Shuffle and Mix: On the Diffusion of Randomness in Threshold Implementations of Keccak

Felix Wegener, Christian Baiker and Amir Moradi

Ruhr University Bochum, Horst Görtz Institute for IT Security, Germany
{firstname.lastname}@rub.de

Abstract. Threshold Implementations are well-known as a provably first-order secure Boolean masking scheme even in the presence of glitches. A precondition for their security proof is a uniform input distribution at each round function, which may require an injection of fresh randomness or an increase in the number of shares. However, it is unclear whether violating the uniformity assumption causes exploitable leakage in practice. Recently, Daemen undertook a theoretical study of lossy mappings to extend the understanding of uniformity violations. We complement his work by entropy simulations and practical measurements of Keccak's round function. Our findings shed light on the necessity of mixing operations in addition to bit-permutations in a cipher's linear layer to propagate randomness between S-boxes and prevent exploitable leakage. Finally, we argue that this result cannot be obtained by current simulation methods, further stressing the continued need for practical leakage measurements.

1 Introduction

Ensuring the integrity of a message is one of the central objectives in many cryptographic applications. It can be achieved by using a hash algorithm in conjunction with a secret key to compute a message authentication code (MAC). As the integrity of a MAC depends on the secrecy of the key, the need to protect against side-channel analysis (SCA), e.g. Differential Power Analysis (DPA) [14], arises. To thwart DPA in hardware implementations of cryptographic algorithms Nikova *et al.* [17] introduced Threshold Implementations (TI), a provable first-order secure Boolean masking scheme.

Later, Bertoni *et al.* [4] developed the Keccak-family[1] of sponge-based hash-functions and suggested a three-share Threshold Implementation for their quadratic non-linear layer χ. Subsequently, Bilgin *et al.* [6] noted that the suggested TI violates the uniformity property and introduced two methods to alleviate this flaw. First, the injection of four bits of fresh-randomness per invocation of the non-linear building block χ. Second, the expansion to four shares, which allows the authors to find a uniform TI. Orthogonally, Daemen [10] investigated the implications of uniformity violation on the overall entropy

[1] standardized for selected parameters as SHA-3 in 2015

in KECCAK and the local entropy of individual bits and suggested a cheap method to re-mask the state bits with other state bits to prevent any exploitable leakage. Later, Daemen [11] suggested a re-masking scheme called *Changing of the Guards* to achieve uniformity of an arbitrary bijective S-box layer and noted the applicability to KECCAK.

Recently, De Meyer *et al.* [15] pointed out that uniformity is not a necessary condition for first-order security. In fact, information leakage takes place when any distribution observable by the attacker differs based on the unmasked secret value. In the setting of infeasible exhaustive computations, they suggest to evaluate this effect based on the χ^2-Test. Previously, Moradi *et al.* [16] demonstrated the applicability of the χ^2-test in leakage detection both for simulated traces of noisy Hamming-weight leakages and in practical measurements.

Our Contribution. We investigate the practical relevance of the diffusion layer to counteract the uniformity loss in masked KECCAK-f. In fact, we find that diffusion between S-boxes solely based on bit-permutations (ρ, π) does not prevent first-order leakage originating from the non-uniformity, while the mixing part (θ) of KECCAK-f alone is sufficient to counteract observable leakage through an FPGA evaluation. Further, we show that this effect cannot be revealed with state-of-the-art simulations, thereby indicating the need for practical SCA evaluations. To our knowledge, this is the first practical analysis of uniformity loss thereby complementing the theoretical foundation laid out by Daemen. [10].

Organization of the Paper. In Section 2, we describe our notation, recall the specification of KECCAK, describe Threshold Implementations and total imbalance. In Section 3 we give an overview of the recent TI designs of KECCAK. In Section 4, we analyze the probability distributions of S-box inputs with the methods of [10] and [15]. We describe the architecture of our hardware implementation in Section 5, and our practical evaluations in Section 6.

2 Preliminaries

In this section we introduce relevant definitions and our notation for the rest of the paper.

Introduction to Keccak KECCAK [4] is a sponge-based hash function that operates on a state of $b = 25 \cdot 2^l$ bits for l between 0 and 6. We use the same terminology as the authors to refer to individual parts of the state (cf. Figure 1 of [18]). Its core is the permutation KECCAK-f[b] which iterates the round function R a fixed number of times. The round function

$$R = \iota \circ \chi \circ \pi \circ \rho \circ \theta$$

consists of five sub-functions which are defined in the following:

– *Theta*: XORs the parity of two columns to each bit of a different column to improve diffusion between columns.

$$\theta : a[x][y][z] = a[x][y][z] \oplus \bigoplus_{y'=0}^{4} a[x-1][y'][z] \oplus \bigoplus_{y'=0}^{4} a[x+1][y'][z-1]$$

– *Rho*: performs a circular shift of all lanes, by a fixed constant per lane.

$$\rho : a[x][y][z] = a[x][y][z - \text{const}(x, y)]$$

– *Pi*: creates diffusion between rows in one slice.

$$\pi : a[x][y] = a[x'][y'], \quad \text{with } x = y', y = 2 \cdot x' + 3 \cdot y'$$

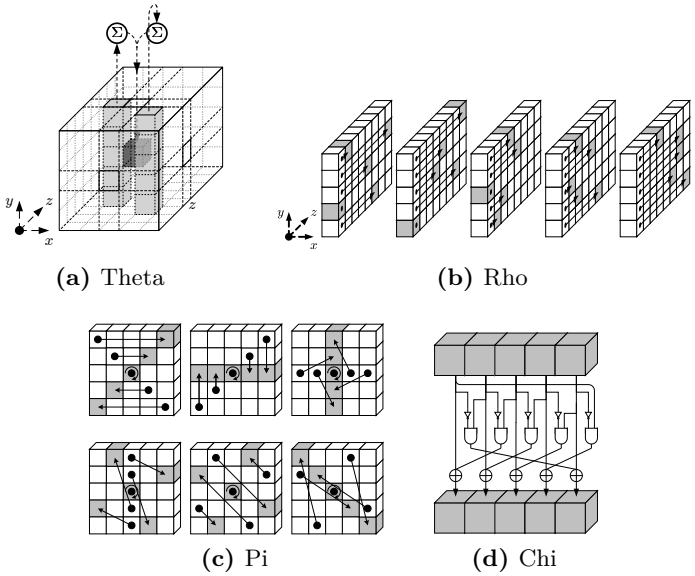
– *Chi*: is the only non-linear function. It operates as a 5-bit quadratic bijection on each row individually.

$$\chi : a[x] = a[x] \oplus (\mathbf{1} \oplus a[x+1]) \cdot a[x+2]$$

– *Iota*: XORs a round constant to the first lane.

$$\iota : a[0][0] = a[0][0] \oplus RC[r]$$

A visual illustration of all steps can be seen in Figure 1. For the remainder of this paper we focus on Keccak-f[200], which consists of 18 iterations of R.



(a) Theta **(b)** Rho **(c)** Pi **(d)** Chi

Figure 1: The Keccak subfunctions (a) θ, (b) ρ, (c) π, (d) χ, taken from [4]

Threshold Implementations For brevity, we limit ourselves to three shares and first-order security in the following introduction to Threshold Implementations [17].

Let $x \in \mathbb{F}_2^n$, we call $X = (a, b, c) \in \mathbb{F}_2^{3n}$ a sharing or masking of x if

$$x = a \oplus b \oplus c.$$

Each part a, b and c is called a share. We denote $\int(x)$ for the set of all such sharings. A sharing is called uniform, if all elements from $\int(x)$ occur with equal likelihood.

Consider a Boolean function $\mathsf{f} : \mathbb{F}_2^n \to \mathbb{F}_2^n$, we call $\mathsf{F} = (\mathsf{F}^A, \mathsf{F}^B, \mathsf{F}^C) : \mathbb{F}_2^{3n} \to \mathbb{F}_2^{3n}$ a *Threshold Implementation* if the following properties are present.

- *Correctness*: XORing all output shares reveals the output of the orignal function.

$$\forall x : \forall \mathsf{X} \in \int(x) : \mathsf{F}^A(X) \oplus \mathsf{F}^B(X) \oplus \mathsf{F}^C(X) = \mathsf{f}(x)$$

- *Non-completeness*: Each output share is independent of at least one input share:

$$\begin{aligned} \mathsf{F}^A(X) &= \mathsf{F}^A(b, c) \\ \mathsf{F}^B(X) &= \mathsf{F}^B(c, a) \\ \mathsf{F}^C(X) &= \mathsf{F}^C(a, b) \end{aligned}$$

Provable security is achieved through the central theorem of TI [17] which states: Let f be a Boolean function and F a TI of f. Let $X_1, \ldots, X_T$ be a sequence of sharings of the value x, each uniformly drawn from $\int(x)$. Then, the evaluations of $\mathsf{F}(X_i)$ do not reveal first-order information about x.

To ensure a uniform input share distribution during each round of an iterated cipher, it is beneficial to demand a third property of TI:

- *Uniformity*: F maps a uniform input distribution to a uniform output distribution.

$$\exists k : \forall x \in \mathbb{F}_2^n : \forall X \in \int(x) : \forall Y \in \int(\mathsf{f}(x)) : Pr(F(X) = Y) = k$$

For Keccak's non-linear function χ a uniform TI with four shares is known, while a uniform TI with three shares is either not possible or has not been found yet. Indeed, no statements about the existence of a uniform three share TI can be made due to the high computational complexity of an exhaustive search over all correction terms [5].

Entropy Study We recall several definitions from Daemen's [9] work: Let P be a probability distribution over $\mathbb{F}_2^n$ and $v \in \mathbb{F}_2^n$ be a mask. The imbalance of

P is defined as the Walsh-transformation of P and the total imbalance as its summation:

$$\widetilde{P}(v) := \sum_x P(x)(-1)^{v^\top x}, \qquad \phi_P := \sum_{\forall v \neq 0} \widetilde{P}(v)^2.$$

The evaluation of $\widetilde{P}$ in zero is omitted, as $\widetilde{P}(0) = 1$ regardless of the distribution of P. It can be shown that ϕ_P is zero if and only if P is a uniform distribution.

The chance that two elements drawn according to the probability distribution P over $\mathbb{F}_2^n$ are identical, is called the collision property $Pr_{\text{coll}}(P)$, which is connected to ϕ_P via the relation

$$\phi_P = 2^n Pr_{\text{coll}}(P) - 1.$$

It follows that $\phi_P \in [0, 2^n - 1]$ can be used as a metric to estimate the non-uniformity of a probability distribution P.

Pearson's χ^2-Test Pearson's χ^2 test allows a comparison between categorical observations of multiple random variables. Consider a table T in which each column (j) corresponds to a category and each row (i) to a variable. The integer value in cell $T_{i,j}$ expresses the number of times the realization of variable i has been observed to adopt category j. To decide whether all variables follow the same distribution (which forms the null-hypothesis $\mathcal{H}_0$) we define the test statistic

$$X = \sum_{i=0}^{r-1} \sum_{j=0}^{c-1} \frac{(T_{i,j} - E_{i,j})^2}{E_{i,j}}$$

with the expected number of occurrences

$$E_{i,j} := \frac{(\sum_{k=0}^{c-1} T_{i,k})(\sum_{k=0}^{r-1} T_{k,j})}{\sum_{k=0}^{c-1} \sum_{l=0}^{r-1} T_{k,l}} \quad .$$

The test statistic X follows a χ^2-distribution

$$X \sim \sum_{i=1}^{\mathsf{df}} N_i^2, \qquad \mathsf{df} = (c-1)(r-1)$$

where N_i are independent, standard normal random variables and df denotes the degrees of freedom. To determine a confidence level, we compute the cumulative distribution for X from the density function:

$$f(x, df) = \begin{cases} \frac{x^{\frac{\mathsf{df}}{2}-1} e^{-\frac{x}{2}}}{2^{\frac{\mathsf{df}}{2}} \Gamma(\frac{\mathsf{df}}{2})} & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \qquad p = \int_x^\infty f(x, \mathsf{df})$$

Under the assumption that $\mathcal{H}_0$ holds, p describes the likelihood that the observations in table T could have occurred. We reject the null-hypothesis to the level $p < 10^{-5}$, which constitutes the common threshold in leakage assessments.

3 TI of Keccak

In this section, we summarize different shared constructions of χ. They all share the properties of correctness and non-completeness, hence they constitute valid TIs. The constructions differ in whether and how they achieve uniformity. In the following we indicate the i-th bit of x by x_i.

Original TI. As the non-linear χ was designed to enable efficient masking with TI by limiting the algebraic degree to two, Bertoni *et al.* [4] also introduced a three-share masking scheme, defined as $\chi' : \mathbb{F}_2^{15} \to \mathbb{F}_2^{15}, (A, B, C) = \chi'(a, b, c)$ with

$$\begin{aligned} A_i &= b_i \oplus (b_{i+1} \oplus 1) \cdot b_{i+2} \oplus b_{i+1} \cdot c_{i+2} \oplus c_{i+1} \cdot b_{i+2} \\ B_i &= c_i \oplus (c_{i+1} \oplus 1) \cdot c_{i+2} \oplus c_{i+1} \cdot a_{i+2} \oplus a_{i+1} \cdot c_{i+2} \\ C_i &= a_i \oplus (a_{i+1} \oplus 1) \cdot a_{i+2} \oplus a_{i+1} \cdot b_{i+2} \oplus b_{i+1} \cdot a_{i+2} \end{aligned}$$

Contra to the original belief of the authors, the given TI is not uniform. Hence, an iterated application reduces entropy. In the following we recall several methods repairing χ' to achieve uniformity.

Re-masking. A naive approach is to re-mask the entire output of χ' according to the equations

$$\begin{aligned} A_i &= {\chi'}_i^A(b, c) \oplus r_i^b \oplus r_i^c \\ B_i &= {\chi'}_i^B(c, a) \oplus r_i^c \\ C_i &= {\chi'}_i^C(a, b) \oplus r_i^b. \end{aligned}$$

This scheme requires 10 bits of fresh randomness (r^b and r^c) for every invocation of χ', which can easily surpass the available randomness in an embedded system, in case several instances of χ' are implemented to operate in parallel (e.g., a round-based implementation).

Better Re-masking. Bilgin *et al.* [6] observed that only some bits require re-masking. More precisely any choice of two successive bits to be re-masked

yields a uniform sharing.

$$
\begin{aligned}
A_i &= {\chi'}_i^A(b,c) \oplus r_i^b \oplus r_i^c && i = 0,1 \\
B_i &= {\chi'}_i^B(c,a) \oplus r_i^c && i = 0,1 \\
C_i &= {\chi'}_i^C(a,b) \oplus r_i^b && i = 0,1 \\
A_i &= {\chi'}_i^A(b,c) && i = 2,3,4 \\
B_i &= {\chi'}_i^B(c,a) && i = 2,3,4 \\
C_i &= {\chi'}_i^C(a,b) && i = 2,3,4
\end{aligned}
$$

Subsequently, the constructions by Daemen re-mask the same bits, but partially [10] or fully [11] recycle randomness to achieve uniformity with reduced fresh randomness. Further, Bilgin *et al.* [6] introduced a uniform four-share TI of χ.

In the following, our focus is to study the original non-uniform three-share TI χ' interleaved with parts of the linear layer of Keccak-f to determine the practical impact of uniformity violation.

4 Simulations

In this section we characterize χ' as a lossy mapping by determining the number of sharings and total imbalance after a given number of successive iterations. Then, we sample the input distribution of χ' from simulations of Keccak-200 with different linear layers.

Iterating χ' alone. We iterated the three-share TI $\chi' : \mathbb{F}_2^{15} \to \mathbb{F}_2^{15}$ by feeding its output back into the function as an input, until the number of observed different sharings reached its minimum (5363) and the total imbalance its maximum (56.66). The extremes are attained after 54 iterations (cf. Figure 2). In comparison, a uniform mapping would maintain a total imbalance of zero and a constant number of 2^{15} possible sharings.

The figure clearly shows that the violation of uniformity from one round to the next has a compounding effect over many rounds. In Section 6 we show that this reduction of entropy is sufficient to practically exhibit leakage. Fortunately, Keccak-f consists of more functions than χ, namely a linear layer with three subfunctions:

- ρ - a bit-permutation for inter-slice diffusion
- π - a bit-permutation for intra-slice diffusion
- θ - a parity function to accelerate diffusion across columns

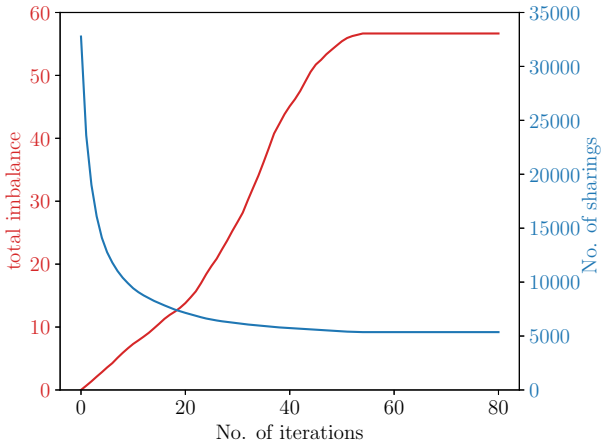


Figure 2: Illustration of the rise of total imbalance over the number of iterations of χ' (red) in comparison with the decrease in the number of sharings (blue).

While ι is also a part of the linear layer, we disregard it for our analysis since it consists only of an addition with a round constant to counteract slide attacks and has limited relevance to SCA[2].

Keccak-200. In our simulations we model the view of an attacker based on the glitch-extended 1-probing model [12, 20]. More specifically, the attacker may observe one output wire of the shared function χ' after a given number of rounds, which corresponds to observing the noise free values on two input wires to χ' resembling 1024 different potential observations. We determine whether the distribution seen on these wires is different between a fixed group consisting of sharings of the 200-bit all-zero plaintext and a random group in which the shared plaintext is chosen uniformly at random. Unfortunately, it is computationally infeasible to conduct an exhaustive computation of the distribution over all 3-sharing of 200 bits[3]. Hence, we follow the suggestion of [15] to conduct a χ^2-test on the histograms of the input values to determine whether a difference in the distributions of both groups is statistically significant. The results for 18 and 1800 iterations of variations of Keccak-f and 200 million samples are displayed in Table 1. We simulated Keccak-f with its original linear layer, only the bit-permuting part (ρ, π), only the mixing-part (θ) and without any linear layer. Note that in the first three cases a diffusion between all 40 instances of non-linear χ is achieved, while only in the last case no diffusion is present.

[2]The addition of round constants would further increase the total imbalance in the χ'-only scenario, but it is of no interest for the investigation of full Keccak.

[3]As it is already computationally infeasible for Keccak-25, we kept the consistency between measurements and simulations by evaluating Keccak-200.

Our simulations succeeded in finding the uniformity violation in the last (very obvious) case without diffusion. In the other three cases the null hypothesis that the input distribution is identical for both groups cannot be rejected given the common threshold of $p = 10^{-5}$ in a statistical test. Moreover, the null hypothesis cannot even be rejected given a very weak threshold of $p = 10^{-2}$. While the results clearly indicate that a linear layer is necessary to counteract the effects of uniformity violations, it remains unclear which parts of the linear layer are crucial and which are dispensable from an SCA perspective.

enabled	p_{18}	p_{1800}
χ', π, ρ, θ	0.021	0.022
χ', π, ρ	0.018	0.016
χ', θ	0.022	0.020
χ'	0.000	0.000

Table 1: χ^2-Test with degree of freedom $\mathsf{df} = 1023$ for 200 million samples. Only the uniformity violation of applying χ' alone is detected.

5 Implementation

Although a round-based implementation would be a natural choice to implement KECCAK and would lead to short SCA traces (hence accelerating the evaluation), it would potentially increase the noise since the combinatorial circuit involving all KECCAK subfunctions would be active at all clock cycles. To achieve a compromise between a high signal-to-noise ratio (SNR) and a fast leakage evaluation, we chose to implement all variants of KECCAK in a slice-serial manner by having five instances of χ' in parallel.

Slice-serial Implementation. In 2011 Jungk and Apfelbeck [13] introduced an area/latency trade-off for KECCAK by computing only eight slices in parallel instead of all 64 slices in a full round. Later Bilgin *et al.* [6] introduced a fully slice-serial architecture, which processes 25 state bits per clock cycle corresponding to the simultaneous execution of five χ functions (cf. Figure 3). It contains a shift register for the state that operates on 25-bit chunks and an additional 5-bit register to keep track of the parity of the previously processed slice to realize θ. A specialty is the application of θ to the first slice, which happens as the last step of each round in parallel to processing the last slice, as it requires the parity of the last slice. We implemented KECCAK with a state size of 200 bits, which requires 144 clock cycles to process a given input for 18 rounds.

Sharing the Implementation. We implemented several variants of three-share designs according to the χ constructions described in Section 3. Following the

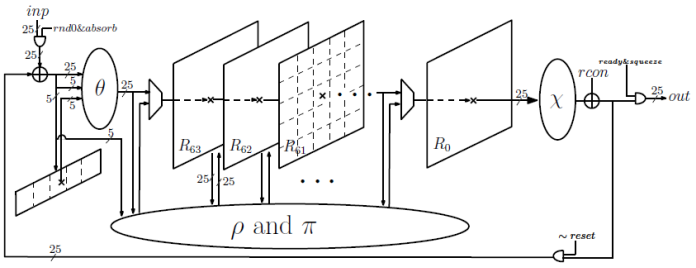


Figure 3: Serial KECCAK-200 architecture [6], one of eight slices is processed per clock cycle. The computation completes after 18 rounds corresponding to 144 clock cycles.

uncompressed design of [6], we maintained three shares throughout the entire computations. As ρ, π and θ are linear functions, they can be applied to each share of the state individually without modifications.

6 Practical Analysis

Measurement Setup. We synthesized our VHDL design in ISE Design Suite with the KEEP_HIERACHY attribute to ensure that non-completeness is maintained throughout the Place&Route process. For the practical evaluation, we used the SAKURA-G Side-Channel Evaluation Board [1] which includes two SPARTAN-6 FPGAs to separate controller and target functionality. We recorded power traces at a sampling rate of 625 MS/s by a Picoscope 6402 and an external amplifier (in addition to the amplifier embedded on the SAKURA-G board). Following the methodology of [21] we performed a non-specific t-test "fixed vs. random" [4] over 100 million traces of the last round of KECCAK while operating the FPGA at a clock frequency of 1.5 MHz.

Results. A measurement of 100 million traces of full KECCAK-200 with non-uniform χ' and 18 rounds did not reveal first-order leakage as can be seen in Figure 4. Even a drastic increase of the number of rounds to 1800 did not lead to first-order leakage (cf. Figure 5). However, the removal of θ leads to detectable first-order leakage after 80 million traces (cf. Figure 6), while a removal of the ρ and π does not indicate first-order leakage as illustrated in Figure 7. All measurements show leakage at orders two and three. Removing the entire linear layer causes each χ output to be taken as an input in the following round. Hence, an additional register is required to avoid transitional leakage, i.e., the leakage depending on the input of χ' being replaced by its output. This doubles the number of clock cycles to 288. Figure 8 shows the evaluation of 18 rounds of the non-uniform χ' function with 100 million traces. We observed, that the first-order

[4] The groups fixed vs. random are formed over the entire 200-bit state.

t-value exceeds the threshold of 4.5 by far. We can also see an increase of the t-value along the time axis.

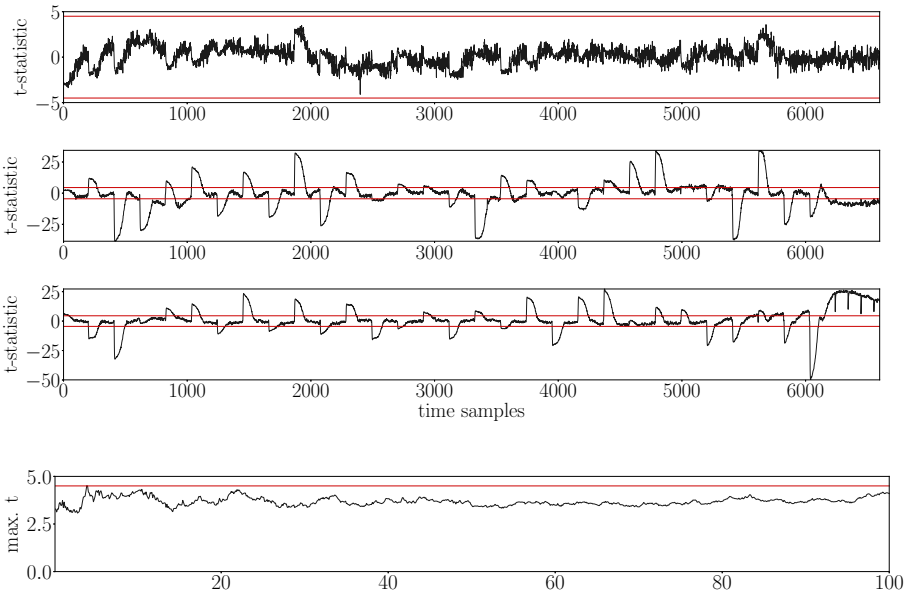


Figure 4: 18 round KECCAK-f. top to bottom: t-test results first to third order over time axis. Maximal t-values first order over trace axis. Entire last round.

Summary. Table 2 summarizes our practical leakage investigation. Based on the results of our simulations in Section 4, we expected χ' alone without re-masking to show excessive first-order leakage which increases over time - this turned out to be true in practice. We also expected 18-round KECCAK-200 and 1800-round KECCAK-200 to show similar leakage behavior, which is also the case. Despite similar simulation results (see Table 1), the omission of permutations ρ and π led to no detectable first-order leakage, while leakage can be observed if θ is omitted. This indicates that although such theoretical analysis can be considered as the very first step, the results in practice might be slightly different.

Our results indicate that a diffusion between χ functions (S-boxes) should not solely employ bit-permutations to cope with uniformity loss. Instead, a good diffusion layer should apply additional linear mixing functions, that lead to partial re-masking by means of uncorrelated state bits.

Discussion. The linear layer of KECCAK has proven to mend a small violation of uniformity in Threshold Implementations of χ. However, extending this result

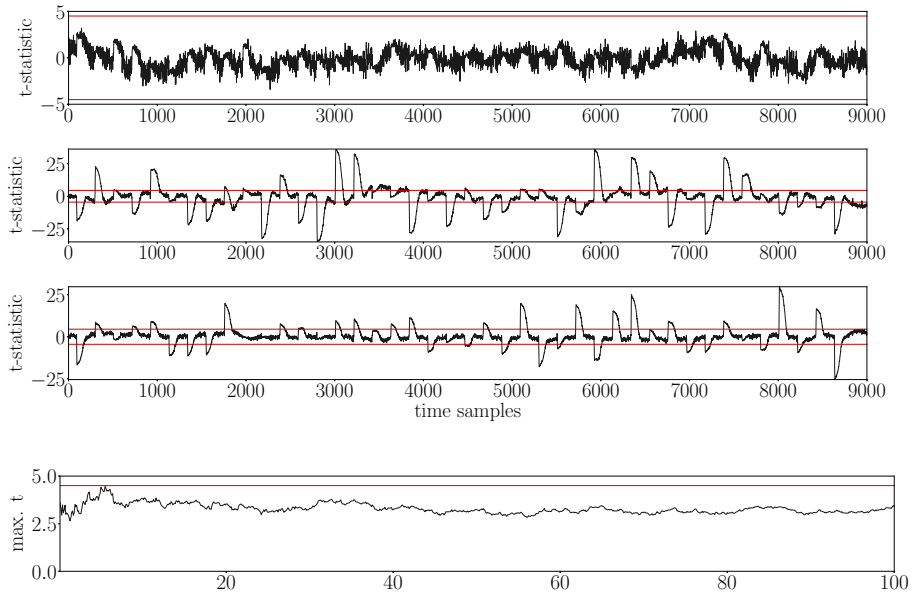


Figure 5: 1800 round KECCAK-f. top to bottom: t-test results first to third order over time axis. Maximal t-values first order over trace axis. Entire last round.

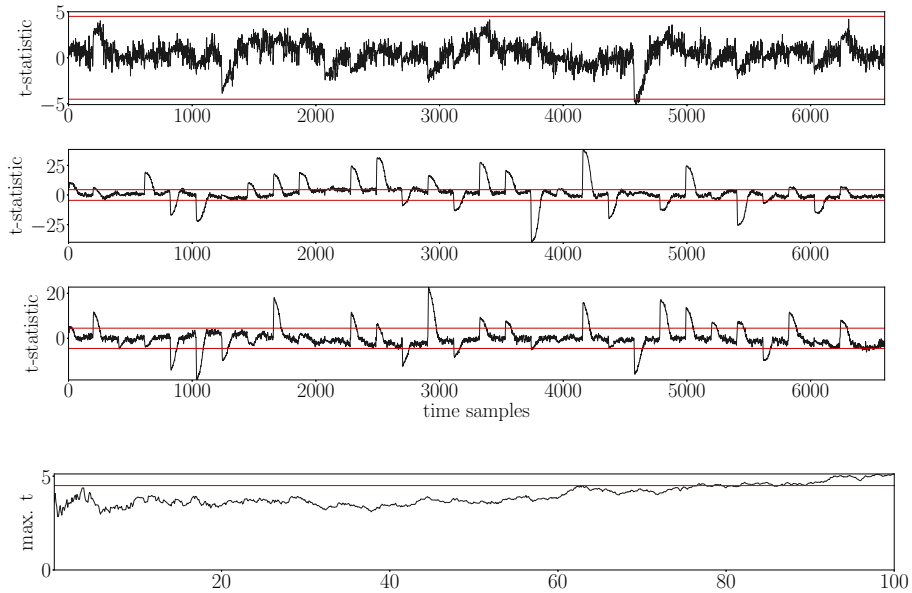


Figure 6: 18 rounds of ρ, π and χ' (i.e., *KECCAK-f* without θ). top to bottom: t-test results first to third order over time axis. Maximal t-values first order over trace axis. Entire last round.

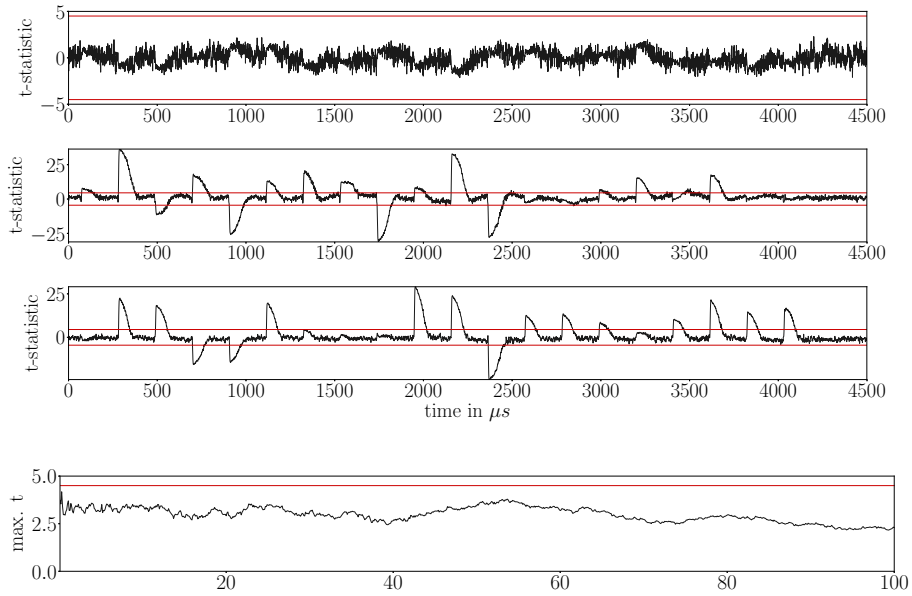


Figure 7: 18 rounds of θ and χ' (i.e., *KECCAK-f* without ρ and π). top to bottom: t-test results first to third order over time axis. Maximal t-values first order over trace axis. Entire last round.

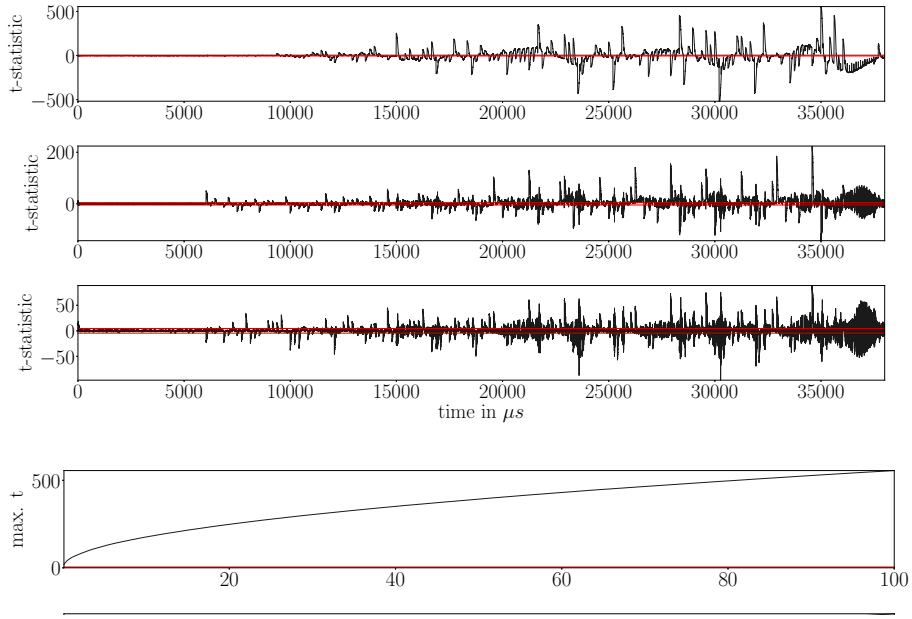


Figure 8: 18 rounds of χ' alone. top to bottom: t-test results first to third order over time axis. Maximal t-values first to third order over trace axis. Entire last round.

Table 2: Summary of practical first-order evaluations.

Active Layers	Leakage?
χ', ρ, π, θ	No
χ', θ	No
χ', ρ, π	**Yes**
χ'	**Yes**

to other security primitives with a non-linear of higher than quadratic degree is challenging. Consider the case of PRESENT [8], the cubic S-box can be decomposed into two quadratic bijections f, g each possessing a non-uniform TI F, G [19]. The non-uniformity caused by F cannot be alleviated by diffusion, before causing leakage in the evaluation of G. Hence, a strictly uniform TI remains important for decomposed non-linear layers.

While simulations of leakage behavior have already proven their utility in finding non-completeness violations in state-of-the-art implementations [2] and in known insecure constructions [3,7,15], finding a flaw based on uniformity violations can be computationally more intensive. On one hand, finding a uniformity flaw between S-box stages is easily possible by exhaustive computation [22]. On the other hand, any simulation of an entire round has to constrain itself to merely sampling the target distribution. It remains an open question how to obtain useful results with few samples. Hence, practical measurements stay a crucial part of leakage investigations.

7 Conclusion

We extended Daemen's [10] theoretical study of lossy mappings with entropy simulations and practical leakage evaluations of different variants of masked Keccak-f. We conclude that Keccak-f achieves practical first-order security even with the non-uniform three-share TI χ' [4] since the diffusion property of its linear layer is sufficient to counteract the loss of entropy. We especially highlight the role of the mixing part (θ) in alleviating the non-uniformity in practical evaluations, whereas shuffling alone (ρ, π) cannot counteract the uniformity loss. Finally, a sampling-based simulation of input distributions is a fast method to falsify security claims, but cannot (and does not aim to) be a substitute for practical evaluation to intensify an indication of leakage absence.

Acknowledgments

The work described in this paper has been supported in part by the German Federal Ministry of Education and Research BMBF (grant nr. 16KIS0666 SysKit_HW).

References

1. Side-channel AttacK User Reference Architecture. http://satoh.cs.uec.ac.jp/SAKURA/index.html.
2. Victor Arribas, Svetla Nikova, and Vincent Rijmen. Vermi: Verification tool for masked implementations. *IACR Cryptology ePrint Archive*, 2017:1227, 2017.
3. Gilles Barthe, Sonia Belaïd, Pierre-Alain Fouque, and Benjamin Grégoire. maskverif: a formal tool for analyzing software and hardware masked implementations. *IACR Cryptology ePrint Archive*, 2018:562, 2018.
4. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Keccak. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conf. on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 313–314. Springer, 2013.
5. Tim Beyne and Begül Bilgin. Uniform first-order threshold implementations. In Roberto Avanzi and Howard M. Heys, editors, *Selected Areas in Cryptography - SAC 2016 - 23rd International Conference, St. John's, NL, Canada, August 10-12, 2016, Revised Selected Papers*, volume 10532 of *Lecture Notes in Computer Science*, pages 79–98. Springer, 2016.
6. Begül Bilgin, Joan Daemen, Ventzislav Nikov, Svetla Nikova, Vincent Rijmen, and Gilles Van Assche. Efficient and First-Order DPA Resistant Implementations of Keccak. In *CARDIS 2013*, volume 8419 of *Lecture Notes in Computer Science*, pages 187–199. Springer, 2014.
7. Roderick Bloem, Hannes Groß, Rinat Iusupov, Bettina Könighofer, Stefan Mangard, and Johannes Winter. Formal verification of masked hardware implementations in the presence of glitches. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, volume 10821 of *Lecture Notes in Computer Science*, pages 321–353. Springer, 2018.
8. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In *CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007.
9. Joan Daemen. On non-uniformity in threshold sharings. In Begül Bilgin, Svetla Nikova, and Vincent Rijmen, editors, *Proc. of the ACM Workshop on Theory of Implementation Security, TIS@CCS 2016 Vienna, Austria, October, 2016*, page 41. ACM, 2016.
10. Joan Daemen. Spectral characterization of iterating lossy mappings. In Claude Carlet, M. Anwar Hasan, and Vishal Saraswat, editors, *Security, Privacy, and Applied Cryptography Engineering - 6th International Conference, SPACE 2016, Hyderabad, India, December 14-18, 2016, Proceedings*, volume 10076 of *Lecture Notes in Computer Science*, pages 159–178. Springer, 2016.
11. 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 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 137–153. Springer, 2017.
12. Yuval Ishai, Amit Sahai, and David A. Wagner. Private circuits: Securing hardware against probing attacks. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO*

2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481. Springer, 2003.
13. Bernhard Jungk and Jürgen Apfelbeck. Area-efficient FPGA implementations of the SHA-3 finalists. In Peter M. Athanas, Jürgen Becker, and René Cumplido, editors, *2011 International Conference on Reconfigurable Computing and FPGAs, ReConFig 2011, Cancun, Mexico, November 30 - December 2, 2011*, pages 235–241. IEEE Computer Society, 2011.
14. Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In *CRYPTO 1999*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.
15. Lauren De Meyer, Begül Bilgin, and Oscar Reparaz. Consolidating security notions in hardware masking. *IACR Cryptology ePrint Archive*, 2018:597, 2018.
16. Amir Moradi, Bastian Richter, Tobias Schneider, and François-Xavier Standaert. Leakage detection with the x2-test. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1):209–237, 2018.
17. 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, 8th International Conference, ICICS 2006, Raleigh, NC, USA, December 4-7, 2006, Proceedings*, volume 4307 of *Lecture Notes in Computer Science*, pages 529–545. Springer, 2006.
18. National Institute of Standards and Technology. Sha-3 standard: Permutation-based hash and extendable-output functions. *FIPS Publikcation*, 2015:1–37, 2015.
19. Axel Poschmann, Amir Moradi, Khoongming Khoo, Chu-Wee Lim, Huaxiong Wang, and San Ling. Side-channel resistant crypto for less than 2, 300 GE. *J. Cryptology*, 24(2):322–345, 2011.
20. 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 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 764–783. Springer, 2015.
21. Tobias Schneider and Amir Moradi. Leakage Assessment Methodology - A Clear Roadmap for Side-Channel Evaluations. In *CHES 2015*, volume 9293 of *Lecture Notes in Computer Science*, pages 495–513. Springer, 2015.
22. Felix Wegener and Amir Moradi. A first-order SCA resistant AES without fresh randomness. In Junfeng Fan and Benedikt Gierlichs, editors, *Constructive Side-Channel Analysis and Secure Design - 9th International Workshop, COSADE 2018, Singapore, April 23-24, 2018, Proceedings*, volume 10815 of *Lecture Notes in Computer Science*, pages 245–262. Springer, 2018.