A Low-Randomness Second-Order Masked AES

Tim Beyne, Siemen Dhooghe, Adrián Ranea, and Danilo Šijačić

imec-COSIC, ESAT, KU Leuven, Belgium
name.lastname@esat.kuleuven.be

Abstract. We propose a second-order masking of the AES in hardware that requires an order of magnitude less random bits per encryption compared to previous work. The design and its security analysis are based on recent results by Beyne *et al.* from Asiacrypt 2020. Applying these results to the AES required overcoming significant engineering challenges by introducing new design techniques. Since the security analysis is based on linear cryptanalysis, the masked cipher needs to have sufficient diffusion and the S-box sharing must be highly nonlinear. Hence, in order to apply the changing of the guards technique, a detailed study of its effect on the diffusion of the linear layer becomes important. The security analysis is automated using an SMT solver. Furthermore, we propose a sharpening of the glitch-extended probing model that results in improvements to our concrete security bounds. Finally, it is shown how to amortize randomness costs over multiple evaluations of the masked cipher.

Keywords: Hardware · Linear Cryptanalysis · Masking · Probing Security · Side-Channel Analysis · Threshold Implementations

1 Introduction

The Advanced Encryption Standard (AES) [10] has been an important building block for many cryptographic applications. For over twenty years, the cipher has largely withstood cryptanalytic attacks. However, just like any other symmetric primitive, naive implementations of the AES are vulnerable to side-channel attacks such as Differential Power Analysis (DPA) due to Kocher *et al.* [17]. To counter these attacks, several adversarial models and side-channel countermeasures have been developed during the past two decades. Masking methods are a common theme among different countermeasures. These methods split all key-dependent variables in the circuit into $d+1$ or more random shares and provide security against d^{th}-order DPA attacks.

Over the years, several first- and even higher-order secure maskings of the AES have appeared in the literature. In particular, several second-order maskings have been proposed: a higher-order threshold implementation [11], a private circuits variant [15], and a multiplicative masking [12]. Despite these advances, all of these works still require significant randomness resources for each evaluation of the round function, namely more than ten-thousand random bits. There are several important downsides to strong randomness requirements. The security

requirements for embedded random number generators used by masking schemes are currently not well understood. As mentioned in NIST's threshold cryptography project roadmap [6], random number generators can be single points of failure. As a result, considerable efforts were made to reduce the randomness costs of maskings. This research culminated in the development of first-order maskings of the AES that did not require fresh randomness [24, 26]. So far, it is not known how to similarly reduce the randomness requirements for higher-order secure maskings of the AES.

Threshold Implementations, proposed by Nikova *et al.* [21], are key to the design of first-order low-randomness maskings. Until recently, this method has had limited success in the higher-order setting as it was only secure against univariate attacks [22]. At Asiacrypt 2020, Beyne *et al.* [2] demonstrated how to design multivariate secure threshold implementations without significantly increasing the randomness costs. Their approach uses linear cryptanalysis to show that the information obtained by second-order probing adversaries cannot be reliably exploited with a finite but large number of queries to the masked cipher.

Although the work of Beyne *et al.* represents an important step towards secure higher-order threshold implementations, it is still quite theoretical and its application was limited to a 7-share masking of the block cipher LED. In addition, it imposes strong requirements such as uniformity and higher-order non-completeness on each shared function. However, there is currently no known uniform sharing of the AES S-box. In the first-order case, the "changing of the guards" method of Daemen [7] can be used to achieve uniformity without fresh randomness. However, as noted by Beyne *et al.*, a direct application of the changing of the guards method would "alter the diffusion of the shared cipher and consequently demand a more detailed security analysis" [2, §8.2]. Finally, for more complicated maskings, the security analysis becomes cumbersome without the use of automated tools.

Contribution. This paper applies the techniques from [2] to design a second-order masking of the AES in hardware. This requires overcoming the difficulties outlined above. The design is based on four shares and requires an order of magnitude fewer random bits per encryption operation than previous work, namely we require only 1800 random bits per encryption including the sharing of the plaintext and key. This randomness cost can be compared with second-order designs such as the one by Groß *et al.* [15] requiring a total of 11312 bits or the work by De Meyer *et al.* [12] requiring 11112 bits.

After reviewing the necessary preliminary material in Section 2, the proposed design is described in Section 3. Several novel design concepts are introduced along the way. It is shown how the second-order non-completeness requirement can be relaxed by means of additional randomness. However, by using the techniques from [2], we are able to show that this randomness can be reused across all S-boxes. In order to maintain strong diffusion even when the changing of the guards method is used to ensure the uniformity of the S-box layer, the "guards in formation" technique is introduced. This technique relies on a detailed analysis

of the interaction between the changing of the guards structure and the linear layer of the cipher.

The design choices made in Section 3 pay off in the security analysis, which is presented in Section 4. As a result, a concrete upper bound on the advantage of bounded-query second-order probing adversaries is obtained. In addition, Section 4 shows how Satisfiability Modulo Theories (SMT) solvers can be used to automate a large portion of the security analysis by identifying optimal linear trails in the masked cipher.

Section 5 investigates the glitch-extended probing model of Faust *et al.* [13] and proposes a sharpened variant. This sharpening results in significant improvements to the security bound of our second-order masking of the AES. We adapt the bounded-query probing model appropriately and apply the necessary changes to the theoretical results from [2]. Our proposals are based on realistic simulations of the behavior of glitches in the masked AES S-box.

Finally, Section 6 proposes a technique to amortize randomness over multiple masked AES calls by extracting randomness during its execution. Moreover, it is shown that this extraction process can be used several thousands of times without a significant security loss. A concrete upper bound on the advantage of bounded-query probing adversaries is derived. This technique allows us to reduce the total number of random bits required to 840 bits per masked encryption call.

2 Preliminaries

This section introduces the bounded-query probing model and the key results from [2] related to the security analysis of higher-order threshold implementations. For convenience, all random variables in this paper are denoted in boldface.

2.1 The Bounded-Query Probing Model

This section introduces the bounded-query probing model of Beyne *et al.* [2] and the main theorem that can be used to prove the security of higher-order masked implementations in this model.

Threshold Probing A d^{th}-order (or d-threshold) probing adversary $\mathcal{A}$, as first proposed by Ishai *et al.* [16], can view up to d gates or wires in a circuit. This circuit encodes an operation, such as a cipher call, and consists of gates, such as AND or XOR gates, and wires. The adversary $\mathcal{A}$ is computationally unbounded, and must specify the location of the probes before querying the circuit. However, the adversary can change the location of the probes over multiple circuit queries. The adversary's interaction with the circuit is mediated through encoder and decoder algorithms, neither of which can be probed.

In the bounded query model, the security of a circuit C with input k against a d^{th}-order probing adversary is quantified by means of the left-or-right security game. The challenger picks a random bit b and provides an oracle $\mathcal{O}^b$, to which adversary $\mathcal{A}$ is given query access. The adversary queries the oracle by choosing

up to d wires to probe – we denote this set of probe positions by $\mathcal{P}$ – and sends it to the oracle along with chosen inputs k_0 and k_1. The oracle responds with the probed wire values of $C(k_b)$. After a total of q queries, the adversary responds to the challenger with a guess for b. For $b \in \{0,1\}$, denote the result of the adversary after interacting with the oracle $\mathcal{O}^b$ using q queries by $\mathcal{A}^{\mathcal{O}^b}$. The left-or-right advantage of the adversary $\mathcal{A}$ is then as defined as

$$\mathrm{Adv}_{\text{-}\mathsf{thr}}(\mathcal{A}) = |\Pr[\mathcal{A}^{\mathcal{O}^0} = 1] - \Pr[\mathcal{A}^{\mathcal{O}^1} = 1]|\,.$$

The above model is extended to capture the effect of glitches on hardware. Whereas a probe normally results in the value of a single wire, a glitch-extended probe allows observing all value used in the calculation of the probed wire up to the previous register layer. More information can be found in the work by Faust *et al.* [13].

Security Analysis The main theoretical result of [2] is that the bounded-query probing security of a masked cipher can be related to its linear cryptanalysis. The first step towards this result is provided by Theorem 1 below, which relates the security of the masked cipher to the Fourier transform of the probability distribution of wire values obtained by probing. The link with linear cryptanalysis will be developed in detail in Section 2.4.

The Fourier transform of a function $V \to \mathbb{C}$, where V is a subspace of $\mathbb{F}_2^n$, can be defined as in Definition 1 below. For the purposes of this section, only probability mass functions on $\mathbb{F}_2^n$ need be considered. Despite this, Definition 1 considers more general functions on an arbitrary subspace $V \subseteq \mathbb{F}_2^n$. Since any vector space over $\mathbb{F}_2$ is isomorphic to $\mathbb{F}_2^n$ for some n, this generalization is mostly a matter of notation. Nevertheless, this extended notation will be convenient in Section 2.4.

Definition 1 ([2], §2.1). *Let $V \subseteq \mathbb{F}_2^n$ be a vector space and $f : V \to \mathbb{C}$ a complex-valued function on V. The Fourier transformation of f is a function $\widehat{f} : \mathbb{F}_2^n/V^\perp \to \mathbb{C}$ defined by*

$$\widehat{f}(u) = \sum_{x \in V} (-1)^{u^\top x} f(x),$$

where we write u for $u + V^\perp$. Equivalently, $\widehat{f}$ is the representation of f in the basis of functions $x \mapsto (-1)^{u^\top x}$ for $u \in \mathbb{F}_2^n/V^\perp$.

Recall that the orthogonal complement $V^\perp$ of a subspace V of $\mathbb{F}_2^n$ is the vector space $V^\perp = \{x \in \mathbb{F}_2^n \mid \forall v \in V : v^\top x = 0\}$. The quotient space $\mathbb{F}_2^n/V^\perp$ is by definition the vector space of cosets of $V^\perp$. For convenience, an element $x + V^\perp \in \mathbb{F}_2^n/V^\perp$ will simply be denoted by x. For $x \in \mathbb{F}_2^n/V^\perp$ and $v \in V$, the expression $x^\top v$ is well-defined. Consequently, the above definition is proper.

The main theorem on the advantage of an adversary in the bounded-query probing model can now be stated. It relies on the observation that, for a bounded-query probing secure circuit, all probed wire values either closely resemble uniform randomness or reveal nothing about the secret input.

Theorem 1 ([2], §4). *Let $\mathcal{A}$ be a t-threshold-probing adversary for a circuit C. Assume that for every query made by $\mathcal{A}$ on the oracle $\mathcal{O}^b$, there exists a partitioning (depending only on the probe positions) of the resulting wire values into two random variables $\mathbf{x}$ ('good') and $\mathbf{y}$ ('bad') such that*

1. *The conditional probability distribution $p_{\mathbf{y}|\mathbf{x}}$ satisfies $\mathbb{E}_{\mathbf{x}}\|\widehat{p}_{\mathbf{y}|\mathbf{x}} - \delta_0\|_2^2 \leq \varepsilon$ with δ_0 the Kronecker delta function,*
2. *Any t-threshold-probing adversary for the same circuit C and making the same oracle queries as $\mathcal{A}$, but which only receives the 'good' wire values (*i.e. *corresponding to $\mathbf{x}$) for each query, has advantage zero.*

The advantage of $\mathcal{A}$ can be upper bounded as

$$\mathrm{Adv}_{t\text{-thr}}(\mathcal{A}) \leq \sqrt{2\,q\,\varepsilon}\,,$$

where q is the number of queries to the oracle $\mathcal{O}^b$.

The advantage of a probing adversary against the circuit can be upper bounded in terms of $\|\widehat{p}_{\mathbf{z}} - \delta_0\|_2$ where $p_{\mathbf{z}}$ is the probability distribution of any measured set of 'bad' wire values, possibly conditioned on several 'good' wire values. The conditioning on 'good' values simply corresponds to fixing some variables in the circuit to constants. Section 2.4 provides the essential link between $\widehat{p}_{\mathbf{z}}$ and the linear cryptanalysis of the shared circuit that will enable us to upper bound the quantity $\|\widehat{p}_{\mathbf{z}} - \delta_0\|_2$ for concrete masked ciphers.

2.2 Boolean Masking and Threshold Implementations

Boolean masking is a technique based on splitting each secret variable $x \in \mathbb{F}_2$ in the circuit into shares $\bar{x} = (x^1, x^2, \ldots, x^{s_x})$ such that $x = \sum_{i=1}^{s_x} x^i$ over $\mathbb{F}_2$. A random Boolean masking of a fixed secret is uniform if all sharings of that secret are equally likely.

There are several approaches to masking a circuit. In this work, we make use of threshold implementations, proposed by Nikova *et al.* [21]. This approach has been extended to capture higher-order univariate attacks by Bilgin *et al.* [4]. In the following, the main properties of threshold implementations are reviewed.

A threshold implementation consists of several layers of Boolean functions. As for any masked implementation, a black-box encoder function generates a uniform random sharing of the input before it enters the shared circuit and the output shares are recombined by a decoder function. At the end of each layer, synchronization is ensured by means of registers.

Let $\bar{F}$ be a layer in the threshold implementation corresponding to a part of the circuit $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$. The function $\bar{F} : \mathbb{F}_2^{ns_x} \to \mathbb{F}_2^{ms_y}$, where we assume s_x shares per input bit and s_y shares per output bit, will be called a *sharing* of F. The i^{th} share of the function $\bar{F}$ is denoted by $F^i : \mathbb{F}_2^{ns_x} \to \mathbb{F}_2^m$, for $i \in \{1, .., s_y\}$. Sharings can have a number of properties that are relevant in the security argument for a threshold implementation; these properties are summarized in Definition 2.

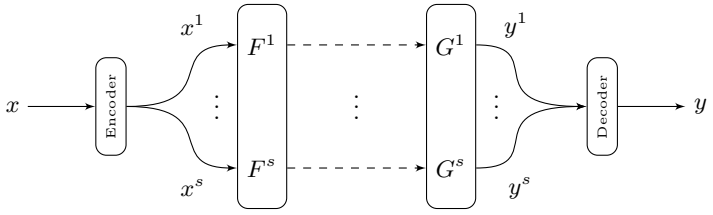


Fig. 1: Schematic illustration of a threshold implementation assuming an equal number of input and output shares.

Definition 2 (Properties of sharings [4, 21]). *Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ be a function and $\bar{F} : \mathbb{F}_2^{ns_x} \to \mathbb{F}_2^{ms_y}$ be a sharing of F. The sharing $\bar{F}$ is said to be*

1. correct *if $\sum_{i=1}^{s_y} F^i(x^1, \ldots, x^{s_x}) = F(x)$ for all $x \in \mathbb{F}_2^n$ and for all shares $x^1, \ldots, x^{s_x} \in \mathbb{F}_2^n$ such that $\sum_{i=1}^{s_x} x^i = x$,*
2. d^{th}-order non-complete *if any function in d or fewer shares F^i depends on at most $s_x - 1$ input shares,*
3. uniform *if $\bar{F}$ maps a uniform random sharing of any $x \in \mathbb{F}_2^n$ to a uniform random sharing of $F(x) \in \mathbb{F}_2^m$.*

2.3 Changing of the Guards

The changing of the guards method proposed by Daemen [7] is a technique that transforms a non-complete sharing into a uniform and non-complete sharing. The technique works by embedding the sharing into a Feistel-like structure. In this paper, we slightly generalize the method by considering a (higher-order) probing secure sharing. Such a sharing potentially requires multiple register stages and extra randomness to guarantee its security. However, the changing of the guards method still ensures the uniformity of the output. An example of the method with four shares is shown in Figure 2.

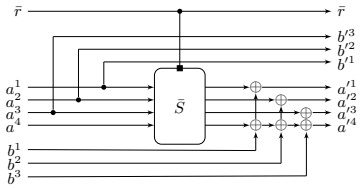


Fig. 2: Changing of the guards method with four shares where the shared S-box $\bar{S}$ uses the randomness $\bar{r}$.

We show that the adapted changing of the guards construction is uniform. This is equivalent to showing that it is invertible for any fixed secret values a

and b. In other words, given secrets a and b and the outputs a'^1, a'^2, a'^3, a'^4, b'^1, b'^2, b'^3, $\bar{r}$, it must be possible to reconstruct the inputs $a^1, a^2, a^3, a^4, b^1, b^2, b^3$ and $\bar{r}$. The derivation is straightforward. Since the secret a is given, the shares a^1, a^2, a^3, a^4 can be recovered from b'^1, b'^2, b'^3. Subtracting a'^1, a'^2, a'^3 from the output of the $\bar{S}$ function yields b^1, b^2, b^3. Since the value $\bar{r}$ was already given and a, b were taken arbitrarily, the construction is indeed invertible.

Additionally, the above construction is still probing secure. Thus, the adapted changing of the guards method allows for the transformation of any probing secure sharing into a uniform one which allows the re-use of the randomness used in the S-box.

2.4 Cryptanalysis of Higher-Order Threshold Implementations

As discussed in Section 2.1, Theorem 1 allows proving the security of higher-order threshold implementations given an upper bound on the Fourier coefficients of probability distributions of wire values obtained by probing. This section shows how such an upper bound can be obtained using linear cryptanalysis.

For any linear masking scheme, there exists a vector space $\mathbb{V} \subset \mathbb{F}_2^\ell$ of valid sharings of zero. More specifically, an $\mathbb{F}_2$-linear secret sharing scheme is an algorithm that maps a secret $x \in \mathbb{F}_2^n$ to a random element of a corresponding coset of the vector space $\mathbb{V}$. Let $\rho : \mathbb{F}_2^n \to \mathbb{F}_2^\ell$ be a map that sends secrets to their corresponding coset representative. For convenience, we denote $\mathbb{V}_a = a + \mathbb{V}$.

Let $\bar{G}$ be a correct sharing of a function $G : \mathbb{F}_2^n \to \mathbb{F}_2^n$ in the sense of Definition 2. Fix any $x \in \mathbb{F}_2^n$ and let $a = \rho(x)$ and $b = \rho(G(x))$. The correctness property implies that $\bar{G}(\mathbb{V}_a) \subseteq \mathbb{V}_b$. It follows that the restriction $F : \mathbb{V}_a \to \mathbb{V}_b$ of $\bar{G}$ defined by $F(x) = \bar{G}(x)$ is a well defined function.

Linear cryptanalysis is closely related to the propagation of the Fourier transformation of a probability distribution under a function $F : \mathbb{V}_a \to \mathbb{V}_b$. This leads to the notion of correlation matrices due to Daemen *et al.* [8]. The action of F on probability distributions can be described by a linear operator. The coordinate representation of this operator with respect to the standard basis $\{\delta_x\}_{x \in V}$ may be called the *transition matrix* of F. Following [1], the correlation matrix of F is then the same operator expressed with respect to the Fourier basis. The correlation matrix of a sharing can be defined as follows. Note that it only depends on the spaces $\mathbb{V}_a$ and $\mathbb{V}_b$, not on the specific choice of the representatives a and b.

Definition 3 (Correlation matrix). *For a subspace $\mathbb{V} \subseteq \mathbb{F}_2^\ell$, let $F : \mathbb{V}_a \to \mathbb{V}_b$ be a function. The correlation matrix C^F of F is a real $|\mathbb{V}_b| \times |\mathbb{V}_a|$ matrix with coordinates indexed by elements $u, v \in \mathbb{F}_2^n / \mathbb{V}^\perp$ and equal to*

$$C^F_{v,u} = \frac{1}{|\mathbb{V}|} \sum_{x \in \mathbb{V}_a} (-1)^{u^\top x + v^\top F(x)} .$$

The relation between Definition 3 and linear cryptanalysis is as follows: the coordinate $C^F_{v,u}$ is equal to the correlation of a linear approximation over F

with input mask u and output mask v. That is, $C^F_{v,u} = 2\Pr[v^\top F(\mathbf{x}) = u^\top \mathbf{x}] - 1$ for $\mathbf{x}$ uniform random on $\mathbb{V}_a$. An important difference with ordinary linear cryptanalysis is that, for shared functions, the masks u and v correspond to equivalence classes. This formalizes the intuitive observation that masks which differ by a vector orthogonal to the space $\mathbb{V}$ lead to identical correlations.

From this point on, we restrict to second-order probing adversaries. The description of the link with linear cryptanalysis presented in [2], is completed by Theorem 2 below. It shows that the coordinates of $\widehat{p}_{\mathbf{z}}$ are entries of the correlation matrix of the state-transformation between the specified probe locations. In Theorem 2, the restriction of $x \in \mathbb{V}_a$ to an index set $I = \{i_1, \ldots, i_m\}$ is denoted by $x_I = (x_{i_1}, \ldots, x_{i_m}) \in \mathbb{F}_2^{|I|}$. This definition depends on the specific choice of the representative a, but the result of Theorem 2 does not.

Theorem 2 ([2], §5.2). *Let $F : \mathbb{V}_a \to \mathbb{V}_b$ be a function with $\mathbb{V} \subset \mathbb{F}_2^\ell$ and $I, J \subset \{1, \ldots, \ell\}$. For $\mathbf{x}$ uniform random on $\mathbb{V}_a$ and $\mathbf{y} = F(\mathbf{x})$, let $\mathbf{z} = (\mathbf{x}_I, \mathbf{y}_J)$. The Fourier transformation of the probability mass function of $\mathbf{z}$ then satisfies*

$$|\widehat{p}_{\mathbf{z}}(u, v)| = |C^F_{\widetilde{v},\, \widetilde{u}}|,$$

where $\widetilde{u}, \widetilde{v} \in \mathbb{F}_2^\ell/\mathbb{V}^\perp$ are such that $\widetilde{u}_I = u$, $\widetilde{u}_{[\ell]\setminus I} = 0$, $\widetilde{v}_J = v$ and $\widetilde{v}_{[\ell]\setminus J} = 0$.

Theorem 2 relates the linear approximations of F to $\widehat{p}_{\mathbf{z}}(u)$ and hence provides a method to upper bound $\|\widehat{p}_{\mathbf{z}} - \delta_0\|_2$ based on linear cryptanalysis. Upper bounding the absolute correlations $|C^F_{\widetilde{v},\, \widetilde{u}}|$ is nontrivial in general. However, the piling-up principle [18, 25] can be used to obtain heuristic estimates.

Importantly, Theorem 2 relates to linear cryptanalysis with respect to $\mathbb{V}$ rather than $\mathbb{F}_2^\ell$. The differences are mostly minor, but there is a subtle difference in relation to the important notion of 'activity'. In standard linear cryptanalysis, an S-box is said to be active if its output mask is nonzero. The same definition applies for linear cryptanalysis with respect to $\mathbb{V}$, but one must take into account that the mask is now an element of the quotient space $\mathbb{F}_2^\ell/\mathbb{V}^\perp$. In particular, if the mask corresponding to the shares of a particular bit can be represented by an all-one vector $(1, 1, \ldots, 1)^\top$, it may be equivalently represented by the zero vector. It is still true that a valid linear approximation for a permutation must have either both input masks equivalent to zero or neither equivalent to zero. More generally, this condition is ensured by any uniform sharing.

3 A Low-Randomness Second-Order Secure AES

In this section, our second-order masking of the AES is introduced.

3.1 Description of the AES

The AES is a family of iterative block ciphers due to Daemen and Rijmen [10] with a 128-bit state that is commonly represented as a 4×4 array of bytes. In this paper, we will only consider the most commonly used variant, known as

AES-128, which has a 128-bit key and consists of 10 rounds. The round function consists of a subkey addition, a bricklayer of S-boxes, a `ShiftRows` operation, and a `MixColumns` step. The AES S-box consists of the map $x \mapsto x^{254}$ in a field $\mathbb{F}_{2^8}$, followed by an affine transformation. A visual representation of the round function is shown in Figure 3.

The key schedule consists of 32-bit operations working on the columns of the key state. Each column is added onto the next, apart from the last column where its bytes are rotated, send through a bricklayer of AES S-Boxes, and added to constants.

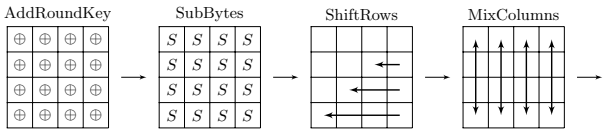


Fig. 3: The AES round function.

3.2 Masking Details

Masking State and Key For the sharing of the AES state and key, we use classical Boolean masking. The 128-bit state is shared using four shares per bit, requiring a total of $128 \times 3 = 384$ random bits. The 128-bit key is also shared using four shares, and this also costs 384 random bits. Finally, we extend the state by an additional column where each cell contains three shares of randomness. This requires an additional $32 \times 3 = 96$ random bits and is necessary for the "guards in formation" technique that will be described in Section 3.4. It will be used over the rows of the state for the S-box layer. An overview of the shared AES round function is shown in Figure 4. The following sections discuss further aspects of this sharing.

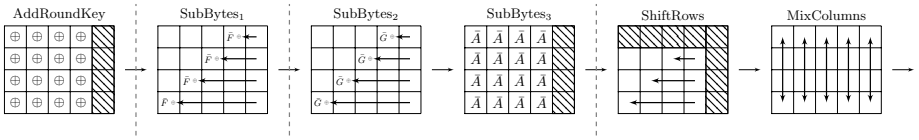


Fig. 4: One round of the masked AES. The locations of the registers are indicated by dashed lines. The nonlinear operations require an additional register stage, which is not shown on the figure. Hatched cells remain unchanged through the operation.

Sharing the Affine Transformations The masking of the linear transformations `ShiftRows` and `MixColumns` is simply done share-wise. Constants are added to the first share of the relevant variable.

Sharing the S-Box The AES S-box consists of an inversion S over $\mathbb{F}_{2^8}$ and an affine layer A. Similar to the other linear layers of the AES, the affine layer is masked using a share-wise approach. Following the work by Wegener *et al.* [26], the inversion ($x \mapsto x^{254}$) is decomposed into two cubic functions. Specifically, we consider

$$x^{254} = \left(x^{26}\right)^{49} = G(F(x)) \,.$$

The sharings $\bar{F}$ and $\bar{G}$ of F and G respectively are chosen such that they are first-order non-complete and second-order probing secure. That is, none of the output shares depends on all of the input shares and placing two probes in the sharing does not reveal any secret values. While the need for second-order probing security is clear, non-completeness is required such that, even when randomness is re-used, a security analysis over multiple rounds remains possible. More details are given in Section 4.2.

Non-complete sharings of F and G can be achieved by using four input shares. More specifically, we use the direct four-sharing of F and G as defined in the thesis of Bilgin [3, pg. 36]. To achieve second-order probing security, the sharings $\bar{F}$ and $\bar{G}$ are split into two stages separated by a register such that $\bar{F} = \bar{F}_2 \circ \bar{F}_1$ and $\bar{G} = \bar{G}_2 \circ \bar{G}_1$ with $\bar{F}_1$, $\bar{F}_2$, $\bar{G}_1$ and $\bar{G}_2$ second-order non-complete. In particular, $\bar{F}_2$ and $\bar{G}_2$ merely implement a linear compression of shares into four output shares. To ensure the second-order probing security of $\bar{F}$ and $\bar{G}$, randomness is added at the end of the stages $\bar{F}_1$ and $\bar{G}_1$. This is depicted in Figure 5. Section 3.3 discuss the specific choice of $\bar{F}_1$ and $\bar{G}_1$ in more detail.

Finally, the sharing is made uniform using the changing of the guards approach of Daemen [7] which was recalled in Section 2.3.

Key Schedule The key schedule is masked similar to the state. Meaning that linear layers are masked share-wise and the masked S-box follows the method above.

Using the linear cryptanalysis tool introduced in Section 4, we find that using the above masked AES S-box with changing of the guards over the four S-boxes does not result in a secure masking of the key-schedule. In fact, one can easily find trails with nonzero correlation over few rounds with a small number of active S-boxes. One such trail is shown in Appendix A. Hence, we instantiate the additional cell due to the changing of the guards technique with fresh randomness in every evaluation. This costs 24 random bits for every $\bar{F}$ or $\bar{G}$ layer for a total randomness cost of $20 \times 24 = 480$ bits plus an additional 384 random bits for the initial sharing of the master key. The 456 random bits to ensure the local second-order probing security of the masked S-box can be re-used from the masked S-boxes in the state.

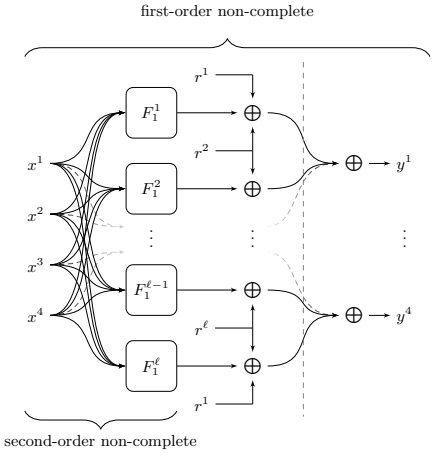


Fig. 5: Sharing of the cubic function F (or G). The Boolean functions $F_1^1, \ldots, F_1^\ell$ are the shares of the first layer $\bar{F}_1$ (or $\bar{G}_1$). The gray dashed line denotes a register stage.

3.3 Optimizing the S-Box Sharing

As explained in Section 3.2, the S-box sharing is realized using a particular technique to ensure second-order non-completeness over each register stage and first-order non-completeness over both stages. This section shows how to minimize the randomness costs by reducing the number of output shares in the second-order non-complete expansion layers $\bar{F}_1$ and $\bar{G}_1$.

We start with a straightforward method to choose $\bar{F}_1$ and $\bar{G}_1$. Denote the output bits of $\bar{F}$ (similarly $\bar{G}$) by $y_i^1, y_i^2, y_i^3, y_i^4$ for $i \in \{1, ..., 8\}$. Each output bit is a function of at most three shares of each input bit due to $\bar{F}$ being non-complete. One can again find a second-order non-complete sharing for each of these three-shared functions. This would indeed result in a decomposition of $\bar{F}$ that is second-order non-complete in each stage, while maintaining the first-order non-completeness of $\bar{F}$.

The above method has the downside that it results in functions $\bar{F}_1$ with a large number of shares. This can be optimized further. Instead of using a second-order non-complete sharing of each F^i (or G^i), we can re-share some functions using a first-order non-complete covering scheme. The optimized covering scheme is shown in Table 1. The third column shows which input shares can be combined in $\bar{F}_1$ (or $\bar{G}_1$). For example, F^1 is re-shared such that each output bit can use only either the first and second share of an input bit or the first and third. This covering scheme is verified to be second-order non-complete. For example, it is clear that one probe can never learn the second and third share of an input bit.

While Table 1 shows a second-order non-complete covering scheme for $\bar{F}_1$ and $\bar{G}_1$, we still need to share $\bar{F}$ or $\bar{G}$ following those requirements. For this, we use the covering schemes from the work by Bozilov [5]. This work provides a

Table 1: This table depicts the share dependencies of an input bit for $\bar{F}$, $\bar{G}$ or $\bar{F}_1$, $\bar{G}_1$. It also shows which random bits are added to the output of $\bar{F}_1$ or $\bar{G}_1$.

Output share	$\bar{F}$ or $\bar{G}$	$\bar{F}_1$ or $\bar{G}_1$	Used random bits
1	$\{x^1, x^2, x^3\}$	$\{x^1, x^2\}, \{x^1, x^3\}$	$r_1, ..., r_{96}$
2	$\{x^2, x^3, x^4\}$	$\{x^2\}, \{x^3\}, \{x^4\}$	$r_{97}, ..., r_{456}$
3	$\{x^1, x^3, x^4\}$	$\{x^1, x^3\}, \{x^1, x^4\}$	$r_1 + r_{97}, ..., r_{96} + r_{192}$
4	$\{x^1, x^2, x^4\}$	$\{x^1, x^2\}, \{x^1, x^4\}$	$r_1 + r_{193}, ..., r_{96} + r_{288}$

covering such that each function F^1, F^3, and F^4 is re-shared to have 12 output shares and F^2 can be re-shared to have 45 output shares. Thus, we have a total of $45 \cdot 8 + 24 \cdot 12 = 648$ output shares for $\bar{F}_1$ (similarly $\bar{G}_1$).

In order to ensure the second-order probing security of the sharings of $\bar{F}$ and $\bar{G}$, a total of 648 random bits are added to the second-order non-complete sharing before re-compression. We can reduce this number by observing that the security condition boils down to requiring that all values seen by the probing adversary need to be masked with a unique random bit. In Table 1, we show a reduction of this randomness to 456 bits. By probing, an adversary can see all random bits related to an output share listed in the table. One can see that, even if two output shares are probed, all observed bits in the expansion are masked with a unique random bit. For example, probing output shares 2 and 3, the adversary observes $r_{97}, ..., r_{456}$ and $r_1 + r_{97}, ..., r_{96} + r_{193}$. The observations are then still masked by the unique random bits $r_{97}, ..., r_{456}$ and $r_1, ..., r_{96}$.

3.4 Guards in Formation

This section discusses the application of the changing of the guards technique to the S-box layer. Recall from Figure 2 that three out of four shares of one input (b^1, b^2, b^3) are used to re-mask the other branch (a'^1, a'^2, a'^3, a'^4) in the Feistel structure. However, when used in a straightforward way, this operation is not second-order non-complete as three shares are used to mask the fourth share of the other branch. In order to make the re-masking operation second-order probing secure, it can be spread across the two stages of the sharing $\bar{F}$ (or $\bar{G}$).

It is also important to consider how the changing of the guards structure links the different cells of the state. As the cryptanalytic properties of our masking affect its security, the diffusion resulting from the linear layer plays an important role. However, from the perspective of linear-cryptanalysis, placing a changing of the guards structure over the S-box layer reduces diffusion. To improve diffusion while keeping the cost minimal, we look for inspiration in the Rijndael-160 cipher [9].

A traditional application of the changing of the guards method would result in one additional state cell. This extra cell is instantiated with a random sharing of zero at the start of execution. One has several options on how to lay out the changing of the guards structure in this case, one example is shown in the left-most illustration in Figure 6. Since the changing of the guards method now mixes

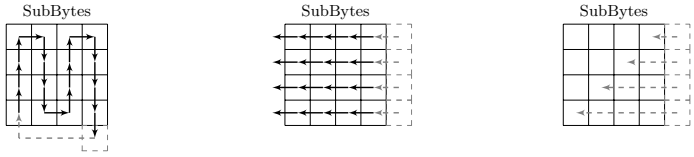


Fig. 6: Three examples of diffusion patterns using the changing of the guards technique. Additional cells are outlined by dashed gray lines. The third example chains all cells in a row.

shares from different cells, it affects the diffusion of the shared cipher. Using the security analysis tool that will be introduced in Section 4, one finds that the example in Figure 6 results in linear trails with few active S-boxes. Consequently, the security bound obtained using Theorem 1 would not be satisfactory.

Instead, we shorten the chain of cells linked by the changing of the guards structure by increasing the number of additional cells. Specifically, the changing of the guards method is applied to the S-boxes in each of the rows of the state independently. This is illustrated in the middle illustration of Figure 6. Furthermore, the diffusion properties over the enlarged state are improved by also applying a `MixColumns` operation over the four extra cells. Finally, to complete the analogy with Rijndael-160, a 'shift' is introduced in the changing of the guards structure. The shifting offset depends on the row number as depicted in the rightmost part of Figure 6. The final result is shown in Figure 4.

4 Security Analysis

This section determines an upper bound on the advantage of second-order probing adversaries for the masked AES construction from Section 3.

4.1 Single Round

In this section, we argue that one round of the masked AES is second-order probing secure. Recall that, without loss of generality, it may be assumed that all probes are placed right before a register stage. Probes placed in linear layers only return one share per input bit. Probing two linear layers is trivially secure, and probing a linear layer and an S-box can be reduced to the following cases where two S-boxes are probed. Hence, only two cases must be considered: both probes placed within the same shared S-box (either $\bar{F}$ or $\bar{G}$), or probes positioned in two different shared S-boxes.

For the first case, consider that the adversary places both probes in $\bar{F}$ (the argument is the same for $\bar{G}$). If both probes are placed in $\bar{F}_1$, the adversary does not receive all shares of any secret bit since $\bar{F}_1$ is a second-order non-complete sharing. Thus, consider the case where the adversary probes the compression layer $\bar{F}_2$. As shown in Table 1, each bit at the input of the expansion layer is masked with a unique random bit. Thus, whatever the choice of the second probe

position, the adversary cannot infer any information about an input secret of $\bar{F}$. As a result, the sharings $\bar{F}$ and $\bar{G}$ are second-order probing secure.

Consider the second case, *i.e.* the two probes are placed in different S-boxes. If both probes are placed in the $\bar{F}$-part of the S-box sharings (similarly $\bar{G}$) then, due to first-order non-completeness, the adversary does not learn any secret bits. Hence, we consider the case where one probe is placed in $\bar{F}$ and one is placed in $\bar{G}$. Consider that $\bar{b}$ denotes the additional branch used for the changing of the guards technique. This branch $\bar{b}$ is a uniform random sharing of zero and is used to re-mask the output of $\bar{F}$. From the probe in $\bar{F}$, the adversary learns at most one output share of $\bar{F}$. We argue that the probe in $\bar{G}$ cannot reveal all of the other output shares. The argument is based on the consistent use of the same covering scheme for both sharings:

- Due to the second-order non-completeness of the expansion layer $\bar{G}_1$ of $\bar{G}$, probing $\bar{G}_1$ does not reveal any of the other three output shares of $\bar{F}$.
- When probing the compression layer $\bar{G}_2$ of $\bar{G}$, the resulting values are masked by the randomness $\bar{r}$. The adversary can only see the same bits of $\bar{r}$ as obtained from the probe in $\bar{F}$ when the indices of the probed output shares of $\bar{F}$ and $\bar{G}$ are the same. The covering scheme used for non-completeness (second column of Table 1) guarantees that the adversary does not obtain all of the output shares of $\bar{F}$.

The above security argument also holds for the key schedule, as it uses the same masked S-box. In particular, the four S-boxes in one round of the key schedule can be considered to be parallel to the 16 S-boxes in the round function.

4.2 Multiple Rounds

Following Theorems 1 and 2, we argue the second-order probing security of multiple rounds of the masked AES by bounding the correlation of all linear trails resulting from two probes placed in different rounds. Specifically, by Theorem 2, bounding the correlation of linear approximations in the masked cipher results in an upper bound on $\|\widehat{p}_{\mathbf{z}} - \delta_0\|_2$ where $p_{\mathbf{z}}$ is the probability distribution of the probed values. In turn, by Theorem 1, this provides a bound on the advantage of second-order probing adversaries.

All of the randomness which is re-used across S-boxes, can be labeled as 'good' in the terminology of Theorem 1. This is safe, since an adversary only probing these values cannot obtain any secret information. Consequently, in the linear cryptanalysis of the masked cipher, these random bits must be considered to be constant. Since F and G are shared in a non-complete way even without the use of randomness, we can assume that any probe placed by the adversary results in a set of non-complete input shares. This is important, since it implies only linear approximations with nonzero input and output masks must be considered.

To ensure the security of the masking, all linear approximations over the sharings $\bar{G}$ and $\bar{F}$ should have a low absolute correlation. Verifying this property is slow since the masked S-box has 32 input bits. Using optimized verification

software, it takes 1500 core hours on an Intel(R) Xeon(R) Gold 6230 CPU with a clock frequency of 2.10 GHz to compute the linear approximation table of a sharing for one input secret (one restriction). The search revealed sharings of the F and G with the following properties.

Claim 1. *Let $\bar{F} : \mathbb{V}_a \to \mathbb{V}_b$ be any restriction of the sharing of F obtained in Section 3.2. Denote its absolute correlation matrix by $|C^{\bar{F}}|$. For any $u, v \in \mathbb{F}_2^\ell / \mathbb{V}^\perp$ not both equal to zero, it holds that $\left|C^{\bar{F}}_{u,v}\right| \leq 2^{-3}$ and, moreover, $\left|C^{\bar{F}}_{0,v}\right| \leq 2^{-3.8}$.*

Claim 2. *Let $\bar{G} : \mathbb{V}_a \to \mathbb{V}_b$ be any restriction of the sharing of G obtained in Section 3.2. Denote its absolute correlation matrix by $|C^{\bar{G}}|$. For any $u, v \in \mathbb{F}_2^\ell / \mathbb{V}^\perp$ not both equal to zero, it holds that $\left|C^{\bar{G}}_{u,v}\right| \leq 2^{-2.6}$ and, moreover, $\left|C^{\bar{G}}_{0,v}\right| \leq 2^{-4}$.*

The above two claims can be verified from the linear approximation table of the sharings. However, the above two results are claims since they were only verified for a couple of secrets.

To upper bound the maximum absolute correlation of linear trails between the observed values (corresponding to nonzero masks), we use a slight refinement of the standard a wide-trail type argument. That is, we search for the best trail activity patterns over the masked AES, but we take into account both cases in Claims 1 and 2. All of this is done using automated tools. Specifically, we encode this search problem as a sequence of Satisfiability Modulo Theories (SMT) problems in the bit-vector theory. These problems are then solved with the off-the-shelf SMT solver Boolector [20]. A similar approach was originally used by Mouha *et al.* [19] to search for activity patterns of the unmasked AES using Mixed Integer Linear Programming. Whereas the approach of Mouha *et al.* is to search for activity patterns with a minimal number of active S-boxes, our approach is to look for activity patterns with maximal absolute correlation. To create and solve these SMT problems we used a development version of ArxPy[1].

To model the correlation of the shared S-boxes, we consider the worst case scenarios from Claims 1 and 2, that is, replacing the inequalities for both cases by equalities. Thus, the correlation of the activity patterns found by our SMT-based method provides an upper bound on the correlation of all linear trails compatible with a specified activity pattern for the input and output masks. An optimal trail is shown in Appendix C. It spans three rounds and has absolute linear correlation at most $2^{-55.20}$. Therefore, the absolute correlation of all relevant linear trails is bounded by $2^{-55.20}$. It follows that the squared 2-norm of the nontrivial Fourier coefficients of the observed bits $\mathbf{z}$ can be upper bounded by

$$\varepsilon := \|\widehat{p}_{\mathbf{z}} - \delta_0\|_2^2 \leq |\operatorname{supp} \widehat{p}_{\mathbf{z}}| \, \|\widehat{p}_{\mathbf{z}} - \delta_0\|_\infty^2 \leq 2^{48} \, 2^{-110.40} = 2^{-62.40},$$

where we have used the inequality $|\operatorname{supp} \widehat{p}_{\mathbf{z}}| \leq 2^{48}$. The latter follows from the fact that the observed value $\mathbf{z}$ consists of at most 48 bits in the glitch-extended probing model: if an output coordinate of $\bar{F}$ or $\bar{G}$ is read, at most 24 shares are

[1] https://github.com/ranea/ArxPy

learned; if an output of the shared linear layer is probed, at most seven shares are observed.

We also built an SMT model for the key schedule. The best trail was found to span eight rounds and activates 21 masked S-boxes. The absolute correlation of the trail is upper bounded by $2^{-63.60}$. Thus, the squared 2-norm of the nontrivial Fourier coefficients of observed bits in the key schedule can be bounded by $\varepsilon \leq 2^{-79.20}$. As the trails through the state transformation have a larger absolute correlation, the upper bound on the maximum advantage of a second-order probing adversary is determined by the bound $\varepsilon \leq 2^{-62.40}$.

4.3 Security Claim

Due to the analysis in Sections 4.1 and 4.2, Theorem 1 can be applied with the upper bound $\varepsilon \leq 2^{-62.40}$. It follows that the following security claim can be made.

Security Claim 1. *For the masked AES described in Section 3, the following bound on the advantage of the adversary (assuming piling-up) in the probing model is claimed:*

$$\mathrm{Adv}_{2\text{-thr}}(\mathcal{A}) \leq \sqrt{\frac{q}{2^{61.4}}}\,.$$

Although the above bound is expressed in terms of the number of probing queries, it can be interpreted in terms of the number of traces taken by an adversary subject to a few assumptions. If the attacker mounts a second-order DPA attack using at most two time samples in a power trace, then the number of queries corresponds to the number of traces. If an adversary does not gather more than 2^{27} (100 million) traces, the above bound shows that the advantage of any attack is at most $2^{-17.2}$. We note that most of the side-channel literature verifies implementations using 100 million traces [11, 12, 14]. The next section shows that this bound can be significantly improved by taking a closer look at the glitch model, leading to a much lower advantage in practice.

5 Sharpening the Glitch Model

In Section 4, an upper bound on the advantage of a second-order probing adversary for the masked AES of Section 3 was obtained. However, we observe that this bound is significantly negatively impacted by the large support of the Fourier transformation of the observed values. Indeed, this leads to an increase in advantage by a factor 2^{24} due to the possibility that a glitch-extended probe may observe 24 bits. In this section, we propose a modified glitch model (supported by simulations) that leads to an improved bound.

We adapt the glitch-extended probing model from Section 2.1. Instead of an adversary which observes the values of all wires that depend on the probed wire (up to the preceding register stage), the new model proposes that the adversary chooses an arbitrary Boolean function of those values. For example, suppose that

the adversary places a glitch-extended probe on a masked function $g(x_1, \ldots, x_n)$. Instead of receiving all of the input values $x_1, \ldots, x_n$, the adversary can choose a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ and receives the value $f(x_1, \ldots, x_n)$ instead. The function f will be referred to as the *glitch function*.

The idea is that, intuitively, the glitch function cannot be completely arbitrary: most Boolean functions in many variables are not easily realized with a small, structured circuit. We formalize this intuition by postulating that the 1-norm of the Walsh-Hadamard transform of the glitch function can not be too large. To exploit this assumption, the following refinement of Theorem 2 is proposed.

Theorem 3. *Let $F : \mathbb{V}_a \to \mathbb{V}_b$ be a permutation with $\mathbb{V} \subset \mathbb{F}_2^\ell$ and $I, J \subset \{1, \ldots, \ell\}$. For $\mathbf{x}$ uniform random on $\mathbb{V}_a$ and $\mathbf{y} = F(\mathbf{x})$, let $\mathbf{z} = (g_1(\mathbf{x}_I), g_2(\mathbf{y}_J))$, $f_1 = 2^{-|I|}(-1)^{g_1}$, and $f_2 = 2^{-|J|}(-1)^{g_2}$. The Fourier transformation of the probability mass function of $\mathbf{z}$ then satisfies*

$$\|\widehat{p}_{\mathbf{z}} - \delta_0\|_2 \leq \|\widehat{f_1}\|_1 \, \|\widehat{f_2}\|_1 \max_{w, w' \in \mathbb{F}_2^\ell / \mathbb{V}^\perp} |C^F_{w,w'}|,$$

where $w, w' \in \mathbb{F}_2^\ell / \mathbb{V}^\perp$ are such that $w'_{[\ell]\setminus I} = 0$ and $w_{[\ell]\setminus J} = 0$.

Proof. Let $\mathbf{z}' = (\mathbf{x}_I, \mathbf{y}_J)$. By Theorem 2, it holds that $\widehat{p}_{\mathbf{z}'}(u, v) = C^F_{\widetilde{v}, \widetilde{u}}$ where $\widetilde{u}, \widetilde{v} \in \mathbb{F}_2^\ell / \mathbb{V}^\perp$ are such that $\widetilde{u}_I = u$, $\widetilde{u}_{[\ell]\setminus I} = 0$, $\widetilde{v}_J = v$, and $\widetilde{v}_{[\ell]\setminus J} = 0$. In addition, it holds that

$$\widehat{p}_{\mathbf{z}}(1,1) = \left[(C^{f_1} \otimes C^{f_2}) \widehat{p}_{\mathbf{z}'} \right]_{1,1} = \sum_{u \in \mathbb{F}_2^{|I|}, \, v \in \mathbb{F}_2^{|J|}} \widehat{f_1}(u) \widehat{f_2}(v) C^F_{\widetilde{v}, \widetilde{u}} \, .$$

Since $\widehat{p}_{\mathbf{z}}(0,0) = 1$, $\widehat{p}_{\mathbf{z}}(0,1) = 0$, and $\widehat{p}_{\mathbf{z}}(1,0) = 0$, it holds that $\|\widehat{p}_{\mathbf{z}} - \delta_0\|_2 = |\widehat{p}_{\mathbf{z}}(1,1)|$. The absolute value of $\widehat{p}_{\mathbf{z}}(1,1)$ can be upper-bounded using the triangle inequality:

$$|\widehat{p}_{\mathbf{z}}(1,1)| \leq \sum_{u \in \mathbb{F}_2^{|I|}, \, v \in \mathbb{F}_2^{|J|}} |\widehat{f_1}(u)| \, |\widehat{f_2}(v)| \, |C^F_{\widetilde{v}, \widetilde{u}}| \leq \|\widehat{f_1}\|_1 \, \|\widehat{f_2}\|_1 \max_{w, w' \in \mathbb{F}_2^\ell / \mathbb{V}^\perp} |C^F_{w,w'}| \, .$$

□

The above theorem shows that we can improve the bound on a probing adversary if we know the 1-norm of the Walsh-Hadamard transformation of the glitch functions. In order to upper bound this 1-norm, one can simulate the effect of glitches on the circuit.

Our simulation setup is based on the work of Šijačić *et al.* [23]. We obtain gate-level netlists using Synopsys DesignCompiler with a 45 nm standard-cell library from NanGate. Composite Current Source (CCS) models provide detailed timing information with 1 ps precision. CCS timing also captures different gate propagation delays for every pin and signal edge (rising or falling). Thus, we include the effects of data-dependent glitches in the simulations. The distribution

of data-dependent glitches is also affected by the routing wires delays and random fluctuations of the operating environment (*e.g.* noise, temperature, and operating voltage). To account for these influences, we annotate delays of all ports and wires using random values drawn from a normal distribution with mean 100 ps and variance $30\,(\text{ps})^2$. We use MentorGraphics QuestaSim for logic simulation. Lastly, we develop custom parsers to obtain continuous identity function traces from logic simulation outputs.

We simulate the masked AES S-box from Section 3 and compute its Walsh-spectrum for different probe positions. However, since the masked AES S-box has 32 input bits, the memory requirements for performing many experiments are large. To provide an additional example, we also simulate a 9-bit XOR. The spectrum and 1-norms for different input delays, initial states, and different probe positions and time samples for the masked AES S-box and 9-bit XOR is given in Appendix B.

Using the observed 1-norm and the above theorem, we can improve the security bound for the masked AES from Section 3. From Figure 11 in Appendix B, we observe that the 1-norm of the Walsh-Hadamard transform of the glitch functions for the masked AES S-box is between 600 and $1635 \approx 2^{10.68}$. By applying Theorem 3, the squared 2-norm of the nontrivial Fourier coefficients of the observed bits $\mathbf{z}$ can be upper bounded by

$$\|\widehat{p}_{\mathbf{z}} - \delta_0\|_2^2 \leq 2^{21.36}\, 2^{-110.40} = 2^{-89.04},$$

This gives the following refined security claim.

Security Claim 2. *For the masked AES described in Section 3, the following bound on the advantage of the adversary (assuming piling-up) in the probing model with refined glitches is claimed:*

$$\mathrm{Adv}_{2\text{-thr}}(\mathcal{A}) \leq \sqrt{\frac{q}{2^{88.04}}}\,.$$

6 Amortizing Randomness Over Multiple Queries

This section introduces a method to safely extract randomness from the state of the masked AES from Section 3 for usage in the next call. This further reduces the requirements on the (true) random number generator used in the implementation.

The technique works by extracting the randomness from the masked state in certain rounds. Here extraction means taking three out of four shares of each bit of the state for subsequent use. Directly using the extracting state bits as randomness in the next masked cipher call would be insecure in the probing model. Instead, we extract randomness from multiple rounds and add it together. A total of twice the shared state size, 960 random bits, is extracted per masked cipher call. This process is depicted in Figure 7.

Suppose that the randomness is refreshed after every l calls to the masked cipher, *i.e.* after l blocks have been encrypted. To assess the security of such

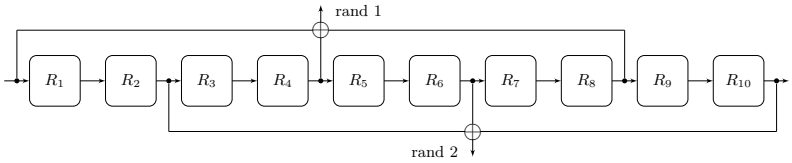


Fig. 7: Extraction of randomness from a masked AES to create the random bits "rand 1" and "rand 2". The i^{th} round function of AES is denoted by R_i.

a construction, one can rely on a variant of the bounded query probing model from Section 2.1. Since the regular circuit oracle only generates new randomness every l calls, it is more convenient to consider an oracle for which each query corresponds to l invocations of the masked cipher. Equivalently, the adversary is given oracle access to a circuit that consists of l blocks which are only connected by the circuitry required to reuse randomness. The adversary is allowed to reposition the probes after each invocation of the masked circuit, so up to $2l$ probes per query are provided. However, only two probes per block are allowed. An adversary with this access structure may be called an l-block 2-threshold adversary. This is illustrated in Figure 8.

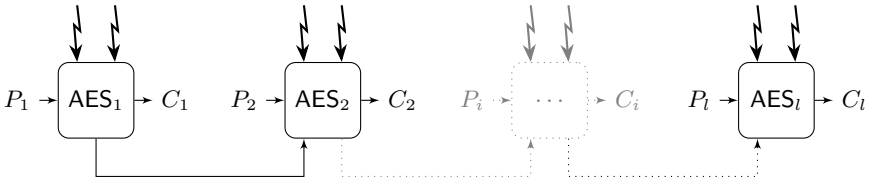


Fig. 8: Illustration of the l-block 2-threshold probing model.

It is straightforward to adapt Theorem 1 to l-block 2-threshold adversaries. The only difference is in the admissible probe positions. However, when applying this result to determine the security bound, care must be taken in the labeling (as 'good' or 'bad') of probed values. In the $l = 1$ case, probed values resulting from two probes placed within the same S-box were marked as 'good'. This helps to avoid corner cases that do not threaten the probing security, but that prevent the direct replacement of these values with uniform randomness in the proof of Theorem 1. For $l \geq 2$, the presence of probes in other blocks makes this labeling incorrect and an additional reduction is required before applying the theorem. In this initial step, the set of values obtained by placing both probes of a block in the same S-box is modified (expanded) to a set of values with a uniform random *marginal* distribution. By simply returning all but one of the input shares of the S-box, this is achieved without any loss in security. The modified values can then be labeled as 'bad'.

To complete the argument, an upper bound on $\|\widehat{p}_{\mathbf{z}} - \delta_0\|_2^2$ must be determined. Here, $\mathbf{z}$ consists of the (modified) probed values conditioned on any values labeled as 'good'. This requires an analysis of linear trails, including trails that run across multiple blocks through the randomness extraction mechanism shown in Figure 7. Suppose that $\varepsilon = 2^b \times c^2$, where b is the maximum number of bits seen by any pair of probes (possibly in different blocks) and c^2 an upper bound on the squared correlation of any linear approximation between such a pair of probes. Grouping masks by the number of active probes, and applying the piling-up principle, one obtains the upper bound

$$\|\widehat{p}_{\mathbf{z}} - \delta_0\|_2^2 \leq \sum_{n=2}^{2l} \binom{2l}{n} \varepsilon^{n-1} = \varepsilon^{-1}\left[(1+\varepsilon)^{2l} - (1+2l\,\varepsilon)\right] \leq l(2l-1)\varepsilon\,.$$

The actual number of probes could be lower than $2l$ due to the initial reduction that essentially combines two probes placed in one S-box into a single probe. It follows that a second-order probing adversary making a total of q queries has advantage bounded by

$$\mathrm{Adv}_{l\text{-blk},2\text{-thr}}(\mathcal{A}) \leq \sqrt{2l-1} \times \sqrt{2\,q\,\varepsilon}\,.$$

That is, a factor of at most $\sqrt{2l-1}$ is lost when randomness is reused over l masked cipher calls. This loss is due to the adversary's capability of placing two probes in each query in arbitrary positions. However, the information gathered in each query does not directly relate to a secret. Instead, many queries are needed to distinguish this information from uniform random. Thus, it is unclear whether these attacks relate to second-order DPA attacks. Hence when factoring in noise, we expect that the above bound can be significantly improved. We pose this improvement as an open problem.

Using SMT-based search tools, we find the best linear trail with two probes in one AES circuit and an arbitrary non-zero mask on the extracted randomness. Optimal trails are shown in Appendix D.[2] The best trail with a non-zero mask on the extracted randomness has absolute correlation at most $2^{-61.0}$. Thus, the dominant trails are still those given in Section 4.2 and we again have the bound $\varepsilon \leq 2^{-62.40}$.

Security Claim 3. *For the masked AES with randomness reuse, the following bound on the advantage of the adversary (assuming piling-up) in the probing model is claimed:*

$$\mathrm{Adv}_{l\text{-blk},2\text{-thr}}(\mathcal{A}) \leq \sqrt{2l-1} \times \sqrt{\frac{q}{2^{61.4}}}\,.$$

[2] We exclude the attack where probes are placed directly on the XOR between the extracted states. We neglect this attack due to an XOR using too low power consumption for it to be usable by the attacker. However, if this attack is of concern, one can use two bits of extracted randomness and XOR them with the rest at the start of the next cipher call to avoid it.

By making use of the sharpened glitch model from Section 5, the above bound can again be improved significantly.

This technique generates a total of 960 random bits per masked cipher call. In total, the masked AES from Section 3 requires 1800 random bits, 936 bits for the state and 864 bits for the key. By choosing $l > 936$, the effective number of random bits for the state is less than one per call. This increases the maximum advantage of a second-order probing adversaries by a factor less than 2^6.

The above security claim should be investigated further as the piling-up principle might not be applicable since the randomness extraction makes the entire masked cipher non-uniform. Due to this, there may be a large number of trails with a similar absolute correlation for a given mask on the extracted randomness.

7 Conclusion

A second-order masked AES using less than 2000 bits of randomness was developed. Furthermore, it was shown how the randomness cost can be amortized over several cipher calls. This resulted in a total cost of less than 1000 random bits per encrypted block. These low randomness costs are the result of careful design choices guided by a detailed security analysis in the bounded-query probing model. In particular, the sharing of the nonlinear layer is based on a variant of the changing of the guards method, which we call "guards in formation" and improves the diffusive properties of the masked cipher. An automated tool based on SMT solvers was used to bound the absolute correlation of linear trails through the masked cipher. This resulted in concrete upper bounds on the advantage of probing adversaries.

The main open problem of this work is the efficiency of the masking. We note that our S-box masking extends the one from Wegener *et al.* [26], who reported that the S-box costs over 20k GE when unrolled. Synthesis of our unrolled S-box shows that it requires over 40k GE. Hence, the presented AES masking will not competitive with the current state-of-the-art in terms of area-requirements. However, we expect that future work will improve the efficiency of the implementation.

Acknowledgment. We thank Vincent Rijmen, Svetla Nikova, Lauren De Meyer, and Victor Arribas for interesting discussions. We also thank Dušan Božilov for his help on covering schemes. Tim Beyne, Siemen Dhooghe, and Adrián Ranea are supported by a PhD Fellowship from the Research Foundation – Flanders (FWO).

References

1. Beyne, T.: Block cipher invariants as eigenvectors of correlation matrices. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018, Part I. LNCS, vol. 11272, pp. 3–31. Springer, Heidelberg (Dec 2018). https://doi.org/10.1007/978-3-030-03326-2_1

2. Beyne, T., Dhooghe, S., Zhang, Z.: Cryptanalysis of masked ciphers: A not so random idea. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part I. LNCS, vol. 12491, pp. 817–850. Springer, Heidelberg (Dec 2020). https://doi.org/10.1007/978-3-030-64837-4_27
3. Bilgin, B.: Threshold implementations : as countermeasure against higher-order differential power analysis. Ph.D. thesis, University of Twente, Enschede, Netherlands (2015), http://purl.utwente.nl/publications/95796
4. Bilgin, B., Gierlichs, B., Nikova, S., Nikov, V., Rijmen, V.: Higher-order threshold implementations. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part II. LNCS, vol. 8874, pp. 326–343. Springer, Heidelberg (Dec 2014). https://doi.org/10.1007/978-3-662-45608-8_18
5. Bozilov, D.: On optimality of d + 1 TI shared functions of 8 bits or less. IACR Cryptol. ePrint Arch. **2020**, 570 (2020), https://eprint.iacr.org/2020/570
6. Brandão, L.T.A.N., Davidson, M., Vassilev, A.: NIST Roadmap Toward Criteria for Threshold Schemes for Cryptographic Primitives. National Institute of Standards and Technology (NIST), U.S. Department of Commerce (July 2020), https://csrc.nist.gov/publications/detail/nistir/8214a/final
7. Daemen, J.: Changing of the guards: A simple and efficient method for achieving uniformity in threshold sharing. In: Fischer, W., Homma, N. (eds.) CHES 2017. LNCS, vol. 10529, pp. 137–153. Springer, Heidelberg (Sep 2017). https://doi.org/10.1007/978-3-319-66787-4_7
8. Daemen, J., Govaerts, R., Vandewalle, J.: Correlation matrices. In: Preneel, B. (ed.) FSE'94. LNCS, vol. 1008, pp. 275–285. Springer, Heidelberg (Dec 1995). https://doi.org/10.1007/3-540-60590-8_21
9. Daemen, J., Rijmen, V.: The block cipher rijndael. In: Quisquater, J., Schneier, B. (eds.) Smart Card Research and Applications, This International Conference, CARDIS '98, Louvain-la-Neuve, Belgium, September 14-16, 1998, Proceedings. Lecture Notes in Computer Science, vol. 1820, pp. 277–284. Springer (1998). https://doi.org/10.1007/10721064_26, https://doi.org/10.1007/10721064_26
10. Daemen, J., Rijmen, V.: Advanced Encryption Standard (AES). National Institute of Standards and Technology (NIST), FIPS PUB 197, U.S. Department of Commerce (Nov 2001)
11. De Cnudde, T., Reparaz, O., Bilgin, B., Nikova, S., Nikov, V., Rijmen, V.: Masking AES with d+1 shares in hardware. In: Gierlichs, B., Poschmann, A.Y. (eds.) CHES 2016. LNCS, vol. 9813, pp. 194–212. Springer, Heidelberg (Aug 2016). https://doi.org/10.1007/978-3-662-53140-2_10
12. De Meyer, L., Reparaz, O., Bilgin, B.: Multiplicative masking for AES in hardware. IACR TCHES **2018**(3), 431–468 (2018). https://doi.org/10.13154/tches.v2018.i3.431-468, https://tches.iacr.org/index.php/TCHES/article/view/7282
13. Faust, S., Grosso, V., Pozo, S.M.D., Paglialonga, C., Standaert, F.X.: Composable masking schemes in the presence of physical defaults & the robust probing model. IACR TCHES **2018**(3), 89–120 (2018). https://doi.org/10.13154/tches.v2018.i3.89-120, https://tches.iacr.org/index.php/TCHES/article/view/7270
14. Groß, H., Mangard, S.: Reconciling d+1 masking in hardware and software. In: Fischer, W., Homma, N. (eds.) CHES 2017. LNCS, vol. 10529, pp. 115–136. Springer, Heidelberg (Sep 2017). https://doi.org/10.1007/978-3-319-66787-4_6
15. Groß, H., Mangard, S., Korak, T.: An efficient side-channel protected AES implementation with arbitrary protection order. In: Handschuh, H. (ed.) CT-

RSA 2017. LNCS, vol. 10159, pp. 95–112. Springer, Heidelberg (Feb 2017). https://doi.org/10.1007/978-3-319-52153-4_6
16. Ishai, Y., Sahai, A., Wagner, D.: Private circuits: Securing hardware against probing attacks. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 463–481. Springer, Heidelberg (Aug 2003). https://doi.org/10.1007/978-3-540-45146-4_27
17. Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M.J. (ed.) CRYPTO'99. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (Aug 1999). https://doi.org/10.1007/3-540-48405-1_25
18. Matsui, M.: Linear cryptanalysis method for DES cipher. In: Helleseth, T. (ed.) EUROCRYPT'93. LNCS, vol. 765, pp. 386–397. Springer, Heidelberg (May 1994). https://doi.org/10.1007/3-540-48285-7_33
19. Mouha, N., Wang, Q., Gu, D., Preneel, B.: Differential and linear cryptanalysis using mixed-integer linear programming. In: Inscrypt. Lecture Notes in Computer Science, vol. 7537, pp. 57–76. Springer (2011)
20. Niemetz, A., Preiner, M., Biere, A.: Boolector 2.0 system description. Journal on Satisfiability, Boolean Modeling and Computation **9**, 53–58 (2015)
21. Nikova, S., Rechberger, C., Rijmen, V.: Threshold implementations against side-channel attacks and glitches. In: Ning, P., Qing, S., Li, N. (eds.) ICICS 06. LNCS, vol. 4307, pp. 529–545. Springer, Heidelberg (Dec 2006)
22. Reparaz, O.: A note on the security of higher-order threshold implementations. Cryptology ePrint Archive, Report 2015/001 (2015), http://eprint.iacr.org/2015/001
23. Sijacic, D., Balasch, J., Yang, B., Ghosh, S., Verbauwhede, I.: Towards efficient and automated side-channel evaluations at design time. J. Cryptogr. Eng. **10**(4), 305–319 (2020)
24. Sugawara, T.: 3-share threshold implementation of AES s-box without fresh randomness. IACR TCHES **2019**(1), 123–145 (2018). https://doi.org/10.13154/tches.v2019.i1.123-145, https://tches.iacr.org/index.php/TCHES/article/view/7336
25. Tardy-Corfdir, A., Gilbert, H.: A known plaintext attack of FEAL-4 and FEAL-6. In: Feigenbaum, J. (ed.) CRYPTO'91. LNCS, vol. 576, pp. 172–181. Springer, Heidelberg (Aug 1992). https://doi.org/10.1007/3-540-46766-1_12
26. Wegener, F., Moradi, A.: A first-order SCA resistant AES without fresh randomness. In: Fan, J., Gierlichs, B. (eds.) COSADE 2018. LNCS, vol. 10815, pp. 245–262. Springer, Heidelberg (Apr 2018). https://doi.org/10.1007/978-3-319-89641-0_14

A Trails in the Second-Order Masked Key Schedule

This appendix provides the activity-pattern of a trail with non-zero correlation over two rounds of the second-order masked key schedule of the AES. This is the key-schedule described in Section 3.2, but *without using fresh randomness* for the changing of the guards technique. The trail is depicted in Figure 9. Recall that the AES S-box is split in two cubic maps and that the additional cell used for the changing of the guards technique is passed on to the next nonlinear operation.

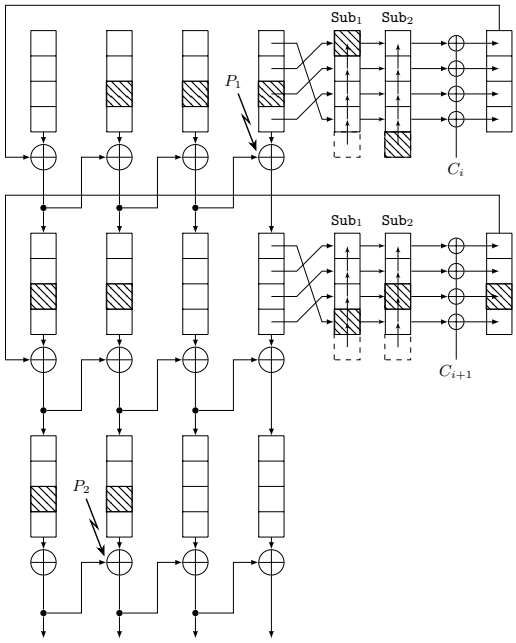
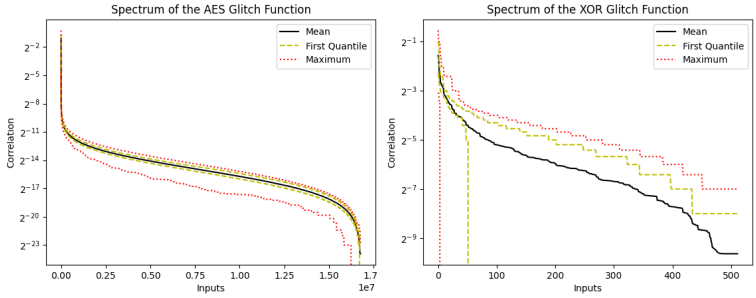


Fig. 9: Activity pattern of a trail with non-zero correlation through two rounds of the second-order masked key schedule of AES caused by two probes. Hatched cells correspond to active cells. The lightning signs denote the two probes.

B Walsh Spectra of the AES and XOR

In this appendix we simulate the masked AES S-box from Section 3 and compute its Walsh-spectrum for different probe positions. The spectrum for a single input delay and initial state but different probe positions and time samples for the masked AES S-box is given in Figure 10a. The spectrum for different input delays, initial states, probe positions, and time samples for the 9-bit XOR is given in Figure 10b. The distributions of the 1-norms are illustrated in Figure 11 using a box plot.



(a) AES S-box. (b) 9-bit XOR.

Fig. 10: Sorted absolute Walsh-spectrum of glitch functions for the AES S-box and the 9-bit XOR for different probe positions, times, inputs delays, and initial states.

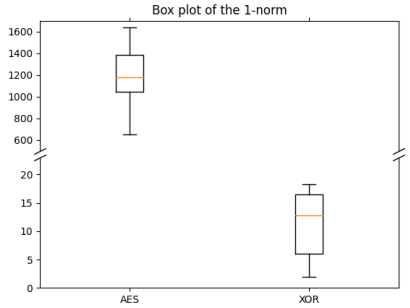


Fig. 11: Box plot of the 1-norm of the Walsh-Hadamard transform of the glitch functions for the masked AES and the XOR, for different probe positions, times, input delays and initial states.

C Trails in the Second-Order Masked State

In this appendix we give trails through the state of the second-order masked AES. We give the best trail with absolute linear correlation at most $2^{-55.20}$ in Figure 12.

D Trails in the Generation of Randomness

In this appendix we give trails of the second-order masked AES where randomness is generated following the method described in Section 6. We give the best trail with absolute linear correlation at most $2^{-61.0}$ in Figure 13. The hatched cells in these figures denote the active cells in the output masks for the indicated transformations.

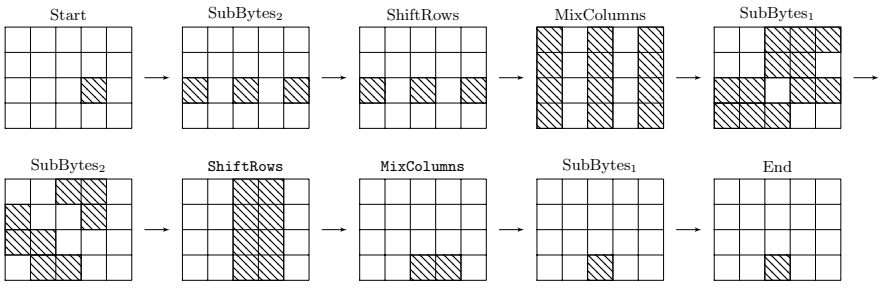


Fig. 12: An optimal trail (for two probes) with non-zero correlation through the second-order masked AES. The hatched cells denote active cells of the output of the operation.

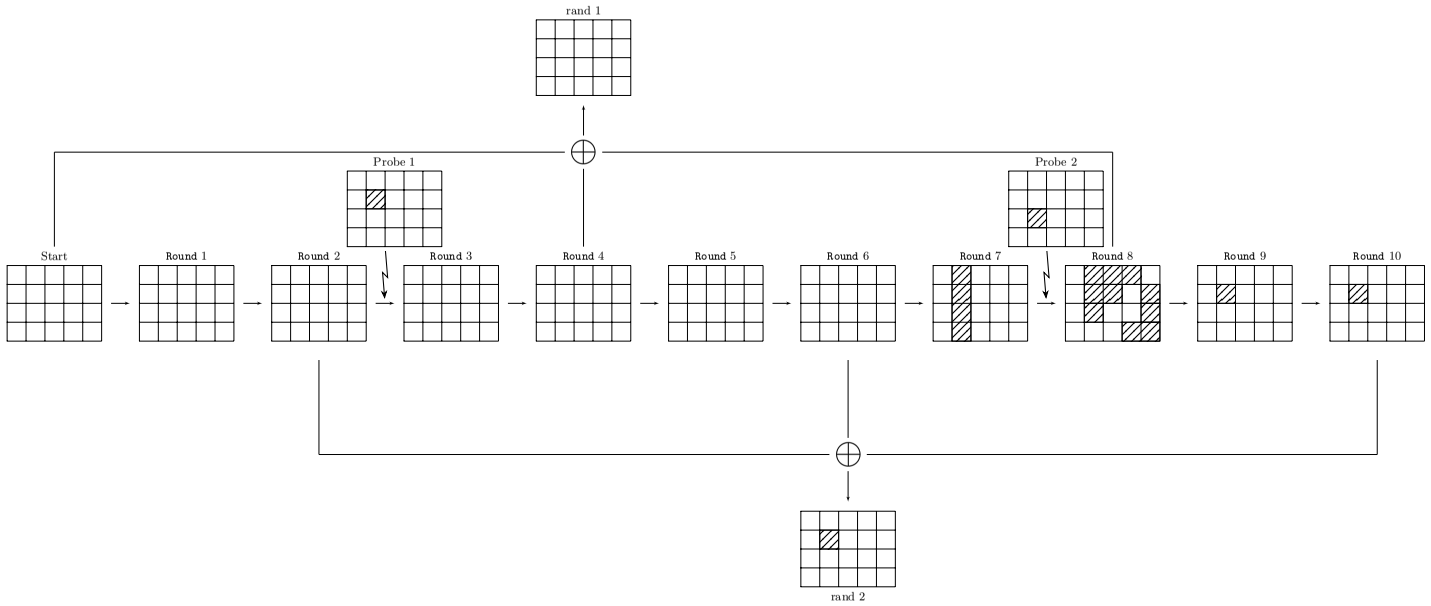


Fig. 13: A trail achieving the highest correlation on the extracted randomness. The trail essentially follows the one given in Figure 12.