Correction on "Further Improving Efficiency of Higher-Order Masking Schemes by Decreasing Randomness Complexity"

Shuang Qiu, Rui Zhang, Yongbin Zhou, Wei Cheng

***Abstract*—Provably secure masking schemes always require too many random generations, which significantly increases the implementation cost. Recently in IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY (TIFS) (DOI:10.1109/TIFS.2017.2713323), Zhang, Qiu, and Zhou improve the efficiency of the CPRR scheme by decreasing the random generations. Recently, Barthe et al. claim that security flaws exist in both proposals and provide the counter-examples. In this paper, we fix these security flaws by changing the addition order. In this way, the two proposals are corrected with no extra random generation.**

***Index Terms*—masking scheme, side-channel attacks, probing model, randomness complexity.**

I. INTRODUCTION

MASKING is the most widely deployed countermeasure against the Side-Channel Attack (SCA). In the scope of higher-order masking, randomness reduction is a crucial and tough task. Recently in the above paper [8], Zhang, Qiu, and Zhou have proposed two variants of the CPRR scheme, which outperform the original CPRR scheme with 50% and 50%-75% randomness reductions, respectively. Furthermore, under the probing model, they prove that the two schemes, called the ZQZ schemes, satisfy SNI and TNI, respectively.

Subsequently, Barthe, Dupressoir, and Grégoire [3] find two security flaws and a typo existing in the ZQZ schemes with the automated verifier MaskVerif [2], [1]:

1) the first proposal (the ZQZ-1 scheme) fail to achieve SNI, as the first output share c_0 shows dependence on the first input share a_0.
2) the second proposal (the ZQZ-2 scheme), which is derived from the ZQZ-1 scheme, cannot achieve TNI.
3) there is a typo in the ZQZ-2 scheme, which makes it unable to be generalized to odd orders d.

After revisiting the two masking schemes, we found that both Problem 1 and Problem 2 are due to one simple mistake. In the ZQZ schemes, the terms $h(r_{i,j}) + h(a_i + r_{i,j})$ and $h(a_i + r_{i,j} + a_j) + h(a_j + r_{i,j})$ are dependent on the input share a_i, as the randomness $r_{i,j}$ is unfortunately counteracted. In fact, this means that the random values are not correctly added and each term in the ZQZ schemes is left unprotected. As a result, the ZQZ-1 scheme cannot even achieve TNI, as all the random variables $r_{i,j}$ are invalid. As the ZQZ-2 scheme is obtained by decreasing the randomness of the ZQZ-1 scheme, the ZQZ-2 scheme cannot achieve TNI, either.

In order to fix the ZQZ schemes, we replace the two terms $h(r_{i,j}) + h(a_i + r_{i,j})$ and $h(a_i + r_{i,j} + a_j) + h(a_j + r_{i,j})$ with the modified terms $h(r_{i,j})$ and $h(a_i + r_{i,j}) + h(a_i + r_{i,j} + a_j) + h(a_j + r_{i,j})$, which are independent of the input shares. In this way, each two terms are protected by one randomness $r_{i,j}$, and thus the security bias is fixed.

II. PRELIMINARIES

A. Notations

Linear function is denoted as $\ell(\cdot)$. The arrow $\leftarrow$ means to assign the value of the right variable to the left variable. $\xleftarrow{\$}$ means to randomly pick one value from the right set and assign this value to the left variable. $x \mapsto y$ means a function which maps from x to y. $+$ denotes bit-xor operation, and $\cdot$ denotes the field multiplication on the finite field $\mathbb{F}_{2^n}$. $\sum_{i=0}^{m}$ represents the xor-sum, namely $\sum_{i=0}^{m} x_i = x_0 + x_1 + \cdots + x_m$.

B. Security Notions

Two security notions are involved in this paper, i.e. Non-Interference (NI) and Strong-Non-Interference (SNI) [1]. Their definitions are based on the notion of simulatability, which is first proposed by Ishai *et al.* in [6] and then utilized by almost all subsequent masking schemes.

Definition 1 (Simulatability): Denote by $V = \{v_1, \ldots, v_m\}$ the set of m variables of a multiplication algorithm. If there exists two sets $I = \{i_1, \ldots, i_t\}$ and $J = \{j_1, \ldots, j_t\}$ of t indices from set $\{0, \ldots, d\}$ and a random function S taking as input $2t$ bits and outputting m bits such that for any fixed bits $(a_i)_{0 \leq i \leq d}$ and $(b_j)_{0 \leq j \leq d}$, the distributions of $\{v_1, \ldots, v_m\}$ and $\{S(a_{i_1}, \ldots, a_{i_t}, b_{j_1}, \ldots, b_{j_t})\}$ are identical, we say the set V can be simulated with at most t shares of each input[1] a_I and b_J.

Definition 2 (d-Tight-Non-Interference): An algorithm satisfies d-Tight-Non-Interference (d-TNI) if and only if every tuple of $t \leq d$ variables can be perfectly simulated with at most t shares of each input.

Definition 3 (d-Strong-Non-Interference): An algorithm satisfies d-Strong-Non-Interference (d-SNI) if and only if for every set $\mathcal{I}$ of variables on intermediate variables (i.e. no

The authors are with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China, and the School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China (*Corresponding author: Rui Zhang, e-mail: r-zhang@iie.ac.cn*).

[1]The set $\{a_{i_1}, \ldots, a_{i_t}\}$ is written as a_I, and the set $\{b_{j_1}, \ldots, b_{j_t}\}$ is written as b_J.

Algorithm 1: ZQZ-1 Scheme.

Input: sharing $(a_i)_{0\le i\le d}$ satisfying $\sum_i a_i = a$, a LUT for $h(a) = a\cdot\ell(a)$
Output: sharing $(c_i)_{0\le i\le d}$ satisfying $\sum_i c_i = a\cdot\ell(a)$

for $i = 0$ **to** d **do**
  for $j = i+1$ **to** d **do**
    $r_{i,j} \xleftarrow{\$} \mathbb{F}_{2^n}$
    $t_{i,j} \leftarrow h(r_{i,j}) + \mathbf{h(a_i + r_{i,j})}$
    $t_{j,i} \leftarrow h(a_i + r_{i,j} + a_j) + h(a_j + r_{i,j})$
for $i = 0$ **to** d **do**
  $c_i \leftarrow h(a_i)$
  for $j = 0$ **to** $d,\ j \neq i$ **do**
    $c_i \leftarrow c_i + t_{i,j}$

output shares) and every set $\mathcal{O}$ of variables on output shares such that $|\mathcal{I}| + |\mathcal{O}| \le d$, the set $\mathcal{I}\cup\mathcal{O}$ can be simulated by only $|\mathcal{I}|$ shares of each input.

C. ZQZ Schemes

In [5], Coron *et al.* propose d-th order masking scheme for the dependent-input multiplication $a\cdot\ell(a)$, i.e. the CPRR scheme. In [8], authors reduce the randomness complexity of the CPRR scheme, and propose two masking schemes, which we call the ZQZ-1 scheme and the ZQZ-2 scheme in the sequel. The description of ZQZ-1 scheme is given in Alg. 1. It is noteworthy that the involved function $h(x) = x\cdot\ell(x)$ is computed by calling a Look-Up-Table (LUT).

The ZQZ-2 scheme is obtained by reusing the random numbers of ZQZ-1, according to the randomness reusing strategy in [4]. For clarity, an illustration of the ZQZ-2 scheme in case $d = 6$ is given in Fig. 1, where $t_{i,j}(r)$ represents term $t_{i,j}$ involving random value r, and the sum of all terms on the i-th line equals the i-th output share c_i. The reused terms are printed in a larger blue font. It is noteworthy that, in the ZQZ-2 scheme, terms $\left[t_{i,j}, t_{i,j-1}\right]$ in one bracket is combined into one term $t_{i,j}$.

III. Security Analysis of ZQZ Schemes

Based on the observation of Barthe *et al.* [3], we revisit the security of the ZQZ schemes. Furthermore, we trace to the source of the security flaws.

A. Counteracted Randomness and Undesirable Dependence

In the CPRR scheme [5] and the ZQZ schemes [8], ordinary multiplications are replaced with quadratic function $h(x) = x\cdot\ell(x)$. Each quadratic function $h(x) = x\cdot\ell(x)$ is implemented by calling a precomputed LUT. In the ZQZ-1 scheme (Alg. 1), $t_{i,j}$ and $t_{j,i}$ satisfies:

$$\begin{aligned} t_{i,j} &= h(r_{i,j}) + h(a_i + r_{i,j}) \\ t_{j,i} &= h(a_i + r_{i,j} + a_j) + h(a_j + r_{i,j})\,. \end{aligned} \tag{1}$$

According to the description of function $h(\cdot)$, term $t_{i,j}$ can be rewritten as

$$h(a_i + r_{i,j}) + h(r_{i,j}) = a_i\ell(r_{i,j}) + r_{i,j}\ell(a_i) + a_i\ell(a_i)\,. \tag{2}$$

According to Eq. (2), when a_i equals zero, $t_{i,j}$ will definitely equal zero[2]. Namely, $t_{i,j}$ can be seen as the product of a_i and a function of $(a_i, r_{i,j})$:

$$t_{i,j} = \mathbf{a_i}\cdot f(a_i, r_{i,j})\,. \tag{3}$$

Obviously, $t_{i,j}$ leaks a_i. Similarly, term $t_{j,i}$ can be rewritten as

$$\begin{aligned} t_{j,i} &= h(a_j + r_{i,j}) + h(a_i + (r_{i,j} + a_j)) \\ &= \mathbf{a_i}\cdot f(a_j + r_{i,j}, a_i)\,. \end{aligned} \tag{4}$$

According to Eq. (4), $t_{j,i}$ also leaks a_i[3].

B. Invalid Assumptions in Security Proofs

Given that $t_{i,j}$ leaks a_i and $t_{j,i}$ leaks a_i ($j > i$), the assumption in the security proof for ZQZ-1 in [8] can no longer hold:

1) variables in the fourth set $h(a_i) + \sum_{j=0}^{j_0}[h(a_i + r_{j,i} + a_j) + h(a_i + r_{j,i})]$ (refer to [8], Page 7, right column, Line 8) cannot be simulated with only a_i, as each term $h(a_i + r_{j,i} + a_j) + h(a_i + r_{j,i})$ leaks a_j. Hence, it should be simulated with $\{a_i, a_0, a_1, \cdots, a_{j0}\}$.
2) the observed output share c_i also leaks $\{a_0, a_1, \cdots, a_i\}$, which contradicts with the security proof (refer to [8], Page 7, right column, Line 42).

Accordingly, the security proof for the ZQZ-1 scheme cannot hold.

C. Counter Example to TNI

In [3], authors propose a counter-example to show that the ZQZ-1 scheme is not SNI. Here, we further propose an example to show that the ZQZ-1 scheme is not even TNI. The last output share c_d can be rewritten as,

$$\begin{aligned} c_d &= h(a_d) + \sum_{j=0}^{d-1}[h(a_j + r_{j,i} + a_i) + h(a_i + r_{j,i})] \\ &= h(a_d) + \sum_{j=0}^{d-1} a_j\cdot f(a_i + r_{j,i}, a_j)\,. \end{aligned} \tag{5}$$

According to Eq. (5), it is easy to figure out that when $a_0, a_1, \cdots, a_d$ are all set to zero, the output share c_d are definitely zero, which implies that the output share c_d show some dependence on the joint distribution of $d+1$ input shares of a. Moreover, as the ZQZ-2 scheme is derived from the ZQZ-1 scheme, the ZQZ-2 scheme can hardly preserve its security level, either.

IV. Fixed Versions of ZQZ-1 and ZQZ-2

By eliminating the undesirable dependence (Sec. III), we fix the security flaws in the ZQZ schemes, and thus obtain the modified ZQZ schemes.

[2] In this paper, the linear function $\ell(\cdot)$ is assumed to be the squaring operation over the finite field. In this case, when a_i equals zero, $\ell(a_i)$ equals zero as well.

[3] Note that $t_{j,i}$ does not leak a_j, as it only relates with $a_j + r_{i,j}$.

$$
\begin{array}{lcccccc}
h(a_0) & \big[t_{0,6}(r_{0,6}) & t_{0,5}(r_5)\big] & \big[t_{0,4}(r_{0,4}) & t_{0,3}(r_3)\big] & \big[t_{0,2}(r_{0,2}) & t_{0,1}(r_1)\big] \\
h(a_1) & \big[t_{1,6}(r_{1,6}) & t_{1,5}(r_5)\big] & \big[t_{1,4}(r_{1,4}) & t_{1,3}(r_3)\big] & \big[t_{1,2}(r_{1,2})\big] & t_{1,0}(r_1) \\
h(a_2) & \big[t_{2,6}(r_{2,6}) & t_{2,5}(r_5)\big] & \big[t_{2,4}(r_{2,4}) & t_{2,3}(r_3)\big] & t_{2,1}(r_{1,2}) & t_{2,0}(r_{0,2}) \\
h(a_3) & \big[t_{3,6}(r_{3,6}) & t_{3,5}(r_5)\big] & \big[t_{3,4}(r_{3,4})\big] & t_{3,2}(r_3) & t_{3,1}(r_3) & t_{3,0}(r_3) \\
h(a_4) & \big[t_{4,6}(r_{4,6}) & t_{4,5}(r_5)\big] & t_{4,3}(r_{3,4}) & t_{4,2}(r_{2,4}) & t_{4,1}(r_{1,4}) & t_{4,0}(r_{0,4}) \\
h(a_5) & \big[t_{5,6}(r_{5,6})\big] & t_{5,4}(r_5) & t_{5,3}(r_5) & t_{5,2}(r_5) & t_{5,1}(r_5) & t_{5,0}(r_5) \\
h(a_6) & t_{6,5}(r_{5,6}) & t_{6,4}(r_{4,6}) & t_{6,3}(r_{3,6}) & t_{6,2}(r_{2,6}) & t_{6,1}(r_{1,6}) & t_{6,0}(r_{0,6})
\end{array}
$$

Fig. 1: Illustration of randomness reusing in the ZQZ-2 scheme for $d = 6$.

Algorithm 2: Modified ZQZ-1 Scheme.

Input: sharing $(a_i)_{0\le i\le d}$ satisfying $\sum_i a_i = a$, a LUT for $h(a) = a \cdot \ell(a)$

Output: sharing $(c_i)_{0\le i\le d}$ satisfying $\sum_i c_i = a \cdot \ell(a)$

for $i = 0$ **to** d **do**
  for $j = i + 1$ **to** d **do**
    $r_{i,j} \xleftarrow{\$} \mathbb{F}_{2^n}$
    $t_{i,j} \leftarrow h(r_{i,j})$
    $t_{j,i} \leftarrow \mathbf{h(a_i + r_{i,j})} + h(a_i + r_{i,j} + a_j) + h(a_j + r_{i,j})$
for $i = 0$ **to** d **do**
  $c_i \leftarrow h(a_i)$
  for $j = 0$ **to** $d,\ j \neq i$ **do**
    $c_i \leftarrow c_i + t_{i,j}$

A. Modified ZQZ-1 Scheme

In order to fix the above security bias of the ZQZ schemes, we first slightly modify the ZQZ-1 scheme, as is given in Alg. 2. In the modified ZQZ-1 scheme, $t_{i,j}$ and $t_{j,i}$ are changed:

$$
\begin{aligned}
&t_{i,j} \leftarrow h(r_{i,j}) \\
&t_{j,i} \leftarrow \left[\mathbf{h(a_i + r_{i,j})} + h(a_i + r_{i,j} + a_j)\right] + h(a_j + r_{i,j})\,.
\end{aligned}
\tag{6}
$$

Obviously, $t_{i,j}$ and $t_{j,i}$ are independent of $\mathbf{a_i}$ and $\mathbf{a_j}$, due to the randomness $r_{i,j}$. Till now, the security bias in the ZQZ-1 scheme has been fixed, and the modified ZQZ-1 scheme achieves d-SNI. The security proof is given in Appendix A.

It is noteworthy that the addition order of $t_{j,i}$ in Eq. (6) should be carefully chosen. During the computation of $t_{j,i}$, there exists one intermediate, where in the case of Eq. (6) it is $h(a_i + r_{i,j}) + h(a_i + r_{i,j} + a_j)$. In order to make the security proof valid, this intermediate variable should be dependent on at most one input share. In this case, $h(a_i + r_{i,j}) + h(a_i + r_{i,j} + a_j)$ can be rewritten as:

$$
\begin{aligned}
h(a_i + r_{i,j}) + h(a_i + r_{i,j} + a_j)& \\
=&(a_i + r_{i,j})\ell(a_j) + a_j\ell(a_i + r_{i,j}) + a_j\ell(a_j) \\
=&a_j \cdot f(a_i + r_{i,j}, a_j)\,.
\end{aligned}
\tag{7}
$$

Hence, $h(a_i + r_{i,j}) + h(a_i + r_{i,j} + a_j)$ only depends on a_j.

Otherwise, if $t_{j,i}$ is computed following the order below:

$$
\begin{aligned}
t_{j,i} =&\left[h(a_i + r_{i,j}) + h(a_j + r_{i,j})\right] \\
&+ h(a_i + r_{i,j} + a_j)\,,
\end{aligned}
\tag{8}
$$

the intermediate variable is $h(a_i + r_{i,j}) + h(a_j + r_{i,j})$, which satisfies:

$$
\begin{aligned}
h(a_i + r_{i,j})+&h(a_j + r_{i,j}) \\
&=a_i \cdot f(a_i, r_{i,j}) + a_j \cdot f(a_j, r_{i,j})\,.
\end{aligned}
\tag{9}
$$

In this case, the intermediate will depend on both a_i and a_j[4], and the masking scheme will be insecure.

B. Modified ZQZ-2 Scheme

In [8], the ZQZ-2 scheme is obtained by decreasing the randomness of the ZQZ-1 scheme, with the randomness reduction strategy proposed in [4]. As shown above, the ZQZ-1 scheme is flawed and cannot achieve d-TNI. Since the ZQZ-2 scheme is a randomness reduction version of the ZQZ-1 scheme, the ZQZ-2 scheme cannot achieve d-TNI as well.

In this section, we obtain the modified ZQZ-2 scheme by applying the randomness reduction strategy (see Fig. 1) to the modified ZQZ-1 scheme. The modified ZQZ-2 is given in Alg. 3. We claim that the modified ZQZ-2 scheme achieves its claimed security level, the d-TNI. The security proof is given in Appendix B.

Moreover, we modify the 14-th line of Alg. 3 and make the description can be generalized to odd orders.

It is noteworthy that the addition order of $t_{i,j}$ (line 9) is carefully chosen. Any change in addition order may lead to security bias. As a counter-example, if the term $t_{i,j}$ is computed according to the following order, where the fourth term and the sixth term switch positions,

$$
\begin{aligned}
t_{i,j} =& \left[h(a_j + r_{i,j}) + h(a_j + r_{i,j} + a_i) + h(a_i + r_{i,j})\right] \\
&+ \left[\mathbf{h(a_i + r_{j-1})} + h(a_{j-1} + r_{j-1} + a_i)\right. \\
&\left.+ \mathbf{h(a_{j-1} + r_{j-1})}\right]\,,
\end{aligned}
\tag{10}
$$

[4]When $a_i = 0$ and $a_j = 0$, the intermediate $h(a_i + r_{i,j}) + h(a_j + r_{i,j}) = 0$ with the probability of 1.

Algorithm 3: Modified ZQZ-2 Scheme.

Input: sharing $(a_i)_{0\le i\le d}$ satisfying $\sum_i a_i = a$, a LUT for $h(a) = a\cdot \ell(a)$

Output: sharing $(c_i)_{0\le i\le d}$ satisfying $\sum_i c_i = a\cdot \ell(a)$

for $i = 0$ **to** d **do**
  for $j = 0$ **to** $d-i-1$ *by* 2 **do**
    $r_{i,d-j} \xleftarrow{\$} \mathbb{F}_{2^n}$
for $j = d-1$ **downto** 1 *by* 2 **do**
  $r_j \xleftarrow{\$} \mathbb{F}_{2^n}$
for $i = 0$ **to** d **do**
  $c_i \leftarrow h(a_i)$
  for $j = d$ **downto** $i+2$ *by* 2 **do**
    $t_{i,j} \leftarrow \mathbf{h(a_j + r_{i,j})} + h(a_j + r_{i,j} + a_i) + h(a_i + r_{i,j}) + \mathbf{h(a_{j-1} + r_{j-1})} + h(a_{j-1} + r_{j-1} + a_i) + h(a_i + r_{j-1})$
    $c_i \leftarrow c_i + t_{i,j}$
  if $i \neq d$ *(mod 2)* **then**
    $t_{i,i+1} \leftarrow \mathbf{h(a_{i+1} + r_{i,i+1})} + h(a_{i+1} + r_{i,i+1} + a_i) + h(a_i + r_{i,i+1})$
    $c_i \leftarrow c_i + t_{i,i+1}$
    if $\mathbf{d = 0}$ **(mod 2) then**
      $c_i \leftarrow c_i + h(r_i)$
  else
    for $j = i-1$ **downto** 0 **do**
      $c_i \leftarrow c_i + h(r_{j,i})$

there will be intermediate $t^0_{i,j}$ during the computation,

$$\begin{aligned} t^0_{i,j} =& \big[h(r_{i,j}) + a_j\ell(a_i) + a_i\ell(a_j)\big] + h(a_i + r_{j-1}) \\ &+ h(a_{j-1} + r_{j-1} + a_i) \\ =& \big[h(r_{i,j}) + a_j\ell(a_i) + a_i\ell(a_j)\big] \\ &+ a_{j-1}\cdot f(a_i + r_{j-1}, a_{j-1})\,. \end{aligned} \tag{11}$$

This intermediate $t^0_{i,j}$ depends on a_i, a_j, a_{j-1}, and $r_{i,j}$. Thus, the joint distribution of two intermediate variables $(t^0_{i,j}, r_{i,j})$ depends on three input shares a_i, a_j, a_{j-1}, which makes the scheme insecure.

In the modified ZQZ-2 scheme, the intermediate sum $t^0_{i,j}$ satisfies

$$\begin{aligned} t^0_{i,j} =& \big[h(r_{i,j}) + a_j\ell(a_i) + a_i\ell(a_j)\big] + h(a_{j-1} + r_{j-1}) \\ &+ h(a_{j-1} + r_{j-1} + a_i) \\ =& \big[h(r_{i,j}) + a_j\ell(a_i) + a_i\ell(a_j)\big] \\ &+ a_i\cdot f(a_{j-1} + r_{j-1}, a_i)\,, \end{aligned} \tag{12}$$

hence the joint distribution of two intermediate variables $(t^0_{i,j}, r_{i,j})$ depends on only two input shares a_i and a_j, which satisfies the requirement of TNI.

V. Conclusion and Perspective

In this paper, we fix the security flaws of the ZQZ schemes. In this way, the randomness reduction expected in the original paper [8] can be achieved. Besides, we suggest that any further randomness reduction strategy for ISW-like schemes, e.g. the new progress in CRYPTO 2017 [7], can also be securely applied to the modified ZQZ-1 scheme, and thus one can obtain a more efficient ZQZ-2 scheme achieving TNI.

Appendix A
Proof of Modified ZQZ-1

Denote a tuple observations $(\mathcal{I}, \mathcal{O})$, where $|\mathcal{I}| + |\mathcal{O}| \le d$. We aim to prove that this scheme is SNI, i.e. one can always simulate $(\mathcal{I}, \mathcal{O})$ utilizing $|\mathcal{I}|$ shares of each input. Hence, this proof consists in constructing set $\mathcal{S}$ of indices in $\{0, 1, \cdots, d\}$ of size at most $|\mathcal{I}|$ and perfectly simulate $(\mathcal{I}, \mathcal{O})$ with the shares $(a_i)_{i\in I}$.

First, we show how to construct set $\mathcal{S}$. Initially, set $\mathcal{S}$ is empty. We fill it in the following specific order according to the possible leaked intermediate variables in $\mathcal{I}$.

1) for any observed variables a_i and $h(a_i)$, add i to $\mathcal{S}$.
2) for any observed variables $r_{i,j}$, $h(r_{i,j})$, $a_i + r_{i,j}$, and $h(a_i + r_{i,j})$, add i to $\mathcal{S}$.
3) for any observed variables $a_i + r_{i,j} + a_j$, $h(a_i + r_{i,j} + a_j)$, $a_j + r_{i,j}$, $h(a_j + r_{i,j})$: if $i \notin \mathcal{S}$, add i to $\mathcal{S}$, otherwise add j to $\mathcal{S}$.
4) for any observed variables $t_{j,i} = h(a_i + r_{i,j}) + h(a_i + r_{i,j} + a_j) + h(a_j + r_{i,j})$: if $i \notin \mathcal{S}$, add i to $\mathcal{S}$, otherwise add j to $\mathcal{S}$.
5) for the observed variable $h(a_i + r_{i,j}) + h(a_i + r_{i,j} + a_j)$, add j to $\mathcal{S}$.
6) for any observed variables $h(a_i) + \sum_{j=0}^{j_0} \big[h(a_j + r_{j,i}) + h(a_j + r_{j,i} + a_i) + h(a_i + r_{j,i})\big]$ with $1 \le j_0 \le i-1$ and $h(a_i) + \sum_{j=0}^{i-1} \big[h(a_j + r_{j,i}) + h(a_j + r_{j,i} + a_i) + h(a_i + r_{j,i})\big] + \sum_{j=i+1}^{j_0} h(r_{i,j})$ with $j_0 < i < d$, add i to $\mathcal{S}$.

The output shares are the final value c_i, which are included in set $\mathcal{O}$.

Now the set $\mathcal{S}$ has been determined. Each observation in $\mathcal{I}$ adds at most one index to set $\mathcal{S}$. Hence, the simulator satisfies $|\mathcal{S}| \le |\mathcal{I}|$. Then, we prove that every observed value can be perfectly simulated with the input shares whose indices are among $\mathcal{S}$.

- any variable in group 1 can be simulated with a_i.
- any variable in group 2 can be simulated with a_i and $r_{i,j}$.
- for each variable in Group 3, we consider two cases. If $i \in \mathcal{S}$ and we add j to $\mathcal{S}$, any variable in Group 3 can be simulated with a_i, a_j, and $r_{i,j}$. If $i \notin \mathcal{S}$ and we add i to $\mathcal{S}$, then $r_{i,j}$ and $a_i + r_{i,j}$ does not enter in the computation of any other variables. Hence, $a_i + r_{i,j} + a_j$ and $a_j + r_{i,j}$ can be assigned to a fresh random value.
- for variables in group 4, $t_{j,i}$ can be rewritten as $h(r_{i,j}) + a_i\ell(a_j) + a_j\ell(a_i)$. If $i \in \mathcal{S}$ and we add j to $\mathcal{S}$, $t_{j,i}$ can be simulated with a_i, a_j, and $r_{i,j}$. If $i \notin \mathcal{S}$ and we add i to $\mathcal{S}$, then $r_{i,j}$ does not enter in the computation of any other variables. Hence, $t_{j,i}$ can be assigned to a fresh random value.
- for variables in group 5, according to Eq. (7), $h(a_i + r_{i,j}) + h(a_i + r_{i,j} + a_j)$ can be rewritten as $a_j \cdot f(a_i + r_{i,j}, a_j)$. If $i \in \mathcal{S}$, $h(a_i + r_{i,j}) + h(a_i + r_{i,j} + a_j)$ can be simulated with a_i, a_j, and $r_{i,j}$. If $i \notin \mathcal{S}$, then $a_i + r_{i,j}$ does not enter in the computation of any other variables.

Hence, $h(a_i + r_{i,j}) + h(a_i + r_{i,j} + a_j)$ can be simulated with a_j and a fresh random value.

- for each variable in group 6, we consider the different terms. The first term $h(a_i)$ can be simulated with a_i. Then, for the sum of $h(a_j + r_{j,i}) + h(a_j + r_{j,i} + a_i) + h(a_i + r_{j,i})$, we consider two cases. If $j \in \mathcal{S}$, this sum can be perfectly simulated with a_i, a_j and $r_{j,i}$. Otherwise, $r_{j,i}$ does not enter in the computation of other variables. Hence, it can be assigned to a fresh random value.

In order to prove SNI, we still have to simulate the observed output values for rows on which no internal values are observed. Remarking that simulating the i-th line also necessarily fixed the value of all random variables appearing in the i-th column (so that dependencies between variables are preserved). After internal observations are simulated, at most $|\mathcal{I}|$ lines of the matrix are fully filled. Therefore, at least $|\mathcal{O}|$ random values are not yet simulated on lines on which no internal observations are made. For each output observation made on one such line (say i), we can therefore pick a different $r_{i,j}$ that we fix so that output i can be simulated using a fresh random value.

APPENDIX B
PROOF OF MODIFIED ZQZ-2

This proof consists in constructing set $\mathcal{S}$ of indices in $\{0, \cdots, d\}$ of size at most d and perfectly simulate any d-tuple observations $\mathcal{I} \cup \mathcal{O}$ of intermediate variables with the shares $(a_i)_{i \in \mathcal{S}}$. As the shares $(a_i)_{i \in \mathcal{S}}$ are independent of the sensitive variable a, any d-tuple of intermediate variables are independent of a. We now describe the construction of $\mathcal{S}$.

First, we show how to construct the set $\mathcal{S}$. Initially, set $\mathcal{S}$ is empty. We fill it in the following specific order according to the possible leaked intermediate variables.

1) for any observed variables a_i and $h(a_i)$, add i to $\mathcal{S}$.
2) for any observed variable r_j, put j to $\mathcal{S}$.
3) for any intermediate sum occurring during the computation of c_i, assign from shortest sums (in terms of number of terms) to longest sums: if $i \notin \mathcal{S}$, add i to $\mathcal{S}$. Otherwise, if c_i involves corrective terms (i.e., randoms not in $r_{i,j}$), consider them successively (from left to right). For a random of the form $r_{j,i}$, if $j \notin J$, add j to $\mathcal{S}$, otherwise, consider the next random. For a random of r_j, if $j \notin \mathcal{S}$, add j to $\mathcal{S}$. If there are no more corrective terms to consider, or if c_i does not involve corrective terms, consider the involved $t_{i,j}$ in reverse order (from right to left). Add to $\mathcal{S}$ the first index j that is not in $\mathcal{S}$.
4) for any observed variables $r_{i,j}$, $a_i + r_{i,j}$, $h(r_{i,j})$, and $h(a_i + r_{i,j})$: if $i \notin \mathcal{S}$, add i to $\mathcal{S}$, otherwise add j to $\mathcal{S}$.
5) for any observed intermediate sum $t^0_{i,j}$ occurring during the computation of $t_{i,j}$ with at most three terms (no r_{j-1}). If $i \notin \mathcal{S}$, add i to $\mathcal{S}$, otherwise add j to $\mathcal{S}$.
6) for any observed intermediate sum $t^0_{i,j}$ occurring during the computation of $t_{i,j}$ with strictly more than three terms (with r_{j-1}). If $j-1 \notin \mathcal{S}$, add $j-1$ to $\mathcal{S}$. Otherwise, add i to $\mathcal{S}$, otherwise add j to $\mathcal{S}$.

Now that the set $\mathcal{S}$ has been determined, and note that each observation adds at most one index in $\mathcal{S}$. With at most d variables, their cardinals hence cannot be greater than d. Before simulating, the following observations are given,

1) all variables involves $r_{i,j}$ are $t_{i,j}$, c_i, and c_j,
2) all variables involves r_{j-1} are $t_{k,j}$, c_{j-1} and c_k, for any $k \leq j-2$,
3) all variables involves both $r_{i,j}$ and r_{j-1} are c_i and $t_{i,j}$.

Then, we prove that every observed value can be perfectly simulated with the input shares whose indices are among $\mathcal{S}$.

1) any variable in group 1 can be trivially simulated with a_i.
2) any variable in group 4 can be trivially simulated with a_i and $r_{i,j}$.
3) any variable r_j (group 2) is assigned to a fresh random value.
4) for any intermediate (group 5) $t^0_{i,j}$ during the computation of $t_{i,j}$ with at most three terms (including $a_j{+}r_{i,j}{+}a_i$ and $a_j{+}r_{i,j}$): if $j \in \mathcal{S}$, intermediate variables can be perfectly simulated with a_i, a_j and $r_{i,j}$. Otherwise, if $j \notin \mathcal{S}$, we show that these observations can be assigned to a random value (variable $h(a_j + r_{i,j}) + h(a_j + r_{i,j} + a_i)$ can be simulated with a_i and a random value). In particular, we show that if they are non-random, we must have $i, j \in \mathcal{S}$. All those intermediate variables involve $r_{i,j}$. This variable can only appear in intermediate variables of group 4, in c_i, in c_j, in $t^0_{i,j}$ of less than three terms part of $t_{i,j}$, or in $t^0_{i,j}$ of more than three terms part of $t_{i,j}$.
   - $r_{i,j}$ appears in group 4: this probe involved $i \in \mathcal{S}$, and hence the probe in group 5 added j to $\mathcal{S}$.
   - $r_{i,j}$ appears in an observed c_i: this probe involved $i \in \mathcal{S}$, and hence the probe of group 5 added j to $\mathcal{S}$.
   - $r_{i,j}$ appears in an observed c_j: this probe involved $j \in \mathcal{S}$, and hence the probe of group 5 added i to $\mathcal{S}$.
   - $r_{i,j}$ appears in an observed $t^0_{i,j}$ of less than three terms: this probe involved $i \in \mathcal{S}$, and hence the probe of group 5 added j to $\mathcal{S}$.
   - $r_{i,j}$ appears in an observed $t^0_{i,j}$ of strictly more than three terms: in this case, this probe also involves the random r_{j-1}. We know that r_{j-1} can either be observed alone, in c_{j-1}, in $t^0_{i,j}$ of more than three terms part of $t_{k,j}$ or in c_k. Once again, considering r_{j-1}, in c_{j-1}, and $t^0_{i,j}$, we get that $j-1, j, i \in \mathcal{S}$. Considering $t^0_{i,j}$ of more than three terms, or c_k, if $k = i$, we have already treated this case and we have $i, j \in \mathcal{S}$, otherwise, the variable involves $r_{k,j}$. All variables whose expression involves $r_{k,j}$ are: $r_{k,j}$, $t_{k,j}$, c_k and c_j. It can be checked that $i, j, k, j-1$ are in $\mathcal{S}$ for each variables that are not part of c_k or $t_{k,j}$. Consequently, each other probe that does not imply $i, j \in \mathcal{S}$ are variables of these kinds. However, each of these variables involve both r_{j-1} and $r_{k,j}$ for a certain k. To summarize, $t^0_{i,j}$ has been queried, which involves only $r_{i,j}$, and the only other possible variables involve r_{j-1} and $r_{l,j}$, which l is the index of the line. Hence, the parity of the number of occurrences of r_{j-1} is different from the parity of the number of occurrences of $r_{l,j}$. This ensures that it is possible to get rid of r_{j-1} and all variables $r_{l,j}$

at the same time. Therefore, in those cases $t^0_{i,j}$ can be assigned to a random value.

5) if $t^0_{i,j}$ is a sum of strictly more than three terms (group 6):
   - if $i, j, j-1 \in \mathcal{S}$, then t can be simulated with a_i, a_j and random numbers.
   - $t^0_{i,j}$ involves $r_{i,j}$ and r_{j-1}. Observations (1) and (2) provide us the variables in which these randomness are involved. For all but four cases, we trivially have $i, j, j-1 \in \mathcal{S}$. These four cases are the queries of $(r_{i,j}, t^0_{k,j})$ with $t^0_{k,j}$ part of $t_{k,j}$ and involving strictly more than three terms, (c^0_i, c^1_i), where c^0_i and c^1_i are part of c_i, $(t^1_{i,j}, t^0_{k,j})$ with $t^1_{i,j}$ part of $t_{i,j}$ and $t^0_{k,j}$ part of $t_{k,j}$, where $t^0_{k,j}$ is assigned before $t^0_{i,j}$, both involving more than three terms, and finally, any other couple involving a part of c_k.
     - the cases $(r_{i,j}, t^0_{k,j})$ and $(t^1_{i,j}, t^0_{k,j})$ imply the involvement of $r_{k,j}$. Thanks to Observation (1), all possible cases can be exhausted, and we obtain $i, j, j-1 \in \mathcal{S}$.
     - the case (c^0_i, c^1_i) is particular. Indeed, we can assume that c^0_i is computed during the computation of c^1_i. We can hence safely assign $t^0_{i,j}$ to a random variable if this is the only case where $r_{i,j}$ and r_{j-1} have been involved.
     - the query of a c^0_k, part of c_k involving r_{j-1} involves the variable $r_{k,j}$. From Observation (i), we can exhaust the possible cases. For each of these cases except five, we have $i, j, j-1 \in \mathcal{S}$. The five remaining cases are (c_j, c_j), (c_j, c_k), $(r_{k,j}, c_k)$, (c_i, c_k), $(t_{i,j}, c_k)$. With the case involving c_j, by construction we have that $r_{k,j}$ and $r_{i,j}$ appear after the addition of all the terms of the form t_{jl}. Consequently, this expression involves the term $r_{j-1,j}$. Using Observation (i), we find out that the only way not to have $i, j, j-1 \in \mathcal{S}$ is to make another probe to c_j. However, this case is similar to the one we just observed: it is safe to randomly assign $t^0_{i,j}$. For any another case, the random $t_{k,j}$ reappears, and we must hence query another variable to get rid of it. The only possibility is to query c_k once more. Hence $t^0_{i,j}$ can be randomly assigned.

6) for each variable in group 3, we consider the different terms. The first term $h(a_i)$ can be simulated with a_i. For term $t_{i,j}$ with r_{j-1} (more than three terms), if $i, j, j-1 \in \mathcal{S}$, it can be perfectly simulated. Otherwise, it can be assigned to a random value. For term $t_{i,j}$ without r_{j-1} (at most three terms), it can be perfectly simulated with $i, j \in \mathcal{S}$. Otherwise, it can be assigned to a random value. □

References

[1] G. Barthe, S. Belaïd, F. Dupressoir, and P.-A. Fouque et al. Strong Non-Interference and Type-Directed Higher-Order Masking. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 116–129. ACM, 2016.

[2] G. Barthe, S. Belaid, F. Dupressoir, P.-A. Fouque, B. Gregoire, and P.-Y. Strub. Verified Proofs of Higher-Order Masking. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015*, volume 9056 of *LNCS*, pages 457–485. Springer, 2015.

[3] G. Barthe, F. Dupressoir, and B. Grégoire. A Note on 'Further Improving Efficiency of Higher-Order Masking Scheme by Decreasing Randomness Complexity'. Cryptology ePrint Archive, Report 2017/1053 (2017), http://eprint.iacr.org/, 2017.

[4] S. Belaïd, F. Benhamouda, A. Passelègue, and E. Prouff et al. Randomness Complexity of Private Circuits for Multiplication. In M. Fischlin and J.-S. Coron, editors, *EUROCRYPT 2016*, volume 9666 of *LNCS*, pages 616–648. Springer, 2016.

[5] J.-S. Coron, E. Prouff, M. Rivain, and T. Roche. Higher-Order Side Channel Security and Mask Refreshing. In S. Moriai, editor, *FSE 2013*, volume 8424 of *LNCS*, pages 410–424. Springer, 2014.

[6] Y. Ishai, A. Sahai, and D. Wagner. Private Circuits: Securing Hardware against Probing Attacks. In D. Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 463–481. Springer, 2003.

[7] A. Passelègue E. Prouff A. Thillard D. Vergnaud S. Belaïd, F. Benhamouda. Private Multiplication over Finite Fields. In J. Katz and H. Shacham, editors, *CRYPTO 2017*, volume 10403 of *LNCS*, pages 397–426. Springer, 2017.

[8] R. Zhang, S. Qiu, and Y. Zhou. Further Improving Efficiency of Higher-Order Masking Schemes by Decreasing Randomness Complexity. *IEEE Transactions on Information Forensics and Security*, 12(11):2590–2598.