal{T}| \leq (\mathcal{B}_{l\text{-box}} - 1)$, then this inclusion does not break the independence assumptions.

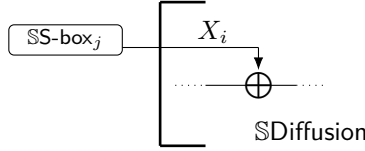


Figure 6.5: A probe on SDiffusion is reduced to a probe on SS-box_j .

Step 4: final bound. Each SS-box added to $\mathcal{T}$ corresponds to at most one probe (direct or via diffusion). As long as $|\mathcal{T}| \leq (\mathcal{B}_{l\text{-box}} - 1)$, we can ensure that the gadgets operate on jointly independent inputs. From Step 2, this yields:

$$\min\{d_{\text{SS-box}}, (\mathcal{B}_{l\text{-box}} - 1)\} \leq d_{\text{comb}} \leq d_{\text{SS-box}}.$$

This completes the proof. $\square$

In what follows, we will use $d_{\text{SS-box}}$ and d_{comb} to denote probing security orders where ambiguity might arise; otherwise, we simply use d .

Security claim for multiple rounds. In practical cipher designs, the combination of two consecutive rounds (connected via a single diffusion layer) typically determines the limiting value of $\mathcal{B}_{l\text{-box}}$. Adding additional rounds generally does not further reduce this quantity. This behavior reflects the *avalanche effect*, a common design goal in cryptographic primitives, where changes to a small number of input bits rapidly propagate across the state.

As a result, the compositional guarantees established in Theorem 6.15 often extend naturally to larger numbers of rounds. However, we leave a formal treatment of this extension to future work.

Use of online randomness in SS-box . The proof of Theorem 6.15 assumes that each SS-box produces independent n -sharings at its output given independent n -shared inputs, and that it satisfies local probing security of order $d_{\text{S-box}}$. However, the theorem imposes no constraints on how these properties are achieved—particularly, it does not require that the computation inside the gadget be deterministic. This flexibility allows the composition result to hold even when the SS-box uses *online* randomness in its internal operations. This generality significantly enhances the practical applicability of our composability theorem.

Application to AES. In the case of AES, the combination of the ShiftRows and MixColumns layers yields a box linear branch number of $\mathcal{B}_{l\text{-box}} = 5$. Therefore, the diffusion layer in AES can support up to $d = \mathcal{B}_{l\text{-box}} - 1 = 4$ probing security order—*provided* that the $\mathbb{S}$ S-box meets the requirements of Theorem 6.15.

Towards probing-secure randomness reuse. So far, our discussion has assumed that secure computation inside each $\mathbb{S}$ S-box is either deterministic—as demonstrated in this work for the case of ASCON—or that each gadget receives its own fresh (online) randomness. A complementary paradigm aims to minimize the need for online randomness by enabling its *reuse* across rounds.

Figure 6.6 illustrates a conceptual architecture where auxiliary randomness, introduced only in the initial round, is propagated forward via a parallel *randomness network*. This structure supports the reuse of offline randomness while aiming to maintain a high level of probing security.

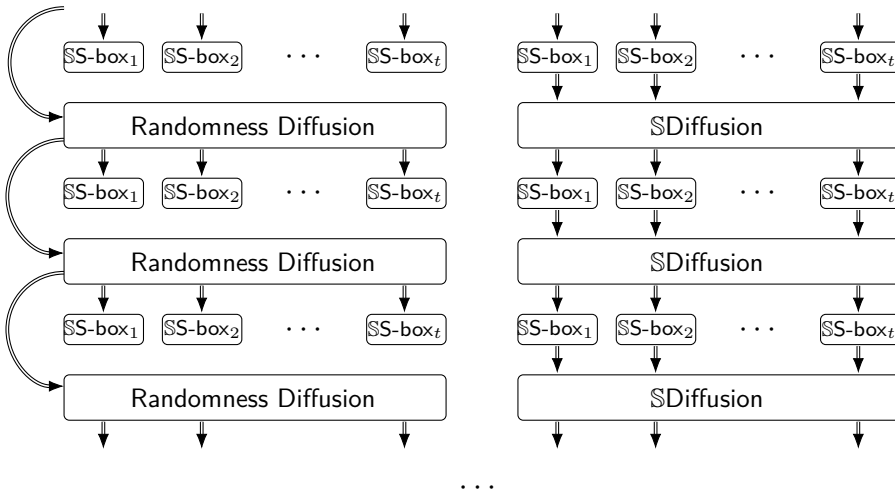


Figure 6.6: A parallel randomness reuse network that mirrors the round structure of the masked cipher. The initial invocation of Randomness Diffusion receives fresh randomness; subsequent rounds reuse the transformed outputs.

The problem of expanding a small amount of true randomness into a larger amount of pseudo-random bits suitable for side-channel protection has been previously considered in the literature [CGZ20, IKL⁺13]. However, the specific

architecture depicted in Figure 6.6—which mirrors the round structure of the cipher while avoiding nonlinear operations—has not been studied. Such a structure may offer both implementation simplicity and improved efficiency.

Mathematically, the behavior of the randomness reuse network mirrors that of diffusion layer self-composition, as discussed in Section 6.5.1. There, we examined how to determine the number of jointly independent outputs at the bit level. Here, a similar analysis can be extended to groupings that align with S-box boundaries.

Modeling the problem. Let M_R and M_C represent the diffusion matrices for the randomness network and the cipher, respectively. The inputs to each SS-box derive from the outputs of both M_R and M_C . To evaluate how many SS-boxes receive jointly independent inputs, we define the following set of polynomials over the inputs $\{X_1, \dots, X_b\}$:

$$\mathcal{F}^r = \{X_1, \dots, X_b, g_1, \dots, g_b, f_1, \dots, f_{rb}\},$$

where the polynomials are defined as follows:

$$g_\beta = \bigoplus_{i=1}^b M_C[\beta, i] X_i, \quad f_{\alpha b + \beta} = \bigoplus_{i=1}^b M_R^{\alpha+1}[\beta, i] X_i,$$

for $1 \leq \beta \leq b$ and $0 \leq \alpha < r$. We assume that the input lengths match; otherwise, missing inputs in M_R can be accommodated by inserting zero columns at the appropriate positions to align with the indexing of the X_i . In this setup, $f_{\alpha b + \beta}$ polynomials are computing the randomness that are delivered to the SS-boxes at round $\alpha + 1$, and g_β is representing possible mix of the randomness via the masked cipher round.

We define $\mathcal{B}_{l\text{-box}}^r$ as the minimal number of S-boxes involved in a parity relation over $\mathcal{F}^r$. Let each SS-box be $d_{\text{SS-box}}$ -secure and produce n -shared outputs from n -shared inputs for any fixed auxiliary randomness.

We conjecture that the probing security order d_{comb} of the r -round composition satisfies:

$$\min\{d_{\text{SS-box}}, (\mathcal{B}_{l\text{-box}}^r - 1)\} \leq d_{\text{comb}} \leq d_{\text{SS-box}}.$$

This conjecture is based on the structural analogy with our deterministic composition result (Theorem 6.15), where $\mathcal{B}_{l\text{-box}}$ governs the effective independence across masked components. Extending the same reasoning to the auxiliary randomness setting appears promising, but proving this bound remains an open problem. We leave its rigorous analysis to future work.

6.6 Case Study on Ascon

In this section, we briefly review the structure of ASCON and investigate the feasibility of applying deterministic masking to it. The value of its box linear branch number is $\mathcal{B}_{l\text{-box}} = 4$, which permits probing security up to order $d = 3$.

In this chapter, we focus on designing and analyzing SS-box gadgets for security orders $d \in \{1, 2\}$. A construction for $d = 3$ security order will be given in the next chapter.

For the specific case of $d = 2$, we additionally present an alternative, *direct* method for verifying the probing security of the round composition.

We conclude this section by discussing the resulting randomness savings and reporting on software implementations of the protected ASCON design for $d \in \{1, 2\}$.

6.6.1 Ascon's Structure

ASCON [TMC⁺24, DEMS21] has a bricklayer structure operating on a state of $b = 320$ bits. It employs a *sponge-based* [BDPV07] mode of operation for authenticated encryption. The key, tag, and nonce are each 128 bits. As outlined in Figure 6.7, the encryption process is divided into four phases: (1) *Initialization*: The state is initialized with the (private) key K , the (public) nonce N , and a fixed initialization vector (IV). (2) *Associated Data Processing*: The state is updated with associated data blocks A_i . Associated data is not confidential and is therefore not encrypted. However, it is absorbed by the sponge to verify its integrity during transit. (3) *Plaintext Processing*: Plaintext blocks P_i are injected into the state, and ciphertext blocks C_i are extracted. (4) *Finalization*: The key K is injected again, and the tag T is extracted for authentication.

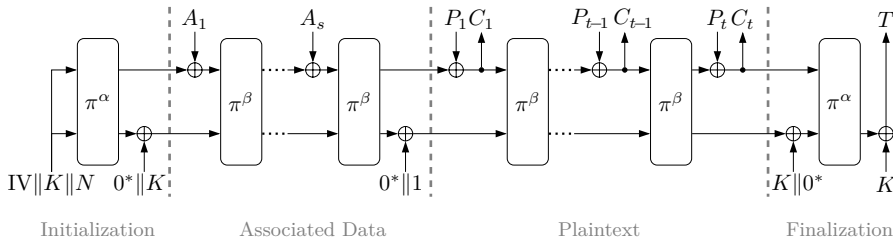


Figure 6.7: ASCON. π^α and π^β are application of the permutation π in α and β rounds.

Ascon's permutation. The cryptographic permutation of ASCON is called ASCON-p, which we here for convenience denote with π . The permutation π is an iterative application of a round transformation on the 320-bit state $\mathbf{s}$. The state is represented as a 5×64 table, $\mathbf{s} = [s_1, s_2, s_3, s_4, s_5]^\top$, where each s_i is 64 bits long. The round transformation comprises three steps: (1) *Addition of Round Constants*: XORs a round-specific 1-byte constant to the state. (2) *Nonlinear Layer*: Applies a 5-bit S-box 64 times in parallel. For $1 \leq i \leq 64$, the S-box is applied to $(s_1[i], s_2[i], s_3[i], s_4[i], s_5[i])$, where $[i]$ denotes the i -th bit. (3) *Diffusion Layer*: XOR rotated parts of the state. Specifically, for $1 \leq j \leq 5$, s_j is updated as:

$$s_j \leftarrow s_j \oplus (s_j \lll \theta_j) \oplus (s_j \lll \theta'_j), \quad (6.5)$$

where the rotation amounts θ_j and θ'_j are specified in [DEMS21].

Ascon's S-box. The S-box is χ_5 (defined below) between two thin 5-bit linear and affine layers, as depicted in Figure 6.8. The round function contains 64 S-boxes, one for each i .

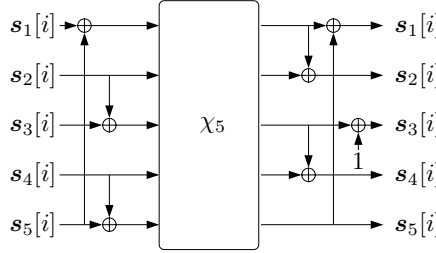


Figure 6.8: ASCON's S-box.

The map χ_m . The nonlinear map $\chi_m: \{0,1\}^m \rightarrow \{0,1\}^m$ is defined as follows: For binary input vector $(X^1, \dots, X^m)$, the output Y^i , for $1 \leq i \leq m$, is computed as:

$$Y^i = X^i \oplus (1 \oplus X^{i+1})X^{i+2},$$

where the boundary values X^{m+1} and X^{m+2} are substituted by X^1 and X^2 , respectively.

The map χ_m was investigated by Daemen in [Dae95]. For odd m , it is bijective. This map is deployed in several other important cryptographic algorithms. For instance, χ_5 is used in SHA-3 [oST15] and χ_3 is used in XOODYAK [DHP⁺20]. Notably, in these algorithms, χ_m provides the only source of non-linearity in the round function.

Protecting the permutation. The addition of the key K at the end of the *initialization* phase and the start of the *finalization* phase, as shown in Figure 6.7, prevents side-channel state recovery attacks (carried out during Associated Data and Plaintext Processing) from escalating into *key recovery* or *tag forgery* attacks [DEMS21, BBC⁺20].

To specifically mitigate key recovery attacks, only the initialization and finalization phases require protection. Considering that both phases consist of a single invocation of the permutation π , our focus will be on securing the computations of π .

6.6.2 Protected Implementation of the Permutation

The permutation consists of two main building blocks: the Diffusion layer and the S-box. While Diffusion is affine and thus straightforward to mask at any order, the nonlinear S-box presents a greater challenge. In the following, we focus on existing proposals for *deterministic* masking of the S-box for orders $d \in \{1, 2\}$.

Ascon’s deterministic SS-boxes. ASCON’s S-box (Figure 6.8) implements the χ_5 function, enclosed between two linear-affine layers. In the corresponding SS-box construction, these layers are trivially masked using standard Boolean masking techniques and then composed with a $\mathbb{S}_{\chi_5}$ gadget.

For $n \in \{2, 3\}$, the literature provides deterministic, $(n - 1)$ -order probing-secure, and bijective $\mathbb{S}_{\chi_5}$ gadgets that preserve the independence of n -shared outputs when fed with independent n -shared inputs. We briefly review these gadgets below.

A $\mathbb{S}_{\chi_5}$ gadget. Daemen et al. [DDE⁺20] introduced a two-share, first-order secure $\mathbb{S}_{\chi_5}$ gadget. They formally verified the security of the resulting SS-box (built using this gadget) using the maskVerif tool [BBD⁺15a, BBC⁺19].⁹

⁹At the time of verification, maskVerif followed the methodology of [BBD⁺15a]. The more recent version [BBC⁺19], which relies on simulation-based rules, cannot verify the probing security of gadget or its associated SS-box.

This design has been implemented by the official ASCON team in a software setting,¹⁰ using a *bit-sliced* representation combined with *share rotation* to enhance resistance against side-channel attacks [GMSP23].

SM- $\mathbb{S}_{\chi_5}$ gadgets. Shahmirzadi and Moradi [SM21] employed a computerized search to identify $\mathbb{S}_{\chi_5}$ gadgets for $(n = 2, d = 1)$ and $(n = 3, d = 2)$. Their constructions provide glitch resistance and achieve probing security without requiring any auxiliary randomness. We refer to these designs as SM1- $\mathbb{S}_{\chi_5}$ and SM2- $\mathbb{S}_{\chi_5}$, respectively. Slightly simplified versions of their internal computations, adapted for a probing-security focus.

SM1- $\mathbb{S}_{\chi_5}$ gadget. Let $\{A, B, C, D, E\}$ denote the five native input bits. For a first-order probing secure implementation with $n = 2$ shares, we write their corresponding shares as:

$$\{A_1, A_2, B_1, B_2, C_1, C_2, D_1, D_2, E_1, E_2\}.$$

Recall that the non-shared (unmasked) χ_5 function for five bits is:

$$\begin{aligned} A' &= A \oplus C \oplus (BC), & B' &= B \oplus D \oplus (CD), & C' &= C \oplus E \oplus (DE), \\ D' &= D \oplus A \oplus (EA), & E' &= E \oplus B \oplus (AB). \end{aligned}$$

In the shared version, the output shares $\{A'_i, B'_i, C'_i, D'_i, E'_i\}$ for $i \in \{1, 2\}$ can be computed as follows:

$$\begin{aligned} A'_1 &= C_1 B_1 \oplus A_1 \oplus C_1 B_2 \oplus C_1, & A'_2 &= C_2 B_1 \oplus A_2 \oplus C_2 B_2 \oplus C_2, \\ B'_1 &= D_1 C_1 \oplus B_1 \oplus D_1 C_2 \oplus D_1, & B'_2 &= D_2 C_2 \oplus D_2 \oplus B_2 \oplus D_2 C_1, \\ C'_1 &= E_1 D_2 \oplus E_1 \oplus C_1 \oplus E_1 D_1, & C'_2 &= E_2 D_1 \oplus E_2 \oplus C_2 \oplus E_2 D_2, \\ D'_1 &= A_2 E_1 \oplus D_1 \oplus A_1 E_1, & D'_2 &= A_1 E_2 \oplus D_2 \oplus A_1 \oplus A_2 E_2 \oplus A_2, \\ E'_1 &= B_1 A_1 \oplus E_1 \oplus B_1 \oplus B_1 A_2, & E'_2 &= B_2 A_1 \oplus E_2 \oplus B_2 \oplus B_2 A_2. \end{aligned}$$

SM2- $\mathbb{S}_{\chi_5}$ gadget. For second-order implementations ($n = 3$ shares) of χ_5 function, let the set $\{A_1, A_2, A_3, \dots, E_1, E_2, E_3\}$ denote the three shares of

¹⁰<https://github.com/ascon/simpleserial-ascon>

inputs $\{A, B, C, D, E\}$. The output shares of the χ_5 function can be computed as:

$$\begin{aligned}
A'_1 &= C_1 B_1 \oplus C_2 \oplus C_2 B_1 \oplus C_3 \oplus A_1 \oplus C_3 B_1, \\
A'_2 &= C_1 \oplus C_1 B_2 \oplus C_2 B_2 \oplus C_3 \oplus A_2 \oplus C_3 B_2, \\
A'_3 &= A_3 \oplus C_1 B_3 \oplus C_2 B_3 \oplus C_3 \oplus C_3 B_3, \\
\\
B'_1 &= D_1 \oplus D_1 C_1 \oplus D_2 C_1 \oplus B_3 \oplus D_3 C_1, \\
B'_2 &= D_1 \oplus D_1 C_2 \oplus D_2 C_2 \oplus D_3 \oplus B_1 \oplus D_3 C_2, \\
B'_3 &= D_1 \oplus B_2 \oplus D_1 C_3 \oplus D_2 \oplus D_2 C_3 \oplus D_3 C_3, \\
\\
C'_1 &= E_1 \oplus C_3 \oplus E_1 D_1 \oplus E_2 D_1 \oplus E_3 \oplus E_3 D_1, \\
C'_2 &= E_1 \oplus E_1 D_2 \oplus E_2 D_2 \oplus C_1 \oplus E_3 D_2, \\
C'_3 &= E_1 \oplus C_2 \oplus E_1 D_3 \oplus E_2 \oplus E_2 D_3 \oplus E_3 D_3, \\
\\
D'_1 &= A_1 \oplus D_1 \oplus A_1 E_1 \oplus A_2 \oplus A_2 E_1 \oplus A_3 E_1, \\
D'_2 &= A_1 \oplus A_1 E_2 \oplus A_2 E_2 \oplus A_3 \oplus D_3 \oplus A_3 E_2, \\
D'_3 &= A_1 \oplus A_1 E_3 \oplus A_2 E_3 \oplus D_2 \oplus A_3 E_3, \\
\\
E'_1 &= B_1 \oplus E_1 \oplus B_1 A_1 \oplus B_2 \oplus B_2 A_1 \oplus B_3 A_1, \\
E'_2 &= B_1 \oplus B_1 A_2 \oplus B_2 A_2 \oplus E_3 \oplus B_3 A_2, \\
E'_3 &= B_1 \oplus B_1 A_3 \oplus B_2 A_3 \oplus B_3 \oplus E_2 \oplus B_3 A_3.
\end{aligned}$$

Important note: The order of operations (from left to right) is critical for achieving probing security. The original threshold implementations in [SM21] ensure security even in the presence of glitches, while the simplified expressions given here are probing-secure and suffice for scenarios considered in our work.

Using the SILVER tool [KSM20], we verified the probing security and bijectivity of these SM- $\mathbb{S}_{\chi_5}$ gadgets, as well as the full SS-box constructions that incorporate them. These gadgets are also used in our practical side-channel leakage evaluations.

6.6.3 Application of Our Composition Rules

The SS-box constructions based on SM- $\mathbb{S}_{\chi_5}$ satisfy the conditions of Lemma 6.6, and thus provide first-order probing security by composition. In particular, the gadget SM2- $\mathbb{S}_{\chi_5}$ achieves $d_{\text{SS-box}} = 2$. Hence, when used within a bricklayer

design, the achievable probing security order d_{comb} satisfies:

$$\min\{2, 3\} \leq d_{\text{comb}} \leq 2,$$

which implies $d_{\text{comb}} = 2$.

We additionally present an alternative, direct argument based on probe elimination to support the probing security of ASCON's permutation across multiple rounds. The key observation is that when using SM2- $\mathbb{S}\chi_5$ gadgets, it is possible to derive a representation that decouples the transformation of shares in each round from the evolution of their underlying native variables (secrets). Specifically, while the unshared native state evolves nonlinearly, the corresponding shares undergo only linear transformations. This separation simplifies the task of identifying how each $\mathbb{S}\mathbb{S}$ -box input remains blinded by the randomness in the initial input shares throughout all rounds.

6.6.4 Direct Analysis of the $d = 2$ Case

Linear representation of output shares. A key property of the SM2- $\mathbb{S}\chi_5$ gadget is that, for a *fixed* native state, each output share can be written as a linear function of the input shares. For example, the output share A'_1 satisfies:

$$A'_1 = C_1 B_1 \oplus C_2 \oplus C_2 B_1 \oplus C_3 \oplus A_1 \oplus C_3 B_1,$$

which simplifies to

$$A'_1 = C B_1 \oplus C_2 \oplus C_3 \oplus A_1,$$

where $C = C_1 \oplus C_2 \oplus C_3$ is a fixed function of the native state. Thus, the expression is linear in the input shares when the state is treated as constant.

Adopting the convention that native variables (e.g., C) are functions of the initial state S , we can rewrite A'_1 schematically as:

$$A'_1 = (A_1 \oplus C_2 \oplus C_3) \oplus (B_1 \cdot f(S)),$$

where $f(S)$ denotes some function of the native state whose exact form is not tracked.

This structure holds for all output shares of the SM2- $\mathbb{S}\chi_5$ gadget. In what follows, we exploit this linearity to trace how the shares evolve across rounds. From a side-channel perspective, the precise nonlinear evolution of the native state S is unimportant—as long as the shares correctly mask the state, the security objective is met.

Tracing round-input formats. ASCON’s permutation operates on a 320-bit state ($b = 320$). Under an $n = 3$ sharing, the protected state comprises $nb = 960$ shares. For notational convenience, we label these shares as $\{X_1, X_2, \dots, X_{nb}\}$, grouping together the shares of each bit in the native state.

To track how these shares propagate through the layers of **SS-box** and **SDiffusion**, we model each input share Y to round r as a polynomial expression of the form:

$$Y = \left(\bigoplus_{i \in \mathcal{I}} X_i \right) \oplus \left(\bigoplus_{j \in \mathcal{J}} X_j \cdot f_j(S) \right), \quad (6.6)$$

where $\mathcal{I}, \mathcal{J} \subseteq \{1, \dots, nb\}$ are disjoint sets of indices, and each $f_j(S)$ is a Boolean function of the native state S . Since these functions serve only to modulate the input shares via native-state dependence, we do not track their explicit forms. Instead, our analysis follows only the indices $\mathcal{I}$ and $\mathcal{J}$.

Blinding with initial input shares. Due to the symmetry of the cipher structure, we focus our second-order security analysis on showing that the n -shared inputs to any **SS-box** gadget beyond the first are independent of the initial inputs to the first one. Specifically, we consider the initial 15 shares $\{X_1, \dots, X_{15}\}$ that serve as input to the first **SS-box**, and we aim to show that later-round inputs are statistically independent of these shares and of the state S .

We do this by inspecting the expressions in Equation (6.6). If, for a given Y , there exists at least one index $i \in \mathcal{I}$ such that $i \notin \{1, \dots, 15\}$, then Y is blinded by a share independent of the first **SS-box** input. When analyzing multiple shares Y , we further ensure that each is blinded by a *distinct* X_i in order to satisfy joint independence. This argument follows the probe-elimination reasoning from Section 6.3.

Note that only $n - 1$ shares per native variable are uniformly random; using all n shares would compromise blinding. Hence, our approach ensures that each X_i used for blinding comes from a separate native variable or avoids exhausting all its shares.

We carried out this procedure for up to $r = 5$ rounds and verified that the inputs to all **SS-box** gadgets remained independent of $\{X_1, \dots, X_{15}\}$ and of the state S . For rounds beyond $r = 5$, each polynomial in Equation (6.6) effectively reduces to the following form:

$$Y = \bigoplus_{j=1}^{nb} X_j \cdot f_j(S), \quad (6.7)$$

where again the goal is to ensure that for some $j \notin \{1, \dots, 15\}$, the function $f_j(S)$ evaluates to 1. Since each $f_j(S)$ is Boolean and state-dependent, the probability that all $f_j(S)$ vanish over $j \in \{16, \dots, nb\}$ is negligibly small. Hence, with overwhelming probability, the share Y is blinded by a variable independent of $\{X_1, \dots, X_{15}\}$, ensuring second-order security across multiple rounds.

6.6.5 Randomness Savings

In ASCON’s masked implementation, the only nonlinear component per round is the χ_5 layer. Each round processes the 320-bit state using 64 instances of χ_5 , and each instance involves five AND operations. Thus, each round contains a total of 320 AND operations.

A typical SAND gadget (e.g., from [ISW03, CS20]) requires $\frac{d(d+1)}{2}$ bits of *online* randomness per invocation to achieve d -order probing security. For ASCON, this leads to an online randomness cost of approximately $160d(d+1)$ bits *per round*, assuming randomness is only used within nonlinear components. This cost scales linearly with the number of rounds.

In contrast, our deterministic masking approach eliminates this overhead entirely: no online randomness is required during the execution of the masked permutation. The initial input shares encapsulate all necessary randomness, and this offline entropy is recycled across rounds.

6.6.6 Implementation Results

We implemented the proposed deterministic masking scheme for ASCON.¹¹ Our starting point was the existing software-protected implementation of ASCON, available at [DDM⁺22], which we adapted to suit the deterministic masking framework presented in this chapter.

Specifically, we replaced the original $\mathbb{S}_{\chi_5}$ gadgets with the SM1- $\mathbb{S}_{\chi_5}$ and SM2- $\mathbb{S}_{\chi_5}$ constructions, targeting first- and second-order probing security, respectively. We eliminated all dependencies on auxiliary randomness, ensuring that the entire implementation relies solely on offline randomness embedded in the initial input shares. To prevent compiler optimizations from merging or simplifying masked values (e.g., by combining shares), we implemented the critical $\mathbb{S}_{\chi_5}$ computations directly in ARM assembly. Unlike the original code base, our version is explicitly backed by a formal composability theorem.

Our implementation preserves several useful features from the original repository, including share rotation, device-specific leakage mitigations, and

¹¹<https://github.com/vahid-jahandideh/deterministic-masking-ascon.git>

a side-channel secure tag comparison mechanism. Additionally, we introduced word-level interleaving within the $\mathbb{S}\mathbb{S}$ -box computations to reduce Hamming distance leakage stemming from register-level value transitions.

Execution and measurement setup. All experiments were performed on a ChipWhisperer CW308 board with an STM32F303 UFO target,¹² featuring a 32-bit ARM Cortex-M4 microcontroller. Power traces were captured using a ChipWhisperer-Lite at a sampling rate four times the device’s clock frequency. Initial randomness for input sharing was generated using a modified `randombytes.c` file, based on the standard `rand()` and `srand()` functions from `stdlib.h`, as used in the original framework.

Share rotation and Hamming distance leakage. Share rotation is a relatively recent and effective technique introduced in [GD23] to mitigate Hamming distance leakage. The core idea is to ensure that different shares are loaded into registers with varying amounts of rotation. This reduces the likelihood that transitions between share values result in data-dependent leakage—especially in designs where no cross-share operations occur.

However, within nonlinear computations such as the $\mathbb{S}\mathbb{S}$ -box, registers may be overwritten without proper share rotation, potentially introducing side-channel leakage. One common mitigation strategy is to explicitly clear registers (e.g., using `mov R0, 0`) before writing new values. Yet, clearing low-index registers (like `R0–R4`) can interfere with correct computation due to register allocation constraints.

To resolve this, our implementation combines share rotation with 32-bit *word-level interleaving* inside the 64-bit $\mathbb{S}\mathbb{S}$ -box computations. This approach eliminates visible leakage peaks in the T-test measurements, albeit at the cost of slightly increased cycle count.

Speed comparison. Table 6.1 compares the running time (in terms of clock cycles per encrypted byte) for three protection levels: unprotected ($d = 0$), first-order ($d = 1$), and second-order ($d = 2$). We benchmark both our deterministic masking implementation and the reference implementation from [DDM⁺22]. While our approach eliminates the need for a randomness reuse network, the $\mathbb{S}\mathbb{S}$ -box computations are heavier and incorporate word-level interleaving. As a result, despite some architectural simplifications, our implementation consumes roughly 25% more cycles. Importantly, our design is backed by a formal theorem for the 2-round composition (Theorem 6.15)

¹²<https://github.com/newaetech/chipwhisperer>

and complemented by a direct/probabilistic analysis for the multi-round permutation (Section 6.6.4). The 25% overhead observed even in the unprotected ($d = 0$) case is primarily due to the word-level interleaving countermeasure.

Table 6.1: Cycle count per byte under different protection orders

Protection Order	Source	1 Byte	Long Message
$d = 0$	Ascon Team [DDM ⁺ 22]	4,274	99
	This Work	5,109	124
$d = 1$	Ascon Team [DDM ⁺ 22]	11,085	310
	This Work	13,339	388
$d = 2$	Ascon Team [DDM ⁺ 22]	20,883	585
	This Work	25,414	745

6.6.7 Power Analysis

Simple power analysis (SPA). Figure 6.9 shows the average power consumption over 1000 traces during the initialization phase of ASCON. The $\alpha = 12$ initialization rounds (see Figure 6.7) are clearly identifiable. The first ~ 1050 clock cycles correspond to the loading and unpacking of the shared nonce, key, and other input data transmitted from the host computer (via Python scripts) to the device. Each round of the permutation takes ~ 460 clock cycles to execute.

Attack point and first-order T-Test results. As shown in Figure 6.7, the ASCON initialization phase involves injecting the secret key, a public nonce, and a fixed initialization vector (IV) into the permutation. We target this phase for our experimental leakage evaluation.

To conduct the first-order T-test, we partition the collected power traces into two groups:

- **Group 1:** Fixed key, *fixed* nonce, fixed IV.
- **Group 2:** Fixed key, *random* nonce, fixed IV.

To understand the signal-to-noise behavior of our setup and highlight possible leakage, we first perform a sanity check by deliberately reducing the randomness. Specifically, we set the randomness used for sharing the key and nonce to zero, while preserving full randomness for the IV. Under this setting,

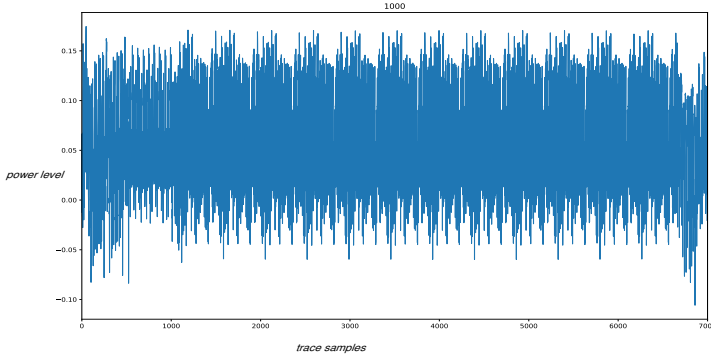


Figure 6.9: Average power consumption during the initialization phase of ASCON (12 rounds) using the proposed deterministic masking with $n = 2$ shares ($d = 1$ security). The x-axis denotes clock cycles; the y-axis shows power consumption.

the first few rounds are only partially protected, and the IV—being properly shared—gradually blends into the state, eventually restoring protection.

The results, depicted in Figure 6.10, are based on 50 power traces. The observed T-test peaks confirm that our measurement setup exhibits low noise. This is expected due to factors such as synchronized sampling and execution on a single, stable measurement platform.

Figure 6.11 presents the T-test results for the $d = 1$ security case using 500K traces. All values remain within the ± 4.5 threshold, indicating no detectable leakage at this level. However, when increasing the number of traces to 1 million, small peaks begin to appear in the early clock cycles. A closer inspection reveals that these peaks correspond to register transitions not fully mitigated by the implemented countermeasures.

In comparison, the original code from [DDM⁺22] exhibits noticeable peaks exceeding the ± 4.5 threshold with as few as 10K traces for the $d = 1$ case. Upon examining the final assembly code, we observed that their register-cleaning mechanism (e.g., `mov Rd, 0`) does not consistently clear registers containing secret-dependent values. However, their reported T-test results suggest that the fix is effective in suppressing leakage at this level. Similar to our implementation, their design also encounters leakage when the number of traces exceeds 1 million.

A well-known mitigation strategy for transition leakage—especially in the

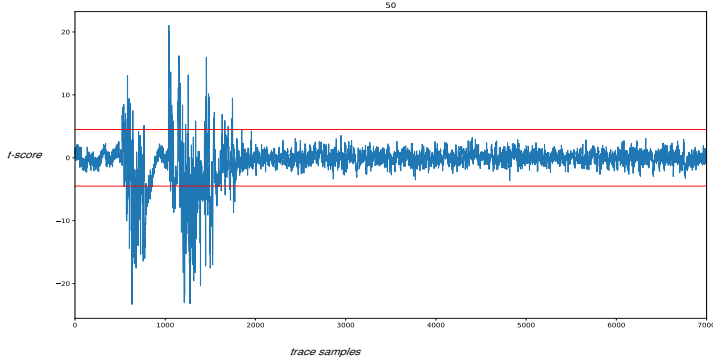


Figure 6.10: T-test on the initialization of ASCON with $n = 2$ shares. Randomness for sharing the key and nonce is set to zero. The x-axis denotes clock cycles and the y-axis is T-test output.

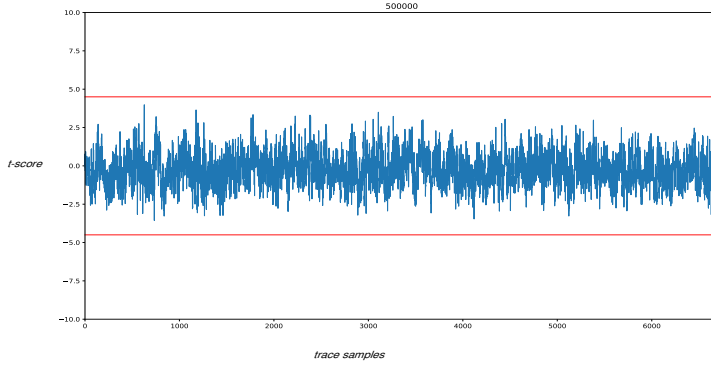


Figure 6.11: T-test applied to the initialization of ASCON (12 rounds) with $n = 2$ shares using the proposed deterministic masking for $d = 1$ security. Results are based on 500K traces.

presence of hidden architectural registers—is to increase the masking order. In our experiments, the $d = 2$ implementation showed significantly more robustness. Even without additional transition-focused countermeasures, it did not exhibit visible leakage in our T-test measurements.

Second-order T-test analysis. To evaluate the effectiveness of our $d = 2$ implementation, we performed a second-order T-test targeting selected regions of the trace. Figure 6.12 compares the test outcomes for both the $d = 1$ and $d = 2$ cases, focusing on operations from the same round.

The heatmap on the left reveals prominent out-of-threshold spots (green) in the $d = 1$ case, indicating detectable leakage even with only 30K traces, confirming that first-order protection is insufficient against second-order attacks. In contrast, the heatmap on the right, corresponding to our $d = 2$ implementation, shows no such peaks with 1 million traces, confirming the absence of detectable second-order leakage in our setup.

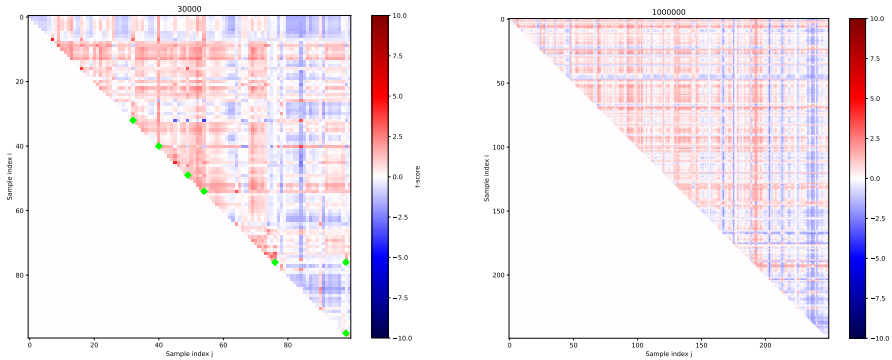


Figure 6.12: Second-order T-test results focused on round operations. Left: $n = 2$ shares ($d = 1$ protection), with green spots indicating values outside the ± 4.5 threshold. Right: $n = 3$ shares ($d = 2$ protection), showing no leakage. X- and Y-axes correspond to selected sample indices.

Chapter 7

Clustering Approach for Higher-Order Deterministic Masking

This chapter builds on Chapter 6 and is based on work currently under submission [JSB25]. We present a deterministic scheme for securely computing finite-field multiplications without requiring additional (fresh) online randomness. Our starting point is the first-order Threshold Implementation (TI) methodology of Nikova et al. (see Section 2.9), which achieves first-order glitch resistance via non-completeness but does not directly extend to higher orders in a fully deterministic manner.

Our main technical tool is the concept of *share clustering*, inspired by finite affine geometry. Clustering enables us to enforce security and (when desired) uniformity constraints simultaneously, yielding deterministic multiplication gadgets beyond first order. We further show how the resulting gadgets can be used to obtain higher-order deterministic masking for ASCON. By eliminating online randomness inside the protected circuit, our approach broadens the practical applicability of higher-order masking in resource-constrained environments.

7.1 Introduction

We continue the deterministic masking line of Chapter 6, where randomness is confined to the *initial* sharing of the circuit inputs. Our deterministic masking of ASCON in this chapter is analyzed in the standard probing model. At the gadget level, however, we design and prove our deterministic multiplication gadgets in the *glitch-extended* probing model, in order to capture transient hardware effects.

7.1.1 Higher-Order Deterministic AND Gadgets

In masked implementations, most online randomness is consumed by nonlinear operations; the simplest such primitive is the AND (multiplication in $\mathbb{F}_{2^n}$) function on two secrets. As a foundational result, Ishai et al. [ISW03] introduced a masked AND gadget that uses $n(n-1)/2$ random values to achieve probing security order $d = \lfloor (n-1)/2 \rfloor$. Subsequent works increased the achievable security to $d = n-1$ [RP10], while reducing the randomness requirement to $n^2/4 + \mathcal{O}(n)$ [BBP⁺16]. Meanwhile, Belaïd et al. [BBP⁺16] proved that achieving $d = n-1$ *without* online randomness is impossible.

This impossibility result does not rule out deterministic gadgets at smaller security orders. Indeed, Nikova et al. [NRR06] constructed a deterministic ($n = 4, d = 1$) AND gadget. Whether deterministic AND gadgets exist for larger parameter pairs (n, d) (with $d > 1$), while maintaining glitch resistance and uniformity, has remained an open question in masking theory.

7.1.2 Our Contribution

We address the longstanding open problem of constructing a deterministic masked AND operation that achieves $d > 1$ probing security, while preserving both glitch resistance and output uniformity. In contrast to prior approaches that rely on exhaustive search, our method systematically clusters shares using tools from finite affine geometry.

In terms of computational complexity, our multiplication gadget requires only n field multiplications and achieves $d = \sqrt{n} - 1$ (glitch-extended) probing security. For a detailed comparison of AND algorithms, see Table 7.1. In our construction, n must be the square of a prime power. Therefore, for $d < 10$, our method applies to the following parameter sets (n, d) : $(4, 1)$, $(9, 2)$, $(16, 3)$, $(25, 4)$, $(49, 6)$, $(64, 7)$, and $(81, 8)$.

In addition, we propose a deterministic masking circuit for the map χ_m that achieves up to $d = \sqrt{n} + 1$ (standard) probing security. We verify the gadget's bijectivity for $n = 4$ through exhaustive enumeration. This $n = 4$ case is particularly relevant, as it provides third-order security and applies to χ_m in ASCON [DEMS21, TMC⁺24], as well as other widely used cryptographic primitives such as SHA-3 [oST15] and Xoodoo [DHP⁺20]. Moreover, we have implemented¹ and analyzed the side-channel security of ASCON using this third-order secure χ_m gadget.

For a comparison of our χ_m masking with existing approaches, see Table 7.2. Table 7.3 compares deterministic methods for masking ASCON.

¹<https://github.com/vahid-jahandideh/deterministic-masking-ascon.git>

AND Algorithm	(A)	(B)	(C)	(D)	(E)	(F)	(G)	(H)
ISW [ISW03]	n	$\frac{n(n-1)}{2}$	$\lfloor \frac{n-1}{2} \rfloor$	✓	$2(n^2 - n)$	n^2	×	✓
Rivain and Prouff [RP10]	n	$\frac{n(n-1)}{2}$	$n - 1$	✓	$2(n^2 - n)$	n^2	×	✓
Belaïd et al. [BBP ⁺ 16]	n	$\approx \frac{n^2+2n}{4}$	$n - 1$	✓	$\approx \frac{7n^2}{4}$	n^2	×	✓
Nikova et al. [NRR06]	4	0	1	✓	24	4	✓	×
De Cnudde et al. [DCBR ⁺ 16]	6	7	2	✓	29	36	✓	×
SAND-DN [This work]	n	0	$\sqrt{n} - 1$	×	$2(n - \sqrt{n})$	n	✓	×
SAND-DU [This work]	n	0	$\sqrt{n} - 1$	✓	$2\sqrt{n}(n-1)$	n	✓	×

Table 7.1: Comparison of masking algorithms for secure computation of the AND operation. The columns indicate: (A) number of shares; (B) required online randomness; (C) probing security order; (D) uniform output; (E) number of XORs; (F) number of field multiplications; (G) glitch resistance; (H) whether the security proof avoids assuming input share independence.

Candidate	(A)	(B)	(C)	(D)	(E)	(F)	(G)
Unprotected [Dae95]	1	0	0	odd m	✓	$2m$	m
Bilgin et al. [BDN ⁺ 14]	4	0	1	5	✓	80	30
Daemen et al. [DDE ⁺ 20]	2	1 bit	1	5	×	22	20
Shahmirzadi and Moradi [SM21]	3	0	2	5	✓	80	45
$\mathbb{S}_{\chi_m}$ [This work]	4	0	3	5 & 3	✓	85 ($m = 5$)	80 ($m = 5$)

Table 7.2: Comparison of dedicated masked algorithms for the maps χ_m . The columns indicate: (A) sharing order n ; (B) required online randomness; (C) probing security order; (D) supported values of m ; (E) uniform output (i.e., mn -bit permutation); (F) number of XORs; (G) number of ANDs.

A central component of our approach is a novel grouping strategy grounded in finite geometry. This technique has the potential to benefit other masking schemes beyond the specific constructions proposed in this work. While Petrides [Pet19] also explored a grouping-based strategy, their method was limited to achieving second-order probing security, whereas our approach generalizes beyond this constraint.

7.2 Share Clustering

Before going into the design of masking schemes, we introduce a fundamental concept for grouping the shares in an n -sharing. This new method lays the

Candidate	(A)	(B)	(C)	(D)
Daemen et al. [DDE ⁺ 20]	2	✓	1	Verified for one round (using maskVerif [BBD ⁺ 15a])
Gigerl et al. [GMSP23]	3	✓	2	Verified for one round (using CoCo [GHP ⁺ 21])
Jahandideh et al. [JMB25b]	2,3	✓	1,2	Proved for multiple rounds
This work	4	✓	3	Proved using the composition theorem of [JMB25b]

Table 7.3: Comparison of deterministic masking methods for ASCON. The columns indicate: (A) sharing order; (B) no requirement for online randomness; (C) probing security order; (D) type of security proof.

groundwork for ensuring both probing security and uniformity. We focus on the case where n is a perfect square (the smallest example being $n = 4$). Consequently, we can write $n = s^2$ and partition the n shares into s lists (or *multi-shares*), each containing s elements. We refer to each such partition as a *cluster*.

Definition 7.1 (Cluster). Let $\mathbf{X} = [X_0, \dots, X_{n-1}]$ be an n -sharing. A *cluster* is a partition $\mathcal{C} = [\mathbf{A}_0, \dots, \mathbf{A}_{s-1}]$ of $\mathbf{X}$ into s *multi-shares*, each containing exactly s elements. In other words, $\mathcal{C}$ satisfies:

- Each multi-share $\mathbf{A}_i$ has s elements: $|\mathbf{A}_i| = s$.
- The union of all multi-shares equals the original sharing:

$$\bigcup_{i=0}^{s-1} \mathbf{A}_i = \mathbf{X}.$$

- Any two distinct multi-shares $\mathbf{A}_i$ and $\mathbf{A}_j$ are disjoint:

$$\mathbf{A}_i \cap \mathbf{A}_j = \emptyset \quad \text{for } i \neq j.$$

Example 7.1 ($n = 4$). Let $\mathbf{X} = [X_0, X_1, X_2, X_3]$ be a 4-sharing (so $s = 2$). One possible cluster is

$$\mathcal{C}^0 = [\mathbf{A}_0^0 = [X_0, X_1], \mathbf{A}_1^0 = [X_2, X_3]]. \quad (7.1a)$$

Another distinct cluster for the same set of shares is

$$\mathcal{C}^1 = [\mathbf{A}_0^1 = [X_0, X_2], \mathbf{A}_1^1 = [X_1, X_3]]. \quad (7.1b)$$

We will see that these two clusters are *Maximally Non-Overlapping (MNO)*.

Definition 7.2 (MNO Clusters). Two clusters $\mathcal{C}^1$ and $\mathcal{C}^2$ are called *MNO* if, for every multi-share $\mathbf{A}_i^1$ in $\mathcal{C}^1$ and every multi-share $\mathbf{A}_j^2$ in $\mathcal{C}^2$, their intersection contains exactly one element:

$$|\mathbf{A}_i^1 \cap \mathbf{A}_j^2| = 1 \quad \text{for all } 0 \leq i, j < s.$$

In Example 7.1, observe how

$$\mathbf{A}_0^0 \cap \mathbf{A}_0^1 = [X_0] \quad \text{and} \quad \mathbf{A}_0^0 \cap \mathbf{A}_1^1 = [X_1],$$

demonstrating the MNO property for $\mathcal{C}^0$ and $\mathcal{C}^1$. Moreover, a third cluster,

$$\mathcal{C}^2 = [\mathbf{A}_0^2 = [X_0, X_3], \mathbf{A}_1^2 = [X_1, X_2]], \quad (7.1c)$$

can be formed such that any two clusters in the set

$$\mathfrak{C} = \{\mathcal{C}^0, \mathcal{C}^1, \mathcal{C}^2\}$$

are MNO.

Definition 7.3 (Simultaneous MNO Clusters). Let $\mathfrak{C}$ be a collection of clusters, $\mathfrak{C} = \{\mathcal{C}^0, \dots, \mathcal{C}^{|\mathfrak{C}|-1}\}$. The clusters $\mathcal{C}^0, \dots, \mathcal{C}^{|\mathfrak{C}|-1}$ are called *simultaneous MNO* (SMNO) if every pair of clusters in $\mathfrak{C}$ is MNO.

7.2.1 Maximum Number of SMNO Clusters

For the smallest non-trivial case $n = 4$, it is impossible to construct more than three SMNO clusters. This observation motivates an investigation into the maximum number of SMNO clusters for larger square values of n .

Example 7.2 ($n = 9$). To visualize potential cluster arrangements for $n = 9$, we place the nine shares in a 3×3 square, as shown in Figure 7.1. This representation aids in identifying how shares can be grouped into clusters that satisfy the SMNO property.

In this 9-sharing scenario, one can construct four SMNO clusters by taking rows, columns, diagonals, and anti-diagonals of the share square. Specifically, these clusters are given by:

$$\mathcal{C}^0 = [X_0, X_1, X_2], [X_3, X_4, X_5], [X_6, X_7, X_8], \quad (7.2a)$$

$$\mathcal{C}^1 = [X_0, X_3, X_6], [X_1, X_4, X_7], [X_2, X_5, X_8], \quad (7.2b)$$

$$\mathcal{C}^2 = [X_0, X_4, X_8], [X_1, X_5, X_6], [X_2, X_3, X_7], \quad (7.2c)$$

$$\mathcal{C}^3 = [X_0, X_5, X_7], [X_1, X_3, X_8], [X_2, X_4, X_6]. \quad (7.2d)$$

Other Values of n . For $n = 4^2 = 16$, one can form five SMNO clusters (see Figure 7.2). In contrast, for $n = 6^2 = 36$, we did not find more than three SMNO clusters. The number of SMNO clusters is therefore not a straightforward (or even monotonic) function of n .

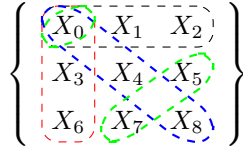


Figure 7.1: First multi-share of each of the four SMNO clusters of $\mathbf{X}$ for $n = 9$.

Remark 7.4. For any $n = s^2$ with $s \geq 2$, the clusters formed by rows, columns, and diagonals of an $s \times s$ arrangement always exist and are SMNO. Hence, for each square n , there are at least three SMNO clusters. It is easily seen that the anti-diagonal cannot be added for all s to have a set of SMNO clusters, e.g., when $s = 2$ or $s = 4$.

In the following sections, we will make use of *all* the SMNO clusters in our design. We, therefore, want to establish the maximum number of such clusters and to understand how they can be constructed.

A systematic clustering structure. One convenient way to form clusters in an n -sharing is to lay out the shares in an $s \times s$ square, where $s = \sqrt{n}$. Let us denote the row-based cluster (indexed by 0) as follows:

$$\mathbf{A}_j^0 = \{X_l \mid l = i + js, 0 \leq i \leq s - 1\} \quad \text{for } 0 \leq j \leq s - 1.$$

Subsequent clusters are generated using

$$\mathbf{A}_j^h = \{X_l \mid l = is + \text{mod}(hi + j - i, s), 0 \leq i \leq s - 1\},$$

where $h \in \{1, \dots, s\}$ identifies the cluster, $\text{mod}(a, b)$ is the remainder of a divided by b , and $0 \leq j \leq s - 1$. This construction yields $s + 1$ potential clusters² for each s .

Lemma 7.5. *In an n -sharing with $s = \sqrt{n}$ prime, there are exactly $s + 1$ SMNO clusters following this construction. Furthermore, if s is composite with prime factorization $s = \prod_{i=1}^r p_i^{e_i}$ (where $p_1 < p_2 < \dots < p_r$), then there exist at least $p_1 + 1$ clusters that are SMNO.*

Proof. The proof here gives a construction to obtain SMNO clusters. We arrange shares in a square of side length s (as illustrated in Figure 7.1). The rows form one cluster. For this cluster, the shares $0 \leq i \leq s - 1$ in the j -th multi-share are denoted by:

$$\mathbf{A}_j^0[i] = X_{i+js}. \quad (7.3)$$

Subsequent clusters are formed as follows:

$$\mathbf{A}_j^h[i] = X_{is + \text{mod}(hi + j - i, s)}. \quad (7.4)$$

Here, each $h \in \{1, \dots, s\}$ labels a cluster.

Clusters covering $\mathbf{X}$. We show that each cluster covers all shares in $\mathbf{X}$. For a given h , any share X_l with $0 \leq l < s^2$ must be in one multi-share of this cluster. Setting $l = is + \text{mod}(hi + j - i, s)$ yields $i = \lfloor l/s \rfloor$ and $\text{mod}(l, s) = \text{mod}(hi + j - i, s)$, because $\text{mod}(hi + j - i, s) < s$. This leads to a unique $j = \text{mod}(l - hi + i, s)$ for each h .

SMNO property. Using the clusters constructed above, we prove that each multi-share from cluster h_1 intersects exactly once with each multi-share from cluster h_2 : $|\mathbf{A}_{j_1}^{h_1} \cap \mathbf{A}_{j_2}^{h_2}| = 1$. For any two multi-shares j_1 and j_2 , there exists a pair of indices (i_1, i_2) satisfying:

$$i_1 s + \text{mod}(h_1 i_1 + j_1 - i_1, s) = i_2 s + \text{mod}(h_2 i_2 + j_2 - i_2, s). \quad (7.5)$$

Applying $\lfloor \cdot / s \rfloor$ to both sides gives $i_1 = i_2$. Replacing i_1 and i_2 with i , we get:

$$is + \text{mod}(h_1 i + j_1 - i, s) = is + \text{mod}(h_2 i + j_2 - i, s). \quad (7.6)$$

Solving for i yields

$$i = (h_1 - h_2)^{-1}(j_2 - j_1) \pmod{s}.$$

²Not necessarily all SMNO.

With s being prime, $(h_1 - h_2)^{-1}$ is always unique, leading to: $|\mathbf{A}_{j_1}^{h_1} \cap \mathbf{A}_{j_2}^{h_2}| = 1$.

Furthermore, the rows cluster and each cluster defined by (7.4) are MNO. The r -th multi-share in the rows cluster, which is the r -th row, contains shares from indices rs to $rs + s - 1$. Each multi-share from other clusters includes exactly one share in the range rs to $rs + s - 1$, located at index $i = r$. Thus, each multi-share from the rows cluster intersects exactly once with each multi-share from other clusters.

There are no other SMNO clusters. Using this method, there are no more clusters. For the sake of contradiction, assume there exists an unaccounted cluster $\mathcal{C}'$, and $\mathbf{B}'$ is one of its multi-shares. Imagine shares in a square of side length s . $\mathbf{B}'$ must have one intersection with the first row and one with the second row. Let c_1 and c_2 be the respective column indices of these intersections. Using the structure defined in (7.4), we can show that in the cluster with $h = \text{mod}(c_2 - c_1 + 1, s)$, the first two members of c_1 -th multi-share coincide with $\mathbf{B}_0$. Consequently,

$$|\mathbf{A}_{c_1}^{\text{mod}(c_2 - c_1 + 1, s)} \cap \mathbf{B}_0| > 1.$$

This contradiction indicates that no additional cluster $\mathcal{C}'$ exists outside the defined structure.

Composite s . For non-prime s , we seek an as large as possible set $\mathbf{T}$ where $\text{gcd}(h_1 - h_2, s) = 1$ for any distinct $h_1, h_2 \in \mathbf{T}$, ensuring that $h_1 - h_2$ is invertible modulo s . We always have $0, 1 \in \mathbf{T}$. A third value h is in $\mathbf{T}$ if $\text{gcd}(h, s) = 1$ and $\text{gcd}(h - 1, s) = 1$. Hence, if 2 does not divide s , then $2 \in \mathbf{T}$. Following up on this, let p be the smallest prime factor of a composite s . Then all values up to the smallest p are included in $\mathbf{T}$, incrementally. Let $k < p$ be such that $0, 1, \dots, k - 1 \in \mathbf{T}$. Then we need $\text{gcd}(k - 0, s) = \text{gcd}(k - 1, s) = \text{gcd}(k - 2, s) = \dots = \text{gcd}(k - (k - 1), s) = 1$. That $\text{gcd}(k, s) = 1$ is obvious, since $k < p$. All other gcds in the above line are because $k - 1, k - 2, \dots, 1 \in \mathbf{T}$. Any value h in $\mathbf{T}$ defines an MNO cluster per (7.4). That $h_1 - h_2$ is invertible modulo s for distinct $h_1, h_2 \in \mathbf{T}$ ensures that these clusters are SMNO. $\square$

Example 7.3. For instance, with $n = 16$, we have $s = 2^2$ and $p_1 = 2$. Hence, the number of SMNO clusters, that we can create with the method from Lemma 7.5 is three. In general, since 2 is the smallest prime, we have at least three clusters that are SMNO for any value s . This gives a different argument to the one in Remark 7.4 for having at least three SMNO clusters. (We see, in Figure 7.2, that we can create five SMNO clusters for $n = 16$.) $\square$

The SMNO clusters introduced here play a central role in our subsequent discussion. By organizing shares according to SMNO clusters, we achieve desirable properties related to both probing security and uniformity.

Interestingly, these constructions also connect to finite affine planes in *finite affine geometry*, further enriching the theoretical perspective on share clustering.

7.2.2 Finite Affine Geometry

In this section, we show how to view SMNO clusters through finite affine geometry. For an introduction to the topic see [Bat97]. If the number of shares is $n = s^2$ with s prime, then a maximum set of clusters that is SMNO constitutes a finite affine plane of prime order s . In such a plane, there are s^2 points and $s^2 + s$ lines of s points each. Each point, moreover, lies on $s + 1$ lines. These statements follow from the axioms for a finite affine plane. The three axioms for a finite affine plane are:

1. For every two distinct points, there is a unique line that contains both;
2. For each line L and point X not on L , there is precisely one line L' that contains X and is disjoint from L ;
3. It is possible to choose a set of four points, such that no three of them lie on a single line.

The second axiom is often called Playfair's axiom [Pla36] and can be seen as a replacement of Euclid's fifth postulate. In terminology of clusters, any class of s parallel lines forms a cluster. Any two lines L, L' from different sets of parallel lines intersect. Hence the number of SMNO clusters is the number of classes of parallel lines. For example, see Figure 7.2, where five classes of parallel lines (each coloured differently).

Proof of the properties of a finite affine plane for SMNO clusters.

The first axiom that there is a unique line that contains both distinct points, is just a combination of having the union of all multi-shares being $\mathbf{X}$ and having all clusters be SMNO; $|\mathbf{A}_i^k \cap \mathbf{A}_j^l| \leq 1$ for all i, j, k, l , provided we do not have $i = j$ and $k = l$. The second axiom is already satisfied within a single cluster if the first line is $\mathbf{A}_i^k$, and $X \notin \mathbf{A}_i^k$, then there is some $j \neq i$ such that $X \in \mathbf{A}_j^k$. The third axiom works by taking the set of four points to be

$$\{\mathbf{A}_0^0[1], \mathbf{A}_0^0[2], \mathbf{A}_1^0[1], \mathbf{A}_2^0[1]\}.$$

By design, these are not all on any line $\mathbf{A}_j^0$, since $\mathbf{A}_j^0 \cap \mathbf{A}_0^0 = \emptyset$ for all $j \neq 0$. Furthermore, these four points are not all on any line $\mathbf{A}_j^k$ with $k \neq 0$, since then $|\mathbf{A}_0^0 \cap \mathbf{A}_j^k| \geq 2$. $\square$

Remark 7.6. We see, from this method, that we need to have a square number of shares to obtain a maximal number of SMNO clusters. A heuristic way to consider this, is by seeing that for, say, 3rd order security, we need to have at least 3 rows and at least 3 columns. Therefore, the minimum number of shares is 9.

It is an open conjecture in the field of finite geometry that a finite plane of order s (such that each line has s points) exists if and only if s is a prime power. In Lemma 7.5, s is a prime number, so this follows the theory. As described in [LM98], we have:

Proposition 7.7. *If s is any prime power, there exists a finite affine plane of s^2 points, hence we have at least $s + 1$ SMNO clusters.*

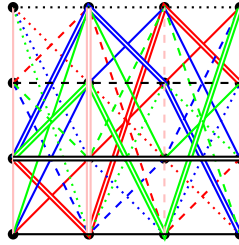


Figure 7.2: Affine plane of order 4, or, multi-shares of each of the five SMNO clusters for $n = 16$.

The method to obtain them, however does not follow the descending diagonal construction in Lemma 7.5, as that only works for prime numbers and gives a lower bound for the number of SMNO clusters for composite numbers as described there. It does work similar, by writing down algebraic expressions for the lines in the finite plane. We present an example for $s = 4$ of the five SMNO clusters that we can have in Figure 7.2. Here, all lines of a single color are parallel (they do not intersect in a point). The points are coordinates in $\mathbb{F}_4 \times \mathbb{F}_4$ with at the both axes, $0, 1, X, X + 1$, starting from bottom left. All lines are of one of the following forms $\ell_{\lambda,b} = \{(x,y) \in \mathbb{F}_4^2 \mid y = \lambda x + b\}$ or $\ell_a = \{(a,y) \mid y \in \mathbb{F}_4\}$ (with $\lambda, a, b, \in \mathbb{F}_4$). For example, the red double line is $\ell_{1,0}$, and the blue double line is $\ell_{X+1,1}$.

In particular it shows that our clustering approach for designing multi-shares works as well in the case where $s = \sqrt{n}$ is any prime power, instead of s being a prime number.

Remark 7.8. We did not find an equivalence between SMNO clusters and finite affine planes. It, therefore, theoretically is possible that there might be a method giving more SMNO clusters that does not use finite affine planes.

7.2.3 Non-Completeness of Multi-Shares

The clustering approach satisfies the following non-completeness property that will help in the probing security proofs of the gadgets later in this chapter.

Lemma 7.9. *Let $\mathbf{X}$ be an n -sharing with $n = s^2$. Suppose the adversary probes all-but-one multi-shares of a single cluster. Then, if the total number of probed multi-shares is less than $2s - 1$, the adversary cannot recover all n shares of $\mathbf{X}$.*

Proof. By hypothesis most $s - 1$ of multi-shares of a single cluster are covered. Thus we miss precisely s shares (the last multi-share of that cluster). By the MNO property, each other multi-share contains at most one of these s shares. Hence, fewer than $2s - 1$ probed multi-shares will not cover all n shares. $\square$

7.2.4 Linear Independence of Clusters

Clusters constructed for odd prime $s = \sqrt{n}$ exhibit an algebraic structure that is instrumental in proving the uniformity of the output shares in our proposed gadgets. By Lemma 7.5, there are $s + 1$ SMNO clusters when s is prime. Each cluster $\mathcal{C}^h$ consists of s multi-shares, and each multi-share $\mathbf{A}_j^h$ contains exactly s individual shares from the n -sharing $\mathbf{X}$. We use the notation $\oplus \mathbf{A}_j^h$ to denote the bitwise sum (i.e., XOR) of all shares in the multi-share $\mathbf{A}_j^h$. For instance, when $n = 9$, the expression $\oplus \mathbf{A}_0^0$ equals $X_0 \oplus X_1 \oplus X_2$.

Key observation: maximal linear independence. Consider the set of sums

$$\left\{ \oplus \mathbf{A}_j^h \mid 0 \leq h \leq s, 0 \leq j \leq s - 1 \right\},$$

which forms $s(s + 1)$ parity equations. A crucial property is that these sums exhibit *maximal* linear independence, with only some dependencies introduced

by the fact that the total sum of each cluster is the same. Concretely, for each cluster $\mathcal{C}^h$, we have

$$\bigoplus \mathcal{C}^h = \bigoplus_{j=0}^{s-1} \left(\bigoplus \mathbf{A}_j^h \right) = \bigoplus_{i=0}^{n-1} X_i = X,$$

where X is secret of $\mathbf{X}$. Consequently, for any two clusters h_1 and h_2 , one finds

$$\bigoplus_{j=0}^{s-1} \left(\bigoplus \mathbf{A}_j^{h_1} \right) = \bigoplus_{j=0}^{s-1} \left(\bigoplus \mathbf{A}_j^{h_2} \right).$$

Thus, each additional cluster beyond the first introduces one dependent relation among the sums. As there are $s + 1$ clusters, the number of such dependencies is s . Therefore, the net rank of the system of $s(s + 1)$ parity equations is

$$(s + 1)s - s = s^2 = n.$$

Example 7.4. (*Case $n = 9$, $s = 3$*) Recall the clusters from (7.2). Since $s = 3$ is an odd prime, we expect a total rank of 9. We construct the following set of 9 independent equations, omitting one relation from each new cluster (after the first) to avoid introducing linear dependence:

$$\begin{cases} X_0 \oplus X_1 \oplus X_2 = \oplus \mathbf{A}_0^0, & X_3 \oplus X_4 \oplus X_5 = \oplus \mathbf{A}_1^0, & X_6 \oplus X_7 \oplus X_8 = \oplus \mathbf{A}_2^0, \\ \textcolor{red}{X}_0 \oplus \textcolor{red}{X}_3 \oplus \textcolor{red}{X}_6 = \oplus \mathbf{A}_0^1, & \textcolor{red}{X}_1 \oplus \textcolor{red}{X}_4 \oplus \textcolor{red}{X}_7 = \oplus \mathbf{A}_1^1, & \textcolor{blue}{X}_0 \oplus \textcolor{blue}{X}_4 \oplus \textcolor{blue}{X}_8 = \oplus \mathbf{A}_0^2, \\ \textcolor{blue}{X}_1 \oplus \textcolor{blue}{X}_5 \oplus \textcolor{blue}{X}_6 = \oplus \mathbf{A}_1^2, & \textcolor{green}{X}_0 \oplus \textcolor{green}{X}_5 \oplus \textcolor{green}{X}_7 = \oplus \mathbf{A}_0^3, & \textcolor{green}{X}_1 \oplus \textcolor{green}{X}_3 \oplus \textcolor{green}{X}_8 = \oplus \mathbf{A}_1^3. \end{cases}$$

These nine equations are linearly independent, giving the system a rank of 9. $\square$

General case for odd prime s . This linear independence argument applies for all odd prime powers $s = p^k$. The formal statement appears below.

Lemma 7.10. *Let s be an odd prime power, and let $n = s^2$. Consider all sums*

$$\bigoplus \mathbf{A}_j^h \quad \text{for } 0 \leq h \leq s, \quad 0 \leq j \leq s - 1,$$

obtained from the SMNO clusters of an n -sharing $\mathbf{X}$. Then the rank of the corresponding system of parity relations is n .

Proof. For each share X_i in $\mathbf{X}$, we identify a specific subset of these parity sums that reconstructs X_i . Exclude the *row* cluster ($\mathcal{C}^0$) and let $\mathcal{B}_i$ be the set of all remaining multi-shares that contain X_i :

$$\mathcal{B}_i = \{ \mathbf{A}_j^h \mid 1 \leq h \leq s, 0 \leq j \leq s-1, X_i \in \mathbf{A}_j^h \}.$$

Since X_i appears in exactly one multi-share of each of these s clusters, $\mathcal{B}_i$ has s elements. Any two multi-shares in $\mathcal{B}_i$ intersect exactly in $[X_i]$, so combining all elements of $\mathcal{B}_i$ yields s copies of X_i and one copy of each additional share in the relevant rows. Let j_0 be such that $X_i \in \mathbf{A}_{j_0}^0$ in the row cluster. Because s is odd, we obtain

$$X_i = \left(\bigoplus_{\substack{j=0 \\ j \neq j_0}}^{s-1} \oplus \mathbf{A}_j^0 \right) \oplus \left(\bigoplus_{\mathbf{A}_j^h \in \mathcal{B}_i} \oplus \mathbf{A}_j^h \right). \quad (7.7)$$

Since each X_i can be expressed in this manner, the system of all sums $\bigoplus \mathbf{A}_j^h$ must have rank at least n . On the other hand, there are only n variables total, so the rank is exactly n . $\square$

Even Prime Case ($s = 2$). For example, when $s = 2$, the rank of the corresponding system of parity sums is 3, and for $s = 4$, it is 9. However, the statement of Lemma 7.10 for odd prime powers is sufficient for our constructions in the next section.

Example 7.5. (*Detail for $n = 9$, X_5*) Referring again to the clusters in (7.2), consider the share X_5 . The set

$$\mathcal{B}_5 = \{ \mathbf{A}_2^1, \mathbf{A}_1^2, \mathbf{A}_0^3 \}$$

collects all multi-shares from clusters 1, 2, 3 that contain X_5 . Summing their parities,

$$\oplus \mathcal{B}_5 = (\oplus \mathbf{A}_2^1) \oplus (\oplus \mathbf{A}_1^2) \oplus (\oplus \mathbf{A}_0^3) = X_2 \oplus X_5 \oplus X_8 \oplus X_1 \oplus X_6 \oplus X_0 \oplus X_7.$$

Since X_5 lies in the row-based multi-share $\mathbf{A}_1^0$, equation (7.7) for $i = 5$ becomes

$$X_5 = (\oplus \mathbf{A}_0^0) \oplus (\oplus \mathbf{A}_2^0) \oplus (\oplus \mathcal{B}_5).$$

Hence, X_5 can be recovered from the row cluster (excluding its own row) and the parity sums in $\mathcal{B}_5$. $\square$

7.3 Higher-Order Deterministic Multiplication

With our building blocks in place, we now turn to designing gadgets that perform higher-order deterministic AND operation. We start with a simpler gadget that is glitch resistant but is not uniform.

A SAND gadget takes two n -sharings $\mathbf{X}$ and $\mathbf{Y}$ as inputs and computes an n -sharing $\mathbf{Z}$ such that the native value Z satisfies $Z = XY$.

Binomial grouping. Given input n -sharings $\mathbf{X} = [X_0, \dots, X_{n-1}]$ and $\mathbf{Y} = [Y_0, \dots, Y_{n-1}]$, for the native X and Y we can write:

$$Z = XY = \bigoplus_{i=0}^{n-1} X_i \bigoplus_{j=0}^{n-1} Y_j = \bigoplus_{i=0}^{n-1} \bigoplus_{j=0}^{n-1} X_i Y_j \stackrel{(\Gamma)}{=} \bigoplus_{k=0}^{n-1} Z_k. \quad (7.8)$$

Our goal is to partition the n^2 products $\{X_i Y_j\}$ into n disjoint *bins*, each assigned to a share Z_k . Intuitively, by carefully grouping these products into each Z_k , we aim to maximize the gadget's glitch-extended probing security order.

Maximizing the probing security order. Let $\mathbf{L}_k$ be the set of all pairs (i, j) such that $X_i Y_j$ is included in Z_k . In other words,

$$Z_k = \bigoplus_{(i,j) \in \mathbf{L}_k} X_i Y_j,$$

and the partitioning condition implies

$$\bigcup_{k=0}^{n-1} \mathbf{L}_k = \{0, \dots, n-1\}^2 \quad \text{and} \quad \mathbf{L}_k \cap \mathbf{L}_{k'} = \emptyset \quad (\text{for } k \neq k').$$

A glitch-extended probe on Z_k reveals *all* pairs $\{(X_i, Y_j) \mid (i, j) \in \mathbf{L}_k\}$. Denoting

$$\mathbf{I}_k = \{i \mid \exists j \text{ with } (i, j) \in \mathbf{L}_k\} \quad \text{and} \quad \mathbf{J}_k = \{j \mid \exists i \text{ with } (i, j) \in \mathbf{L}_k\},$$

we have $|\mathbf{L}_k| \leq |\mathbf{I}_k| \cdot |\mathbf{J}_k|$. Summing over all k , we get

$$n^2 = \sum_{k=0}^{n-1} |\mathbf{L}_k| \leq \sum_{k=0}^{n-1} |\mathbf{I}_k| |\mathbf{J}_k|. \quad (7.9)$$

To ensure high security order, we want each $|\mathbf{L}_k|$ to be as small as possible—equivalently, we want $|\mathbf{I}_k|$ and $|\mathbf{J}_k|$ to be bounded. Setting $I = \max_{0 \leq k \leq n-1} |\mathbf{I}_k|$ and $J = \max_{0 \leq k \leq n-1} |\mathbf{J}_k|$, it follows from (7.9) that

$$n^2 \leq \sum_{k=0}^{n-1} |\mathbf{I}_k| |\mathbf{J}_k| \leq n I J \leq n (\max\{I, J\})^2 \implies \sqrt{n} \leq \max\{I, J\}. \quad (7.10)$$

Thus, the minimum possible value of $\max\{I, J\}$ is $\sqrt{n}$, which corresponds to the maximum achievable glitch-extended security order.

Moreover, one can show that if this bound is met with equality (i.e., $\max\{I, J\} = \sqrt{n}$), then all sets $\mathbf{L}_k$ must have equal size.

A clustering-based construction. To achieve the bound $\max\{I, J\} = \sqrt{n}$, we employ the SMNO clustering approach introduced in Section 7.2. Recall that n is assumed to be a perfect square, say $n = s^2$. Following our construction over a set of SMNO clusters, we define, for each $k \in \{0, \dots, n-1\}$:

$$\mathbf{L}_k = \mathbf{I}_k \times \mathbf{J}_k,$$

where

$$\begin{aligned} \alpha &= \lfloor k/s \rfloor, & \mathbf{X}|_{\mathbf{I}_k} &= \mathbf{A}_\alpha^{h_1}(\mathbf{X}), \\ \beta &= \text{mod}(k, s), & \mathbf{Y}|_{\mathbf{J}_k} &= \mathbf{A}_\beta^{h_2}(\mathbf{Y}). \end{aligned} \quad (7.11)$$

Here, h_1 and h_2 refer to two distinct SMNO cluster types; for example, one might assign the “row” cluster to $\mathbf{X}$ and the “column” cluster to $\mathbf{Y}$. The indices α and β range over $\{0, \dots, s-1\}$, yielding $n = s^2$ unique combinations that cover the full $s \times s$ product space.

By construction, we have $|\mathbf{I}_k| = |\mathbf{J}_k| = s$, and hence:

$$I = J = s = \sqrt{n},$$

which achieves equality in the bound from (7.10).

Each output share Z_k is then defined as:

$$\begin{aligned} Z_k &= \mathbf{A}_\alpha^{h_1}(\mathbf{X}) \otimes \mathbf{A}_\beta^{h_2}(\mathbf{Y}) \\ &\stackrel{\text{def}}{=} \bigoplus_{i=0}^{s-1} \bigoplus_{j=0}^{s-1} \left(\mathbf{A}_\alpha^{h_1}(\mathbf{X})[i] \right) \left(\mathbf{A}_\beta^{h_2}(\mathbf{Y})[j] \right) \\ &= \left(\bigoplus_{i=0}^{s-1} \mathbf{A}_\alpha^{h_1}(\mathbf{X})[i] \right) \cdot \left(\bigoplus_{j=0}^{s-1} \mathbf{A}_\beta^{h_2}(\mathbf{Y})[j] \right). \end{aligned} \quad (7.12)$$

This results in an n -sharing $\mathbf{Z} = [Z_0, \dots, Z_{n-1}]$ that satisfies correctness by construction and simultaneously maximizes the probing security order under glitch-extended probes.

Remark 7.11. In our notation, $\mathbf{A}_\alpha^h(\mathbf{X})[i]$ refers to the i th share within the multi-share α of cluster h applied to $\mathbf{X}$.

7.3.1 The SAND-DN Gadget

The output $\mathbf{Z}$ of the multiplication gadget described in the previous section is not uniformly distributed. We denote this construction by SAND-DN, where “D” stands for “deterministic” and “N” denotes “non-uniform.”

Algorithm 16 formally describes the SAND-DN gadget for given cluster choices h_1 and h_2 . As an example, for $s = 2$ and $(h_1, h_2) = (0, 1)$ —following the cluster assignment in Example 7.1—the output shares of Z are:

$$\begin{cases} Z_0 = (X_0 \oplus X_1)(Y_0 \oplus Y_2), \\ Z_1 = (X_0 \oplus X_1)(Y_1 \oplus Y_3), \\ Z_2 = (X_2 \oplus X_3)(Y_0 \oplus Y_2), \\ Z_3 = (X_2 \oplus X_3)(Y_1 \oplus Y_3). \end{cases}$$

Algorithm 16 The SAND-DN gadget

Input: $\mathbf{X}, \mathbf{Y} \in (\mathbb{F}_{2^u})^n$; cluster indices h_1, h_2

Output: $\mathbf{Z} \in (\mathbb{F}_{2^u})^n$ such that $Z = XY$

```

1: for  $k = 0$  to  $n - 1$  do
2:    $\alpha \leftarrow \lfloor k/s \rfloor$ 
3:    $\beta \leftarrow \text{mod}(k, s)$ 
4:    $Z_k \leftarrow \mathbf{A}_\alpha^{h_1}(\mathbf{X}) \otimes \mathbf{A}_\beta^{h_2}(\mathbf{Y})$ 
5: return  $\mathbf{Z} = [Z_0, Z_1, \dots, Z_{n-1}]$ 

```

Correctness. To verify that the gadget correctly computes $Z = XY$, observe that the index pair $(\alpha, \beta) = (\lfloor k/s \rfloor, \text{mod}(k, s))$ for $0 \leq k < n$ systematically enumerates all combinations of multi-shares from clusters h_1 and h_2 under the $\otimes$ operation.

As a result, each product term $X_i Y_j$ (for $0 \leq i, j < n$) appears exactly once in one of the Z_k shares. Therefore,

$$\bigoplus_{k=0}^{n-1} Z_k = \bigoplus_{i=0}^{n-1} \bigoplus_{j=0}^{n-1} (X_i Y_j) = X \cdot Y,$$

which confirms that the gadget preserves the correctness of the native multiplication.

7.3.2 Properties of the Proposed SAND-DN Gadget

Probing security. In this gadget, any collection of up to $d = s - 1$ glitch-extended probes reveals only a *non-complete* subset of the input shares, due to the SMNO-based clustering. Consequently, such probes remain statistically independent of the secret values X , Y , and thus also of Z . However, a set of $d + 1$ glitch-extended probes may leak all shares of both $\mathbf{X}$ and $\mathbf{Y}$.

Concretely, consider the following set of $d + 1$ probes: $\mathbf{P} = \{Z_{\gamma s + \gamma} \mid 0 \leq \gamma < s\}$.

From (7.12), each such probe evaluates to:

$$P_\gamma = Z_{\gamma s + \gamma} = \left(\bigoplus_{i=0}^{s-1} A_\gamma^{h_1}(\mathbf{X})[i] \right) \cdot \left(\bigoplus_{i=0}^{s-1} A_\gamma^{h_2}(\mathbf{Y})[i] \right). \quad (7.13)$$

Hence, the adversary learns the partial sums:

$$\bigoplus_{i=0}^{s-1} A_\gamma^{h_1}(\mathbf{X})[i] \quad \text{and} \quad \bigoplus_{i=0}^{s-1} A_\gamma^{h_2}(\mathbf{Y})[i],$$

for all $0 \leq \gamma < s$. Summing over γ yields:

$$\bigoplus_{\gamma=0}^{s-1} \bigoplus_{i=0}^{s-1} A_\gamma^{h_1}(\mathbf{X})[i] = X, \quad \text{and} \quad \bigoplus_{\gamma=0}^{s-1} \bigoplus_{i=0}^{s-1} A_\gamma^{h_2}(\mathbf{Y})[i] = Y.$$

Therefore, if $\mathbf{X}$ and $\mathbf{Y}$ are independently shared, the gadget is secure against up to $d = s - 1$ glitch-extended probes. Each such probe reveals only a single multi-share of either $\mathbf{X}$ or $\mathbf{Y}$, and fewer than s multi-shares are insufficient to reconstruct the corresponding secret.

Non-uniformity of the output shares. For the binary field $\mathbb{F}_2$, each individual output share Z_k of the SAND-DN gadget is biased: for $0 \leq k \leq n - 1$, we have $\Pr(Z_k = 0) = \frac{3}{4}$.

Proof. From (7.12), we can write

$$Z_k = \left(\bigoplus_{i=0}^{s-1} A_{\lfloor k/s \rfloor}^{h_1}(\mathbf{X})[i] \right) \cdot \left(\bigoplus_{i=0}^{s-1} A_{\text{mod}(k,s)}^{h_2}(\mathbf{Y})[i] \right).$$

In $\mathbb{F}_2$, each of these two sums is an independent, uniform random bit, so their product is zero except when both bits are 1. Hence, the probability that $Z_k = 0$ is $\frac{3}{4}$. $\square$

This gadget primarily serves as a stepping stone toward our main construction, which will address the issue of uniformity. Its main purpose is to illustrate the core clustering strategy and how it enables higher-order probing security. In the following subsection, we build upon this foundation to construct a gadget that also ensures uniform output distribution.

7.3.3 A Uniform Deterministic Multiplication Gadget: SAND-DU

Nikova et al. [NRR06] introduced the first known example of a deterministic multiplication circuit that is first-order probing-secure, glitch-resistant, and *uniform* (see also Section 2.9). Extending this construction to higher orders has remained an open challenge. In this section, we contribute to this line of work by presenting a new gadget, denoted SAND-DU, and show that for prime power s , it achieves both *uniformity* and $(d = s - 1)$ -order glitch resistance.

Achieving a uniform $\mathbf{Z}$. The central idea is to construct an n -sharing $\mathbf{R}$ of the constant value 0 and add it to the output $\mathbf{Z}$ without compromising probing security. Our proofs rely on the principle of *non-completeness*, so each R_k is computed using only the multi-shares involved in the corresponding Z_k . This ensures that adding $\mathbf{R}$ does not allow the adversary to “complete” any missing shares of $\mathbf{X}$ or $\mathbf{Y}$.

To guarantee that $\mathbf{R}$ indeed shares the secret value 0, each share X_i (or Y_i) must appear an *even* number of times across the components of $\mathbf{R}$. This ensures that X_i (and similarly Y_i) cancels out when summing all R_k , yielding:

$$\bigoplus_{k=0}^{n-1} R_k = 0.$$

This construction ensures that each individual output share $Z_k + R_k$ is uniformly distributed. However, for the entire sharing $\mathbf{Z}$ to be uniform, the random variables R_k must be *jointly* uniform. That is, the shares of $\mathbf{R}$ must be sufficiently (linearly) independent.³ We ensure this by carefully designing the structure of $\mathbf{R}$ to avoid any unintended dependencies among its components. In the remainder of this section, we detail how $\mathbf{R}$ is constructed and integrated to obtain our SAND-DU gadget.

³If there exists a subset of $\mathbf{R}$ of size r whose sum is zero, then $\mathbf{R}$ cannot be jointly uniform.

Design of SAND-DU. Given the input n -sharings $\mathbf{X}$ and $\mathbf{Y}$, the gadget computes each output share Z_k as the XOR of two terms: a multiplication term W_k and an auxiliary term R_k . Unlike the simpler SAND-DN construction, this design uses more than two SMNO clusters. For $0 \leq k < n$, define:

$$\begin{aligned}\alpha &= \lfloor k/s \rfloor, \\ \beta &= \text{mod}(k, s), \\ R_k &= \bigoplus_{i=0}^{s-1} \left(\mathbf{A}_\alpha^0(\mathbf{X})[i] \oplus \mathbf{A}_\beta^{\alpha+1}(\mathbf{X})[i] \oplus \mathbf{A}_\alpha^0(\mathbf{Y})[i] \oplus \mathbf{A}_\beta^{\alpha+1}(\mathbf{Y})[i] \right), \quad (7.14) \\ W_k &= \mathbf{A}_\alpha^0(\mathbf{X}) \otimes \mathbf{A}_\beta^{\alpha+1}(\mathbf{Y}), \\ Z_k &= R_k \oplus W_k.\end{aligned}$$

Recall that $\mathbf{A}_j^h(\cdot)$ denotes the j -th multi-share from the SMNO cluster indexed by h . By Proposition 7.7, if s is a prime power, then $s + 1$ mutually disjoint SMNO clusters exist.

Example 7.6 (Case $n = 4$). Using the four-share SMNO clusters from Example 7.1, the output shares of SAND-DU are:

$$\begin{cases} Z_0 = (X_0 \oplus X_1)(Y_0 \oplus Y_2) \oplus X_1 \oplus X_2 \oplus Y_1 \oplus Y_2, \\ Z_1 = (X_0 \oplus X_1)(Y_1 \oplus Y_3) \oplus X_0 \oplus X_3 \oplus Y_0 \oplus Y_3, \\ Z_2 = (X_2 \oplus X_3)(Y_0 \oplus Y_3) \oplus X_0 \oplus X_2 \oplus Y_0 \oplus Y_2, \\ Z_3 = (X_2 \oplus X_3)(Y_1 \oplus Y_2) \oplus X_1 \oplus X_3 \oplus Y_1 \oplus Y_3. \end{cases}$$

□

Algorithm and correctness. Algorithm 17 presents the SAND-DU construction. Correctness—that is, ensuring $Z = XY$ —follows from two key properties:

- $\mathbf{W} = [W_0, \dots, W_{n-1}]$ is an n -sharing of XY ;
- $\mathbf{R} = [R_0, \dots, R_{n-1}]$ is an n -sharing of 0.

We first verify that $\mathbf{R}$ indeed shares the zero value, i.e., $\bigoplus_{k=0}^{n-1} R_k = 0$. We demonstrate this for the $\mathbf{X}$ terms; the steps for $\mathbf{Y}$ are analogous. From the definition of R_k :

$$\bigoplus_{k=0}^{n-1} R_k = \bigoplus_{k=0}^{n-1} \bigoplus_{i=0}^{s-1} \left(\mathbf{A}_\alpha^0(\mathbf{X})[i] \oplus \mathbf{A}_\beta^{\alpha+1}(\mathbf{X})[i] \right), \quad (7.15)$$

where $k = \alpha s + \beta$ for $0 \leq \alpha, \beta < s$.

Since there is a one-to-one correspondence between k and the pair (α, β) , we can rewrite the outer sum as a double sum:

$$\bigoplus_{k=0}^{n-1} (\cdot) = \bigoplus_{\alpha=0}^{s-1} \bigoplus_{\beta=0}^{s-1} (\cdot).$$

Substituting this into (7.15) yields:

$$\begin{aligned} \bigoplus_{k=0}^{n-1} R_k &= \left(\bigoplus_{\beta=0}^{s-1} \bigoplus_{\alpha=0}^{s-1} \bigoplus_{i=0}^{s-1} \mathbf{A}_{\alpha}^0(\mathbf{X})[i] \right) \oplus \left(\bigoplus_{\alpha=0}^{s-1} \bigoplus_{\beta=0}^{s-1} \bigoplus_{i=0}^{s-1} \mathbf{A}_{\beta}^{\alpha+1}(\mathbf{X})[i] \right) \\ &\stackrel{\text{(I)}}{=} \left(\bigoplus_{\beta=0}^{s-1} X \right) \oplus \left(\bigoplus_{\alpha=0}^{s-1} X \right) = 0. \end{aligned} \quad (7.16)$$

The identity (I) uses the fact that for any SMNO cluster h , we have:

$$\bigoplus_{j=0}^{s-1} \bigoplus_{i=0}^{s-1} \mathbf{A}_j^h(\mathbf{X})[i] = X.$$

For the multiplication component $\mathbf{W}$, we compute:

$$\begin{aligned} \bigoplus_{k=0}^{n-1} W_k &= \bigoplus_{k=0}^{n-1} \left(\mathbf{A}_{\alpha}^0(\mathbf{X}) \otimes \mathbf{A}_{\beta}^{\alpha+1}(\mathbf{Y}) \right) \\ &= \bigoplus_{\alpha=0}^{s-1} \bigoplus_{\beta=0}^{s-1} \left(\left(\bigoplus_{i=0}^{s-1} \mathbf{A}_{\alpha}^0(\mathbf{X})[i] \right) \left(\bigoplus_{i=0}^{s-1} \mathbf{A}_{\beta}^{\alpha+1}(\mathbf{Y})[i] \right) \right) \\ &= \bigoplus_{\alpha=0}^{s-1} \left(\left(\bigoplus_{i=0}^{s-1} \mathbf{A}_{\alpha}^0(\mathbf{X})[i] \right) Y \right) = \left(\bigoplus_{\alpha=0}^{s-1} \bigoplus_{i=0}^{s-1} \mathbf{A}_{\alpha}^0(\mathbf{X})[i] \right) Y = XY. \end{aligned} \quad (7.17)$$

Combining the above, we conclude that:

$$Z = \bigoplus_{k=0}^{n-1} Z_k = \left(\bigoplus_{k=0}^{n-1} W_k \right) \oplus \left(\bigoplus_{k=0}^{n-1} R_k \right) = XY.$$

This proves correctness. $\square$

Algorithm 17 The SAND-DU gadget

Input: $\mathbf{X}, \mathbf{Y} \in (\mathbb{F}_{2^u})^n$
Output: $\mathbf{Z} \in (\mathbb{F}_{2^u})^n$ such that $\mathbf{Z} = \mathbf{X} \cdot \mathbf{Y}$

```

1: for  $k = 0$  to  $n - 1$  do
2:    $\alpha \leftarrow \lfloor k/s \rfloor$ 
3:    $\beta \leftarrow \text{mod}(k, s)$ 
4:    $R_k \leftarrow 0$  ▷ Compute the zero-sharing  $R_k$ .
5:   for  $i = 0$  to  $s - 1$  do
6:      $R_k \leftarrow R_k \oplus A_\alpha^0(\mathbf{X})[i] \oplus A_\beta^{\alpha+1}(\mathbf{X})[i] \oplus A_\alpha^0(\mathbf{Y})[i] \oplus A_\beta^{\alpha+1}(\mathbf{Y})[i]$ 
▷ Compute the multiplication share  $W_k$ .
7:    $W_k \leftarrow A_\alpha^0(\mathbf{X}) \otimes A_\beta^{\alpha+1}(\mathbf{Y})$ 
8: for  $k = 0$  to  $n - 1$  do
9:    $Z_k \leftarrow W_k \oplus R_k$ 
10: return  $\mathbf{Z} = [Z_0, \dots, Z_{n-1}]$ 

```

7.3.4 Properties of the Proposed SAND-DU Gadget

Probing security. We show that the SAND-DU gadget is secure up to glitch-extended probing order $d = s - 1$, assuming the sharings $\mathbf{X}$ and $\mathbf{Y}$ are independent. Thanks to the clustering-based design, the argument is straightforward.

The key observation is that computing up to d probed values involves only a *non-complete* subset of the shares in $\mathbf{X}$ and $\mathbf{Y}$. As a result, the probed values remain statistically independent of the native secrets X and Y .

From Algorithm 17, each output share Z_k depends on two multi-shares: $A_\alpha^0(\cdot)$ and $A_\beta^{\alpha+1}(\cdot)$, where $\alpha = \lfloor k/s \rfloor$ and $\beta = \text{mod}(k, s)$. That is, each Z_k involves one multi-share from cluster 0 and one from cluster $\alpha + 1$, with $\alpha + 1 > 0$.

By Lemma 7.9, any collection of $s - 1$ multi-shares of cluster 0 and $s - 1$ multi-shares from other SMNO clusters is non-complete. Thus, for $d = s - 1$, the probed values cannot collectively involve all n shares of either input, thereby preserving the non-completeness condition and ensuring the privacy of X and Y .

Uniformity of SAND-DU. For $s \in \{2, 3\}$, one can directly verify uniformity via exhaustive enumeration. Specifically, by constructing a truth table of all possible inputs and outputs and counting the frequency of each output $\mathbf{Z}$, one checks that:

- All $\mathbf{Z}$ with secret $Z = \bigoplus_{i=0}^{n-1} Z_i = 0$ appear with the same frequency.
- All $\mathbf{Z}$ with secret $Z = 1$ also appear with the same frequency.

This shows uniformity of the output shares. For larger s , enumerating all possible inputs grows infeasible. Instead, we use an *indirect counting* method. We prove that any subset of $\mathbf{Z}$ of size $n - 1$ (denoted $\mathbf{Z}_1$) is uniformly distributed over $(\mathbb{F}_{2^u})^{n-1}$. Concretely, for each fixed realization $\mathbf{z}_1$, the number of input pairs $(\mathbf{X}, \mathbf{Y})$ that produce $\mathbf{Z}_1 = \mathbf{z}_1$ is the same, regardless of $\mathbf{z}_1$.

The main difficulty is that $\mathbf{Z}$ depends on $\mathbf{X}, \mathbf{Y}$ via a *nonlinear* mapping. To overcome this, we introduce a technique to *linearize* the relevant parity relations, allowing us to apply linear algebra arguments to compute the number of preimages for any given $\mathbf{z}_1$. Shares in $\mathbf{Z}$ are nonlinearly related to $\mathbf{X}$ and $\mathbf{Y}$. To work around this, let us fix the summations of the shares in the multi-shares of cluster 0 of $\mathbf{X}$. These summations, i.e., values $\oplus \mathbf{A}_i^0(\mathbf{X})$, are the only terms creating nonlinear dependencies. With this trick, $\mathbf{Z}$ will be linearly related to $\mathbf{X}$ and $\mathbf{Y}$, and consequently, we can determine the shape of its distribution.

Let, for $0 \leq i < s$, the sum of shares in the i -th multi-share of cluster 0 of $\mathbf{X}$ be fixed as: $\theta_i = \oplus_{j=0}^{s-1} \mathbf{A}_i^0(\mathbf{X})[j] \stackrel{\text{def}}{=} \oplus \mathbf{A}_i^0(\mathbf{X})$.

By substituting the values θ_i into (7.14), for Z_k , $0 \leq k \leq n - 1$, we have:

$$\alpha = \lfloor k/s \rfloor, \quad \beta = \text{mod}(k, s),$$

$$Z_k = \theta_\alpha \left(\oplus \mathbf{A}_\beta^{\alpha+1}(\mathbf{Y}) \right) \oplus \theta_\alpha \oplus \left(\oplus \mathbf{A}_\beta^{\alpha+1}(\mathbf{X}) \right) \oplus \left(\oplus \mathbf{A}_\alpha^0(\mathbf{Y}) \right) \oplus \left(\oplus \mathbf{A}_\beta^{\alpha+1}(\mathbf{Y}) \right). \quad (7.18)$$

The resulting system of relations is *linear* in both $\mathbf{X}$ and $\mathbf{Y}$. We now show that for each $(n - 1)$ -element set $\mathbf{Z}_1 \subset \mathbf{Z}$ with realization $\mathbf{z}_1$ and *any* values $\{\theta_0, \dots, \theta_{s-1}\}$, there are exactly $2^{u(n-s+1)}$ tuples $(\mathbf{x}, \mathbf{y})$ that map to $\mathbf{z}_1$. Recall that u is the bit-width of the underlying field.

We demonstrate that the collection of all equations of the form $\oplus \mathbf{A}_i^0(\mathbf{X}) = \theta_i$ and at most $n - 1$ of the equations Z_k are linearly independent.

In Equation (7.18), if we just care about $\mathbf{X}$, Z_k has only one parity sum: $\oplus \mathbf{A}_\beta^{\alpha+1}(\mathbf{X})$. By Lemma 7.10, we know that among $s^2 + s$ parity relations in the system of equations: $\{\oplus \mathbf{A}_\beta^{\alpha+1}(\mathbf{X}), \oplus \mathbf{A}_i^0(\mathbf{X})\}$ there are s^2 independent relations. Specifically, the following set of parity equations are linearly independent.

$$\mathcal{B}_X = \{\oplus \mathbf{A}_\beta^{\alpha+1}(\mathbf{X}), \oplus \mathbf{A}_i^0(\mathbf{X}) \mid 0 \leq \alpha, i < s, \ 0 \leq \beta < s - 1\}.$$

$\mathcal{B}_X$ is defined by omitting the last multi-share of clusters > 0 that are clusters other than the rows cluster. Therefore, we have $|\mathcal{B}_X| = s(s - 1) + s = s^2$.

Summing properly chosen Z_k equations, we build other $s - 1$ linear equations that annihilate the $\mathbf{X}$ -shares and, consequently, are independent of $\mathcal{B}_X$.

We call the set of these equations $\mathcal{B}_Y$. We will show that $s - 1$ of these equations are linearly independent of each other. $\mathcal{B}_Y$ is as follows.

$$\mathcal{B}_Y = \left\{ \bigoplus_{k=0}^{s-1} Z_k, \bigoplus_{k=s}^{2s-1} Z_k, \dots, \bigoplus_{k=s^2-s}^{s^2-1} Z_k \right\}.$$

For $0 \leq i < s$, the i -th parity relation in $\mathcal{B}_Y$ is

$$\mathcal{B}_Y[i] = \bigoplus_{k=is}^{(i+1)s-1} Z_k = \theta_i Y \oplus \theta_i \oplus X \oplus (\oplus \mathbf{A}_i^0(\mathbf{Y})) \oplus Y.$$

Note that $X = \oplus_{i=0}^{s-1} \theta_i$. Hence, $\mathcal{B}_Y[i]$ only has variables of $\mathbf{Y}$ as its unknowns. This makes $\mathcal{B}_X$ and $\mathcal{B}_Y$ linearly independent.

What remains is to show that there are $s - 1$ independent relations in $\mathcal{B}_Y$ for any set of θ_i values. If there was no term such as $\theta_i Y \oplus Y = (1 \oplus \theta_i)Y$ in the $\mathcal{B}_Y[i]$ relations, then the required independence relation was obvious, since each $\mathcal{B}_Y[i]$ is using one multi-share of $\mathbf{Y}$ and these multi-shares are disjoint (they are all from the same cluster). Addition of $(1 \oplus \theta_i)Y$ term can at most decrease the rank of $\mathcal{B}_Y$ by one. Therefore, we identified $s^2 + s - 1 = n + s - 1$ linearly independent equations for given $\{\theta_0, \dots, \theta_{s-1}\}$ and $n - 1$ values of $\mathbf{z}_1$ based on $2n$ variables in $\mathbf{X}$ and $\mathbf{Y}$. So, there $2n - (n + s - 1) = n - s + 1$ free variables in $\mathbf{X}$ and $\mathbf{Y}$. That implies there are $2^{u(n-s+1)}$ preimages for any set $\{\theta_0, \dots, \theta_{s-1}\}$ and values of $\mathbf{z}_1$. Considering only preimages of $\mathbf{z}_1$, we can conclude that for any realization $\mathbf{z}_1$, there are $2^{us} 2^{u(n-s+1)} = 2^{u(n+1)}$ preimages, a number which is independent of actual values of $\mathbf{z}_1$, and this proves the aimed uniformity. $\square$

7.4 Higher-Order Deterministic SToffoli Gadget

To demonstrate how the clustering framework generalizes to complete cipher constructions, we introduce an additional gadget. Our target cipher is ASCON, whose core nonlinearity is the χ_5 function. In the next section, we develop a full $\mathbb{S}_{\chi_5}$ gadget. Here, we present a smaller building block that corresponds to a Toffoli gate [Tof80].

A Toffoli gate maps $(X, Y, Z) \in (\mathbb{F}_{2^u})^3$ to $(X, Y, W = Z \oplus XY)$. Accordingly, an SToffoli gadget takes three n -sharings $[\mathbf{X}, \mathbf{Y}, \mathbf{Z}]$ and computes an n -sharing $\mathbf{W}$ such that: $W = Z \oplus XY$.

Construction. We leverage the clustering framework to build a $\mathcal{SToffoli}$ gadget, described in Algorithm 18. The construction uses three mutually SMNO clusters, assigning one to each input. By Remark 7.4, such a choice is always possible for any $s \geq 2$. We omit the correctness proof, as it closely parallels that of $\mathcal{SAND-DN}$.

Algorithm 18 The $\mathcal{SToffoli}$ gadget

Input: $[X, Y, Z] \in (\mathbb{F}_{2^u}^n)^3$,
cluster indices (h_X, h_Y, h_Z)
Output: $[X, Y, W] \in (\mathbb{F}_{2^u}^n)^3$ with $W = Z \oplus XY$

- 1: **for** $k = 0$ to $n - 1$ **do**
- 2: $\alpha \leftarrow \lfloor k/s \rfloor, \quad \beta \leftarrow \text{mod}(k, s)$
- 3: $A_{\alpha}^{h_Z}(W)[\beta] \leftarrow A_{\alpha}^{h_Z}(Z)[\beta]$
- 4: **for** $i = 0$ to $s - 1$ **do**
- 5: **for** $j = 0$ to $s - 1$ **do**
- 6: $A_{\alpha}^{h_Z}(W)[\beta] \leftarrow A_{\alpha}^{h_Z}(W)[\beta] \oplus (A_{\alpha}^{h_X}(X)[i] \cdot A_{\beta}^{h_Y}(Y)[j])$
- 7: **return** $[X, Y, W]$

7.4.1 Properties of the $\mathcal{SToffoli}$ Gadget

We analyze security separately in the standard and glitch-extended probing models.

Standard probing security. Let P be a probe targeting W_k , computed as $P = Z_k \oplus F$, where F depends on intermediate values. As discussed in Section 6.3.3, P can be eliminated from the adversary's view if the following hold:

1. The share Z_k does not appear in F or any other probe;
2. Not all n shares of Z appear across the probe set;
3. The sharings X , Y , and Z are independent (Definition 6.3).

If two or more probes target the same W_k , probe elimination may not apply. In this case, we conservatively assume that the adversary learns the shares used in computing F . Each such pair of probes reveals:

one share from Z and one multi-share from each of X and Y .

Since $\mathbf{X}$ and $\mathbf{Y}$ each have s multi-shares, and single probes can be eliminated, the maximum number of disjoint pairs before completeness is:

$$\left\lfloor \frac{d}{2} \right\rfloor < s \implies d \leq 2s - 1.$$

Each probe also reveals at most one share of $\mathbf{Z}$, preserving non-completeness for $\mathbf{Z}$ as well.

For $n = 4$ ($s = 2$), this yields $d \leq 3$, which is optimal: no masking scheme can exceed $d = n - 1$ probing security.⁴ For $n = 9$ ($s = 3$), the bound becomes $d \leq 5$.

Glitch-extended probing security. In the glitch-extended setting, a probe on W_k reveals not only one share of $\mathbf{Z}$ but also one multi-share from each of $\mathbf{X}$ and $\mathbf{Y}$. Thus, each probe directly recovers three components. To preserve non-completeness, we require:

$$d < s \implies d \leq s - 1.$$

Hence, the gadget achieves glitch-extended probing security of order $d = s - 1$.

The SAND-XOR gadget. A Toffoli gate outputs three values $(X, Y, Z \oplus XY)$, copying the first two inputs to the output directly. In some applications (including those in the next section), only the combination $Z \oplus XY$ is needed at the output. Thus, we define the SAND-XOR gadget simply by omitting the X and Y outputs from the $\mathbb{S}$ Toffoli gadget. Formally, it maps three n -sharings $[\mathbf{X}, \mathbf{Y}, \mathbf{Z}]$ into one n -sharing for $[\mathbf{Z} \oplus \mathbf{X}\mathbf{Y}]$, using the same cluster arrangement and inheriting the security properties just described.

7.5 Case Study on χ_m

Building on the foundations established in previous sections, we now construct a deterministic, higher-order probing-secure masking scheme for the maps χ_m . Our design achieves a standard probing security order of $d = s + 1 = \sqrt{n} + 1$.

Due to computational limitations, we verify uniformity for the specific cases $(n = 4, m = 5)$ and $(n = 4, m = 3)$. These configurations correspond to the smallest square number of shares and match the instantiations of χ_m used in widely deployed ciphers such as ASCON [DEMS21], SHA-3 [oST15], and Xoodoo [DHP⁺20]. Importantly, for $n = 4$, our construction achieves $d =$

⁴An adversary with n probes can fully reconstruct any n -sharing.

$\sqrt{n} + 1 = 3$, which is particularly relevant, as this is the maximum achievable probing order—even for masking schemes that allow online randomness.

Uniformity challenge in masking χ_m . When m is odd, the map χ_m is a permutation, which implies that its masked counterpart $\mathbb{S}_{\chi_m}$ could, in principle, also act as a permutation over the mn -bit domain. However, a straightforward instantiation using the multiplication gadget of Nikova et al. [NRR06] (for $n = 4$) does *not* yield a bijective realization of $\mathbb{S}_{\chi_m}$. Earlier attempts at constructing uniform, deterministic masking gadgets for χ_m were similarly unsuccessful [BDN⁺14].

To address this, Daemen [Dae17] introduced the “Changing of the guards” technique, which leverages randomness from neighboring $\mathbb{S}_{\chi_m}$ instances to help uniformize the output. Independently, Shahmirzadi and Moradi [SM21] conducted an exhaustive search that yielded a $\mathbb{S}_{\chi_5}$ gadget achieving both first- and second-order probing security and uniformity (bijectivity).

In this work, we extend these ideas by constructing $\mathbb{S}_{\chi_5}$ and $\mathbb{S}_{\chi_3}$ gadgets that achieve *third-order* probing security while preserving uniformity for the case $n = 4$.

7.5.1 The $\mathbb{S}_{\chi_m}$ Gadget

The map χ_m . For m binary inputs $(X^0, \dots, X^{m-1})$, the transformation χ_m defines the outputs $(Y^0, \dots, Y^{m-1})$ as: $Y^i = X^i \oplus (1 \oplus X^{i+1}) X^{i+2}$, where indices are taken modulo m .

A corresponding gadget $\mathbb{S}_{\chi_m}$ takes m input n -sharings $[\mathbf{X}^0, \dots, \mathbf{X}^{m-1}]$ and produces m output n -sharings $[\mathbf{Y}^0, \dots, \mathbf{Y}^{m-1}]$, ensuring that the native values correspond to the definition of χ_m .

To compute $1 \oplus X$ from an n -sharing $\mathbf{X}$, it suffices to XOR 1 to a single share—e.g., X_0 . Each $\mathbf{Y}^i$ can then be securely computed using a masked AND-XOR gadget, as detailed in Algorithm 19.

This gadget requires three distinct SMNO clusters. Each n -sharing appears in exactly three different SAND-XOR gadgets, each using a distinct cluster. The *shake* function reverses the order of shares within each multi-share of a cluster and does not affect correctness or probing security. However, its inclusion is necessary for uniformity: without it, $\mathbb{S}_{\chi_m}$ would not be a permutation. This was verified for $(n = 4, m = 5)$ and $(n = 4, m = 3)$. Bijectivity for other values of (n, m) remains an open problem.

Probing security (standard probes). Assume the input n -sharings are independent (see Definition 6.3), and that the clusters h_0, h_1, h_2 are distinct.

Algorithm 19 The $\mathbb{S}_{\chi_m}$ gadget

Input: $[X^0, \dots, X^{m-1}] \in (\mathbb{F}_2^n)^m$, cluster indices $\{h_0, h_1, h_2\}$
Output: $[Y^0, \dots, Y^{m-1}] \in (\mathbb{F}_2^n)^m$

```

1: function neg( $U$ )
2:    $V \leftarrow U$ 
3:    $V_0 \leftarrow 1 \oplus U_0$ 
4:   return  $V$ 

5: function shake( $U, h$ )
6:    $V \leftarrow U$ 
7:   for  $j = 0$  to  $s - 1$  do
8:     for  $i = 0$  to  $s - 1$  do
9:        $A_j^h(V)[s - 1 - i] \leftarrow A_j^h(U)[i]$ 
10:  return  $V$ 

11: for  $i = 0$  to  $m - 2$  do
12:    $Y^i \leftarrow \text{SAND-XOR}\left(X^{i+2}, \text{neg}(X^{i+1}), X^i\right) \triangleright Y^i = X^i \oplus (1 \oplus X^{i+1})X^{i+2}.$ 
13:  $Y^{m-1} \leftarrow \text{SAND-XOR}\left(X^1, \text{neg}(X^0), \text{shake}(X^{m-1}, h_2)\right)$ 
14: return  $[Y^0, \dots, Y^{m-1}]$ 

```

From Algorithm 19, each X^i appears in exactly three consecutive SAND-XOR gadgets, each time under a *different* cluster index. The intermediate values in these gadgets take the following forms:

$$\begin{cases} D_1 = A_{j_1}^{h_2}(X^i)[j_2] \oplus \left(\bigoplus_{l_1=0}^{r_1} \bigoplus_{l_2=0}^{r_2} A_{j_1}^{h_0}(X^{i+1})[l_1] \cdot A_{j_2}^{h_1}(X^{i+2})[l_2]\right), \\ D_2 = A_{j_3}^{h_2}(X^{i-1})[j_4] \oplus \left(\bigoplus_{l_1=0}^{r_3} \bigoplus_{l_2=0}^{r_4} A_{j_3}^{h_0}(X^i)[l_1] \cdot A_{j_4}^{h_1}(X^{i+1})[l_2]\right), \\ D_3 = A_{j_5}^{h_2}(X^{i-2})[j_6] \oplus \left(\bigoplus_{l_1=0}^{r_5} \bigoplus_{l_2=0}^{r_6} A_{j_5}^{h_0}(X^{i-1})[l_1] \cdot A_{j_6}^{h_1}(X^i)[l_2]\right), \end{cases} \quad (7.19)$$

where $r_\ell, j_\ell \in \{0, \dots, s-1\}$ index the multi-share structure. The blue color highlights the targeted input (here, X^i). For clarity, the effects of **neg** and **shake** are omitted.

Single-Probe Case. Suppose a single probe targets an intermediate such as D_3 :

$$D_3 = \underbrace{A_{j_5}^{h_2}(X^{i-2})[j_6]}_{\text{blinding term}} \oplus \dots$$

The additive share from cluster h_2 serves as a blinding term. If no other probe reveals this same share, then by the probe elimination rule (Section 6.3.3), the adversary gains no information, and the probe can be eliminated.

Two or More Probes. When two or more probes target D_3 -type intermediates, the elimination condition may no longer hold. For example, two such probes can jointly reveal:

- one share from $A_{j_5}^{h_2}(\mathbf{X}^{i-2})$ (blinding term);
- one multi-share of $\mathbf{X}^{i-1}$; and
- one multi-share of $\mathbf{X}^i$.

Once the adversary obtains a multi-share of $\mathbf{X}^{i-1}$, they may switch to probing a D_2 -type gadget, where the blinding term may now overlap with previously revealed values. Hence, such a probe cannot be eliminated, and we conservatively assume that each new probe beyond the second leaks one additional multi-share of the targeted input $\mathbf{X}^i$.

- The first two probes may reveal one multi-share of $\mathbf{X}^i$ under cluster h_1 .
- Each subsequent probe may reveal one new multi-share under cluster h_0 .

Since h_0 and h_1 refer to two distinct (SMNO) clusters, fewer than $2 + (s - 1) = \sqrt{n} + 1$ probes will not expose all n shares of $\mathbf{X}^i$. Therefore, the gadget achieves a probing security order of $d = \sqrt{n} + 1$.

Glitch-extended probing security. For glitch-extended probes, the additive blinding terms do not apply. Instead, we rely solely on non-completeness. Each intermediate in $\mathbb{S}_{\chi_m}$ involves at most one multi-share from each secret. Thus, up to $d = \sqrt{n} - 1$ glitch-extended probes can be tolerated without full exposure of any input. In particular, for $n = 4$, we obtain first-order glitch resistance.

7.6 Application to Ascon

We now demonstrate how the proposed $\mathbb{S}_{\chi_5}$ gadget can be used to construct a masked implementation of the ASCON cipher [DEMS21], achieving $n = 4$ shares and probing security of order $d = 3$, without relying on any online randomness.

7.6.1 A Deterministic $\mathbb{S}$ Ascon

The only nonlinear operation in ASCON is χ_5 . In our deterministic masking approach, we replace χ_5 with the $\mathbb{S}\chi_5$ gadget from Section 7.5, while the diffusion layer and the linear/affine parts of the **S-box** are masked directly—without using any refresh gadgets or online randomness. Our ARM-based software implementation⁵ is based on the GitHub code developed by the original ASCON team [DDM⁺22].

Probing insecurity in [DDM⁺22]. The codebase in [DDM⁺22] includes $d = \{1, 2, 3\}$ order masked versions of the **S-box**. While the cases $d = 1$ and $d = 2$ are linked to prior work, the $d = 3$ implementation is not formally documented, and we found it to be insecure. Given that [DDM⁺22] is widely used in the literature, we believe this issue is worth noting.⁶ In particular, one line in the $d = 3$ $\mathbb{S}\mathbb{S}$ -box implementation computes the output share Z'_j from the input shares $\mathbf{Z} = [Z_0, \dots, Z_3]$, $\mathbf{X}$, and $\mathbf{Y}$ as:

$$Z'_j = Z_j \oplus X_j Y_0 \oplus X_j Y_1 \oplus X_j Y_2 \oplus X_j Y_3.$$

This construction is vulnerable to the two probes $P_1 = Z_j$ and $P_2 = Z'_j$, which together reveal:

$$P_1 \oplus P_2 = X_j(Y_0 \oplus Y_1 \oplus Y_2 \oplus Y_3) = X_j Y,$$

where $Y = \bigoplus_i Y_i$ is the native value.

Our implementation avoids this vulnerability by employing a third-order secure construction, as described in this chapter.

Probing security of composition. For ASCON, as we discussed in Chapter 6, the diffusion layer has box linear branch number $\mathcal{B}_{l\text{-box}} = 4$, and the probing security d of the composed cipher satisfies

$$\min\{d_{\mathbb{S}\mathbb{S}\text{-box}}, \mathcal{B}_{l\text{-box}} - 1\} \leq d \leq d_{\mathbb{S}\mathbb{S}\text{-box}},$$

where $d_{\mathbb{S}\mathbb{S}\text{-box}}$ denotes the local probing security order of each $\mathbb{S}\mathbb{S}$ -box gadget. The cases $d_{\mathbb{S}\mathbb{S}\text{-box}} = \{1, 2\}$ were analyzed in Chapter 6. The $d_{\mathbb{S}\mathbb{S}\text{-box}} = 3$ case is introduced and proven secure in this Chapter.

⁵<https://github.com/vahid-jahandideh/deterministic-masking-ascon.git>

⁶This observation pertains specifically to the $d = 3$ case and does not apply to $d = \{1, 2\}$.

Probing security of SS-box. Let the n -sharings entering $\mathbb{S}\chi_5$ be denoted by $[A, B, C, D, E]$. In ASCON, each S-box input is given by:

$$[A \oplus D \oplus E, B, C \oplus B, D, E \oplus D],$$

which corresponds to the inverse of the preceding linear layer (see Figure 6.8). As a result, each intermediate variable falls under the pattern of (7.19), possibly with additional affine terms. Importantly, every output includes distinct blinding terms, allowing a security argument similar to that of $\mathbb{S}\chi_5$ in Section 7.5. We also translated the SS-box implementation into the input format of the mask verification tool SILVER [KSM20], and verified both its third-order (standard) probing security and its uniformity.

Probing environment. In most software implementations, glitches are less of a concern, so we primarily focus on standard probing security. To further mitigate glitch-related risks, one could insert *synchronization layers* that preserve the effect of additive blinding across all intermediate computations without introducing additional randomness. We omit such extensions here, as our primary objective is to demonstrate the viability of higher-order deterministic masking based on our proposed clustering framework.

7.6.2 Implementation Results

We worked on the codebase in [DDM⁺22] for the $d = 3$ masking. The share-rotation and word-level interleaving countermeasures were preserved. Additionally, to ensure that the compiler does not interfere with the order of operations, the critical computations are implemented in ARM assembly.

Table 7.4 compares the number of clock cycles required to perform encryption under different protection orders. All measurements were taken on a ChipWhisperer CW308 board with an STM32F303 UFO target⁷, featuring a 32-bit ARM Cortex-M4 microcontroller.

The $d = 3$ implementation introduced in this work is approximately three times more expensive, in terms of cycle count, compared to the $d = 2$ implementation in [JMB25b].

In [JMB25b], the $d = 1$ implementation was shown to be vulnerable to second-order T-tests⁸ using approximately 30,000 traces. In our experiments, we observed that the $d = 2$ implementation becomes vulnerable to third-order

⁷<https://github.com/newaetech/chipwhisperer>

⁸An r -th order T-test computes the product of r centered sample points across traces to detect leakage.

Table 7.4: Cycle count per byte under different protection orders

Protection Order	Source	1 Byte	Long Message
$d = 0$	Ascon Team [DDM ⁺ 22]	4,274	99
	Jahandideh et al. [JMB25b]	5,109	124
$d = 1$	Ascon Team [DDM ⁺ 22]	11,085	310
	Jahandideh et al. [JMB25b]	13,339	388
$d = 2$	Ascon Team [DDM ⁺ 22]	20,883	585
	Jahandideh et al. [JMB25b]	25,414	745
$d = 3$	This work	73,522	2,229

T-tests with approximately an order of magnitude more traces. Specifically, around 400,000 traces are required to exceed the ± 4.5 threshold, as shown in Figure 7.3. This motivates the need for $d = 3$ protection.

Our $d = 3$ implementation remains below the threshold in the third-order T-test across 2 million tested traces. While it may eventually be susceptible to higher-order attacks, it is reasonable to expect that such an attack may require several million traces. Collecting and processing such a large dataset is computationally expensive and time-consuming.

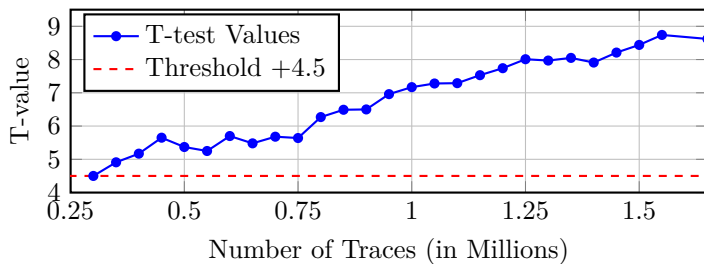


Figure 7.3: Third-order T-test values for a selected set of (i, j, k) samples from the round computations.

Remark 7.12. Our educational ChipWhisperer setup exhibits unusually high signal-to-noise ratio (SNR), due to synchronized sampling and tight coupling of the scope with the ARM Cortex-M4 target on a single board. This environment facilitates successful attacks even on protected implementations. In real-world scenarios, one would typically expect significantly higher trace complexity for similar results.

Chapter 8

Conclusions and Outlook

This thesis investigated how to *evaluate* and *engineer* masking security in leakage models that reflect practical side-channel behavior, and how to push masking to higher orders under tight randomness constraints. Concretely, we pursued two complementary research directions.

(I) Evaluation in noisy leakage and the random probing model. In Chapters 3–5, we developed tools and techniques for assessing masking security beyond high-noise regimes. We introduced an algebraic methodology for the random probing model (RPM) that models post-leakage knowledge via linear relations, yielding scalable evaluation methods for masked *linear* circuits, and effective approximations for structured *nonlinear* ones. This line culminated in (i) an RPM-native composition view that avoids assuming leak-free refresh and (ii) a practical evaluation of an $\mathcal{O}(n^2)$ masked AES design at meaningful leakage rates. Complementing this, we developed a decomposition approach in the noisy leakage model (Chapter 4) that yields tight bounds and clarifies when masking provably helps in low-noise settings, including regimes where classical noisy-to-RPM reductions become vacuous. Finally, in Chapter 5 we refined compositional reasoning in RPM through leakage-diagram analysis, introduced a “bridge” perspective to quantify refresh-induced dependencies, and proposed importance sampling to estimate rare disclosure events efficiently.

A practical outcome of this direction is LAPSE, a toolchain that compiles linear masked gadgets into a vector-space representation and computes/estimates RPM disclosure probabilities and composition parameters (exactly when feasible, and otherwise via Monte Carlo enhanced with importance sampling).

(II) Deterministic higher-order masking for full ciphers. In Chapters 6 and 7, we studied *deterministic masking*, where the protected compu-

tation consumes no fresh online randomness beyond the initial input sharing. We developed probing-security composition results for bricklayer permutations where a linear diffusion layer acts as an implicit refresh, enabling multi-round security without explicit refresh gadgets. We instantiated these results for ASCON using deterministic masked S-boxes and validated the implementations experimentally via first-, second-, and third-order leakage assessments. To extend deterministic masking to higher orders more systematically, we introduced a clustering-based approach (inspired by affine geometry) and used it to construct higher-order deterministic gadgets and to obtain higher-order deterministic masking for ASCON.

Overall message. Taken together, these results show that (i) RPM-native algebraic evaluation can deliver concrete, scalable security estimates and composition reasoning in constant-noise regimes, and (ii) deterministic masking can be made practical for full primitives when the cipher structure supports implicit “refresh-like” mixing, opening a path toward resource-efficient higher-order protections.

Future Research Directions

Impact of the field size q on $E_X(n, \epsilon)$. Recent works advocate masking over prime fields, motivated in part by improved concrete security for a *single* encoding as the prime q grows [CMM⁺23, GMM⁺24, FMM⁺24, BGM⁺25]. This aligns with analyses in noisy leakage models suggesting that prime fields can amplify noise more effectively than extension fields $q = 2^u$ [DFS16]. In contrast, the dependence on q that appears in noisy-to-RPM reductions (already questioned in early work [DFS19]) has been attributed to looseness of the reduction rather than an intrinsic effect of the RPM itself [BCG⁺23, BDF24].

Once working directly in the RPM, however, the relevance of q is less clear. In particular, the linear refresh/sum gadgets studied in this thesis admit the same vector-space (and leakage-diagram) structure over any field $\mathbb{F}_q$. Accordingly, in our LAPSE experiments on several linear refresh gadgets—including SR-multi—we did not observe a meaningful dependence of $E_X(n, \epsilon)$ or $\beta(n, \epsilon)$ on q , and the analytic leakage-diagram bounds derived in this work are also q -independent.

Understanding whether (and when) q has an inherent effect on RPM security for *circuits*—beyond reduction-induced changes in the effective probing rate—remains open. Clarifying this would be valuable for assessing circuit-level benefits of prime-field masking beyond isolated encodings.

Tighter and more general composition theory in the RPM. Our “bridge” view isolates which refresh-induced input-output relations actually create dependence across gadget boundaries. A natural next step is to turn this perspective into a more complete theory of composition in RPM: identifying conditions that are close to necessary (not only sufficient), and deriving systematically tight leakage-rate inflation bounds for broader gadget classes (beyond the refresh families analyzed here). It would also be valuable to understand how this bridge-based picture interacts with other frameworks (e.g., simulation-based notions and their RPM instantiations), and whether one can obtain *cipher-level* security statements that remain stable across many rounds without resorting to conservative union bounds.

Beyond linear structure: extending algebraic evaluation to richer circuit fragments. A core limitation of LAPSE is its restriction to linear circuits. While Chapter 3 already uses linearization ideas to approximate certain nonlinear components, a more systematic extension is open: for example, supporting mixed linear/nonlinear fragments, structured polynomial relations, or hybrid symbolic-probabilistic reasoning that still scales to large n . On the estimation side, importance sampling for RPM leakage patterns looks promising well beyond the linear setting, but designing low-variance tilting strategies for nonlinear disclosure events is largely unexplored.

More on deterministic masking. We initiated deterministic masking at the cipher level, but the current results still rely on (i) specific cipher structures (bricklayer + strong diffusion) and (ii) availability of suitable deterministic masked S-box gadgets at the desired order. Extending the approach to more complex S-boxes (or to permutations with different structural constraints) is an obvious direction. On the theory side, proving broader analytical composition claims—e.g., less tailored sufficient conditions and tighter order propagation bounds—would strengthen the foundations of deterministic masking. On the design side, developing a principled methodology for constructing higher-order deterministic gadgets (beyond the cases addressed by clustering) remains a major challenge.

Finally, our results suggest that diffusion metrics (e.g., branch-number style parameters) are not only relevant for classical cryptanalysis but may also serve as *side-channel-relevant* design criteria: understanding which diffusion properties best promote implicit refresh and compositional probing security could inform future cipher design.

Bridging theory and practice with richer leakage models. The thesis mostly reasons in clean abstractions (RPM, noisy leakage, probing/glitch-extended probing), while practical implementations face additional effects such as transitions, coupling, and micro-architectural leakage sources. A worthwhile direction is to integrate such effects into evaluation pipelines in a modular way: e.g., extending the bridge/leakage-diagram methodology to capture limited transition leakage, or using the same algebraic backbone to study “effective leakage sets” induced by micro-architectural behavior. On the experimental side, connecting these refined models to measurement-based certification (T-tests, SNR estimation, and beyond) could yield a more unified workflow from device characterization to parameterized security guarantees.

Bibliography

- [ABB⁺20] Melissa Azouaoui, Davide Bellizia, Ileana Buhan, Nicolas Debande, Sébastien Duval, Christophe Giraud, Éliane Jaulmes, François Koeune, Elisabeth Oswald, François-Xavier Standaert, and Carolyn Whittall. A Systematic Appraisal of Side Channel Evaluation Strategies. In Thyla van der Merwe, Chris J. Mitchell, and Maryam Mehrnezhad, editors, *Security Standardisation Research – 6th International Conference, SSR 2020, London, UK, November 30 – December 1, 2020, Proceedings*, volume 12529 of *Lecture Notes in Computer Science*, pages 46–66. Springer, 2020.
- [ADF16] Marcin Andrychowicz, Stefan Dziembowski, and Sebastian Faust. Circuit Compilers with $O(1/\log n)$ Leakage Rate. In *Advances in Cryptology – EUROCRYPT 2016*, pages 586–615, Berlin, Heidelberg, 2016. Springer-Verlag.
- [AIS18] Prabhanjan Ananth, Yuval Ishai, and Amit Sahai. Private circuits: A modular approach. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018*, pages 427–455, Cham, 2018. Springer International Publishing.
- [Ajt11] Miklós Ajtai. Secure Computation with Information Leaking to an Adversary. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 715–724, 2011.
- [Bat97] Lynn Margaret Batten. *Combinatorics of Finite Geometries*. Cambridge University Press, 1997.
- [BBC⁺19] Gilles Barthe, Sonia Belaïd, Gaëtan Cassiers, Pierre-Alain Fouque, Benjamin Grégoire, and François-Xavier Standaert. maskVerif: Automated Verification of Higher-Order Masking

- in Presence of Physical Defaults. In Kazue Sako, Steve A. Schneider, and Peter Y. A. Ryan, editors, *Computer Security – ESORICS 2019 – 24th European Symposium on Research in Computer Security, Luxembourg, September 23-27, 2019, Proceedings, Part I*, volume 11735 of *Lecture Notes in Computer Science*, pages 300–318. Springer, 2019.
- [BBC⁺20] Davide Bellizia, Olivier Bronchain, Gaëtan Cassiers, Vincent Grosso, Chun Guo, Charles Momin, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. Mode-Level vs. Implementation-Level Physical Security in Symmetric Cryptography - A Practical Guide Through the Leakage-Resistance Jungle. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020*, volume 12170 of *Lecture Notes in Computer Science*, pages 369–400. Springer, 2020.
- [BBD⁺15a] Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, and Pierre-Yves Strub. Verified Proofs of Higher-Order Masking. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015*, pages 457–485, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.
- [BBD⁺15b] Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, Pierre-Yves Strub, and Rébecca Zucchini. Strong Non-Interference and Type-Directed Higher-Order Masking. Cryptology ePrint Archive, Paper 2015/506, 2015.
- [BBD⁺16] Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, Pierre-Yves Strub, and Rébecca Zucchini. Strong Non-Interference and Type-Directed Higher-Order Masking. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 116–129, 2016.
- [BBP⁺16] Sonia Belaïd, Fabrice Benhamouda, Alain Passelègue, Emmanuel Prouff, Adrian Thillard, and Damien Vergnaud. Randomness Complexity of Private Circuits for Multiplication. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in*

- Cryptology – EUROCRYPT 2016*, volume 9666 of *Lecture Notes in Computer Science*, pages 616–648. Springer, 2016.
- [BBP⁺17] Sonia Belaïd, Fabrice Benhamouda, Alain Passelègue, Emmanuel Prouff, Adrian Thillard, and Damien Vergnaud. Private multiplication over finite fields. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017*, pages 397–426, Cham, 2017. Springer International Publishing.
- [BC25] Sonia Belaïd and Gaëtan Cassiers. PERSEUS - Probabilistic Evaluation of Random probing SEcurity Using efficient Sampling. *IACR Cryptol. ePrint Arch.*, page 1884, 2025.
- [BCG⁺23] Julien Béguinot, Wei Cheng, Sylvain Guilley, Yi Liu, Loïc Masure, Olivier Rioul, and François-Xavier Standaert. Removing the Field Size Loss from Duc et al.’s Conjectured Bound for Masked Encodings. In Elif Bilge Kavun and Michael Pehl, editors, *Constructive Side-Channel Analysis and Secure Design – COSADE 2023*, volume 13979 of *Lecture Notes in Computer Science*, pages 86–104. Springer, 2023.
- [BCGR24] Julien Béguinot, Wei Cheng, Sylvain Guilley, and Olivier Rioul. Formal Security Proofs via Doeblin Coefficients: - Optimal Side-Channel Factorization from Noisy Leakage to Random Probing. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology – CRYPTO 2024*, volume 14925 of *Lecture Notes in Computer Science*, pages 389–426. Springer, 2024.
- [BCP⁺20] Sonia Belaïd, Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, and Abdul Rahman Taleb. Random probing security: Verification, composition, expansion and new constructions. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020*, pages 339–368, Cham, 2020. Springer International Publishing.
- [BCPZ16] Alberto Battistello, Jean-Sébastien Coron, Emmanuel Prouff, and Rina Zeitoun. Horizontal Side-Channel Attacks and Countermeasures on the ISW Masking Scheme. In Benedikt Gierlichs and Axel Y. Poschmann, editors, *Cryptographic Hardware and Embedded Systems – CHES 2016*, volume 9813 of *Lecture Notes in Computer Science*, pages 23–39. Springer, 2016.

- [BDF⁺17] Gilles Barthe, François Dupressoir, Sebastian Faust, Benjamin Grégoire, François-Xavier Standaert, and Pierre-Yves Strub. Parallel implementations of masking schemes and the bounded moment leakage model. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017*, pages 535–566, Cham, 2017. Springer International Publishing.
- [BDF24] Gianluca Brian, Stefan Dziembowski, and Sebastian Faust. From Random Probing to Noisy Leakages Without Field-Size Dependence. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology – EUROCRYPT 2024*, volume 14654 of *Lecture Notes in Computer Science*, pages 345–374. Springer, 2024.
- [BDMS22] Tim Beyne, Siemen Dhooghe, Amir Moradi, and Aein Rezaei Shahmirzadi. Cryptanalysis of Efficient Masked Ciphers: Applications to Low Latency. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2022(1):679–721, 2022.
- [BDN⁺14] Begül Bilgin, Joan Daemen, Ventsislav Nikov, Svetla Nikova, Vincent Rijmen, and Gilles Van Assche. Efficient and first-order DPA resistant implementations of Keccak. In Aurélien Francillon and Pankaj Rohatgi, editors, *Smart Card Research and Advanced Applications*, pages 187–199. Springer International Publishing, 2014.
- [BDPV07] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Sponge Functions. In *Ecrypt Hash Workshop 2007*, May 2007.
- [BDPV13] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Keccak. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 313–314. Springer, 2013.
- [BDZ20] Tim Beyne, Siemen Dhooghe, and Zhenda Zhang. Cryptanalysis of Masked Ciphers: A Not So Random Idea. In Shihō Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020*, volume 12491 of *Lecture Notes in Computer Science*, pages 817–850. Springer, 2020.

- [BFO23] Francesco Berti, Sebastian Faust, and Maximilian Ortl. Provable Secure Parallel Gadgets. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2023(4):420–459, 2023.
- [BGM⁺25] Briec Balon, Lorenzo Grassi, Pierrick Méaux, Thorben Moos, François-Xavier Standaert, and Matthias Johann Steiner. mid-square: Leveraging the strong side-channel security of prime-field masking in software. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2025(4):486–519, 2025.
- [BGN⁺14] Begül Bilgin, Benedikt Gierlichs, Svetla Nikova, Ventzislav Nikov, and Vincent Rijmen. Higher-Order Threshold Implementations. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014*, volume 8874 of *Lecture Notes in Computer Science*, pages 326–343. Springer, 2014.
- [BGP⁺11] Lejla Batina, Benedikt Gierlichs, Emmanuel Prouff, Matthieu Rivain, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. Mutual Information Analysis: a Comprehensive Study. *J. Cryptol.*, 24(2):269–291, 2011.
- [BGR18] Sonia Belaïd, Dahmun Goudarzi, and Matthieu Rivain. Tight private circuits: Achieving probing security with the least refreshing. In Thomas Peyrin and Steven Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018*, pages 343–372. Springer International Publishing, 2018.
- [BM25] Julien Béguinot and Loïc Masure. On the Average Random Probing Model. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2025(3):32–55, 2025.
- [BNN⁺12] Begül Bilgin, Svetla Nikova, Ventzislav Nikov, Vincent Rijmen, and Georg Stütz. Threshold Implementations of All 3×3 and 4×4 S-Boxes. In Emmanuel Prouff and Patrick Schaumont, editors, *Cryptographic Hardware and Embedded Systems – CHES 2012*, volume 7428 of *Lecture Notes in Computer Science*, pages 76–91. Springer, 2012.
- [BNR25] Sonia Belaïd, Victor Normand, and Matthieu Rivain. Masked Circuit Compiler in the Cardinal Random Probing Composability Framework. In Goichiro Hanaoka and Bo-Yin Yang, editors, *Advances in Cryptology – ASIACRYPT 2025*, volume 16246 of

- Lecture Notes in Computer Science*, pages 375–406. Springer, 2025.
- [BRR25] Sonia Belaïd, Matthieu Rivain, and Mélissa Rossi. New Techniques for Random Probing Security and Application to Racoon Signature Scheme. In Serge Fehr and Pierre-Alain Fouque, editors, *Advances in Cryptology – EUROCRYPT 2025*, volume 15608 of *Lecture Notes in Computer Science*, pages 94–123. Springer, 2025.
- [BRT21] Sonia Belaïd, Matthieu Rivain, and Abdul Rahman Taleb. On the Power of Expansion: More Efficient Constructions in the Random Probing Model. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021*, volume 12697 of *Lecture Notes in Computer Science*, pages 313–343. Springer, 2021.
- [BRTV21] Sonia Belaïd, Matthieu Rivain, Abdul Rahman Taleb, and Damien Vergnaud. Dynamic random probing expansion with quasi linear asymptotic complexity. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2021*, pages 157–188, Cham, 2021. Springer International Publishing.
- [BS21] Olivier Bronchain and François-Xavier Standaert. Breaking Masked Implementations with Many Shares on 32-bit Software Platforms or When the Security Order Does Not Matter. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(3):202–234, 2021.
- [CFOS21] Gaëtan Cassiers, Sebastian Faust, Maximilian Ortl, and François-Xavier Standaert. Towards Tight Random Probing Security. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021*, volume 12827 of *Lecture Notes in Computer Science*, pages 185–214. Springer, 2021.
- [CGLS21] Gaëtan Cassiers, Benjamin Grégoire, Itamar Levi, and François-Xavier Standaert. Hardware Private Circuits: From Trivial Composition to Full Verification. *IEEE Transactions on Computers*, 70(10):1677–1690, 2021.
- [CGPZ16] Jean-Sébastien Coron, Aurélien Greuet, Emmanuel Prouff, and Rina Zeitoun. Faster Evaluation of SBoxes via Common Shares.

- In Benedikt Gierlichs and Axel Y. Poschmann, editors, *Cryptographic Hardware and Embedded Systems – CHES 2016*, volume 9813 of *Lecture Notes in Computer Science*, pages 498–514. Springer, 2016.
- [CGZ20] Jean-Sébastien Coron, Aurélien Greuet, and Rina Zeitoun. Side-channel masking with pseudo-random generator. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020*, pages 342–375, 2020.
- [CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Michael Wiener, editor, *Advances in Cryptology — CRYPTO’ 99*, pages 398–412, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.
- [CMM⁺23] Gaëtan Cassiers, Loïc Masure, Charles Momin, Thorben Moos, and François-Xavier Standaert. Prime-Field Masking in Hardware and its Soundness against Low-Noise SCA Attacks. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2023(2):482–518, Mar. 2023.
- [CMM⁺24] Gaëtan Cassiers, Loïc Masure, Charles Momin, Thorben Moos, Amir Moradi, and François-Xavier Standaert. Randomness generation for secure hardware masking – unrolled trivium to the rescue. *IACR Communications in Cryptology*, 1(2), 2024.
- [Cor18] Jean-Sébastien Coron. Formal verification of side-channel countermeasures via elementary circuit transformations. In Bart Preneel and Frederik Vercauteren, editors, *Applied Cryptography and Network Security*, pages 65–82, Cham, 2018. Springer International Publishing.
- [CPRR14] Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, and Thomas Roche. Higher-order side channel security and mask refreshing. In Shiho Moriai, editor, *Fast Software Encryption*, pages 410–424, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.
- [CS20] Gaëtan Cassiers and François-Xavier Standaert. Trivially and Efficiently Composing Masked Gadgets With Probe Isolating Non-Interference. *IEEE Transactions on Information Forensics and Security*, pages 2542–2555, 2020.

- [CS21] Gaëtan Cassiers and François-Xavier Standaert. Provably Secure Hardware Masking in the Transition- and Glitch-Robust Probing Model: Better Safe than Sorry. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2021(2):136–158, Feb. 2021.
- [CT06] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, July 2006.
- [Dae95] Joan Daemen. *Cipher and Hash Function Design Strategies based on linear and differential cryptanalysis*. PhD thesis, Katholieke Universiteit Leuven, 1995.
- [Dae17] Joan Daemen. Changing of the Guards: A Simple and Efficient Method for Achieving Uniformity in Threshold Sharing. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems – CHES 2017*, volume 10529 of *Lecture Notes in Computer Science*, pages 137–153. Springer, 2017.
- [DBG⁺17] Thomas De Cnudde, Begül Bilgin, Benedikt Gierlichs, Ventzislav Nikov, Svetla Nikova, and Vincent Rijmen. Does Coupling Affect the Security of Masked Implementations? In Sylvain Guilley, editor, *Constructive Side-Channel Analysis and Secure Design – COSADE 2017*, volume 10348 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2017.
- [DBR⁺15] Thomas De Cnudde, Begül Bilgin, Oscar Reparaz, Ventzislav Nikov, and Svetla Nikova. Higher-Order Threshold Implementation of the AES S-Box. In Naofumi Homma and Marcel Medwed, editors, *Smart Card Research and Advanced Applications – 14th International Conference, CARDIS 2015, Bochum, Germany, November 4-6, 2015. Revised Selected Papers*, volume 9514 of *Lecture Notes in Computer Science*, pages 259–272. Springer, 2015.
- [DCBR⁺16] Thomas De Cnudde, Begül Bilgin, Oscar Reparaz, Ventzislav Nikov, and Svetla Nikova. Higher-order threshold implementation of the AES s-box. In Naofumi Homma and Marcel Medwed, editors, *Smart Card Research and Advanced Applications*. Springer International Publishing, 2016.

- [DDE⁺20] Joan Daemen, Christoph Dobraunig, Maria Eichlseder, Hannes Groß, Florian Mendel, and Robert Primas. Protecting against Statistical Ineffective Fault Attacks. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(3):508–543, 2020.
- [DDF19] Alexandre Duc, Stefan Dziembowski, and Sebastian Faust. Unifying Leakage Models: From Probing Attacks to Noisy Leakage. *J. Cryptol.*, 32(1):151–177, 2019.
- [DDM⁺22] Florian Dietrich, Christoph Dobraunig, Florian Mendel, Robert Primas, and Martin Schl  ffer. simpleserial-ascon: Protected Ascon implementations. <https://github.com/ascon/simpleserial-ascon>, 2022. Accessed: July 2025.
- [DEM⁺20] Christoph Dobraunig, Maria Eichlseder, Stefan Mangard, Florian Mendel, Bart Mennink, Robert Primas, and Thomas Unterluggauer. Isap v2.0. *IACR Trans. Symmetric Cryptol.*, 2020(S1):390–416, 2020.
- [DEMS21] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schl  ffer. Ascon v1.2: Lightweight Authenticated Encryption and Hashing. *Journal of Cryptology*, 34, 2021.
- [DFS15] Stefan Dziembowski, Sebastian Faust, and Maciej Sk  rski. Noisy Leakage Revisited. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015*, volume 9057 of *Lecture Notes in Computer Science*, pages 159–188. Springer, 2015.
- [DFS16] Stefan Dziembowski, Sebastian Faust, and Maciej Sk  rski. Optimal Amplification of Noisy Leakages. In Eyal Kushilevitz and Tal Malkin, editors, *Theory of Cryptography Conference – TCC 2016*, volume 9563 of *Lecture Notes in Computer Science*, pages 291–318. Springer, 2016.
- [DFS19] Alexandre Duc, Sebastian Faust, and Fran  ois-Xavier Standaert. Making Masking Security Proofs Concrete (Or How to Evaluate the Security of Any Leaking Device), Extended Version. *J. Cryptol.*, 32(4):1263–1297, 2019.
- [DF  19] Stefan Dziembowski, Sebastian Faust, and Karol   ebrowski. Simple refreshing in the noisy leakage model. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology*

- *ASIACRYPT 2019*, pages 315–344, Cham, 2019. Springer International Publishing.
- [DHP⁺20] Joan Daemen, Seth Hoffert, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. Xoodoo, a lightweight cryptographic scheme. *IACR Trans. Symmetric Cryptol.*, 2020(S1):60–87, 2020.
- [DNR19] Siemen Dhooghe, Svetla Nikova, and Vincent Rijmen. Threshold Implementations in the Robust Probing Model. In Begül Bilgin, Svetla Petkova-Nikova, and Vincent Rijmen, editors, *Proceedings of ACM Workshop on Theory of Implementation Security, TIS@CCS 2019, London, UK, November 11, 2019*, pages 30–37. ACM, 2019.
- [DR20] Joan Daemen and Vincent Rijmen. *The Design of Rijndael - The Advanced Encryption Standard (AES), Second Edition*. Information Security and Cryptography. Springer, 2020.
- [DSV14] François Durvaux, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. How to Certify the Leakage of a Chip? In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 459–476. Springer, 2014.
- [FGMDP⁺18] Sebastian Faust, Vincent Grosso, Santos Merino Del Pozo, Clara Paglialonga, and François-Xavier Standaert. Composable Masking Schemes in the Presence of Physical Defaults & the Robust Probing Model. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018(3):89–120, Aug. 2018.
- [FHT03] A.A. Fedotov, P. Harremoës, and F. Topsøe. Refinements of Pinsker’s inequality. *IEEE Transactions on Information Theory*, 49(6):1491–1498, 2003.
- [FMM⁺24] Sebastian Faust, Loïc Masure, Elena Micheli, Maximilian Ortl, and François-Xavier Standaert. Connecting Leakage-Resilient Secret Sharing to Practice: Scaling Trends and Physical Dependencies of Prime Field Masking. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology – EUROCRYPT 2024*, volume 14654 of *Lecture Notes in Computer Science*, pages 316–344. Springer, 2024.

- [FPS17] Sebastian Faust, Clara Paglialonga, and Tobias Schneider. Amortizing Randomness Complexity in Private Circuits. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017*, pages 781–810. Springer International Publishing, 2017.
- [GBTP08] Benedikt Gierlichs, Lejla Batina, Pim Tuyls, and Bart Preneel. Mutual Information Analysis. In Elisabeth Oswald and Pankaj Rohatgi, editors, *Cryptographic Hardware and Embedded Systems – CHES 2008*, volume 5154 of *Lecture Notes in Computer Science*, pages 426–442. Springer, 2008.
- [GD23] John Gaspoz and Siemen Dhooghe. Threshold Implementations in Software: Micro-architectural Leakages in Algorithms. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2023(2):155–179, Mar. 2023.
- [GHP⁺21] Barbara Gigerl, Vedad Hadzic, Robert Primas, Stefan Mangard, and Roderick Bloem. Coco: Co-Design and Co-Verification of Masked Software Implementations on CPUs. In Michael D. Bailey and Rachel Greenstadt, editors, *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, pages 1469–1486. USENIX Association, 2021.
- [GMM⁺24] Lorenzo Grassi, Loïc Masure, Pierrick Méaux, Thorben Moos, and François-Xavier Standaert. Generalized Feistel Ciphers for Efficient Prime Field Masking. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology – EUROCRYPT 2024*, volume 14653 of *Lecture Notes in Computer Science*, pages 188–220. Springer, 2024.
- [GMPO20] Si Gao, Ben Marshall, Dan Page, and Elisabeth Oswald. Share-slicing: Friend or foe? *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(1):152–174, 2020.
- [GMSP23] Barbara Gigerl, Florian Mendel, Martin Schläffer, and Robert Primas. Efficient Second-Order Masked Software Implementations of Ascon in Theory and Practice. *NIST LWC Workshop*, 2023.
- [Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

- [IKL⁺13] Yuval Ishai, Eyal Kushilevitz, Xin Li, Rafail Ostrovsky, Manoj Prabhakaran, Amit Sahai, and David Zuckerman. Robust Pseudorandom Generators. In Fedor V. Fomin, Rūsiņš Freivalds, Marta Kwiatkowska, and David Peleg, editors, *Automata, Languages, and Programming*, pages 576–588, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
- [ISW03] Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, pages 463–481, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.
- [IUH22] Akira Ito, Rei Ueno, and Naofumi Homma. On the Success Rate of Side-Channel Attacks on Masked Implementations: Information-Theoretical Bounds and Their Practical Usage. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, pages 1521–1535. ACM, 2022.
- [JMB24] Vahid Jahandideh, Bart Mennink, and Lejla Batina. An Algebraic Approach for Evaluating Random Probing Security With Application to AES. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2024(4):657–689, 2024.
- [JMB25a] Vahid Jahandideh, Bart Mennink, and Lejla Batina. A decomposition approach for evaluating security of masking. In Goichiro Hanaoka and Bo-Yin Yang, editors, *Advances in Cryptology – ASIACRYPT 2025*, volume 16245 of *Lecture Notes in Computer Science*, pages 580–612. Springer, 2025.
- [JMB25b] Vahid Jahandideh, Bart Mennink, and Lejla Batina. Probing Secure Composability Without Fresh Randomness: Theory and Application to Ascon. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2025(4):972–1009, Sep. 2025.
- [JMB26] Vahid Jahandideh, Bart Mennink, and Lejla Batina. Leakage diagrams, importance sampling, and composition in the random probing model. *Cryptology ePrint Archive*, Paper 2026/444, 2026.

- [JPST17] J  r  my Jean, Thomas Peyrin, Siang Meng Sim, and Jade Tourteaux. Optimizing implementations of lightweight building blocks. *IACR Trans. Symmetric Cryptol.*, 2017(4):130–168, 2017.
- [JSB25] Vahid Jahandideh, Jan Schoone, and Lejla Batina. Clustering approach for higher-order deterministic masking. Cryptology ePrint Archive, Paper 2025/273, 2025.
- [JWMB] Vahid Jahandideh, L  o Weissbart, Bart Mennink, and Lejla Batina. A New Leakage Exploitation Framework and Its Application to Authenticated Encryption. Accepted paper presented at the NIST Lightweight Cryptography Workshop 2023 (virtual), June 21–22, 2023. Available at <https://csrc.nist.gov/csrc/media/Events/2023/lightweight-cryptography-workshop-2023/documents/accepted-papers/05-a-new-leakage-exploitation-framework.pdf>.
- [KJJ99] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael Wiener, editor, *Advances in Cryptology — CRYPTO’ 99*, pages 388–397, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.
- [Koc96] Paul C. Kocher. Timing attacks on implementations of diffie-hellman, RSA, DSS, and other systems. In Neal Kobitz, editor, *Advances in Cryptology — CRYPTO ’96*, pages 104–113, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.
- [KSM20] David Knichel, Pascal Sasdrich, and Amir Moradi. SILVER - Statistical Independence and Leakage Verification. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020*, volume 12491 of *Lecture Notes in Computer Science*, pages 787–816. Springer, 2020.
- [Kv78] Teun Kloek and Herman K. van Dijk. Bayesian Estimates of Equation System Parameters: An Application of Integration by Monte Carlo. *Econometrica*, 46(1):1–19, 1978.
- [LDP20] Steven J Leon and Lisette De Pillis. Linear algebra with applications, 10th Edition, 2020.
- [LM98] Charles F. Laywine and Gary L. Mullen. *Discrete Mathematics Using Latin Squares*. John Wiley & Sons, 1998.

- [LS16] Meicheng Liu and Siang Meng Sim. Lightweight MDS Generalized Circulant Matrices. In Thomas Peyrin, editor, *Fast Software Encryption – FSE 2016*, volume 9783 of *Lecture Notes in Computer Science*, pages 101–120. Springer, 2016.
- [MBR19] Lauren De Meyer, Begül Bilgin, and Oscar Reparaz. Consolidating Security Notions in Hardware Masking. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(3):119–147, 2019.
- [MMSS19] Thorben Moos, Amir Moradi, Tobias Schneider, and François-Xavier Standaert. Glitch-Resistant Masking Revisited or Why Proofs in the Robust Probing Model are Needed. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(2):256–292, 2019.
- [Moo19] Thorben Moos. Static Power SCA of Sub-100 nm CMOS ASICs and the Insecurity of Masking Schemes in Low-Noise Environments. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(3):202–232, 2019.
- [MOP07] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards (Advances in Information Security)*. Springer-Verlag, Berlin, Heidelberg, 2007.
- [MPL⁺11] Amir Moradi, Axel Poschmann, San Ling, Christof Paar, and Huaxiong Wang. Pushing the Limits: A Very Compact and a Threshold Implementation of AES. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 69–88. Springer, 2011.
- [MRS22] Loïc Masure, Olivier Rioul, and François-Xavier Standaert. A Nearly Tight Proof of Duc et al.’s Conjectured Security Bound for Masked Implementations. In Ileana Buhan and Tobias Schneider, editors, *Smart Card Research and Advanced Applications – 21st International Conference, CARDIS 2022, Birmingham, UK, November 7-9, 2022, Revised Selected Papers*, volume 13820 of *Lecture Notes in Computer Science*, pages 69–81. Springer, 2022.
- [MS23] Loïc Masure and François-Xavier Standaert. Prouff and Rivain’s Formal Security Proof of Masking, Revisited - Tight

- Bounds in the Noisy Leakage Model. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023*, volume 14083 of *Lecture Notes in Computer Science*, pages 343–376. Springer, 2023.
- [NRR06] Svetla Nikova, Christian Rechberger, and Vincent Rijmen. Threshold implementations against side-channel attacks and glitches. In Peng Ning, Sihan Qing, and Ninghui Li, editors, *Information and Communications Security*, pages 529–545, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
- [oST01] National Institute of Standards and Technology. FIPS PUB 197: Advanced Encryption Standard (AES). Technical report, National Institute of Standards and Technology, 2001.
- [oST15] National Institute of Standards and Technology. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. Technical Report FIPS PUB 202, National Institute of Standards and Technology, August 2015.
- [PAB⁺23] Enrico Piccione, Samuele Andreoli, Lilya Budaghyan, Claude Carlet, Siemen Dhooghe, Svetla Nikova, George Petrides, and Vincent Rijmen. An Optimal Universal Construction for the Threshold Implementation of Bijective S-Boxes. *IEEE Trans. Inf. Theory*, 69(10):6700–6710, 2023.
- [Pet19] George Petrides. On Non-Completeness in Threshold Implementations. In *Proceedings of ACM Workshop on Theory of Implementation Security Workshop, TIS’19*, pages 24–28, New York, NY, USA, 2019. Association for Computing Machinery.
- [PGMP19] Thomas Prest, Dahmun Goudarzi, Ange Martinelli, and Alain Passelègue. Unifying Leakage Models on a Rényi Day. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, volume 11692 of *Lecture Notes in Computer Science*, pages 683–712. Springer, 2019.
- [Pla36] J. Playfair. *The element of geometry*. W.E. Dean, 1836.
- [PP02] A. Papoulis and S.U. Pillai. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill series in electrical and computer engineering. McGraw-Hill, 2002.

- [PR13] Emmanuel Prouff and Matthieu Rivain. Masking against Side-Channel Attacks: A Formal Security Proof. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 142–159. Springer, 2013.
- [RBN⁺15] Oscar Reparaz, Begül Bilgin, Svetla Nikova, Benedikt Gierlichs, and Ingrid Verbauwhede. Consolidating Masking Schemes. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology – CRYPTO 2015*, volume 9215 of *Lecture Notes in Computer Science*, pages 764–783. Springer, 2015.
- [Rep15] Oscar Reparaz. A note on the security of Higher-Order Threshold Implementations. Cryptology ePrint Archive, Paper 2015/001, 2015.
- [RP10] Matthieu Rivain and Emmanuel Prouff. Provably secure higher-order masking of AES. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems – CHES 2010*, pages 413–427, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
- [SM21] Aein Rezaei Shahmirzadi and Amir Moradi. Second-Order SCA Security with almost no Fresh Randomness. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(3):708–755, 2021.
- [TMC⁺24] Meltem Sönmez Turan, Kerry McKay, Donghoon Chang, Jinkeon Kang, and John Kelsey. Ascon-Based Lightweight Cryptography Standards for Constrained Devices, (NIST initial public draft), 2024.
- [Tof80] Tommaso Toffoli. Reversible computing. In Jaco de Bakker and Jan van Leeuwen, editors, *Automata, Languages and Programming*, pages 632–644, Berlin, Heidelberg, 1980. Springer Berlin Heidelberg.
- [WJZY23] Weijia Wang, Fanjie Ji, Juelin Zhang, and Yu Yu. Efficient Private Circuits with Precomputation. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2023(2):286–309, 2023.
- [Wyn75] A. D. Wyner. The wire-tap channel. *The Bell System Technical Journal*, 54(8):1355–1387, 1975.

Research Data Management

This thesis research has been carried out under the research data management policy of the Institute for Computing and Information Science of Radboud University, The Netherlands.¹

The following research datasets were produced during this PhD research:

- Chapters 3 and 5. The initial technique introduced in Chapter 3 was later improved in Chapter 5, resulting in the tool LAPSE. This tool is available online at <https://doi.org/10.34973/vj8e-dr78>.
- Chapters 6 and 7. These chapters developed a deterministic masking strategy. As a proof of concept, we applied it to ASCON, and the corresponding code is available online at <https://doi.org/10.34973/cn0m-x558>.

¹ru.nl/en/institute-for-computing-and-information-sciences/research, last accessed December 19th, 2024.

Summary

Side-channel attacks exploit physical effects—such as power consumption or electromagnetic radiation—to recover secrets from cryptographic implementations. *Masking* is the standard principled countermeasure: it encodes every sensitive intermediate into n randomized *shares* so that an adversary must combine several leakages before learning anything useful. While masking is supported by strong compositional theorems in proof-oriented models, two practical bottlenecks remain. First, assessing the *concrete* security of a full masked implementation is difficult in realistic, low-noise settings, where leakage can be strong enough that pessimistic bounds or loose reductions become vacuous. Second, higher-order masking typically relies on large amounts of *fresh online randomness* to repeatedly “refresh” shares during execution, which is costly on constrained devices.

This thesis investigates how to *evaluate* and *engineer* masking security under models that better reflect practice, and how to push masking to higher orders under tight randomness constraints. The contributions are organized along two complementary research lines.

Part I: Noisy leakage and the Random Probing Model (Chapters 3–5).

The first part develops theory and tools for analyzing masking in the *random probing model* (RPM), where each intermediate leaks independently with some probability ϵ , and for connecting this view to more physical *noisy leakage* models. Chapter 3 introduces an algebraic methodology for RPM evaluation. It models the adversary’s post-leakage knowledge as systems of linear (parity) relations among intermediates, which enables scalable security estimation for masked *linear* circuits and informative approximations for structured nonlinear computations. A key outcome is an RPM-native composition perspective that does *not* assume leak-free refresh: refresh effects are captured by explicit boundary parameters rather than by idealized assumptions. On the constructive side, the chapter refines a multiplication gadget and yields an

$\mathcal{O}(n^2)$ masked design for AES with guaranteed security at constant leakage rates in the practically relevant range.

Chapter 4 addresses the fundamental question of *when* masking provably helps in low-noise regimes. It develops a decomposition framework that reduces leakage over extension fields to binary projections, leading to tight success-rate bounds where classical reductions can become loose. As a central theoretical result, it proves a conjecture of Dziembowski et al. [DFS16] that characterizes a tight noise threshold for security amplification by masking over finite abelian groups. For masking over $\mathbb{F}_{2^u}$, the chapter further derives an improved noisy-to-RPM reduction for $\mathbb{F}_2$ -linear computations that remains meaningful in regimes where generic field-size-dependent bounds are vacuous.

Chapter 5 complements these results with additional tooling tailored to constant-noise, large-scale implementations. It instantiates the decomposition approach for noisy Hamming-weight leakage and derives concrete RPM probing rates as functions of SNR for $\mathbb{F}_2$ -linear computations, clarifying which measurement regimes correspond to meaningful ϵ parameters. To reason compositionally about extremely small failure probabilities, it adapts *importance sampling* to efficiently estimate rare disclosure events in RPM. It also develops explicit *leakage-diagram* bounds for refresh gadgets, isolating “bridges” created by refresh leakage as the main mechanism that induces inter-round dependence. A practical outcome is LAPSE, a toolchain that compiles linear masked gadgets into vector-space representations and computes/estimates disclosure probabilities and composition parameters (exactly when feasible, otherwise via Monte Carlo enhanced with importance sampling).

Part II: Deterministic masking (Chapters 6–7). The second part explores the extreme setting of *deterministic masking*, where the protected computation consumes no fresh online randomness beyond the initial input sharing. Chapter 6 shows that for round-based *bricklayer* permutations, a linear diffusion layer can act as an *implicit refresh*: it mixes the offline randomness embedded in sharings produced by deterministic gadgets and can restore the independence properties needed for secure composition. The chapter provides composition theorems (and matching conditions) for first-order composability, and relates higher-order preservation of independence to the diffusion layer’s *linear branch number*. It also clarifies why standard probe-propagation frameworks inherently rely on fresh randomness, and proposes refinements that enable simulation-flavored reasoning in deterministic contexts. These ideas are instantiated in software for the NIST-standardized cipher ASCON, using deterministic masked S-box constructions and validating the approach experimen-

tally.

Chapter 7 advances deterministic masking beyond first order by addressing a longstanding obstacle: building deterministic nonlinear gadgets (in particular, multiplication/**AND**) that are simultaneously higher-order secure, *uniform*, and robust in glitch-extended probing models. The chapter introduces *share clustering*, inspired by finite affine geometry, as a systematic design principle that controls overlaps between groups of shares. This yields deterministic multiplication gadgets that use no online randomness, require only n field multiplications, and achieve probing security of order roughly $d \approx \sqrt{n}$ for suitable square sharing sizes. Building on these gadgets, the chapter constructs deterministic masked circuits for the χ_m maps that appear in standardized permutations, obtaining higher-order deterministic nonlinearity with provable security and uniformity properties. Combined with the deterministic composition principles from Chapter 6, these constructions lead to higher-order deterministically masked implementations of the ASCON permutation, supported by implementation results and leakage assessments.

Samenvatting

Zijkanaalaanvallen benutten fysieke effecten—zoals stroomverbruik of elektromagnetische straling—om geheimen te achterhalen uit cryptografische implementaties. *Maskering* is hierbij de standaard, principiële tegenmaatregel: elke gevoelige tussenwaarde wordt gecodeerd in n gerandomiseerde *shares* (delen), zodat een aanvallers meerdere lekkages moet combineren voordat er bruikbare informatie zichtbaar wordt. Hoewel maskering wordt ondersteund door sterke compositiestellingen in bewijs-georiënteerde modellen, blijven er twee praktische knelpunten. Ten eerste is het lastig om de *concrete* beveiliging van een volledige gemaskeerde implementatie te beoordelen in realistische, laag-ruis settings, waar lekkage sterk kan zijn en pessimistische grenzen of losse reducties betekenisloos worden. Ten tweede vereist hogere-orde maskering doorgaans veel *verse online randomisatie* om shares tijdens de uitvoering telkens opnieuw te “verversen” (refresh), wat op beperkte apparaten duur is.

Dit proefschrift onderzoekt hoe men maskerbeveiliging kan *evalueren* en *engineeren* in modellen die praktischer zijn, en hoe men maskering naar hogere orde kan brengen onder strikte randomisatiebeperkingen. De bijdragen zijn georganiseerd langs twee complementaire onderzoekslijnen.

Deel I: Noisy leakage en het Random Probing Model (Hoofdstukken 3–5). Het eerste deel ontwikkelt theorie en hulpmiddelen voor analyse in het *random probing model* (RPM), waarin elke tussenwaarde onafhankelijk lekt met kans ϵ , en voor het verbinden van dit perspectief met fysisch gemotiveerde *noisy-leakage* modellen. Hoofdstuk 3 introduceert een algebraïsche methode voor RPM-evaluatie. Het modelleert de kennis van de aanvallers na lekkage als stelsels van lineaire (pariteits-)relaties tussen intermediates, wat schaalbare schattingen oplevert voor gemaskeerde *lineaire* circuits en informatieve benaderingen voor gestructureerde niet-lineaire berekeningen. Een belangrijk resultaat is een compositieperspectief binnen RPM dat *geen* lekvrije refresh aanneemt: refresh-effecten worden vastgelegd via expliciete grensparameters

in plaats van via geïdealiseerde aannames. Constructief verfijnt het hoofdstuk bovendien een vermenigvuldigingsgadget en leidt het tot een $\mathcal{O}(n^2)$ -ontwerp voor gemaskeerde AES met gegarandeerde beveiliging bij constante lekrates in het praktisch relevante bereik.

Hoofdstuk 4 behandelt de fundamentele vraag *wanneer* maskering aantoonbaar helpt in laag-ruis regimes. Het ontwikkelt een decompositie-raamwerk dat lekkage over uitbreidingslichamen reduceert tot binaire projecties, en zo strakke succes-kansen geeft waar klassieke reducties te los kunnen worden. Als centraal theoretisch resultaat bewijst het een conjectuur van Dziembowski et al. [DFS16] die een scherpe ruisdrempel karakteriseert voor beveiligingsversterking door maskering over eindige abelse groepen. Voor maskering over $\mathbb{F}_{2^u}$ leidt het hoofdstuk daarnaast een verbeterde noisy-naar-RPM reductie af voor $\mathbb{F}_2$ -lineaire berekeningen, die informatief blijft in regimes waar generieke grenzen die afhangen van de veldgrootte geen betekenis meer hebben.

Hoofdstuk 5 vult dit aan met extra tooling gericht op constante ruis en grootschalige implementaties. Het instantieert de decompositie-aanpak voor noisy Hamming-weight lekkage en leidt concrete RPM-probing rates af als functie van SNR voor $\mathbb{F}_2$ -lineaire berekeningen, waardoor duidelijk wordt welke meetregimes overeenkomen met zinvolle ϵ -parameters. Om compositie te kunnen redeneren met extreem kleine faalkansen past het *importance sampling* toe om zeldzame disclosure events in RPM efficiënt te schatten. Verder ontwikkelt het expliciete *leakage-diagram* grenzen voor refresh-gadgets, waarbij “bruggen” door refresh-lekkage worden geïdentificeerd als het centrale mechanisme dat afhankelijkheden tussen ronden introduceert. Een praktisch resultaat is LAPSE, een toolchain die lineaire gemaskeerde gadgets compileert naar vector-ruimte representaties en disclosure-kansen en compositieparameters berekent/schat (exact wanneer mogelijk, anders via Monte Carlo versneld met importance sampling).

Deel II: Deterministische maskering (Hoofdstukken 6–7). Het tweede deel onderzoekt de extreme setting van *deterministische maskering*, waarbij de berekening geen verse online randomisatie gebruikt behalve de initiële input-sharing. Hoofdstuk 6 laat zien dat bij ronde-gebaseerde *bricklayer* permutaties een lineaire diffusielaag kan fungeren als een *impliciete refresh*: zij mengt de offline randomisatie die in deterministische sharings besloten ligt en kan de onafhankelijkheidseigenschappen herstellen die nodig zijn voor veilige compositie. Het hoofdstuk geeft compositiestellingen (en bijbehorende voorwaarden) voor eerste-orde composability en relateert behoud van onafhankelijkheid in hogere orde aan het *lineaire branch number* van de diffusielaag. Ook maakt

het duidelijk waarom standaard kaders voor probe-propagation intrinsiek leunen op verse randomisatie, en stelt het verfijningen voor die simulatie-achtig redeneren mogelijk maken in deterministische contexten. Deze ideeën worden geïntantieerd in software voor het door NIST gestandaardiseerde versleutingsalgoritme ASCON, met deterministische gemaskeerde S-box constructies en experimentele validatie.

Hoofdstuk 7 brengt deterministische maskering voorbij eerste orde door een hardnekkig obstakel aan te pakken: het bouwen van deterministische niet-lineaire gadgets (met name vermenigvuldiging/AND) die tegelijk hogere-orde veilig, *uniform* en robuust zijn in glitch-uitgebreide probing modellen. Het hoofdstuk introduceert *share clustering*, geïnspireerd door eindige affiene meetkunde, als systematisch ontwerpbeginnsel dat overlap tussen groepen shares controleert. Dit levert deterministische vermenigvuldigingsgadgets op die geen online randomisatie gebruiken, slechts n veldvermenigvuldigingen vereisen, en probing security behalen van orde grofweg $d \approx \sqrt{n}$ voor geschikte vierkante aantallen shares. Op basis hiervan construeert het hoofdstuk deterministisch gemaskeerde circuits voor de χ_m -maps die in gestandaardiseerde permutaties voorkomen, en verkrijgt zo hogere-orde deterministische niet-lineariteit met bewezen beveiligings- en uniformiteitseigenschappen. In combinatie met de deterministische compositieprincipes uit Hoofdstuk 6 leidt dit tot hogere-orde deterministisch gemaskeerde implementaties van de ASCON-permutatie, ondersteund door implementatieresultaten en lekkagebeoordelingen.

About the Author

Vahid Jahandideh was born in 1986 in Tabriz, Iran. He received his BSc in 2008 from the University of Tabriz and his MSc in 2010 from Sharif University of Technology, both in Electrical Engineering. From 2010 to 2021, he worked at the Electronics Research Center at Sharif University of Technology, focusing on secure communication engineering. In 2022, he started his PhD at the Digital Security group at Radboud University Nijmegen under the supervision of Lejla Batina and Bart Mennink. This thesis presents results obtained during his PhD research.

List of Publications

1. Vahid Jahandideh, Bart Mennink, and Lejla Batina. Leakage-diagrams, importance sampling, and composition in the random probing model. Cryptology ePrint Archive, Paper 2026/444, 2026
2. Vahid Jahandideh, Bart Mennink, and Lejla Batina. A decomposition approach for evaluating security of masking. In Goichiro Hanaoka and Bo-Yin Yang, editors, *Advances in Cryptology – ASIACRYPT 2025*, volume 16245 of *Lecture Notes in Computer Science*, pages 580–612. Springer, 2025
3. Vahid Jahandideh, Jan Schoone, and Lejla Batina. Clustering approach for higher-order deterministic masking. Cryptology ePrint Archive, Paper 2025/273, 2025
4. Vahid Jahandideh, Bart Mennink, and Lejla Batina. Probing Secure Composability Without Fresh Randomness: Theory and Application to Ascon. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2025(4):972–1009, Sep. 2025

5. Vahid Jahandideh, Bart Mennink, and Lejla Batina. An Algebraic Approach for Evaluating Random Probing Security With Application to AES. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2024(4):657–689, 2024

Notes

Vahid Jahandideh, Léo Weissbart, Bart Mennink, and Lejla Batina. A New Leakage Exploitation Framework and Its Application to Authenticated Encryption. Accepted paper presented at the NIST Lightweight Cryptography Workshop 2023 (virtual), June 21–22, 2023. Available at <https://csrc.nist.gov/csrc/media/Events/2023/lightweight-cryptography-workshop-2023/documents/accepted-papers/05-a-new-leakage-exploitation-framework.pdf>