A Note on 'Further Improving Efficiency of Higher-Order Masking Schemes by Decreasing Randomness Complexity'

"Almost" is Like "Not At All"

Gilles Barthe[1], François Dupressoir[2], and Benjamin Grégoire[3]

[1]IMDEA Software Institute
[2]University of Surrey (fdupress@gmail.com)
[3]Inria Méditerranée

Abstract. Zhang, Qiu and Zhou [5] propose two optimised masked algorithms for computing functions of the form $x \mapsto x \cdot \ell(x)$ for any linear function ℓ. They claim security properties. We disprove their first claim by exhibiting a first order flaw that is present in their first proposed algorithm scheme at all orders. We put their second claim into question by showing that their proposed algorithm, as published, is not well-defined at all orders, making use of variables before defining them. We then also exhibit a counterexample at order 2, that we believe generalises to all even orders.

Coron, Prouff, Rivain and Roche [3] (CPRR) introduced specialised masked algorithms for computing a particular functionality of interest that had so far been problematic from the point of view of security: functions of the form $x \mapsto x \cdot \ell(x)$ for some lineary function ℓ. Their proposed algorithm–if it improves time complexity–does not, however, improve on the randomness complexity of the simpler option of simply composing a gadget for ℓ and a multiplication gadget, taking care to insert the necessary mask refreshing operation.

Zhang, Qiu and Zhou [5] (ZQZ) recently proposed two masked algorithms for computing these functionalities that have reduced randomness complexity. They claim that their algorithms enjoy particular *probing security* properties. In this short note, we give counterexamples to these claims.

We first recall some definitions before detailing the ZQZ proposals and claims, and detailing our own claims and counterexamples.

1. Definitions

We limit our definitions to the relevant case here: that of gadgets with a single input and a single output.

Definition 1 (t-Non-Interference). *A gadget is* t-Non-Interfering *(or* t-NI*) whenever the joint distribution of any* $d \leq t$ *of its intermediate variables can be perfectly simulated using at most* t *shares of its input.*

Definition 2 (t-Strong-Non-Interference). *A gadget is* t-Strongly-Non-Interfering *(or* t-SNI*) whenever the joint distribution of any* t_1 *of its intermediate variables and* t_2 *of its output shares (with* $t_1 + t_2 \leq t$*) can be perfectly simulated using at most* t_1 *shares of its input.*

We use the "simulation" terminology of Ishai, Sahai and Wagner [4] and others, but note that the simulations here are not meant in the usual cryptographic sense. When we write that a distribution $d(a_0, \ldots, a_t)$ (expressed as a function of the input shares a_0, ..., a_t since it is the joint distribution of adversary probes) can besimulated using at most $n < t$ shares of a gadget's input, we mean that there exists an n-ary function d' such that for all a_0, ..., a_t, we have $d'(a_{\pi_1}, \ldots, a_{\pi_n}) = d(a_0, \ldots, a_t)$. (In other words, the distribution is fully determined by the value of at most n shares of the input.) We note in particular that this property makes no assumption on the distribution of inputs, but rather keeps both the real and simulated distribution dependent on it.

With this in mind, in order to break claims of NI or SNI security, it is sufficient to exhibit a set of intermediate variables whose distribution clearly depends on more than the allowed number of input shares. This is the approach we take here.

2. First Proposal

Algorithm 1 shows the first ZQZ proposal, which they claim is t-SNI for all t (and any linear function ℓ). The algorithm, as the original proposal by Coron et al. [3], assumes leak-free lookups in a table h such that $h[x] = x \cdot \ell(x)$ for any x.

2.1. A First-Order Flaw

We now exhibit a first-order flaw in this algorithm that is present at any order, and in any (boolean) ring or field. In order to show that Algorithm 1 is not 1-SNI, it is sufficient to exhibit either:

1. An intermediate variable whose distribution cannot be perfectly simulated (or predicted) given only one share of the input; or
2. An output variable whose distribution cannot be perfectly simulated (or predicted) without any knowledge about the input shares.

Here we explicitly exhibit only a flaw of the second kind, and therefore do not disprove that Algorithm 1 is t-NI, simply focusing on disproving that it is t-SNI as claimed by

Algorithm 1 The First ZQZ Proposal [5] (Algorithm 2)

```
function $H_t^1(\mathbf{a}, h)$
    for $i = 0$ to $t$ do
        for $j = i + 1$ to $t$ do
            $r_{i,j} \xleftarrow{\$} \mathbb{F}_{2^n}$
            $t_{i,j} \xleftarrow{\$} h[r_{i,j}] + h[\mathbf{a}_i + r_{i,j}]$
            $t_{j,i} \xleftarrow{\$} h[\mathbf{a}_i + r_{i,j} + \mathbf{a}_j] + h[\mathbf{a}_j + r_{i,j}]$
    for $i = 0$ to $t$ do
        $\mathbf{c}_i \leftarrow h[\mathbf{a}_i]$
        for $j = 0$ to $t$, $j \neq i$ do
            $\mathbf{c}_i \leftarrow \mathbf{c}_i + t_{i,j}$
    return $\mathbf{c}$
```

ZQZ.[1]

The final value of $\mathbf{c}_0$ at order t can be expressed as follows:

$$\mathbf{c}_0 = \mathbf{a}_0 \cdot \ell(\mathbf{a}_0) + \sum_{j=1}^{t} r_{0,j} \cdot \ell(r_{0,j}) + (\mathbf{a}_0 + r_{0,j}) \cdot \ell(\mathbf{a}_0 + r_{0,j})$$

We now show that the distribution of this expression cannot be predicted without knowledge of $\mathbf{a}_0$. First observe that, for $\mathbf{a}_0 = 0$, the distribution of $\mathbf{c}_0$ is the point distribution that gives probability 1 to element 0. Indeed, note that:

$$\begin{aligned} \mathbf{c}_0 &= \mathbf{a}_0 \cdot \ell(\mathbf{a}_0) + \textstyle\sum_{j=1}^{t} r_{0,j} \cdot \ell(r_{0,j}) + (\mathbf{a}_0 + r_{0,j}) \cdot \ell(\mathbf{a}_0 + r_{0,j}) \\ &= \quad 0 \quad + \textstyle\sum_{j=1}^{t} r_{0,j} \cdot \ell(r_{0,j}) + \qquad r_{0,j} \cdot \ell(r_{0,j}) \\ &= \quad 0 \end{aligned}$$

It is also easy to see that, if the same (constantly 0) distribution is obtained for other values of $\mathbf{a}_0$, then an adversary can obtain the value of $\mathbf{c} = \mathbf{a} \cdot \ell(\mathbf{a})$ by observing the remaining t shares of $\mathbf{c}$ ($\mathbf{c}_1, \ldots, \mathbf{c}_t$) and recombining them.

Therefore, it is the case that either i. the distribution of $\mathbf{c}_0$ depends on $\mathbf{a}_0$ (and in particular, an adversary observing a value of $\mathbf{c}_0 \neq 0$ could infer that $\mathbf{a}_0 \neq 0$); or ii. there is a set of t observations that reveals the value of $\mathbf{a}$.

Either of these situations is a counterexample to ZQZ's security claim. We exhibit a detailed breakdown of the flaw with $t = 1$, $n = 2$ and instantiating ℓ as the function that swaps the bits of its arguments, in Appendix A.

3. Second proposal

Algorithm 2 shows what we understand is the second ZQZ proposal [5], which they claim is t-NI for all t (and all ℓ).

[1] Although we do not prove that Algorithm 1 is not t-NI, this should not be taken as evidence that it provides any sort of probing security.

We first note that the Algorithm is not well-defined for odd values of t. Indeed, in such cases, the loop at Line 5 initialises only those r_j for *even* values of j such that $0 < j < t$. However, Line 17 makes use of r_i for all *odd* values of i such that $0 < i \leq t$ (regardless of the order). In the following, we do not pretend to fix this proposal, and only consider the security of Algorithm 2 for even values of t.[2]

Algorithm 2 The Second ZQZ Proposal [5] (Algorithm 3)

```
1:  function $H_t^2(\mathbf{a}, h)$
2:      for $i = 0$ to $t$ do
3:          for $j = 0$ to $t - i - 1$ by 2 do
4:              $r_{i,t-j} \overset{\$}{\leftarrow} \mathbb{F}_{2^n}$
5:      for $j = t - 1$ downto 1 by 2 do
6:          $r_j \overset{\$}{\leftarrow} \mathbb{F}_{2^n}$
7:      for $i = 0$ to $t$ do
8:          $\mathbf{c}_i \leftarrow h[\mathbf{a}_i]$
9:          for $j = t$ downto $i + 2$ by 2 do
10:             $t_{i,j} \leftarrow h[\mathbf{a}_i + r_{i,j}] + h[r_{i,j}] + h[\mathbf{a}_i + r_{j-1}] + h[r_{j-1}]$
11:             $\mathbf{c}_i \leftarrow \mathbf{c}_i + t_{i,j}$
12:         if $i \mod 2 \neq t \mod 2$ then
13:             $t_{i,i+1} \leftarrow h[\mathbf{a}_i + r_{i,i+1}] + h[r_{i,i+1}]$
14:             $\mathbf{c}_i \leftarrow \mathbf{c}_i + t_{i,i+1}$
15:         if $i \mod 2 = 1$ then
16:             for $j = i - 1$ downto 0 do
17:                 $t_{i,j} \leftarrow h[\mathbf{a}_i + r_i + \mathbf{a}_j] + h[\mathbf{a}_i + r_i]$
18:                 $\mathbf{c}_i \leftarrow \mathbf{c}_i + t_{i,j}$
19:         else
20:             for $j = i - 1$ downto 0 do
21:                 $t_{i,j} \leftarrow h[r_{j,i}] + h[\mathbf{a}_i + r_{j,i}]$
22:                 $\mathbf{c}_i \leftarrow \mathbf{c}_i + t_{i,j}$
23:     return $\mathbf{c}$
```

3.1. Counterexample

We instantiate Algorithm 2 with $t = 2$ (as Algorithm 3).

Consider the final distribution of $\mathbf{c}_1$ and note that, since ℓ is linear and the distribution of r_1 (resp. $r_{1,2}$) is the same as that of $\mathbf{a}_1 + r_1$ (resp. $\mathbf{a}_1 + r_{1,2}$, we have, denoting "equality of distributions knowing that r is distributed uniformly" using $\cong_r$:

[2]It is likely that the loop at Line 5 can be fixed to always initialise r_j with j odd. The presence of security flaws in the scheme makes investigating its correctness less useful.

Algorithm 3 Algorithm 2 with $t = 1$

```
  function $H_2^2(\mathbf{a}, h)$
      $r_{0,2} \overset{\$}{\leftarrow} \mathbb{F}_{2^n}$
      $r_{1,2} \overset{\$}{\leftarrow} \mathbb{F}_{2^n}$
      $r_1 \overset{\$}{\leftarrow} \mathbb{F}_{2^n}$
      $\mathbf{c}_0 \leftarrow h[\mathbf{a}_0]$
      $t_{0,2} \leftarrow h[\mathbf{a}_0 + r_{0,2}] + h[r_{0,2}] + h[\mathbf{a}_0 + r_1] + h[r_1]$
      $\mathbf{c}_0 \leftarrow \mathbf{c}_0 + t_{0,2}$
      $\mathbf{c}_1 \leftarrow h[\mathbf{a}_1]$
      $t_{1,2} \leftarrow h[\mathbf{a}_1 + r_{1,2}] + h[r_{1,2}]$
      $\mathbf{c}_1 \leftarrow \mathbf{c}_1 + t_{1,2}$
      $t_{1,0} \leftarrow h[\mathbf{a}_1 + r_1 + \mathbf{a}_0] + h[\mathbf{a}_1 + r_1]$
      $\mathbf{c}_1 \leftarrow \mathbf{c}_1 + t_{1,0}$
      $\mathbf{c}_2 \leftarrow h[\mathbf{a}_2]$
      $t_{2,1} \leftarrow h[r_{1,2}] + h[\mathbf{a}_2 + r_{1,2}]$
      $\mathbf{c}_2 \leftarrow \mathbf{c}_2 + t_{2,1}$
      $t_{2,0} \leftarrow h[r_{0,2}] + h[\mathbf{a}_2 + r_{0,2}]$
      $\mathbf{c}_2 \leftarrow \mathbf{c}_2 + t_{2,0}$
      return $\langle \mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2 \rangle$
```

$$\begin{aligned}
\mathbf{c}_1 = \;& \mathbf{a}_1 \cdot \ell(\mathbf{a}_1) + (\mathbf{a}_1 + r_{1,2}) \cdot \ell(\mathbf{a}_1 + r_{1,2}) + r_{0,2} \cdot \ell(r_{0,2}) \\
& +(\mathbf{a}_1 + r_1 + \mathbf{a}_0) \cdot \ell(\mathbf{a}_1 + r_1 + \mathbf{a}_0) + (\mathbf{a}_1 + r_1) \cdot \ell(\mathbf{a}_1 + r_1) \\
\cong_{r_{1,2}} \;& \mathbf{a}_1 \cdot \ell(\mathbf{a}_1) + r_{1,2} \cdot \ell(r_{1,2}) + r_{0,2} \cdot \ell(r_{0,2}) \\
& +(\mathbf{a}_1 + r_1 + \mathbf{a}_0) \cdot \ell(\mathbf{a}_1 + r_1 + \mathbf{a}_0) + (\mathbf{a}_1 + r_1) \cdot \ell(\mathbf{a}_1 + r_1) \\
\cong_{r_1} \;& \mathbf{a}_1 \cdot \ell(\mathbf{a}_1) + r_{1,2} \cdot \ell(r_{1,2}) + r_{0,2} \cdot \ell(r_{0,2}) + (r_1 + \mathbf{a}_0) \cdot \ell(r_1 + \mathbf{a}_0) + r_1 \cdot \ell(r_1) \\
= \;& \mathbf{a}_0 \cdot \ell(\mathbf{a}_0) + \mathbf{a}_1 \cdot \ell(\mathbf{a}_1) + r_1 \cdot \ell(\mathbf{a}_0) + \mathbf{a}_0 \cdot \ell(r_1) + r_{1,2} \cdot \ell(r_{1,2}) + r_{0,2} \cdot \ell(r_{0,2})
\end{aligned}$$

Tables 1 and 2 detail the distribution of $\mathbf{c}_1$ as a function of $\mathbf{a}_0$ and $\mathbf{a}_1$ with $t = 2$, $n = 2$, and $\ell = x \mapsto x \ll 1$. When distributions are single-valued, we simply give that value. In other cases we describe the probability mass function, omitting elements of probability 0. Detailed computations for the distribution of $r_1 \cdot (\mathbf{a}_0 \ll 1) + \mathbf{a}_0 \cdot (r_1 \ll 1)$ are given in Table 3, in an Appendix which also includes a table for the computation of $\cdot$ in $\mathbb{F}_{2^2}$ (Table 4). We also note that, for any x, we have

$$x \cdot (x \ll 1) = \begin{cases} 00 & \text{if } x = 00 \\ 10 & \text{otherwise.} \end{cases}$$

As a consequence, the distribution of $r_{1,2} \cdot \ell(r_{1,2}) + r_{0,2} \cdot \ell(r_{0,2})$, independent of $\mathbf{a}_0$ and $\mathbf{a}_1$ is simply expressed as $\{00 \mapsto \frac{5}{8}; 10 \mapsto \frac{3}{8}\}$.

The flaw here is more subtle: we need to show that predicting the distribution of $\mathbf{c}_1$ requires information on both $\mathbf{a}_0$ and $\mathbf{a}_1$.

When $\mathbf{a}_0 = 00$, the distribution of $\mathbf{c}_1$ clearly depends on $\mathbf{a}_1$ (as shown in the first quarter of Table 1).

$\mathbf{a}_0$	$\mathbf{a}_1$	$\mathbf{a}_0 \cdot (\mathbf{a}_0 \lll 1)$	$\mathbf{a}_1 \cdot (\mathbf{a}_1 \lll 1)$	$r_1 \cdot (\mathbf{a}_0 \lll 1) + \mathbf{a}_0 \cdot (r_1 \lll 1)$	$\mathbf{c}_1$
00	00	00	00	00	$\{00 \mapsto \frac{5}{8}; 10 \mapsto \frac{3}{8}\}$
00	01	00	10	00	$\{00 \mapsto \frac{3}{8}; 10 \mapsto \frac{5}{8}\}$
00	10	00	10	00	$\{00 \mapsto \frac{3}{8}; 10 \mapsto \frac{5}{8}\}$
00	11	00	10	00	$\{00 \mapsto \frac{3}{8}; 10 \mapsto \frac{5}{8}\}$
01	00	10	00	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$
01	01	10	10	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$
01	10	10	10	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$
01	11	10	10	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$
10	00	10	00	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$
10	01	10	10	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$
10	10	10	10	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$
10	11	10	10	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$
11	00	10	00	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$
11	01	10	10	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$
11	10	10	10	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$
11	11	10	10	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$

Table 1: Distribution of $\mathbf{c}_1$ as a function of $\mathbf{a}_0$.

$\mathbf{a}_1$	$\mathbf{a}_0$	$\mathbf{a}_1 \cdot (\mathbf{a}_1 \lll 1)$	$\mathbf{a}_0 \cdot (\mathbf{a}_0 \lll 1)$	$r_1 \cdot (\mathbf{a}_0 \lll 1) + \mathbf{a}_0 \cdot (r_1 \lll 1)$	$\mathbf{c}_1$
00	00	00	00	00	$\{00 \mapsto \frac{5}{8}; 10 \mapsto \frac{3}{8}\}$
00	01	00	10	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$
00	10	00	10	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$
00	11	00	10	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$
01	00	10	00	00	$\{00 \mapsto \frac{3}{8}; 10 \mapsto \frac{5}{8}\}$
01	01	10	10	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$
01	10	10	10	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$
01	11	10	10	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$
10	00	10	00	00	$\{00 \mapsto \frac{3}{8}; 10 \mapsto \frac{5}{8}\}$
10	01	10	10	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$
10	10	10	10	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$
10	11	10	10	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$
11	00	10	00	00	$\{00 \mapsto \frac{3}{8}; 10 \mapsto \frac{5}{8}\}$
11	01	10	10	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$
11	10	10	10	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$
11	11	10	10	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$	$\{00 \mapsto \frac{1}{2}; 10 \mapsto \frac{1}{2}\}$

Table 2: Distribution of $\mathbf{c}_1$ as a function of $\mathbf{a}_1$.

Similarly, whatever the value of $\mathbf{a}_1$, the distribution of $\mathbf{c}_1$ varies with the value of $\mathbf{a}_0$. This is shown more clearly in Table 2.

The leak exhibited here is clearly more subtle than that on the first proposal, for several reasons:

1. although we suspect similar flaws exist at higher orders, in other structures and for other instantiations for ℓ, we only explicitly exhibit it for $t = 2$ and in the case where $n = 2$ and ℓ is the 1-bit rotation function;
2. rather than a single observation leaking information about the input, the flaw on the second proposal requires the adversary to estimate the distribution of $\mathbf{c}_1$ (by performing multiple measurements) to infer the information.

The bias we exhibit is quite small, but exists, and therefore contradicts the claims of perfect probing security made by the original authors.

4. Analysis, Discussion and Conclusions

In this short note, we have exhibited counterexamples to both of Zhang, Qiu and Zhou's core theorems [5]. However, their compositional proof for the AES SBox would remain valid given a t-NI gadget with reduced randomness complexity that computes $x \mapsto x{\cdot}\ell(x)$, and searching for such a gadget remains a valuable research objective.

We do not claim that the leakage we exhibit here could be exploited in practice, but simply wish to bring attention to the fact that the gadgets proposed by Zhang, Qiu and Zhou are not as secure as claimed, and may therefore not be suitable for practical security-critical applications, despite their improved randomness complexity.

The counterexamples shown here were found using Barthe et al.'s maskVerif tools [1, 2]. Since the proof techniques used in those tools are incomplete (that is, they may fail to prove true statements), the counterexamples were further verified and refined by hand (and the first flaw generalised). This short note and its use of formal verification tools further illustrate, if it was still needed, the value of formal methods and automated verification tools in the domain of provably secure masked algorithms, for which proofs are notoriously tedious, error prone and difficult to check.

References

[1] Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, and Pierre-Yves Strub. Verified proofs of higher-order masking. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 457–485. Springer, Heidelberg, April 2015.

[2] Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, Pierre-Yves Strub, and Rébecca Zucchini. Strong non-interference and type-directed higher-order masking. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 16*, pages 116–129. ACM Press, October 2016.

[3] Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, and Thomas Roche. Higher-order side channel security and mask refreshing. In Shiho Moriai, editor, *FSE 2013*, volume 8424 of *LNCS*, pages 410–424. Springer, Heidelberg, March 2014.

[4] Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 463–481. Springer, Heidelberg, August 2003.

[5] Rui Zhang, Shuang Qiu, and Yongbin Zhou. Further improving efficiency of higher order masking schemes by decreasing randomness complexity. *IEEE Transactions on Information Forensics and Security*, 12(11), November 2017.

A. A Detailed Look at the First Proposal

We now give a detailed account of a particular instance of the flaw we exhibit on ZQZ's first proposal (Algorithm 1. We instantiate Algorithm 1 for $t = 1$ (as Algorithm 4) and exhibit a single output share whose distribution varies with $\mathbf{a}_0$, and therefore cannot be simulated without knowledge about that input share. This violates the 1-SNI property claimed by ZQZ [5].

Algorithm 4 Algorithm 1 with $t = 1$

```
function $H_1^1(\mathbf{a} = (\mathbf{a}_0, \mathbf{a}_1), h)$
    $r_{0,1} \overset{\$}{\leftarrow} \mathbb{F}_{2^n}$
    $t_{0,1} \leftarrow h[r_{0,1}] + h[\mathbf{a}_0 + r_{0,1}]$
    $t_{1,0} \leftarrow h[\mathbf{a}_0 + r_{0,1} + \mathbf{a}_1] + h[\mathbf{a}_1 + r_{0,1}]$
    $\mathbf{c}_0 \leftarrow h[\mathbf{a}_0]$
    $\mathbf{c}_0 \leftarrow \mathbf{c}_0 + t_{0,1}$
    $\mathbf{c}_1 \leftarrow h[\mathbf{a}_1]$
    $\mathbf{c}_1 \leftarrow \mathbf{c}_1 + t_{1,0}$
    return $\langle \mathbf{c}_0, \mathbf{c}_1 \rangle$
```

Consider the distribution of the final value of $\mathbf{c}_0$, and note that, since ℓ is linear, we have:

$$\begin{aligned}
\mathbf{c}_0 &= \mathbf{a}_0 \cdot \ell(\mathbf{a}_0) + r_{0,1} \cdot \ell(r_{0,1}) + (\mathbf{a}_0 + r_{0,1}) \cdot \ell(\mathbf{a}_0 + r_{0,1}) \\
&= \mathbf{a}_0 \cdot \ell(\mathbf{a}_0) + r_{0,1} \cdot \ell(r_{0,1}) + \mathbf{a}_0 \cdot \ell(\mathbf{a}_0) + \mathbf{a}_0 \cdot \ell(r_{0,1}) + r_{0,1} \cdot \ell(\mathbf{a}_0) + r_{0,1} \cdot \ell(r_{0,1}) \\
&= \mathbf{a}_0 \cdot \ell(r_{0,1}) + r_{0,1} \cdot \ell(\mathbf{a}_0)
\end{aligned}$$

For example, choosing $n = 2$ and $\ell(x) = x \lll 1$ (the left rotation on binary words), it is clear that the distribution of $\mathbf{c}_0$ does in fact depend on the value of $\mathbf{a}_0$. For example, if $\mathbf{a}_0 = 00$, then $\mathbf{c}_0 = 00$ for all possible values of $r_{0,1}$, whereas other values of $\mathbf{a}_0$ yield the same two-valued distribution on $\mathbf{c}_0$ (see Table 3; Table 4 recalls multiplication in $\mathbb{F}_{2^2}$).

This proves that Algorithm 1 is not t-SNI for all t and for all ℓ.

$\mathbf{a}_0$	$r_{0,1}$	$\mathbf{a}_0 \lll 1$	$r_{0,1} \lll 1$	$\mathbf{a}_0 \cdot (r_{0,1} \lll 1)$	$r_{0,1} \cdot (\mathbf{a}_0 \lll 1)$	$r_{0,1} \cdot (\mathbf{a}_0 \lll 1) + \mathbf{a}_0 \cdot (r_{0,1} \lll 1)$
00	00	00	00	00	00	00
00	01	00	10	00	00	00
00	10	00	01	00	00	00
00	11	00	11	00	00	00
01	00	10	00	00	00	00
01	01	10	10	10	10	00
01	10	10	01	01	11	10
01	11	10	11	11	01	10
10	00	01	00	00	00	00
10	01	01	10	11	01	10
10	10	01	01	10	10	00
10	11	01	11	01	11	10
11	00	11	00	00	00	00
11	01	11	10	01	11	10
11	10	11	01	11	01	10
11	11	11	11	10	10	00

Table 3: Distribution of $\mathbf{c}_0$ as a function of $\mathbf{a}_0$: detailed computation.

$\cdot$	00	01	10	11
00	00	00	00	00
01	00	01	10	11
10	00	10	11	01
11	00	11	01	10

Table 4: Multiplication in $\mathbb{F}_{2^2}$.