# Side-Channel Attacks on Threshold Implementations using a Glitch Algebra

Serge Vaudenay

EPFL
CH-1015 Lausanne, Switzerland
http://lasec.epfl.ch

**Abstract.** Threshold implementations allow to implement circuits using secret sharing in a way to thwart side-channel attacks based on probing or power analysis. It was proven they resist to attacks based on glitches as well. In this report, we show the limitations of these results. Concretely, this approach proves security against attacks which use the average power consumption of an isolated circuit. But there is no security provided against attacks using a non-linear function of the power traces (such as the mean of squares or the majority of a threshold function), and there is no security provided for cascades of circuits, even with the power mean. We take as an example the threshold implementation of the AND function by Nikova, Rechberger, and Rijmen with 3 and 4 shares. We further consider a proposal for higher-order by Bilgin *et al.*

## 1 Introduction

Since the late 1990's, many side-channel attacks based on either power analysis or probing have been presented. We consider essentially two types of attacks. In Differential power attacks (DPA), the adversary collects many samples of the sum of the power used by all gates of the circuit with noise. In Probing attacks, the adversary gets a few intermediate values of the computation by probing the circuit. All measures are subject to noise and can be modeled [2]. Duc *et al.* have shown that these two attacks are essentially equivalent [4].

One devastating type of attack is based on "glitches". It takes into account that electric signals are not necessarily a classical 0/1 signal but a real function over a clock period which is non constant. For instance, the signal can be intermediate between 0 and 1, or switching several times between 0 and 1 during the clock period, or a signal with a very short switching peak, etc. The CMOS technology uses very little power. Signals switching in between clock periods use power. Essentially, only signal switches use power. So, a glitch induces an abnormal power consumption which is visible during a clock period [5].

To avoid these attacks, *masking* is a common method. Essentially, instead of running the computations based on inputs $x$ and $y$ to obtain a result $z$, we first use a secret sharing for $x$ and $y$ to split it into $n$ random shares $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_n)$ and run the computation on the shares to obtain a sharing $(z_1, \ldots, z_n)$ of $z$. Usually, the secret sharing is the simple $(n, n)$-scheme in which

$x = x_1 \oplus \cdots \oplus x_n$, $y = y_1 \oplus \cdots \oplus y_n$, and $z = z_1 \oplus \cdots \oplus z_n$. Trichina, Korkishko, and Lee [11] proposed an implementation of an AND gate with $n = 2$.

In [7], Nikova, Rechberger, and Rijmen proposed the *threshold implementation* which transforms a gate (such as an AND gate) into a circuit which resists to probing attacks with a single probe or DPA based on the average of the power consumption. One construction uses $n = 3$ and another one with $n = 4$ has the property that output shares are always balanced. In [1], Bilgin *et al.* extend this method to higher orders, to make circuits resisting to 2 probes or DPA based on a 2nd order moment of the power consumption. They propose an implementation of an AND gate with $n = 5$ but this implementation requires internal flip/flop registers, thus induce latencies, just to have a secure AND circuit. These constructions were recently consolidated in [9].

*Our results.* As the glitch propagation model highly depends on concrete implementations, in this paper, we consider several models for accounting glitches obtained by the XOR of two glitched signals. We do not advertise any model to be better but rather show how little influence the model has on the security results. In a first model, the "double-glitch" simply counts as twice a normal glitch. In this model, the mean power for the construction with $n = 2$ does not leak. In a second model, the double-glitch counts as a normal one. In a third model, the two glitches cancel each other and do not count. In the two latter models, the construction with $n = 2$ leaks from the mean power.

In the mentioned constructions using $n > 2$, we show that two probes leak, that some non-linear function of the power (such as the mean of squares or the majority of a threshold function) leak, and that by composing two circuits implementing two AND gates, one probe leaks.

Finally we show that in the three models, the AND construction using $n = 5$ (the one resisting 2nd order attacks) does not resist to an attack with two probes when we do not add internal flip/flop registers.

The security claims coming with these implementations from the literature are of the form "if [*conditions*] then we have security". We do not contradict any of these results. In this paper, we complement them by showing that when the conditions are not met, we clearly have insecurity. So, these conditions are not only sufficient: they are also necessary.

## 2 The Theory

### 2.1 The Glitch Algebra

Algebra is "the part of mathematics in which letters and other general symbols are used to represent numbers and quantities in formulae and equations". Herein, we propose to represent glitches as well and to do operations on glitches.

In what follows we use the following conventions: a "signal" is a function from a clock cycle $[0, \tau]$ to $\mathbf{R}$; we consider real numbers as constant signals, we consider bits as real numbers in $\{0, 1\}$; $+$ and $\times$ denote the addition and multiplication of reals; $\oplus$, $\vee$, and $\wedge$ denote the XOR, OR, and AND of signals.

A signal "represents" a bit. To avoid confusion, from now on we denote with regular letters a signal and we denote with a bar the bit it is supposed to represent. We say that a signal $x$ has no glitch if it is constant and equal to the bit $\bar{x}$ it represents. The functions $\oplus$, $\vee$, and $\wedge$ are defined by the gates implementing these functions. We only know that they match what we know about bits: $a \oplus b = a + b \bmod 2$, $a \vee b = \max(a, b)$, and $a \wedge b = ab$ when $a$ and $b$ have no glitch. Furthermore, we define a function $\mathsf{glitch}$ giving the "number of glitches" in a signal and a function $\mathsf{power}$ giving the power consumption of a gate. We assume that $\mathsf{glitch}(x) = 0$ if $x$ has no glitch. The function $\mathsf{glitch}$ applies to a signal but the function $\mathsf{power}$ applies to a gate. Concretely, a gate $g = \mathsf{op}(a, b)$ with output signal $c$ corresponds to $\mathsf{power}(g) = \mathsf{glitch}(c) p_{\mathsf{op}}$ where $p_{\mathsf{op}}$ is a constant. So, $\mathsf{power}(g) = 0$ if $\mathsf{op}(a, b)$ has no glitch. Actually, this is an approximation. Essentially, it is assumed that a stable signal uses very little power while a glitch induces a high power consumption, like in the CMOS technology [5]. The assumption on the influence of glitches on the power consumption may be a bit arbitrary. In the sequel, we take for granted that when $y$ has no glitch, then $x \oplus y$ has the same glitch as $x$. When $y$ has no glitch and $\bar{y} = 0$, we assume that $x \wedge y$ has no glitch either (due to the AND with 0). When $y$ has no glitch and $\bar{y} = 1$, we assume that $x \wedge y$ has the same glitch as $x$. So,

$$\mathsf{glitch}(x \wedge y) = \begin{cases} 0 & \text{if } \mathsf{glitch}(y) = 0 \text{ and } \bar{y} = 0 \\ \mathsf{glitch}(x) & \text{if } \mathsf{glitch}(y) = 0 \text{ and } \bar{y} = 1 \end{cases}$$
$$\mathsf{glitch}(x \oplus y) = \mathsf{glitch}(x) \text{ if } \mathsf{glitch}(y) = 0$$

We further define $\Sigma\mathsf{power}$ as the sum of $\mathsf{power}(g)$ for all gates $g$ in a circuit.

It is not quite clear how to define $\mathsf{glitch}(x \wedge y)$ for two glitched signals $x$ and $y$ in general. Even for $\mathsf{glitch}(x \oplus y)$, we may take one of the following assumptions:

$$\mathsf{glitch}(x \oplus y) = \mathsf{glitch}(x) + \mathsf{glitch}(y) \quad (1)$$
$$\mathsf{glitch}(x \oplus y) = \max(\mathsf{glitch}(x), \mathsf{glitch}(y)) \quad (2)$$
$$\mathsf{glitch}(x \oplus y) = \mathsf{glitch}(x) \oplus \mathsf{glitch}(y) \quad (3)$$

These assumptions are quite reasonable in theory. (1) accounts for glitches which cumulate, for instance because they occur at different time in a clock period. (2) assumes that a glitch can be hidden by another one. (3) comes from saying that two perfectly identical glitches should cancel each other in a XOR. However, reality is more complex and probably a mixture of these three models:

$$\mathsf{glitch}(x \oplus y) = F(\mathsf{glitch}(x), \mathsf{glitch}(y))$$

for some symmetric function $F$. For simplicity, we will study these simple assumptions. We will see that nearly all assumptions give the same results. Each defines some kind of "glitch algebra" on which we can do computations.

In this report, we consider two types of side-channel attacks based on glitches.

- Power analysis: the adversary can see $\Sigma\mathsf{power}$ with noise.
- Probing attack: for a gate $g$, the adversary can get $\mathsf{glitch}(g)$ with some noise.

Duc *et al.* have shown that these two attacks are equivalent [4].

### 2.2 Side-Channel Attack with Noise

In side-channel attack, we measure a quantity $S$ in a discrete domain $\mathcal{D}$ but the measurement comes with noise so we obtain $Z = S + \mathsf{noise}$. We assume that $S$ follows a distribution $P_b^S$ depending on a secret bit $b$. We want to make a guess $X$ for $b$. An algorithm taking some random input and giving $X$ as output is a distinguisher. The Type I error is $\alpha = \Pr[X = 1|b = 0]$. The Type II error is $\beta = \Pr[X = 0|b = 1]$. The error probability is $P_e = \Pr[X \neq b] = \alpha \Pr[b = 0] + \beta \Pr[b = 1]$ so depends on the distribution of $b$. The advantage of the distinguisher is $\mathsf{Adv} = |\Pr[X = 1|b = 0] - \Pr[X = 1|b = 1]| = |\alpha + \beta - 1|$.

If $P_b^Z$ denotes the obtained distribution for $Z$. We know that the largest advantage using one single sample $Z$ is $\mathsf{Adv} = d(P_0^Z, P_1^Z)$ defined by the statistical distance between $P_0^Z$ and $P_1^Z$.

$$d(P_0^Z, P_1^Z) = \frac{1}{2} \sum_z |\Pr[Z = z|b = 0] - \Pr[Z = z|b = 1]|$$

**Theorem 1 (Precision amplification).** *Given an elementary distinguisher computing $X$ from $Z$, with Type I error probability $\alpha \leq \frac{1}{2}$ and Type II error probability $\beta \leq \frac{1}{2}$, for any $N$ we can construct a distinguisher such that from i.i.d. samples $Z_1, \ldots, Z_N$ we compute $X$ with error probability*

$$P_e' \leq e^{-N\left(\frac{1}{2} - \min(\alpha,\beta)\right)^2} \tag{4}$$

*Taking $N = 2\left(\frac{1}{2} - \min(\alpha, \beta)\right)^{-2}$, we obtain $P_e' \leq e^{-2} \approx 13\%$.*

*Proof.* We use the elementary distinguisher to compute the $X_1, \ldots, X_N$ corresponding to $Z_1, \ldots, Z_N$. Then, we compute $X = \mathsf{majority}(X_1, \ldots, X_N)$.

Using the Chernoff bound (Lemma 2 below), we obtain a new distinguisher with errors $\alpha_N$ and $\beta_N$ such that $\alpha_N \leq e^{-N\left(\frac{1}{2}-\alpha\right)^2}$ and $\beta_N \leq e^{-N\left(\frac{1}{2}-\beta\right)^2}$. So, the error probability $P_e' = \alpha_N \Pr[b = 0] + \beta_N \Pr[b = 1]$ obtained by taking the majority vote decreases exponentially fast with $N$. As $\min(\alpha, \beta) \leq \alpha, \beta \leq \frac{1}{2}$, we obtain the result. □

**Lemma 2 (Chernoff [3]).** *Let $X_1, X_2, \ldots, X_N$ be $N$ independent boolean variables such that that $E(X_i) = p$ for all $i$. We define $X = \mathsf{majority}(X_1, \ldots, X_N)$. For all $p < \frac{1}{2}$, we have*

$$\Pr[X \neq 0] \leq e^{-N\left(\frac{1}{2}-p\right)^2}$$

*For all $p > \frac{1}{2}$, we have*

$$\Pr[X \neq 1] \leq e^{-N\left(\frac{1}{2}-p\right)^2}$$

In what follows, we assume that the noise is Gaussian, centered, independent from $S$, and that the ratio of the standard deviation of the noise and of $S$ is a given value $\sigma$. So, $\mathsf{noise}$ has a variance of $\sigma^2 V(S)$. Hence,

$$\Pr[\mathsf{noise} \leq -x] = \frac{1}{2}\mathsf{erfc}\left(\frac{x}{\sqrt{2\sigma^2 V(S)}}\right)$$

*Threshold distinguisher.* We consider the distinguisher computing $X = 1_{Z\leq\tau}$. In the Gaussian noise model, the Type I error is

$$\alpha = \Pr[X = 1|b = 0] = \sum_s \Pr[S = s|b = 0]\Pr[\mathsf{noise} \leq \tau - s]$$

by symmetry of the noise distribution, the Type II error is

$$\beta = \Pr[X = 0|b = 1] = \sum_s \Pr[S = s|b = 1]\Pr[\mathsf{noise} \leq s - \tau]$$

By adjusting $\tau$ so that $\alpha = \beta = P_e$, we obtain that $N = 2\left(\frac{1}{2} - P_e\right)^{-2}$ is enough to reach $P'_e \approx 13\%$.

*Case study for $S = 1 - b$ and $b$ uniform.* If $S = 1 - b$ and $b$ is uniform, we have $V(S) = \frac{1}{4}$. We adjust $\tau = \frac{1}{2}$ and obtain $\alpha = \beta = \frac{1}{2}\mathsf{erfc}\left(\frac{0.5}{\sigma}\sqrt{2}\right)$. We obtain from Th. 1 that $N = 2\left(\frac{1}{2} - \frac{1}{2}\mathsf{erfc}\left(\frac{0.5}{\sigma}\sqrt{2}\right)\right)^{-2}$. For instance, with $\sigma = 1$, we have $P_e \approx 16\%$ and $N = 17$. With $\sigma = 2$, we have $P_e \approx 31\%$ and $N = 55$. We obtain that $N \sim_{\sigma\to+\infty} 4\pi\sigma^2$ using $\mathsf{erfc}(t) = 1 - \frac{2t}{\sqrt{\pi}} + o(t)$ for $t \to 0$. So, we see that $N = \mathcal{O}(\sigma^2)$ is enough to guess $b$ with error limited to a constant. This is a quite favorable attack as we can measure $b$ directly.

*Attack for $n = 1$.* As an example, given an AND gate $z = x\wedge y$ (with no threshold protection, or equivalently $n = 1$), assuming that $y$ is stable equal to some secret $\bar{y}$ and that $\mathsf{glitch}(x) = 1$, we have $\mathsf{glitch}(z) = \bar{y}$. So, an attack measuring $S = \mathsf{glitch}(z)$ deduces $\bar{y}$ trivially. We are in the case where $S = \mathsf{glitch}(z) = \bar{y}$ is binary and balanced. So, the above equation governs the complexity $N$ of recovering $\bar{y}$ using no threshold implementation and noise characterized by $\sigma$.

*Pushing to higher order measures.* We can wonder what happens if, for some reasons, $S$ does not leak but $S^2$ leaks. Then, we should look at $Z^2$ instead of $Z$. But $Z^2 = S^2 + \mathsf{noise}'$ with $\mathsf{noise}' = 2S\mathsf{noise} + \mathsf{noise}^2$. By neglecting the quadratic noise, we have $V(\mathsf{noise}') \approx 4V(S)V(\mathsf{noise}) = 4\sigma^2V(S)^2$. Assuming $V(S^2) \approx V(S)^2$, we can see that the effect of moving from $S$ to $S^2$ is only in doubling the value of $\sigma$. As we will see, a motivation of threshold cryptography is to prevent leaks at a lower order $S$ to make the adversary look at higher order. This actually penalizes a bit the adversary.

## 3 Implementation with $n = 2$

Trichina *et al.* [11] proposed an implementation of the AND gate to compute $z = x \wedge y$ by using $n = 2$: 1. (secret sharing for $x$) pick $a \in_U \mathbf{Z}_2$ and compute $\tilde{x} = a\oplus x$; 2. (secret sharing for $y$) pick $b \in_U \mathbf{Z}_2$ and compute $\tilde{y} = b\oplus y$; 3. (secret sharing for $z$) pick $c \in_U \mathbf{Z}_2$; 4. compute

$$\tilde{z} = (((c \oplus (a \wedge b)) \oplus (a \wedge \tilde{y})) \oplus (b \wedge \tilde{x})) \oplus (\tilde{x} \wedge \tilde{y})$$

by respecting the order of the parentheses; 5. the output $(\tilde{z}, c)$ shares $z = c \oplus \tilde{z}$.

In [7], Nikova, Rechberger, and Rijmen observe that if the input signal $x$ has a glitch and $y$ is a secret input, then by analyzing the power consumption of the above gate we can easily deduce $y$. Indeed, assuming that an AND or XOR gate uses an abnormal power scheme proportional to the number of "glitch" on their result, the number of gates using an abnormal power scheme depends on $y$. So, we assume that $\mathsf{glitch}(x) = 1$ and $\mathsf{glitch}(\{a, b, c, y\}) = 0$.

We have

$$\begin{aligned} \mathsf{glitch}(\tilde{x}) &= 1 & \mathsf{glitch}(\tilde{x} \wedge \tilde{y}) &= \bar{\tilde{y}} = \bar{b} \oplus \bar{y} \\ \mathsf{glitch}(\tilde{y}) &= 0 & \mathsf{glitch}((c \oplus (a \wedge b) \oplus (a \wedge \tilde{y})) \oplus (b \wedge \tilde{x})) &= \bar{b} \\ \mathsf{glitch}(b \wedge \tilde{x}) &= \bar{b} \end{aligned}$$

so

$$\mathsf{glitch}(z) = \begin{cases} \bar{b} + \bar{\tilde{y}} & \text{with Assumption (1)} \\ \bar{b} \vee \bar{\tilde{y}} & \text{with Assumption (2)} \\ \bar{y} & \text{with Assumption (3)} \end{cases}$$

$$\Sigma\mathsf{power} = (\bar{b} + \bar{\tilde{y}})p_{\mathsf{AND}} + \left\{ \begin{array}{ll} 2\bar{b} + \bar{\tilde{y}} & \text{with Assumption (1)} \\ \bar{b} + \bar{b} \vee \bar{\tilde{y}} & \text{with Assumption (2)} \\ \bar{b} + \bar{y} & \text{with Assumption (3)} \end{array} \right\} .p_{\mathsf{XOR}}$$

If $\bar{y} = 0$, we have $\bar{\tilde{y}} = \bar{b}$ so

$$\Sigma\mathsf{power} = 2\bar{b}p_{\mathsf{AND}} + \left\{ \begin{array}{ll} 3\bar{b} & \text{with Assumption (1)} \\ 2\bar{b} & \text{with Assumption (2)} \\ \bar{b} & \text{with Assumption (3)} \end{array} \right\} .p_{\mathsf{XOR}}$$

For $\bar{y} = 1$, this is $\Sigma\mathsf{power} = p_{\mathsf{AND}} + (1 + \bar{b}).p_{\mathsf{XOR}}$ for Assumptions (1,2,3). So,

$$E(\Sigma\mathsf{power}|\bar{y} = 1) - E(\Sigma\mathsf{power}|\bar{y} = 0) = \left\{ \begin{array}{ll} 0 & \text{with Assumption (1)} \\ \frac{1}{2} & \text{with Assumption (2)} \\ 1 & \text{with Assumption (3)} \end{array} \right\} .p_{\mathsf{XOR}}$$

It is explicitly said in [8, p. 297] that

> "The power consumption caused by the glitch is related to the number of gates that *see* the glitch. It is clear [...] that the energy consumption depends on the values of [$b$ and $\tilde{y}$]. Since the $\underline{\text{mean power consumption}}$ is different for $y = 0$ and $y = 1$, the power consumption leaks information on the value $y$."

The computation done in [8] to analyze the leakage was based on Assumption (1) as we can easily check from [8, Table 1]. So, we contradict this claim for Assumption (1): $E(\Sigma\mathsf{power})$ is independent from $\bar{y}$ in this case. However, it is true that $E(\Sigma\mathsf{power})$ leaks $\bar{y}$ for Assumption (2) and Assumption (3). For this implementation, the choice of the "glitch algebra" gives different conclusions.

Similarly, in attacks based on probing $z$, we can see that $E(\mathsf{glitch}(z)) = 1$ which is independent from $\bar{y}$ in Assumption (1). For Assumption (2), we have

$E(\mathsf{glitch}(z)) = \frac{1}{2}$ which is also independent from $\bar{y}$. For Assumption (3), we have $\mathsf{glitch}(z) = \bar{y}$. In the latter case, we can see that $E(\mathsf{glitch}(z))$ leaks $\bar{y}$ so noisy samples for $\mathsf{glitch}(z)$ leak $\bar{y}$ using the amplification technique of Eq. (4).

This made [7] propose a "threshold implementation" of an AND gate using $n = 3$ or $n = 4$ shares, the above example being an example using $n = 2$ shares. They prove that, contrarily to this example, probing a single gate in the computation leaks no information on any of the input $x$ and $y$, on average. They deduce that their implementations resist to the above attacks based on glitches. We will show the limitations of this result with effective attacks.

## 4 Implementation with $n = 3$

Assuming that $(x_1, x_2, x_3)$ shares $x$, $(y_1, y_2, y_3)$ shares $y$, and $(z_1, z_2, z_3)$ shares $z$, Nikova, Rechberger, and Rijmen [7] propose

$$\begin{aligned} z_1 &= (x_2 \wedge y_2) \oplus ((x_2 \wedge y_3) \oplus (x_3 \wedge y_2)) \\ z_2 &= (x_3 \wedge y_3) \oplus ((x_1 \wedge y_3) \oplus (x_3 \wedge y_1)) \\ z_3 &= (x_1 \wedge y_1) \oplus ((x_1 \wedge y_2) \oplus (x_2 \wedge y_1)) \end{aligned}$$

This construction satisfies the conditions from Nikova et al. [7]. We quote [7]:

> "**Theorem 3.** [...] the mean power consumption of a circuit implementing realization [above] is independent of [$\bar{x}$, $\bar{y}$], even in the presence of glitches or the delayed arrival of some inputs."

Although we do not contradict the independence of the mean with the input values, we show that a probing attack can leak $\bar{y}$ easily. We further show that a cascade of this construction also leaks with the mean of power consumption.

In the attacks, we will assume that none of the $y_i$ variables have a glitch, and that they are independent from the glitches in the $x_i$ variables.

With Assumption (1), we have

$$\begin{aligned} \mathsf{glitch}(x_i \wedge y_j) &= \mathsf{glitch}(x_i)\bar{y}_j \\ \mathsf{glitch}((x_i \wedge y_j) \oplus (x_j \wedge y_i)) &= \mathsf{glitch}(x_i)\bar{y}_j + \mathsf{glitch}(x_j)\bar{y}_i \\ \mathsf{glitch}(z_1) &= \mathsf{glitch}(x_2)(\bar{y}_2 + \bar{y}_3) + \mathsf{glitch}(x_3)\bar{y}_2 \\ \mathsf{glitch}(z_2) &= \mathsf{glitch}(x_3)(\bar{y}_3 + \bar{y}_1) + \mathsf{glitch}(x_1)\bar{y}_3 \\ \mathsf{glitch}(z_3) &= \mathsf{glitch}(x_1)(\bar{y}_1 + \bar{y}_2) + \mathsf{glitch}(x_2)\bar{y}_1 \end{aligned}$$

so

$$\Sigma\mathsf{power} = \sum_{i,j} \mathsf{glitch}(x_i)\bar{y}_j p_{\mathsf{AND}} + \sum_i \mathsf{glitch}(x_i)(2\bar{y}_1 + 2\bar{y}_2 + 2\bar{y}_3 - \bar{y}_i)p_{\mathsf{XOR}}$$

In the $\mathsf{glitch}$ value of each gate, we can see that at least one variable $\bar{y}_i$ is not present (indeed, the construction was made for that). Since the $\bar{y}_i$ are uniformly distributed conditioned to $\bar{y} = \bar{y}_1 \oplus \bar{y}_2 \oplus \bar{y}_3$, no matter the value of $\bar{y}$, the two present $\bar{y}_i$ variables are uniformly distributed. So, the distribution of any $\mathsf{glitch}$ value is independent from $\bar{y}$. Consequently, it is the case for their mean value. Since $\Sigma\mathsf{power}$ is a linear combination of these values, due to the linearity of the mean operator, this is also the case for $\Sigma\mathsf{power}$.

With Assumption (2), by writing $\max(\bar{y}_i, \bar{y}_j) = \bar{y}_i \vee \bar{y}_j$, we have

$$\begin{aligned}
\mathsf{glitch}(x_i \wedge y_j) &= \mathsf{glitch}(x_i)\bar{y}_j \\
\mathsf{glitch}((x_i \wedge y_j) \oplus (x_j \wedge y_i)) &= \max(\mathsf{glitch}(x_i)\bar{y}_j, \mathsf{glitch}(x_j)\bar{y}_i) \\
\mathsf{glitch}(z_1) &= \max(\mathsf{glitch}(x_2)(\bar{y}_2 \vee \bar{y}_3), \mathsf{glitch}(x_3)\bar{y}_2) \\
\mathsf{glitch}(z_2) &= \max(\mathsf{glitch}(x_3)(\bar{y}_3 \vee \bar{y}_1), \mathsf{glitch}(x_1)\bar{y}_3) \\
\mathsf{glitch}(z_3) &= \max(\mathsf{glitch}(x_1)(\bar{y}_1 \vee \bar{y}_2), \mathsf{glitch}(x_2)\bar{y}_1)
\end{aligned}$$

Like above, the mean value of any of these expression is independent from $\bar{y}$.

With Assumption (3), we have

$$\begin{aligned}
\mathsf{glitch}(x_i \wedge y_j) &= \mathsf{glitch}(x_i)\bar{y}_j \\
\mathsf{glitch}((x_i \wedge y_j) \oplus (x_j \wedge y_i)) &= \mathsf{glitch}(x_i)\bar{y}_j \oplus \mathsf{glitch}(x_j)\bar{y}_i \\
\mathsf{glitch}(z_1) &= \mathsf{glitch}(x_2)(\bar{y}_2 \oplus \bar{y}_3) \oplus \mathsf{glitch}(x_3)\bar{y}_2 \\
\mathsf{glitch}(z_2) &= \mathsf{glitch}(x_3)(\bar{y}_3 \oplus \bar{y}_1) \oplus \mathsf{glitch}(x_1)\bar{y}_3 \\
\mathsf{glitch}(z_3) &= \mathsf{glitch}(x_1)(\bar{y}_1 \oplus \bar{y}_2) \oplus \mathsf{glitch}(x_2)\bar{y}_1
\end{aligned}$$

so

$$\Sigma\mathsf{power} = \sum_{i,j} \mathsf{glitch}(x_i)\bar{y}_j p_{\mathsf{AND}} + \left( \sum_i \mathsf{glitch}(x_i)((\bar{y}_i \oplus \bar{y}_{i+1}) + \bar{y}_{i-1}) - \right.$$
$$\left. \sum_i \mathsf{glitch}(x_i)\mathsf{glitch}(x_{i+1})((\bar{y}_i \oplus \bar{y}_{i+1})\bar{y}_i) \right) p_{\mathsf{XOR}}$$

Like above, the mean value of any of these expression is independent from $\bar{y}$.

## 4.1 Power Analysis not Based on the Mean Value (All Assumptions)

We have already seen that no $\mathsf{glitch}$ value has a distribution which depends on $\bar{y}$. So let us focus on the distribution of $\Sigma\mathsf{power}$. With $\mathsf{glitch}(x_1) = 1$ and $\mathsf{glitch}(x_2) = \mathsf{glitch}(x_3) = 0$, we obtain with Assumption (1) that

$$\Sigma\mathsf{power} = (\bar{y}_1 + \bar{y}_2 + \bar{y}_3)(p_{\mathsf{AND}} + 2p_{\mathsf{XOR}}) - \bar{y}_1 p_{\mathsf{XOR}}$$

With Assumption (2), our previous computations simplify to

$$\mathsf{glitch}(x_i \wedge y_j) = \begin{cases} 0 & \text{if } i \neq 1 \\ \bar{y}_j & \text{if } i = 1 \end{cases}$$

$$\begin{aligned}
\mathsf{glitch}((x_2 \wedge y_3) \oplus (x_3 \wedge y_2)) &= 0 & \mathsf{glitch}(z_1) &= 0 \\
\mathsf{glitch}((x_3 \wedge y_1) \oplus (x_1 \wedge y_3)) &= \bar{y}_3 & \mathsf{glitch}(z_2) &= \bar{y}_3 \\
\mathsf{glitch}((x_1 \wedge y_2) \oplus (x_2 \wedge y_1)) &= \bar{y}_2 & \mathsf{glitch}(z_3) &= \bar{y}_1 \vee \bar{y}_2
\end{aligned}$$

so

$$\Sigma\mathsf{power} = (\bar{y}_1 + \bar{y}_2 + \bar{y}_3)p_{\mathsf{AND}} + ((\bar{y}_1 \vee \bar{y}_2) + \bar{y}_2 + 2\bar{y}_3)p_{\mathsf{XOR}}$$

With Assumption (3), we have the same results except $\mathsf{glitch}(z_3) = \bar{y}_1 \oplus \bar{y}_2$. So

$$\Sigma\mathsf{power} = (\bar{y}_1 + \bar{y}_2 + \bar{y}_3)p_{\mathsf{AND}} + ((\bar{y}_1 \oplus \bar{y}_2) + \bar{y}_2 + 2\bar{y}_3)p_{\mathsf{XOR}}$$

We count the number of gates with a glitched output following the two assumptions. We also indicate $\Sigma\mathsf{power}$ assuming that $p_{\mathsf{AND}} = 1$ and $p_{\mathsf{XOR}} = 4$.[1] The results are on Table 1.

**Table 1.** Distributions for a glitch in $x_1$ in the threshold implementation

| | | | | Assumption (1) | | | Assumption (2) | | | Assumption (3) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\bar{y}$ | $\bar{y}_1$ | $\bar{y}_2$ | $\bar{y}_3$ | #AND | #XOR | $\Sigma$power | #AND | #XOR | $\Sigma$power | #AND | #XOR | $\Sigma$power |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 2 | 4 | 18 | 2 | 4 | 18 | 2 | 4 | 18 |
| 0 | 1 | 0 | 1 | 2 | 3 | 14 | 2 | 3 | 14 | 2 | 3 | 14 |
| 0 | 1 | 1 | 0 | 2 | 3 | 14 | 2 | 2 | 10 | 2 | 1 | 6 |
| mean | | | | 1.5 | 2.5 | 11.5 | 1.5 | 2.25 | 10.5 | 1.5 | 2 | 9.5 |
| variance | | | | 0.75 | 2.25 | 46.75 | 0.75 | 2.1875 | 44.75 | 0.75 | 2.5 | 48.75 |
| 1 | 0 | 0 | 1 | 1 | 2 | 9 | 1 | 2 | 9 | 1 | 2 | 9 |
| 1 | 0 | 1 | 0 | 1 | 2 | 9 | 1 | 2 | 9 | 1 | 2 | 9 |
| 1 | 1 | 0 | 0 | 1 | 1 | 5 | 1 | 1 | 5 | 1 | 1 | 5 |
| 1 | 1 | 1 | 1 | 3 | 5 | 23 | 3 | 4 | 19 | 3 | 3 | 15 |
| mean | | | | 1.5 | 2.5 | 11.5 | 1.5 | 2.25 | 10.5 | 1.5 | 2 | 9.5 |
| variance | | | | 0.75 | 2.25 | 46.75 | 0.75 | 1.1875 | 26.75 | 0.75 | 0.5 | 4.75 |
| stat. dist. | | | | 1 | 1 | 1 | 1 | 0.5 | 1 | 1 | 0.5 | 1 |

Clearly, the distributions of $\Sigma\mathsf{power}|\bar{y} = 0$ and $\Sigma\mathsf{power}|\bar{y} = 1$ are very different. For instance, the parity of $\#\mathsf{AND}$ is always equal to $\bar{y}$. The supports of the distributions $\#\mathsf{XOR}|\bar{y} = 0$ and $\#\mathsf{XOR}|\bar{y} = 1$ are disjoint with Assumption (1). So, we can distinguish them with one sample with advantage 1. With Assumption (2) or Assumption (3), the statistical distance of the distributions $\#\mathsf{XOR}|\bar{y} = 0$ and $\#\mathsf{XOR}|\bar{y} = 1$ is $\frac{1}{2}$. So, a trivial statistic with a couple of samples would recover $\bar{y}$ assuming no noise.

We consider several types of distinguishers base on measuring $\#\mathsf{XOR}$. As the impact of the glitched XORs on $\Sigma\mathsf{power}$ is bigger, we can assume we measure it this way. We could also consider other side channel attacks which can separate the XORs from and ANDs.

[1] We took $p_{\mathsf{XOR}} = 4p_{\mathsf{AND}}$ as an example, which justifies by assuming that we use 4 NAND gates to make a XOR gate. But this must only be taken as an example. Note that an AND requires two NAND gates but the second one which is used as a NOT gate can often cancel with subsequent gates.

**Best distinguisher.** With Assumption (1) and $\#\mathsf{XOR}$, the best distinguisher returns 0 if $\#\mathsf{XOR} \in \{0,3,4\}$ and it returns 1 if $\#\mathsf{XOR} \in \{1,2,5\}$. A statistical distance of 1 means that we can guess $\bar{y}$ with an error probability 0. A statistical distance of 0.5 means that we can guess $\bar{y}$ with an error probability $\frac{1}{4}$.

In practice, measuring $\#\mathsf{XOR}$ may give a noisy value making it hard to implement this distinguisher. I.e., 0 and 1 may be too close to be distinguishable, as well as 2 and 3, and 4 and 5.

**Threshold distinguisher.** We consider the distinguisher giving $1_{\#\mathsf{XOR}+\mathsf{noise}\leq\tau}$, i.e. 1 if $\#\mathsf{XOR}$ (rather its noisy value from a side channel) is below a given threshold $\tau$. Assuming that $\mathsf{noise}$ follows an independent normal distribution with mean 0 (w.l.o.g. by adjusting $\tau$) and variance $\sigma^2 V(\#\mathsf{XOR})$, we have

$$\Pr[\mathsf{noise} \leq -x] = \frac{1}{2}\mathsf{erfc}\left(\frac{x}{\sqrt{2\sigma^2 V(\#\mathsf{XOR})}}\right)$$

so the Type I error in guessing $\bar{y}$ is

$$\alpha = \frac{1}{2}\sum_i \mathsf{erfc}\left(\frac{i-\tau}{\sqrt{2\sigma^2 V(\#\mathsf{XOR})}}\right)\Pr[\#\mathsf{XOR} = i|\bar{y}=0]$$

The Type II error is

$$\beta = 1 - \frac{1}{2}\sum_i \mathsf{erfc}\left(\frac{i-\tau}{\sqrt{2\sigma^2 V(\#\mathsf{XOR})}}\right)\Pr[\#\mathsf{XOR} = i|\bar{y}=1]$$

For Assumption (1), we have $V(\#\mathsf{XOR}) = \frac{9}{4}$ and

$$\begin{aligned}\alpha &= \frac{1}{2}\sum_{i=0}^{5} \mathsf{erfc}\left(\frac{i-\tau}{\frac{3\sigma}{2}\sqrt{2}}\right)\Pr[\#\mathsf{XOR} = i|\bar{y}=0]\\ &= \frac{1}{8}\left(\mathsf{erfc}\left(\frac{-\tau}{\frac{3\sigma}{2}\sqrt{2}}\right) + 2\mathsf{erfc}\left(\frac{3-\tau}{\frac{3\sigma}{2}\sqrt{2}}\right) + \mathsf{erfc}\left(\frac{4-\tau}{\frac{3\sigma}{2}\sqrt{2}}\right)\right)\end{aligned}$$

For $\tau = 2.5$, using $\mathsf{erfc}(-x) = 2 - \mathsf{erfc}(x)$, we obtain

$$\alpha = \frac{1}{4} + \frac{1}{8}\left(-\mathsf{erfc}\left(\frac{2.5}{\frac{3\sigma}{2}\sqrt{2}}\right) + 2\mathsf{erfc}\left(\frac{0.5}{\frac{3\sigma}{2}\sqrt{2}}\right) + \mathsf{erfc}\left(\frac{1.5}{\frac{3\sigma}{2}\sqrt{2}}\right)\right)$$

Similarly, we have $\beta = \alpha$. So, we have $P_e = \alpha = \beta$ and

$$P_e = \frac{1}{4} + \frac{1}{8}\left(-\mathsf{erfc}\left(\frac{2.5}{\frac{3\sigma}{2}\sqrt{2}}\right) + \mathsf{erfc}\left(\frac{1.5}{\frac{3\sigma}{2}\sqrt{2}}\right) + 2\mathsf{erfc}\left(\frac{0.5}{\frac{3\sigma}{2}\sqrt{2}}\right)\right) = \frac{1}{2} - \Omega(\sigma^{-3})$$

As $\sigma$ goes from 0 to infinity, $P_e$ grows from $\frac{1}{4}$ to $\frac{1}{2}$. For instance, for $\sigma = \frac{1}{2}$, we have $P_e \approx 38\%$. For $\sigma = 1$, we have $P_e \approx 46\%$. For $\sigma = 2$, we have $P_e \approx 49.35\%$.

So, even with a big noise, we can recover $\bar{y}$ with an interesting advantage with only one sample.

Of course, we can amplify this advantage by using several samples. Since we have $\alpha = \beta = P_e$, by using (4) we obtain a new error probability of $P'_e = e^{-2} \approx 13\%$ with $N = 2\left(\frac{1}{2} - P_e\right)^{-2} = \mathcal{O}(\sigma^6)$. We obtain the following table:

| $\sigma$: | 0.5 | 1 | 1.5 | 2 | 2.5 | 3 | 3.5 | 4 |
|---|---|---|---|---|---|---|---|---|
| $N$: | 143 | 1 417 | 9 979 | 46 765 | 163 627 | 465 879 | 1 141 284 | 2 495 478 |

So, measuring the number of XORs (a number between 0 and 5) with a big noise of standard deviation twice what we want to measure still allows to deduce $\bar{y}$ with less than 50 000 samples with Assumption (1).

**Moment distinguisher.** Instead of computing the average of $\Sigma\mathsf{power}$, we compute the moment $E((\Sigma\mathsf{power})^d)$ of order $d$, just like in Moradi [6,10].

With Assumption (1), we have

$$\begin{aligned}
E(\Sigma\mathsf{power}|y=0) = E(\Sigma\mathsf{power}|y=1) &= \frac{3}{2}p_{\mathsf{AND}} + \frac{5}{2}p_{\mathsf{XOR}} \\
V(\Sigma\mathsf{power}|y=0) = V(\Sigma\mathsf{power}|y=1) &= \frac{3}{4}p_{\mathsf{AND}}^2 + \frac{5}{2}p_{\mathsf{AND}}p_{\mathsf{XOR}} + \frac{9}{4}p_{\mathsf{XOR}}^2
\end{aligned}$$

but the moments of order $d = 3$ differ. So $(\Sigma\mathsf{power})^3$ leaks $\bar{y}$.

With Assumption (2), we have

$$\begin{aligned}
E(\Sigma\mathsf{power}|y=0) = E(\Sigma\mathsf{power}|y=1) &= \frac{3}{2}p_{\mathsf{AND}} + \frac{9}{4}p_{\mathsf{XOR}} \\
V(\Sigma\mathsf{power}|y=0) &= \frac{3}{4}p_{\mathsf{AND}}^2 + \frac{9}{4}p_{\mathsf{AND}}p_{\mathsf{XOR}} + \frac{35}{16}p_{\mathsf{XOR}}^2 \\
V(\Sigma\mathsf{power}|y=1) &= \frac{3}{4}p_{\mathsf{AND}}^2 + \frac{7}{4}p_{\mathsf{AND}}p_{\mathsf{XOR}} + \frac{19}{16}p_{\mathsf{XOR}}^2 \\
V(\Sigma\mathsf{power}|y=0) - V(\Sigma\mathsf{power}|y=1) &= \frac{1}{2}p_{\mathsf{AND}}p_{\mathsf{XOR}} + p_{\mathsf{XOR}}^2
\end{aligned}$$

With Assumption (3), we have

$$\begin{aligned}
V(\Sigma\mathsf{power}|y=0) &= \frac{3}{4}p_{\mathsf{AND}}^2 + 2p_{\mathsf{AND}}p_{\mathsf{XOR}} + \frac{5}{2}p_{\mathsf{XOR}}^2 \\
V(\Sigma\mathsf{power}|y=1) &= \frac{3}{4}p_{\mathsf{AND}}^2 + \frac{1}{2}p_{\mathsf{AND}}p_{\mathsf{XOR}} + \frac{1}{2}p_{\mathsf{XOR}}^2 \\
V(\Sigma\mathsf{power}|y=0) - V(\Sigma\mathsf{power}|y=1) &= \frac{3}{2}p_{\mathsf{AND}}p_{\mathsf{XOR}} + 2p_{\mathsf{XOR}}^2
\end{aligned}$$

Clearly, the mean of $(\Sigma\mathsf{power})^2$ (i.e., $d = 2$) leaks $\bar{y}$ with Assumption (2) and Assumption (3). For Assumption (1), the same holds with $(\Sigma\mathsf{power})^3$.

If $E(\Sigma\mathsf{power})$ is known and we measure $Z = S + \mathsf{noise}$ with $S = \Sigma\mathsf{power} - E(\Sigma\mathsf{power})$ and a centered Gaussian noise of variance $\sigma^2 V(\Sigma\mathsf{power})$, we can compute $Z' = S^2 + \mathsf{noise}'$ with $\mathsf{noise}' = 2S\mathsf{noise} + \mathsf{noise}^2 - \sigma^2 V(\Sigma\mathsf{power})$. So, it is as if we measured $S^2$ with noise $\mathsf{noise}'$. The variance of $\mathsf{noise}'$ is roughly $4\sigma^2 V(S^2)$, so the attack works as if we just doubled $\sigma$. For instance, our previous computation shows that by doubling $\sigma$ we roughly multiply $N$ by 50. With this approach, threshold implementation penalizes the precision of the measurement.

### 4.2 Probing Attack with Two Probes Based on the Mean Value (All Assumptions)

We can further see what probing can yield.

With Assumption (1,3), we have $\mathsf{glitch}(z_2) = \bar{y}_3$ and $\mathsf{glitch}(z_3) = \bar{y}_1 \oplus \bar{y}_2$. So, probing both $z_2$ and $z_3$ is enough to recover $\bar{y}$.

With Assumption (2), this leaks $(\bar{y}_3, \bar{y}_1 \vee \bar{y}_2)$. For $\bar{y} = 0$, the distribution of this couple is $\Pr[(0,0)] = \frac{1}{4}$, $\Pr[(0,1)] = \frac{1}{4}$, $\Pr[(1,1)] = \frac{1}{2}$. For $\bar{y} = 1$, the distribution of this couple is $\Pr[(0,1)] = \frac{1}{2}$, $\Pr[(1,0)] = \frac{1}{4}$, $\Pr[(1,1)] = \frac{1}{4}$. So, the statistical distance is $\frac{1}{2}$ and the probability of error for guessing $\bar{y}$ is $\frac{1}{4}$.

As an example, with Assumption (1) we compute $S = \mathsf{glitch}(z_2) + \mathsf{glitch}(z_3) - \mathsf{glitch}(z_2)\mathsf{glitch}(z_3)$. Assuming that both $\mathsf{glitch}(z_2)$ and $\mathsf{glitch}(z_3)$ are subject to some noise with same parameter $\sigma$, the value we obtain for $S$ is similar to a noisy value with the parameter $\sigma$ multiplied by a constant factor less than 3. In our table, this results in a complexity $N$ multiplied by a factor 300.

We recall that [7] claims no security when probing two values.

### 4.3 Power Analysis and Probing Attack on Two ANDs Based on the Mean Value (Assumption (2) or (3))

We use two consecutive threshold AND gates to compute the AND between $z$ and another shared bit $u$ to obtain $v = x \wedge y \wedge u$. We assume no glitch on the $y$ and $u$ variables. We assume that only $x_1$ has a glitch. We have

$$\begin{aligned}
z_1 &= (x_2 \wedge y_2) \oplus ((x_2 \wedge y_3) \oplus (x_3 \wedge y_2)) & v_1 &= (z_2 \wedge u_2) \oplus ((z_2 \wedge u_3) \oplus (z_3 \wedge u_2)) \\
z_2 &= (x_3 \wedge y_3) \oplus ((x_1 \wedge y_3) \oplus (x_3 \wedge y_1)) & v_2 &= (z_3 \wedge u_3) \oplus ((z_1 \wedge u_3) \oplus (z_3 \wedge u_1)) \\
z_3 &= (x_1 \wedge y_1) \oplus ((x_1 \wedge y_2) \oplus (x_2 \wedge y_1)) & v_3 &= (z_1 \wedge u_1) \oplus ((z_1 \wedge u_2) \oplus (z_2 \wedge u_1))
\end{aligned}$$

With Assumption (1), the linearity of the equations make sure that the expected value of the glitch variables are independent from $\bar{y}$.

Now, under Assumption (2), we have

$$\begin{aligned}
\mathsf{glitch}(z_1) &= 0 & \mathsf{glitch}(v_1) &= \max(\mathsf{glitch}(z_2)(\bar{u}_2 \vee \bar{u}_3), \mathsf{glitch}(z_3)\bar{u}_2) \\
\mathsf{glitch}(z_2) &= \bar{y}_3 & &= \max(\bar{y}_3(\bar{u}_2 \vee \bar{u}_3), (\bar{y}_1 \vee \bar{y}_2)\bar{u}_2) \\
\mathsf{glitch}(z_3) &= \bar{y}_1 \vee \bar{y}_2 & &
\end{aligned}$$

so we can now try to probe $v_1$. We have the 3 following cases:

- $\bar{u}_2 = \bar{u}_3 = 0$ (probability $\frac{1}{4}$): we have $v_1 = 0$, no glitch and nothing leaks.
- $\bar{u}_2 = 0$ and $\bar{u}_3 = 1$ (probability $\frac{1}{4}$): $v_1$ has a glitch if and only if $\bar{y}_3 = 1$.
- $\bar{u}_2 = 1$ (probability $\frac{1}{2}$): $v_1$ has a glitch when $\bar{y}_1 \vee \bar{y}_2 \vee \bar{y}_3 = 1$. For $\bar{y} = 1$, there is always a glitch. For $\bar{y} = 0$, there is a glitch with probability $\frac{3}{4}$.

So, if $\bar{y} = 0$, we observe a glitch in $v_1$ with probability $\frac{1}{4} \times 0 + \frac{1}{4} \times \frac{1}{2} + \frac{1}{2} \times \frac{3}{4} = \frac{1}{2}$. If $\bar{y} = 1$, the probability becomes $\frac{1}{4} \times 0 + \frac{1}{4} \times \frac{1}{2} + \frac{1}{2} \times 1 = \frac{5}{8}$. Hence, the mean value reveals $\bar{y}$. A single sample gives an error probability of $\frac{7}{17}$.

Now, under Assumption (3), we have

$$\begin{array}{ll} \mathsf{glitch}(z_1) = 0 & \mathsf{glitch}(z_3) = \bar{y}_1 \oplus \bar{y}_2 \\ \mathsf{glitch}(z_2) = \bar{y}_3 & \mathsf{glitch}(v_1) = \bar{y}_3(\bar{u}_2 \oplus \bar{u}_3) \oplus (\bar{y}_1 \oplus \bar{y}_2)\bar{u}_2 \end{array}$$

so we can try to probe $v_1$ again. With probability $\frac{1}{4}$, we have $\bar{u}_2 \oplus \bar{u}_3 = \bar{u}_2 = 1$ so $\mathsf{glitch}(v_1) = \bar{y}$. Otherwise, $\mathsf{glitch}(v_1)$ is uniformly distributed. So, for $\bar{y} = 0$, $E(\mathsf{glitch}(v_1)) = \frac{3}{8}$ and for $\bar{y} = 1$, $E(\mathsf{glitch}(v_1)) = \frac{5}{8}$. Again, $\bar{y}$ leaks from the mean value. A single sample gives an error probability of $\frac{3}{8}$.

The attack with noisy values is hardly more complicated than for $n = 1$.

Note that [7] does not claim any security on the composition of two AND gates. But this attacks clearly shows the limitation of this approach.

## 5 Implementation with $n = 4$

Assuming that $(x_1, x_2, x_3, x_4)$ shares $x$, $(y_1, y_2, y_3, y_4)$ shares $y$, and $(z_1, z_2, z_3, z_4)$ shares $z$, Nikova, Rechberger, and Rijmen [7] propose

$$\begin{aligned} z_1 &= ((x_3 \oplus x_4) \wedge (y_2 \oplus y_3)) \oplus y_2 \oplus y_3 \oplus y_4 \oplus x_2 \oplus x_3 \oplus x_4 \\ z_2 &= ((x_1 \oplus x_3) \wedge (y_1 \oplus y_4)) \oplus y_1 \oplus y_3 \oplus y_4 \oplus x_1 \oplus x_3 \oplus x_4 \\ z_3 &= ((x_2 \oplus x_4) \wedge (y_1 \oplus y_4)) \oplus y_2 \oplus x_2 \\ z_4 &= ((x_1 \oplus x_2) \wedge (y_2 \oplus y_3)) \oplus y_1 \oplus x_1 \end{aligned}$$

It was proposed as an improvement to the $n = 3$ scheme as it makes all $z_i$ shares balanced. This property is called *uniformity* in [8]. It was used to address composition. So, we look again at the composition of two AND circuits.

Again, we assume $\mathsf{glitch}(x_1) = 1$, $\mathsf{glitch}(x_2) = \mathsf{glitch}(x_3) = \mathsf{glitch}(x_4) = \mathsf{glitch}(y_i) = 0$ for $i = 1, \ldots, 4$ and $\mathsf{glitch}(x_1) = 1$. So, $\mathsf{glitch}(z_1) = 0$, $\mathsf{glitch}(z_2) = (\bar{y}_1 \oplus \bar{y}_4) + 1$, $\mathsf{glitch}(z_3) = 0$, and $\mathsf{glitch}(z_4) = (\bar{y}_2 \oplus \bar{y}_3) + 1$ with Assumption (1).

We compute $v = z \wedge u = (x \wedge y) \wedge u$ using the threshold implementation with

$$\begin{aligned} v_1 &= ((z_3 \oplus z_4) \wedge (u_2 \oplus u_3)) \oplus u_2 \oplus u_3 \oplus u_4 \oplus z_2 \oplus z_3 \oplus z_4 \\ v_2 &= ((z_1 \oplus z_3) \wedge (u_1 \oplus u_4)) \oplus u_1 \oplus u_3 \oplus u_4 \oplus z_1 \oplus z_3 \oplus z_4 \\ v_3 &= ((z_2 \oplus z_4) \wedge (u_1 \oplus u_4)) \oplus u_2 \oplus z_2 \\ v_4 &= ((z_1 \oplus z_2) \wedge (u_2 \oplus u_3)) \oplus u_1 \oplus z_1 \end{aligned}$$

So, we have

$$\begin{aligned} \mathsf{glitch}(v_1) &= (\bar{y}_2 \oplus \bar{y}_3 + 1)(\bar{u}_2 \oplus \bar{u}_3) + (\bar{y}_1 \oplus \bar{y}_4) + (\bar{y}_2 \oplus \bar{y}_3) + 2 \\ \mathsf{glitch}(v_2) &= \bar{y}_2 \oplus \bar{y}_3 + 1 \\ \mathsf{glitch}(v_3) &= ((\bar{y}_1 \oplus \bar{y}_4) + (\bar{y}_2 \oplus \bar{y}_3) + 2)(\bar{u}_1 \oplus \bar{u}_4) + \bar{y}_1 \oplus \bar{y}_4 + 1 \\ \mathsf{glitch}(v_4) &= ((\bar{y}_1 \oplus \bar{y}_4) + 1)(\bar{u}_2 \oplus \bar{u}_3) \end{aligned}$$

Hence, we can just probe $v_1$ and see if it has a glitch. With probability $\frac{1}{2}$, we have $\bar{u}_2 = \bar{u}_3$ so $\mathsf{glitch}(v_1) = 2(\bar{y}_2 \oplus \bar{y}_3) + (\bar{y}_1 \oplus \bar{y}_4) + 3$. In other cases, we have $\mathsf{glitch}(v_1) = (\bar{y}_1 \oplus \bar{y}_4) + (\bar{y}_2 \oplus \bar{y}_3) + 2$ which is uniformly distributed. So, by repeating enough times, the majority of $\mathsf{glitch}(v_1)$ is $\bar{y}$ with high probability.

The attack with noisy values is hardly more complicated than for $n = 1$.

Computations with Assumptions (2) or (3) are similar.

Note that [7] does not claim any security on the composition of two AND gates. However, the $n = 4$ implementation was made to produce a balanced sharing of the output to address composability through pipelining, meaning by adding a layer of registers between the circuits we want to compose. Here, we consider the composition of two AND gates without pipelining. Indeed, we certainly do not want to add registers in between two single gates! But our attacks shows that the entire layer of circuit that we want to compose through pipelining must be analyzed as a whole, since single gates clearly do not compose well.

## 6 Higher-Order Threshold Implementation with $n = 5$

In [1], Bilgin *et al.* propose an example of higher-order threshold implementation. Equation (1) in [1] implements $\bar{y} = 1 \oplus \bar{a} \oplus \bar{b}\bar{c}$. To obtain the implementation of an AND gate, we just remove the 1 and the $a$ terms and obtain

$$\begin{array}{ll}
y_1 = (b_2 \wedge c_2) \oplus (b_1 \wedge c_2) \oplus (b_2 \wedge c_1) & y_6 = (b_2 \wedge c_4) \oplus (b_4 \wedge c_2) \\
y_2 = (b_3 \wedge c_3) \oplus (b_1 \wedge c_3) \oplus (b_3 \wedge c_1) & y_7 = (b_5 \wedge c_5) \oplus (b_2 \wedge c_5) \oplus (b_5 \wedge c_2) \\
y_3 = (b_4 \wedge c_4) \oplus (b_1 \wedge c_4) \oplus (b_4 \wedge c_1) & y_8 = (b_3 \wedge c_4) \oplus (b_4 \wedge c_3) \\
y_4 = (b_1 \wedge c_1) \oplus (b_1 \wedge c_5) \oplus (b_5 \wedge c_1) & y_9 = (b_3 \wedge c_5) \oplus (b_5 \wedge c_3) \\
y_5 = (b_2 \wedge c_3) \oplus (b_3 \wedge c_2) & y_{10} = (b_4 \wedge c_5) \oplus (b_5 \wedge c_4)
\end{array}$$

Then, Equation (2) in [1] decreases the number of shares to 5 by

$$\begin{array}{ll}
z_1 = (b_2 \wedge c_2) \oplus (b_1 \wedge c_2) \oplus (b_2 \wedge c_1) & z_5 = (b_2 \wedge c_3) \oplus (b_3 \wedge c_2) \oplus (b_2 \wedge c_4) \oplus \\
z_2 = (b_3 \wedge c_3) \oplus (b_1 \wedge c_3) \oplus (b_3 \wedge c_1) & \quad = (b_4 \wedge c_2) \oplus (b_5 \wedge c_5) \oplus (b_2 \wedge c_5) \oplus \\
z_3 = (b_4 \wedge c_4) \oplus (b_1 \wedge c_4) \oplus (b_4 \wedge c_1) & \quad = (b_5 \wedge c_2) \oplus (b_3 \wedge c_4) \oplus (b_4 \wedge c_3) \oplus \\
z_4 = (b_1 \wedge c_1) \oplus (b_1 \wedge c_5) \oplus (b_5 \wedge c_1) & \qquad (b_3 \wedge c_5) \oplus (b_5 \wedge c_3) \oplus (b_4 \wedge c_5) \oplus \\
 & \qquad (b_5 \wedge c_4)
\end{array}$$

This 2nd order implementation is supposed to resist to probing attacks with two probes. Normally, the transform of $(y_1, \ldots, y_{10})$ to $(z_1, \ldots, z_5)$ by $z_i = y_i$ for $i < 5$ and $z_5 = y_5 \oplus \cdots \oplus y_{10}$ must be done with intermediate registers to avoid the propagation of glitches. We wonder what happens without these registers.

Let consider an attack probing $z_4$ and $z_5$. If there is a glitch in $b_5$ and no other input share, we have $\mathsf{glitch}(z_4) = \bar{c}_1$ and

$$\mathsf{glitch}(z_5) = \mathsf{glitch}((b_5 \wedge c_2) \oplus (b_5 \wedge c_3) \oplus (b_5 \wedge c_4) \oplus (b_5 \wedge c_5))$$

With Assumption (1), this is $\mathsf{glitch}(z_5) = \bar{c}_2 + \bar{c}_3 + \bar{c}_4 + \bar{c}_5$. With Assumption (2), this is $\mathsf{glitch}(z_5) = \max(\bar{c}_2, \bar{c}_3, \bar{c}_4, \bar{c}_5)$. With Assumption (3), this is $\mathsf{glitch}(z_5) =$

**Table 2.** Distribution of $(\mathsf{glitch}(z_4), \mathsf{glitch}(z_5))$ for a glitch in $b_5$ in the 2nd order threshold implementation

| $\bar{c}$ | $\bar{c}_1\bar{c}_2\bar{c}_3\bar{c}_4\bar{c}_5$ | A. (1) | A. (2) | A. (3) |
|---|---|---|---|---|
| 0 | 0 0 0 0 0 | $(0,0)$ | $(0,0)$ | $(0,0)$ |
| 0 | 0 0 0 1 1 | $(0,2)$ | $(0,1)$ | $(0,0)$ |
| 0 | 0 0 1 0 1 | $(0,2)$ | $(0,1)$ | $(0,0)$ |
| 0 | 0 1 0 0 1 | $(0,2)$ | $(0,1)$ | $(0,0)$ |
| 0 | 0 0 1 1 0 | $(0,2)$ | $(0,1)$ | $(0,0)$ |
| 0 | 0 1 0 1 0 | $(0,2)$ | $(0,1)$ | $(0,0)$ |
| 0 | 0 1 1 0 0 | $(0,2)$ | $(0,1)$ | $(0,0)$ |
| 0 | 0 1 1 1 1 | $(0,4)$ | $(0,1)$ | $(0,0)$ |
| 0 | 1 0 0 0 1 | $(1,1)$ | $(1,1)$ | $(1,1)$ |
| 0 | 1 0 0 1 0 | $(1,1)$ | $(1,1)$ | $(1,1)$ |
| 0 | 1 0 1 0 0 | $(1,1)$ | $(1,1)$ | $(1,1)$ |
| 0 | 1 1 0 0 0 | $(1,1)$ | $(1,1)$ | $(1,1)$ |
| 0 | 1 0 1 1 1 | $(1,3)$ | $(1,1)$ | $(1,1)$ |
| 0 | 1 1 0 1 1 | $(1,3)$ | $(1,1)$ | $(1,1)$ |
| 0 | 1 1 1 0 1 | $(1,3)$ | $(1,1)$ | $(1,1)$ |
| 0 | 1 1 1 1 0 | $(1,3)$ | $(1,1)$ | $(1,1)$ |
| mean | | $(\frac{1}{2},2)$ | $(\frac{1}{2},\frac{15}{16})$ | $(\frac{1}{2},\frac{1}{2})$ |
| variance | | $(\frac{1}{4},1)$ | $(\frac{1}{2},\frac{15}{256})$ | $(\frac{1}{2},\frac{1}{4})$ |

| $\bar{c}$ | $\bar{c}_1\bar{c}_2\bar{c}_3\bar{c}_4\bar{c}_5$ | A. (1) | A. (2) | A. (3) |
|---|---|---|---|---|
| 1 | 0 0 0 0 1 | $(0,1)$ | $(0,1)$ | $(0,1)$ |
| 1 | 0 0 0 1 0 | $(0,1)$ | $(0,1)$ | $(0,1)$ |
| 1 | 0 0 1 0 0 | $(0,1)$ | $(0,1)$ | $(0,1)$ |
| 1 | 0 1 0 0 0 | $(0,1)$ | $(0,1)$ | $(0,1)$ |
| 1 | 0 0 1 1 1 | $(0,3)$ | $(0,1)$ | $(0,1)$ |
| 1 | 0 1 0 1 1 | $(0,3)$ | $(0,1)$ | $(0,1)$ |
| 1 | 0 1 1 0 1 | $(0,3)$ | $(0,1)$ | $(0,1)$ |
| 1 | 0 1 1 1 0 | $(0,3)$ | $(0,1)$ | $(0,1)$ |
| 1 | 1 0 0 0 0 | $(1,0)$ | $(1,0)$ | $(1,0)$ |
| 1 | 1 0 0 1 1 | $(1,2)$ | $(1,1)$ | $(1,0)$ |
| 1 | 1 0 1 0 1 | $(1,2)$ | $(1,1)$ | $(1,0)$ |
| 1 | 1 1 0 0 1 | $(1,2)$ | $(1,1)$ | $(1,0)$ |
| 1 | 1 0 1 1 0 | $(1,2)$ | $(1,1)$ | $(1,0)$ |
| 1 | 1 1 0 1 0 | $(1,2)$ | $(1,1)$ | $(1,0)$ |
| 1 | 1 1 1 0 0 | $(1,2)$ | $(1,1)$ | $(1,0)$ |
| 1 | 1 1 1 1 1 | $(1,4)$ | $(1,1)$ | $(1,0)$ |
| mean | | $(\frac{1}{2},2)$ | $(\frac{1}{2},\frac{15}{16})$ | $(\frac{1}{2},\frac{1}{2})$ |
| variance | | $(\frac{1}{4},1)$ | $(\frac{1}{2},\frac{15}{256})$ | $(\frac{1}{2},\frac{1}{4})$ |

$\bar{c}_2 \oplus \bar{c}_3 \oplus \bar{c}_4 \oplus \bar{c}_5$. So, we obtain the distributions for $(\mathsf{glitch}(z_4), \mathsf{glitch}(z_5))$ which is on Table 2. As we can see, the mean and the variance do not leak (as intended). However, the distributions are quite far apart.

Indeed, for Assumption (3), we have $\bar{c} = \mathsf{glitch}(z_4) \oplus \mathsf{glitch}(z_5)$ so it is clear that $\bar{c}$ leaks. For Assumption (1), we have $\bar{c} = \mathsf{glitch}(z_4) \oplus (\mathsf{glitch}(z_5) \bmod 2)$ so it is clear that $\bar{c}$ leaks as well. For Assumption (2), the distributions are

| distribution | $(0,0)$ | $(0,1)$ | $(1,0)$ | $(1,1)$ |
|---|---|---|---|---|
| $(\mathsf{glitch}(z_4), \mathsf{glitch}(z_5)) \vert \bar{c} = 0$ | 1/16 | 7/16 | 0/16 | 8/16 |
| $(\mathsf{glitch}(z_4), \mathsf{glitch}(z_5)) \vert \bar{c} = 1$ | 0/16 | 8/16 | 1/16 | 7/16 |

so the statistical distance is $\frac{1}{8}$. This means that from a single value we can deduce $\bar{c}$ with an error probability of $P_e = \frac{1}{2} - \frac{1}{16}$. Of course, this amplifies like in (4) using more samples. Hence, two probes leak quite a lot. So, we clearly see that avoiding the extra registers needed to avoid the number of shares to inflate makes the implementation from [1] insecure.

## 7 Conclusion

We have shown that the threshold implementations are quite weak against many simple attacks: distinguishers based on non-linear functions on the power traces (as simple as a threshold function or a power function), multiple probes, and linear distinguishers for a cascade of circuits. Although they do not contradict the results by their authors, these attacks show severe limitations on this approach.

We have seen that compared to the attack on the AND gate with no protection, the threshold implementation proposals only have the effect to amplify the noise of the side-channel attack by a constant factor. Therefore, we believe that there is no satisfactory protection for attacks based on glitches.

## References

1. B. Bilgin, B. Gierlichs, S. Nikova, V. Nikov, V. Rijmen. Higher-Order Threshold Implementations. In *Advances in Cryptology ASIACRYPT'14*, Kaohsiung, Taiwan, Lecture Notes in Computer Science 8873–8874, pp. 326–343 vol. 2, Springer-Verlag, 2014.
2. S. Chari, C.S. Jutla, J.R. Rao, P Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In *Advances in Cryptology CRYPTO'99*, Santa Barbara, California, U.S.A., Lecture Notes in Computer Science 1666, pp. 398–412, Springer-Verlag, 1999.
3. H. Chernoff. A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the sum of Observations. *Annals of Mathematical Statistics*, vol. 23 (4), pp. 493-507, 1952.
4. A. Duc, S. Dziembowski, S. Faust. Unifying Leakage Models: From Probing Attacks to Noisy Leakage. In *Advances in Cryptology EUROCRYPT'14*, Copenhaguen, Denmark, Lecture Notes in Computer Science 8441, pp. 423–440, Springer-Verlag, 2014.
5. S. Mangard, T. Popp, B.M. Gammel. Side-Channel Leakage of Masked CMOS Gates. In *Topics in Cryptology CT-RSA'05*, San Francisco CA, USA, Lecture Notes in Computer Science 3376, pp. 351–365, Springer-Verlag, 2005.
6. A. Moradi. Statistical Tools Flavor Side-Channel Collision Attacks. In *Advances in Cryptology EUROCRYPT'12*, Cambridge, UK, Lecture Notes in Computer Science 7237, pp. 428–445, Springer-Verlag, 2012.
7. S. Nikova, C. Rechberger, V. Rijmen. Threshold Implementations Against Side-Channel Attacks and Glitches. In *Information and Communication Security ICICS'06*, Raleigh NC, USA, Lecture Notes in Computer Science 4307, pp. 529–545, Springer-Verlag, 2006.
8. S. Nikova, V. Rijmen, M. Schläffer. Secure Hardware Implementation of Nonlinear Functions in the Presence of Glitches. *Journal of Cryptology*, vol. 24, pp. 292–321, 2011.
9. Oscar Reparaz and Begl Bilgin and Svetla Nikova and Benedikt Gierlichs and Ingrid Verbauwhede. Consolidating Masking Schemes. In *Advances in Cryptology CRYPTO'15*, Santa Barbara, California, U.S.A., Lecture Notes in Computer Science 9215–9216, pp. 764–783, Springer-Verlag, 2015.
10. F.-X. Standaert, N. Veyrat-Charvillon, E. Oswald, B. Gierlichs, M. Medwed, M. Kasper, S. Mangard The World Is Not Enough: Another Look on Second-Order DPA. In *Advances in Cryptology ASIACRYPT'10*, Singapore, Lecture Notes in Computer Science 6477, pp. 112–129, Springer-Verlag, 2010.
11. E. Trichina, T. Korkishko, K.H. Lee. Small Size, Low Power, Side Channel-Immune AES Coprocessor: Design and Synthesis Results. In *Advanced Encryption Standard AES'04*, Bonn, Germany, Lecture Notes in Computer Science 3373, pp. 113–127, Springer-Verlag, 2005.