Secure Wire Shuffling in the Probing Model

Jean-Sébastien Coron and Lorenzo Spignoli

University of Luxembourg
jean-sebastien.coron@uni.lu, lorenzo.spignoli@uni.lu

Abstract. In this paper we describe the first improvement of the wire shuffling countermeasure against side-channel attacks described by Ishai, Sahai and Wagner at Crypto 2003. More precisely, we show how to get worst case statistical security against t probes with running time $\mathcal{O}(t)$ instead of $\mathcal{O}(t \log t)$; our construction is also much simpler. Recall that the classical masking countermeasure achieves perfect security but with running time $\mathcal{O}(t^2)$. We also describe a practical implementation for AES that outperforms the masking countermeasure for $t \geq 6\,000$.

1 Introduction

The masking countermeasure. The study of circuits resistant against probing attacks was initiated by Ishai, Sahai and Wagner in [ISW03]. Their construction is based on the masking countermeasure, where each intermediate variable x is shared into $x = x_1 \oplus \cdots \oplus x_n$, and the shares x_i are processed separately. The ISW construction offers perfect security; this means that an adversary with at most $t < n/2$ probes learns nothing about the secret variables. Rivain and Prouff showed in [RP10] how to adapt the ISW construction to AES, by working in $\mathbb{F}_{2^8}$ instead of $\mathbb{F}_2$; in particular, the non-linear part $S(x) = x^{254}$ of the AES SBox can be efficiently evaluated with only 4 non-linear multiplications over $\mathbb{F}_{2^8}$, and a few linear squarings. In the last few years, numerous variants and improvements of the masking countermeasure have been described: for example, high-order evaluation of any SBOX [CGP+12], high-order table re-computation [Cor14], minimization of randomness usage [FPS17] and efficient implementations of high-order masking [JS17, GJRS18].

The main drawback of the masking countermeasure is that the circuit size is quadratic in the maximum number of probes t in the circuit; namely in the ISW construction and its variants every AND gate gets expanded into a gadget of size $\mathcal{O}(t^2)$; hence the initial circuit C gets expanded into a new circuit of size $\mathcal{O}(|C| \cdot t^2)$. One can divide the new circuit into regions corresponding to each gadget, and by appropriate mask refreshing one can let the adversary put t probes per region, instead of t probes in the full circuit; the maximum number of probes then becomes $|C| \cdot t$ instead of t. But the circuit size remains $\mathcal{O}(|C| \cdot t^2)$, that is quadratic in the maximum number of probes t per region.

Statistical security. To improve the previous complexity, the ISW authors introduced a weaker security model with statistical security only [ISW03]. In this model the adversary can still put t probes wherever he wants in the circuit, but he can now learn a secret variable with some non-zero probability (instead of zero probability as in the perfect security model); this probability should be a negligible function of the security parameter k. The authors described a construction in this model with complexity $\mathcal{O}(|C| \cdot t \log t)$ for at most t probes in the circuit. This is only quasi-linear in the number of probes t, so much better than the classical masking countermeasure. In this asymptotic complexity, a factor $\mathsf{poly}(k)$ is actually hidden in the constant, where k is the security parameter; namely, to achieve $2^{-\Omega(k)}$ statistical security, the size of the protected circuit in [ISW03] is actually $\mathcal{O}(|C| \cdot k^{10} \cdot t \log t)$.

The above result holds in the *stateless model*, in which the adversary must put his t probes in a non-adaptive way, that is before the evaluation of the circuit. The authors also considered the more

useful *stateful model*, in which the adversary can move its probes between successive executions of the circuit; however within an execution the model is still non-adaptive. For the stateful model, the authors described a construction with complexity $\mathcal{O}(|C| \cdot t \log t + s \cdot t^3 \log t)$, where s is the number of memory cells in the circuit that must be passed from one execution to the other; for a block-cipher, s would be the number of key bits. Assuming that the circuit size $|C|$ is significantly larger than the key size s, this is again better than the classical masking countermeasure with respect to the number of probes t.

While the masking countermeasure used in the first part of [ISW03] is quite practical and has been widely studied with numerous improvements, the construction in the statistical model, which appears in the second part of [ISW03], has never been investigated up to our knowledge. Our goal in this paper is to describe an improved construction in the statistical model that is better asymptotically and moreover practical, while we argue that the original construction from [ISW03] was essentially unpractical.

The wire shuffling countermeasure from [ISW03]. To achieve the $\mathcal{O}(t \cdot \log t)$ complexity in the statistical model, the ISW paper proceeds in two steps. First, it considers statistical security in the weaker *random probing model*, in which the adversary gets the value of each variable with independent probability p. This is easy to achieve from the classical masking countermeasure. Namely, if we apply the masking countermeasure against $t = k$ probes with $2k + 1$ shares (where k is the security parameter), we get a circuit where each gadget has size at most $c \cdot k^2$ (for some constant c), and secure against k probes per gadget. These k probes per gadget correspond to a fraction $k/(c \cdot k^2) = 1/(c \cdot k)$ of the gadget wires. Hence if we let $p = 1/(10 \cdot c \cdot k)$, then from Chernoff's bound, the probability that in a given gadget the adversary gets more than k probes becomes a negligible function of k; this gives statistical security in the random probing model. Starting from a circuit C, we can therefore obtain an intermediate circuit C' of size $\mathcal{O}(|C| \cdot k^2)$ that is secure in the random probing model with leakage probability $p = \Omega(1/k)$.

In the second step, the ISW paper describes a construction where each wire i of the intermediate circuit C' is expanded into ℓ wires, such that only one of the ℓ wires contains the original signal value v_i from C', while the other wires contain only a dummy value \$; see Figure 1 for an illustration. We call this construction the wire shuffling countermeasure, as it consists in randomly shuffling the position of the signal among those ℓ wires. More precisely, for each execution the position of the signal v_i in the expanded circuit $\tilde{C}$ is selected randomly and independently among the ℓ wires, for each original wire i of C'.

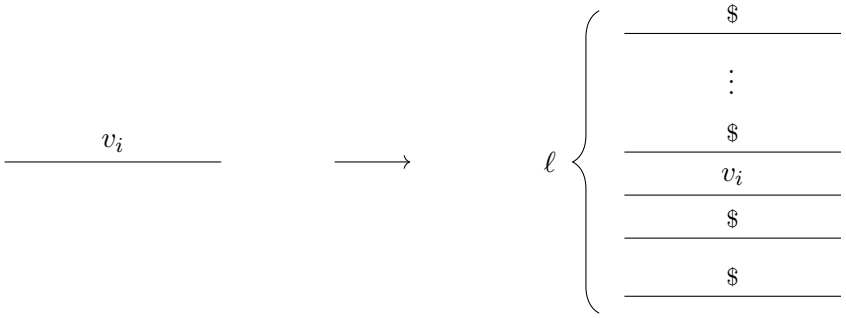

Fig. 1: A wire with signal v_i in C' (left), and the corresponding ℓ wires in $\tilde{C}$ (right); only one of the ℓ wires contain the signal v_i, while the others contain the dummy value \$.

Consider now a gate from the intermediate circuit C'. If the two input wires i and i' from the intermediate circuit C' have their information located at index $j \in [1, \ell]$ and $j' \in [1, \ell]$ in the expanded circuit $\tilde{C}$, one must be able to process the original gate from C' without leaking information on v_i and $v_{i'}$ in the process, except with small probability. One cannot consider all possible index pairs (j, j') as the complexity would be quadratic in ℓ (and eventually quadratic in t). Instead the ISW paper describes a relatively complex construction based on sorting networks with complexity $\mathcal{O}(\ell \log \ell)$; it then proves that with $\ell = \mathcal{O}(t/p^7)$ wires, the probability that each original value v_i is learned by the adversary is at most p. This means that the adversary does not learn more from the worst case probing of the final circuit $\tilde{C}$, than from the p-random probing of the intermediate circuit C'. This implies statistical security for $\tilde{C}$ with circuit size $\mathcal{O}(|C| \cdot t \log t)$, so better than the classical masking countermeasure with complexity $\mathcal{O}(|C| \cdot t^2)$. We stress that for this final circuit $\tilde{C}$, security holds in the worst case probing model as well, where the adversary can freely choose the position of the t probes in the circuit (as opposed to the random probing model where every variable leaks with probability p).

Our contribution. In this paper we describe a construction that achieves worst-case statistical security against t probes in the stateless model with time complexity $\mathcal{O}(|C| \cdot t)$ instead of $\mathcal{O}(|C| \cdot t \log t)$; our construction is also much simpler. Our technique is as follows. As in [ISW03], we randomly shuffle the position of the signal v_i among the ℓ wires, independently for each original wire i of the intermediate circuit C'. However, we now explicitly compute the index position $j_i \in [1, \ell]$ of each signal v_i among the ℓ wires; whereas in ISW this position was only implicitly determined by the value of the ℓ wires, as one of them would contain the signal v_i while the others would get the dummy value \$ (see Fig. 1).

Consider now two wires i and i' in C', for which the signal is located at positions $j \in [1, \ell]$ and $j' \in [1, \ell]$ in the expanded circuit $\tilde{C}$. Since the positions j and j' of the signal are now explicitly computed, we don't need to use a sorting network as in [ISW03] anymore. Instead, we can simply generate a new random index $j'' \in [1, \ell]$, and cyclically shift the information corresponding to wire i by $\Delta = j'' - j$ positions modulo ℓ, and similarly by $\Delta' = j'' - j'$ positions for wire i'. For both inputs the signal is now located at the common position $j + \Delta = j' + \Delta' = j''$, so now the signal will be processed at this position j''. Such cyclic shift can be computed in time $\mathcal{O}(\ell)$ instead of $\mathcal{O}(\ell \log \ell)$, hence we can get statistical security with time complexity $\mathcal{O}(|C| \cdot t)$ instead of $\mathcal{O}(|C| \cdot t \log t)$. Our construction is also much easier to implement in practice, as we can use a simple table look-up for the cyclic shifts, instead of a complex sorting network.

The main difference between our construction and the original ISW is that the index positions of the signal values are now explicitly computed in the final circuit $\tilde{C}$. This means that those index positions can be probed by the adversary, so we may as well assume that the adversary knows all those index positions. Our proof of security crucially relies on the fact that as in [ISW03], the adversary learns those positions only at the evaluation phase, that is *after* he has committed his probes in the circuit. Therefore when the adversary learns the exact locations it is actually too late: we show that he can only learn the signal values with probability at most p. This means that as previously the adversary does not learn more from the worst case probing of the final circuit $\tilde{C}$, than from the p-random probing of the intermediate circuit C'; this gives worst case statistical security for our final circuit $\tilde{C}$.

For the stateful construction we must add some additional countermeasure, because if the adversary knows the position of the signal v_i at the end of one execution, he can directly probe v_i at the beginning of the next execution; this holds for memory cells that must be transmitted from one execution to the next. In ISW this is achieved by using a t-private encoding of a random cyclic shift

for each pack of ℓ wires. Such t-private encoding has complexity $\mathcal{O}(t^2)$, and since for every memory cell this cyclic shift requires a circuit of size $\mathcal{O}(\ell \log \ell)$, the additional complexity is $\mathcal{O}(st^2 \ell \log \ell)$, which gives a complexity $\tilde{\mathcal{O}}(st^3)$ for s memory cells. To get a better complexity we proceed as follows: for each wire i from C' at the end of one execution, we perform a random permutation of the $\ell = \mathcal{O}(t)$ corresponding wires in $\tilde{C}$, but without processing the index location explicitly. For this we use a sequence of $\log_2 \ell$ layers, where in each layer the information in all wires of index j and $j + 2^m$ is randomly swapped, for $0 \leq m < \log_2 \ell$. The complexity is then $\mathcal{O}(s\ell \log \ell) = \mathcal{O}(st \log t)$, and eventually the circuit complexity is $\mathcal{O}(|C| \cdot t \log t)$. We summarize the time and circuit complexities in Table 1. We see that asymptotically in the stateless model our construction improves the time complexity but not the circuit complexity; in the stateful model we improve both the time and circuit complexities.

Finally, we describe an AES implementation of our shuffling countermeasure, which we compare with an AES implementation of the masking countermeasure. In practice our shuffling construction outperforms the masking countermeasure for $t \geq 6\,000$. We provide the source code in [Cor21].

		time complexity (RAM model)	circuit complexity
Stateless model	ISW, Theorem 3	$\mathcal{O}(\lvert C\rvert \cdot t \log t)$	$\mathcal{O}(\lvert C\rvert \cdot t \log t)$
	Theorem 6	$\mathcal{O}(\lvert C\rvert \cdot t)$	$\mathcal{O}(\lvert C\rvert \cdot t \log t)$
Stateful model	ISW, Theorem 8	$\mathcal{O}(\lvert C\rvert \cdot t \log t + s \cdot t^3 \log t)$	$\mathcal{O}(\lvert C\rvert \cdot t \log t + s \cdot t^3 \log t)$
	Theorem 9	$\mathcal{O}(\lvert C\rvert \cdot t + s \cdot t \log t)$	$\mathcal{O}(\lvert C\rvert \cdot t \log t)$

Table 1: Time and circuit complexity of our new construction vs ISW, where s is the number of memory cells that must be passed from one execution to the other.

Software probing model. For a software implementation we will work in the RAM model used in algorithmic analysis; see [MS08, Section 2.2]. In this model, each memory access takes unit time, and every memory cell can store an integer whose bit-size is logarithmic in the input size; for a polynomial-time algorithm, this enables to store array indices in a single cell.

Moreover, in the *software probing model*, we assume that during the execution the adversary can only probe the input address and output value of a RAM cell that is read during a table look-up, but not the content of the internal wires of the circuit implementation of the RAM. This software probing model was already used for example in the high-order table look-up countermeasure from [Cor14]. For simplicity we will still describe our construction in terms of an expanded circuit $\tilde{C}$ as in [ISW03]. For a software implementation our circuit $\tilde{C}$ is therefore augmented with a RAM unit, where the adversary can only probe the input address and the input/output value, but not the internal wires of the RAM. For completeness we also provide in Appendix D a pure circuit description of our countermeasure, with a proof of security in the standard wire probing model.

Related work. In practice, operation shuffling is often used in addition to the masking countermeasure to improve the resistance against side-channel attacks. Operation shuffling consists in randomizing the execution order of the cryptographic blocks when the operations are independent;

we refer to [VMKS12] for a comprehensive study. For example, for AES one can randomize the evaluation of the 16 Sboxes. In [HOM06], the authors describe an 8-bit implementation of first-order masking of AES, combined with SBOX shuffling with a random starting index; the technique was extended to a 32-bit implementation in [THM07]. In [RPD09], the authors investigate the combination of high-order SBOX masking (but with resistance against first-order attacks only) and shuffling by a random permutation. Namely the shuffling prevents a second-order DPA attack against the SBOX masking countermeasure; the authors can then quantify the efficiency of the main attack paths. The authors of [VMKS12] improve the software implementation of random permutation shuffling, with an efficient permutation generator; see also [VML16, Pap18] for a description of shuffling countermeasures with low randomness usage. The main attack against the shuffling countermeasure is the "integrated DPA" introduced in [CCD00]; it consists in summing the signal over a sliding window. If the signal is spread in t positions, the signal will be reduced by a factor $\sqrt{t}$ only, instead of t without the integration; see [VMKS12] for an improved security analysis.

In summary, operation shuffling has been used in numerous previous work to improve the practical resistance of an implementation against side-channel attacks, but not in provable way against t probes for large t. Conversely, the second part of [ISW03] describes a theoretical construction with complexity $\mathcal{O}(t \log t)$ in the statistical model, but it has never been investigated. In this paper, our goal is to describe an improved construction with provable security in the same model, moreover with complexity $\mathcal{O}(t)$ only, and to compare its performance in practice with the classical masking countermeasure.

2 Preliminaries

In this section we first recall the perfect privacy model and the masking-based construction from the first part of [ISW03]. We then recall the statistical security model and the wire shuffling construction from the second part of [ISW03]. For simplicity we first consider stateless circuits only; we will consider stateful circuits in Section 4.

A deterministic circuit C is a directed acyclic graph whose vertices are gates or input/output variables, and whose edges are wires. A randomized circuit is a circuit augmented with gates which have fan-in 0 and output a random bit. The size of a circuit is defined as the number of gates and its depth is the length of the longest path from an input to an output.

2.1 The ISW model for perfect security

In the ISW probing model [ISW03], the adversary is allowed to put at most t probes in the circuit, and must learn nothing from those t probes. For stateless circuits, both inputs and outputs are hidden in every invocation. For this one uses a randomized input *encoder* I and an output *decoder* O; the internal wires of I and O cannot be probed by the adversary.

Definition 1 (Perfect privacy for stateless circuits.). *Let T be an efficiently computable deterministic function mapping a stateless circuit C to a stateless circuit C', and let I, O be as above. We say that (T, I, O) is a t-private stateless transformer if it satisfies:*

1. ***Soundness.*** *The input-output functionality of $O \circ C' \circ I$ (i.e., the iterated application of I, C', O in that order) is indistinguishable from that of C.*
2. ***Privacy.*** *We require that the view of any t-limited adversary, which attacks $O \circ C' \circ I$ by probing at most t wires in C', can be simulated from scratch, i.e. without access to any wire in the circuit. The identity of the probed wires has to be chosen in advance by the adversary.*

2.2 The ISW construction for perfect privacy

We recall the classical ISW construction for achieving perfect privacy. We first consider the stateless model; we then explain how the construction can be adapted to the stateful model in Section 4. For security against t probes, the construction uses a simple secret-sharing scheme with $n = 2t + 1$ shares. The three algorithms Encode, Decode, and Transform are defined as follow:

- Encode I. Each binary input x is mapped to n binary values. First, $n-1$ random bits $r_1, \ldots, r_{n-1}$ are independently generated. The encoding of x is composed by these $n-1$ random values together with $r_n = x \oplus r_1 \oplus \ldots \oplus r_{n-1}$. The circuit I computes the encoding of each input bit independently.
- Decode O. The output returned by $T(C)$ has the form $y_1, \ldots, y_n$. The associated output bit of C computed by O is $y_1 \oplus \ldots \oplus y_n$.
- Transform T. Assume without loss of generality that the original circuit C consists of only XOR and AND gates. The transformed circuit C' maintains the invariant that corresponding to each wire in C will be n wires in C' carrying an n-sharing of the value on that wire of C. More precisely, the circuit C' is obtained by transforming the gates of C as follows.
  For a XOR gate with inputs a, b and output c, let in C' be the corresponding wires $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$. From $c = a \oplus b = \bigoplus_{i=1}^{n} a_i \oplus b_i$, we let $c_i = a_i \oplus b_i$ for $1 \leq i \leq n$.
  Consider an AND gate in C with inputs a, b and output c; we have $c = a \wedge b = \bigoplus_{i,j} a_i b_j$. In the transformation of this gate, intermediate values $z_{i,j}$ for $i \neq j$ are computed. For each $1 \leq i < j \leq n$, $z_{i,j}$ is computed uniformly random, while $z_{j,i}$ is set to $(z_{i,j} \oplus a_i b_j) \oplus a_j b_i$. Now, the output bits $c_1, \ldots, c_n$ in C' are defined to be the sequence $c_i = a_i b_i \oplus \bigoplus_{j \neq i} z_{i,j}$; see Appendix B.1 for an algorithmic description of the AND gadget.

In the transformed circuit $C' = T(C)$, every XOR gate and AND gate in C are therefore expanded to gadgets of size $\mathcal{O}(n)$ and $\mathcal{O}(n^2)$ respectively, and the gadgets in C' are connected in the same way as the gates in C. This completes the description of T.

Theorem 1 (Perfect privacy, stateless model [ISW03]). *The above construction is a perfectly t-private stateless transformer (T, I, O), such that T maps any stateless circuit C of depth d to a randomized stateless circuit of size $\mathcal{O}(|C| \cdot t^2)$ and depth $\mathcal{O}(d \log t)$.*

2.3 The region probing model and t-SNI security

The above privacy result holds in the worst case probing model, where the adversary can freely chose the position of the t probes in the circuit. Alternatively one can consider the weaker random probing model, where each wire leaks with probability p. To prove security in the random probing model, we first need to consider worst-case privacy in the region probing model, where the adversary can put t probes per region [ADF16], instead of t probes in the full circuit. Recall that a circuit C is a directed acyclic graph whose vertices are gates and whose edges are wires. We partition the set of gates of the circuit into a number of regions, and a wire connecting two gates can therefore meet at most two regions.

The region probing model was already considered in [ISW03], with one region per gadget. The authors claimed that security in this model is achieved thanks to the re-randomization property of the outputs of the AND gadget: for each original output bit, the encoded outputs are $(n-1)$-wise independent even given the entire n-encoding of the inputs; this would imply security against a stronger type of adversary who may observe at most t' wires in each gadget, where $t' = \Omega(t)$.

However we argue that this re-randomization property is actually not enough to achieve security in the region probing model: we exhibit in Appendix A a simple counterexample, *i.e.* a gadget achieving the re-randomization property but insecure in the region probing model.

The required property for achieving security in the region probing model is actually the t-SNI notion introduced in [BBD+16]. The authors showed that the notion allows for securely composing masked algorithms; i.e. the t-SNI of a full construction can be proven based on the t-SNI of its component gadgets.

Definition 2 (t-SNI security [BBD+16]). *Let G be a gadget taking as input n shares $(a_i)_{1\leq i\leq n}$ and n shares $(b_i)_{1\leq i\leq n}$, and outputting n shares $(c_i)_{1\leq i\leq n}$. The gadget G is said to be t-SNI secure if for any set of t_1 probed intermediate variables and any subset O of output indices, such that $t_1 + |O| \leq t$, there exist two subsets I and J of input indices which satisfy $|I| \leq t_1$ and $|J| \leq t_1$, such that the t_1 intermediate variables and the output variables $c|_O$ can be perfectly simulated from $a|_I$ and $b|_J$.*

To achieve privacy in the region probing model, we consider the ISW construction from Section 2.2, in which we additionally perform an $(n-1)$-SNI mask refreshing algorithm as inputs of each XOR and AND gadgets. Such mask refreshing can be based on the AND gadget, since as showed in [BBD+16], the AND gadget achieves the $(n-1)$-SNI security property (see Appendix B.2 for a concrete mask refreshing algorithm). We define each region as comprising an AND or XOR gadget, and the mask refreshing of the corresponding output variable, so that each output $z^{(j)}$ is used only once in the next region; see Fig. 2 for an illustration.

Theorem 2 (t-privacy in the region probing model). *Let C be a circuit of fan-out f. Let (I, O, T) be the previous transformer with $n = 2t+1$ shares, where a $(n-1)$-SNI mask refreshing is applied as input of each XOR and AND gadgets. The transformed circuit is t-private secure where the adversary can put at most t probes per regions, each of size $\mathcal{O}(f \cdot t^2)$.*

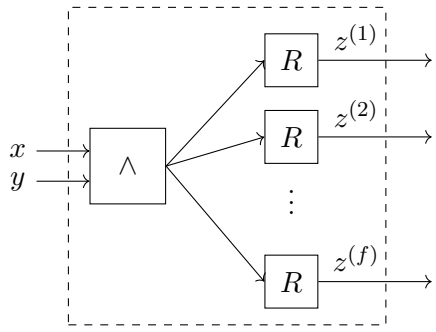

Fig. 2: A region comprises the AND (or XOR) gadget, and the mask refreshing of the output variable.

We provide the proof in Appendix B.4. Note that any circuit C can be converted into a circuit of fan-out $f = 2$; therefore we can always obtain regions of size $\mathcal{O}(t^2)$.

2.4 Security in the random probing model

We recall below the privacy definition when the adversary learns each wire with probability p (average-case security), instead of freely choosing the positions of the probes as above (worst-case security); this is the random probing model [ISW03].

Definition 3 (Random probing model [ISW03]). *A circuit transformer $T = T(C, k)$ is said to be (statistically) p-private in the average case if $C' = T(C, k)$ is statistically private against an adversary which corrupts each wire in C' with independent probability p. That is, the joint distribution of the random set of corrupted wires and the values observed by the adversary can be simulated up to a $k^{-\omega(1)}$ statistical distance.*

From Theorem 2, the ISW circuit transformer from Section 2.2 with $2k + 1$ shares is perfectly private with respect to any adversary corrupting k wires per region. Since each region has size $O(k^2)$, it follows from Chernoff's bound that the view of an adversary corrupting each wire with probability $p = \Omega(1/k)$ can be perfectly simulated, except with negligible failure probability. We provide the proof of Lemma 1 in Appendix C.2.

Lemma 1 (Random probing security [ISW03]). *There exists a circuit transformer $T(C, k)$ producing a circuit C' of size $\mathcal{O}(|C| \cdot k^2)$, such that T is $\Omega(1/k)$-private in the average case.*

2.5 Worst-case statistical security model

The masking countermeasure recalled in Section 2.2 achieves perfect security against t probes with complexity $\mathcal{O}(t^2)$. To obtain a construction with complexity $\mathcal{O}(t \cdot \log t)$ only, the authors of [ISW03] introduced a relaxation of the security model, in which one tolerates a leakage of the secrets, albeit with a negligible probability; this is called the statistical model of security. We stress that with respect to the probes we are still working in the worst case model, in which the adversary can freely chose the position of the t probes (as opposed to the random probing model above in which every wire leaks with probability p). The definition below is similar to the perfect privacy model, except that now the simulation can fail with negligible probability. For this worst-case statistical model, our main goal in this paper is to improve the wire shuffling countermeasure introduced in [ISW03], with a running time $\mathcal{O}(t)$ instead of $\mathcal{O}(t \cdot \log t)$.

Definition 4 (Statistical privacy for stateless circuits). *Let T be an efficiently computable deterministic function mapping a stateless circuit C to a stateless circuit $\tilde{C}$, and let I, O be as above. We say that (T, I, O) is a statistically t-private stateless transformer if it satisfies:*

1. ***Soundness.*** *The input-output functionality of $O \circ \tilde{C} \circ I$ (i.e., the iterated application of I, $\tilde{C}$, O in that order) is indistinguishable from that of C.*
2. ***Privacy.*** *We require that the view of any t-limited adversary, which attacks $O \circ \tilde{C} \circ I$ by probing at most t wires in $\tilde{C}$, can be simulated except with negligible probability. The identity of the probed wires has to be chosen in advance by the adversary.*

2.6 The ISW construction for statistical privacy

We now recall the statistically private construction from [ISW03] that achieves complexity $\mathcal{O}(t \cdot \log t)$. For simplicity we first consider the case of stateless circuits; stateful circuits will be considered in Section 4. The construction proceeds in two steps. First one applies the classical masking countermeasure, namely the circuit transformer $T(C, k)$ guaranteeing p-privacy in the average case, for $p = \Omega(1/k)$ with security parameter k; see Lemma 1 from Section 2.4. Then one transforms its output C' into a larger circuit $\tilde{C}$, where only a fraction of the computation is useful. Namely the circuit $\tilde{C}$ will perform the same computation as C', but only on a small random subset of its wires; the remaining wires of $\tilde{C}$ will contain no useful information for the adversary.

The worst-case probing security of the final circuit $\tilde{C}$ will reduce to the p-random probing security of C' as follows. In the stateless model the adversary must commit its probes before the

circuit evaluation. The subset of useful wires in the final circuit $\tilde{C}$ will be only determined during the invocation of $\tilde{C}$, and therefore it will be independent of the set of corrupted wires. This implies the adversary in $\tilde{C}$ will be able to obtain information about the original wires in C' with small probability only; hence the worst-case probing security of $\tilde{C}$ will follow from the p-random probing security of C'.

Thus, the author's construction transforms the circuit $C' = T(C, k)$ to a circuit $\tilde{C}$ as follows. For each wire i of C' one considers ℓ wires of $\tilde{C}$ labeled $(i, 1), \ldots, (i, \ell)$. Every such wires can take a value from the set $\{0, 1, \$\}$. For every wire i in C' carrying a value $v_i \in \{0, 1\}$, the wires $(i, 1), \ldots, (i, \ell)$ in $\tilde{C}$ will carry the value v_i in a random position (independently of other ℓ-tuples), and the value \$ in the remaining $\ell - 1$ positions; see Figure 1 for an illustration.

Formally we define the Encode' and Decode' algorithms for encoding the input wires and decoding the output wires of the intermediate circuit C'. Note that these algorithms must eventually be composed with Encode and Decode from Section 2.2.

- Encode'. To encode a value v, first generate at random an index $j \leftarrow_{\$} [1, \ell]$ and output an ℓ-tuple in which v will be the j-th element, while the other elements carry a dummy value \$. That is, return $(\$, \ldots, \$, v, \$, \ldots, \$)$, where v is at the j-th position.
- Decode'. Given a ℓ-tuple $(\$, \ldots, \$, v, \$, \ldots, \$)$, return v.

We now describe the transformation applied to every gate of the intermediate circuit C'. Suppose that $v_i = v_{i_1} * v_{i_2}$, i.e., the value of wire i in C' is obtained by applying a boolean operation $*$ to the values of wires i_1, i_2 in C'. Such a gate in C' is replaced with a 2ℓ-input, ℓ-output gadget in $\tilde{C}$. The gadget first puts both values v_{i_1} and v_{i_2} in random but adjacent positions, and then combines them to obtain the value $v_{i_1} * v_{i_2}$ in a randomly defined wire out of the ℓ output ones. For this the gadget makes use of *sorting networks* as a building block. A sorting network is a layered circuit from ℓ integer-valued input wires to ℓ integer-valued output wires, that outputs its input sequence in a sorted order[1]. More technically, the gate is processed as follow:

- **Preprocessing.** Compute $\ell + 1$ uniformly and independently random integers $r, r_1, \ldots, r_\ell$ from the range $[0, 2^k]$, where k is the security parameter. For each $1 \leq j \leq \ell$, use the values $v_{i_1,j}, v_{i_2,j}$ (of wires (i_1, j) and (i_2, j)) to form a pair $(\mathsf{key}_j, \mathsf{val}_j)$ such that:
  1. key_j is set to r_j if $v_{i_1,j} = v_{i_2,j} = \$$ and to r otherwise;
  2. val_j is set to \$ if both $v_{i_1,j}$, $v_{i_2,j}$ are \$; to a bit value b if one of $v_{i_1,j}$,$v_{i_2,j}$ is b and the other is \$, and to $b_1 * b_2$ if $v_{i_1,j} = b_1$ and $v_{i_2,j} = b_2$.
- **Sorting.** A sorting network is applied to the above ℓ-tuple of pairs using key as the sorting key. Let $(u_1, \ldots, u_\ell)$ denote the ℓ-tuple of symbols val_j sorted according to the keys key_j.
- **Postprocessing.** The j^{th} output v_j is obtained by looking at u_j, u_{j+1}, u_{j+2}: if u_j, $u_{j+1} \neq \$$ then $v_j = u_j * u_{j+1}$, if $u_j = u_{j+2} = \$$ and $u_{j+1} \neq \$$ then $v_j = u_{j+1}$, and otherwise $v_j = \$$.

This terminates the description of the construction. The above transformation works because if the input signals v_{i_1} and v_{i_2} are initially located at positions j_1 and j_2 for some $j_1 \neq j_2$, then by definition $\mathsf{key}_{j_1} = \mathsf{key}_{j_2} = r$, and therefore after sorting by key_j the signal values v_{i_1} and v_{i_2} will be contiguous; then at the postprocessing phase the output signal $v_{i_1} * v_{i_2}$ will be computed, and located at some random position j_3. [2] This gadget can be implemented by a circuit of size $\mathcal{O}(k \cdot \ell \log \ell)$.

[1] The authors of [ISW03] use the AKS network [AKS83], which achieves the optimal parameters of $\mathcal{O}(\ell \log \ell)$ size and $\mathcal{O}(\log \ell)$ depth.

[2] The same holds if $j_1 = j_2$.

The following lemma proves the worst-case t-private security of the final circuit $\tilde{C}$, from the p-random probing security of the intermediate circuit C'. A minor difference is that we use $\ell = \mathcal{O}(t/p^7)$ instead of $\ell = \mathcal{O}(t/p^4)$ in [ISW03, Lemma 2]. We claim that this is indeed the correct bound, as it comes from the relative size of the maximal matching of a graph of degree 4, which is at most $1/7$ (and not $1/4$ as used in the proof of [ISW03, Lemma 2], see for example [BDD+04]). Note that this technicality does not change the asymptotic behavior with respect to the number of probes t which is still $\mathcal{O}(t \cdot \log t)$, only the dependence with respect to the security parameter k.

Lemma 2. *Suppose that C' is p-private in the average case. Then the circuit $\tilde{C}$, constructed with $\ell = \mathcal{O}(t/p^7)$, is statistically t-private in the worst case.*

The following theorem proves the worst-case statistical t-privacy of the circuit $\tilde{C}$. It is the same as [ISW03, Theorem 3], except that we make the dependence of the circuit size in the security parameter k more explicit; this is to enable a comparison with our new construction, which has an improved complexity not only with respect to the number of probes t but also with respect to k.

Theorem 3. *There exists a statistically t-private stateless transformer $(\widetilde{T}, \widetilde{I}, \widetilde{O})$, such that $\widetilde{T}(C, k)$ transforms a circuit C to a circuit $\tilde{C}$ of size $\mathcal{O}(|C| \cdot k^{10} \cdot t \cdot (\log k + \log t))$.*

Proof. The worst-case statistical t-privacy of $\tilde{C}$ follows from Lemma 2. The intermediate circuit $C' = T(C, k)$ has complexity $\mathcal{O}(|C| \cdot k^2)$. Then C' is expanded by a factor $\mathcal{O}(k \cdot \ell \log \ell)$; from Lemma 2 and with $p = \Omega(1/k)$, one can take $\ell = \mathcal{O}(t \cdot k^7)$; the expansion factor is therefore $\mathcal{O}(k \cdot (t \cdot k^7) \log(t \cdot k^7)) = \mathcal{O}(k^8 \cdot t \cdot (\log t + \log k))$. The final complexity is therefore $\mathcal{O}(|C| \cdot k^{10} \cdot t \cdot (\log k + \log t))$. □

2.7 Random gate-probing model

For proving the security of our new construction, it will be more efficient to work in a slight variant of the random probing model for the intermediate circuit C', in which we assume that every *gate* of the circuit leaks all its information with probability p, instead of every wire. When a gate is leaking, all its input and output wires are leaked (see Figure 3 for an illustration); we call this variant the *random gate-probing model*. We also assume that the input wires in C' are also leaking with probability p; this is equivalent to considering a "copy gate" applied to each input and also leaking with probability p. Given a circuit C and a set of wires W, we define C_W as the value of the wires in W.

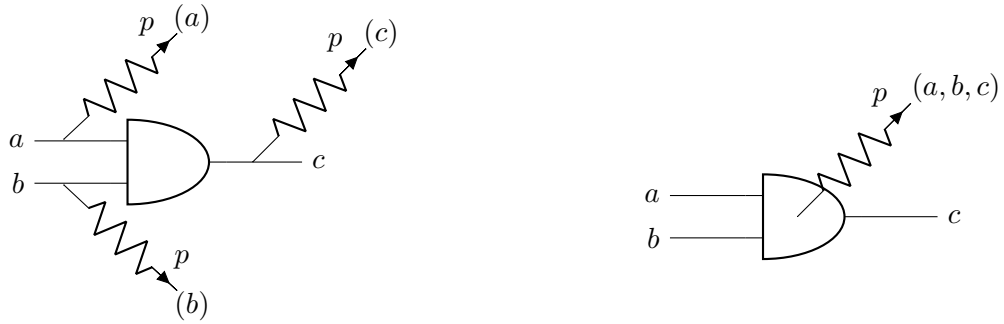

Fig. 3: Random probing model (left) vs random gate-probing model (right).

Definition 5 (Random gate-probing security). *Consider a randomized circuit C' and a random sampling W of its internal wires, where each gate G_i of C' leaks with independent probability p_i. The circuit C' is said (p, ε)-random gate-probing secure if for any $(p_i)_i$ with $p_i \leq p$, there exists a simulator $\mathcal{S}_{C'}$ such that $\mathcal{S}_{C'}(W) \stackrel{id}{=} C'_W(\mathsf{Encode}(\vec{x}))$ for every plain input $\vec{x}$, except with probability at most ε over the sampling of W.*

Note that our above definition is slightly stronger than Definition 3 from [ISW03]. Namely in Definition 3 the simulator produces both the random sampling W and the leaking values, whereas in the above definition the simulator is given W as input and must perfectly simulate the leaking values, except with probability at most ε over the sampling of W. This slightly stronger definition will be more convenient for proving the security of our construction. As in Lemma 1, the masking countermeasure is proven secure in the random gate-probing model via the Chernoff's bound. The proof is essentially the same as the proof of Lemma 1 (see Appendix C.2) and is therefore omitted.

Lemma 3. *There exists a circuit transformer $T(C, k)$ producing a circuit C' of size $\mathcal{O}(k^2|C|)$, such that T achieves $(\Omega(1/k), \varepsilon)$-random gate-probing security, where ε is a negligible function of the security parameter k.*

3 Our new shuffling countermeasure

In this section we describe our new construction that achieves worst-case probing security with running time $\mathcal{O}(t)$ instead of $\mathcal{O}(t \cdot \log t)$ in [ISW03]. For simplicity we consider stateless circuits only; we will consider stateful circuits in Section 4.

3.1 Description

Our construction proceeds in two steps, as in the ISW construction recalled in Section 2.6. First we transform the original circuit C into an intermediate circuit $C' = T(C, k)$ with $n = 2k + 1$ shares, using Theorem 2 from Section 2.3. Then we transform the circuit C' into a circuit $\tilde{C}$ as follows. The main difference with the original ISW construction recalled in Section 2.6 is the usage of a "shuffling index" storing the position of each signal wire from C' in the final circuit $\tilde{C}$.

Wires. For each wire of i of C' we consider ℓ wires of $\tilde{C}$ labeled $(i, 0), \ldots, (i, \ell - 1)$ and an index j_i. Let $a_0, \ldots, a_{\ell-1}$ be the value of the ℓ wires. The circuit $\tilde{C}$ will make the invariant that if wire i in C' has value v, then this value appears at position j_i in $\tilde{C}$, that is $a_{j_i} = v$, while the value of the over wires is arbitrary.

Encoding and decoding. We define the Encode' and Decode' algorithms for encoding the input wires and decoding the output wires of the intermediate circuit C'. As in Section 2.6 these algorithms must be composed with Encode and Decode from the masking countermeasure (Section 2.2). Note that the index position of the signal is computed explicitly; therefore we don't need the dummy element \$ as in Section 2.6 for the $\ell - 1$ other wires, and at the encoding phase we can simply assign them to 0.

- Encode'. To encode a value v, first generate at random an index $j \leftarrow\!\!\$\, [0, \ell - 1]$ and output the encoding $(j, (0, \ldots, 0, v, 0, \ldots, 0))$, where v is at the j-th position.
- Decode'. Given $(j, (a_0, \ldots, a_{\ell-1}))$, return a_j.

Algorithm 1 Gate $*$ processing

Input: Encodings $(j, (a_0, a_1, \ldots, a_{\ell-1}))$ and $(j', (b_0, b_1, \ldots, b_{\ell-1}))$
Output: Index j'' and array $(c_0, c_1, \ldots, c_{\ell-1})$ such that $c_{j''} = a_j * b_{j'}$

1: $j'' \leftarrow_{\$} [0, \ell)$
2: $\Delta = j'' - j$, $\Delta' = j'' - j'$
3: For all $0 \le i < \ell$, let $a'_i \leftarrow a_{i-\Delta}$ and $b'_i \leftarrow b_{i-\Delta'}$ ▷ $a'_{j''} = a_j$, $b'_{j''} = b_{j'}$.
4: For all $0 \le i < \ell$, let $c_i \leftarrow a'_i * b'_i$ ▷ $c_{j''} = a_j * b_{j'}$.
5: **return** $(j'', (c_0, c_1, \ldots, c_{\ell-1}))$

Gates. We consider a gate G in C', taking as input a and b, and outputting $c = a * b$ where $* \in \{\mathsf{XOR}, \mathsf{AND}\}$. We provide a formal description in Algorithm 1 above, where all indices computations are performed modulo ℓ; see also Figure 4 for an illustration. Let $(a_i)_{0 \le i < \ell}$ and $(b_i)_{0 \le i < \ell}$ be the corresponding input wires in $\tilde{C}$, and let j and j' be the corresponding indexes, with $a = a_j$ and $b = b_{j'}$ the signal values in C'. To process the gate in $\tilde{C}$, one generates a random $j'' \leftarrow [0, \ell - 1]$ and then cyclically shifts the ℓ-array (a_i) by $j'' - j$ positions modulo ℓ; similarly the ℓ-array (b_i) is cyclically shifted by $j'' - j'$ positions. The input signals a and b are then located at common position j'', in which the gate G can now be processed; the same gate G is also applied on the other positions that contain arbitrary values; eventually the output signal c is located at position j''.

Finally, a random gate $r \leftarrow \{0, 1\}$ is expanded into a gadget outputting $(j, (r_0, \ldots, r_{\ell-1}))$ with $r_i \leftarrow \{0, 1\}$ for all $0 \le i < \ell$ and $j \leftarrow [0, \ell)$. This terminates the description of the construction.

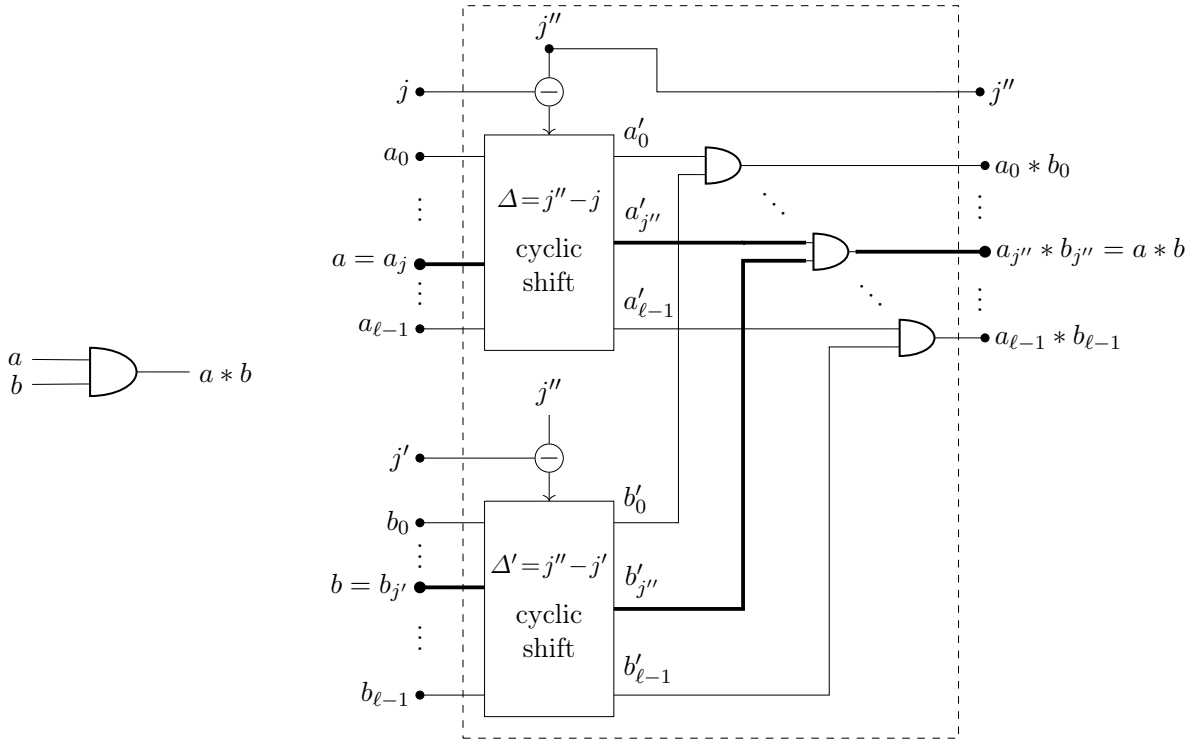

Fig. 4: Original gate in C' (left) and shuffling gadget in $\tilde{C}$ (right). The bold wires contain the original signal value from C'; the other wires contain only dummy values.

Theorem 4. *The transform defined above achieves the soundness property.*

Proof. The intermediate circuit $C' = T(C, k)$ computes the same function as C. Moreover every expanded gate in $\tilde{C}$ computes the same gate as C'. Namely consider the gate $c = a * b$ in C'. In the final circuit $\tilde{C}$ we have $c_{j''} = a'_{j''} * b'_{j''} = a_{j''-\Delta} * b_{j''-\Delta'} = a_j * b_{j'} = a * b = c$ as required. Therefore $\tilde{C} = \tilde{T}(C, k)$ computes the same function as C. □

Note that Algorithm 1 can be implemented via a table look-up. Namely the ℓ wires can be stored in an array $T[i]$ for $0 \leq i < \ell$, with $T[j] = v$ for the signal index j, and the cyclic shift is performed as in Step 3, that is with the loop $T'[i] \leftarrow T[i - \Delta \bmod \ell]$ for each $0 \leq i < \ell$. The running time of Algorithm 1 is $\mathcal{O}(\ell)$ per gadget.

3.2 Shuffling security of a gadget and composition

As in [ISW03], our goal is to show that the adversary does not learn more from the worst-case probing of the final circuit $\tilde{C}$ than from the p-random probing of the intermediate circuit C'. For this we proceed with a similar compositional approach as in [BBD+16]: 1) we introduce a new security definition for a single gadget in the expanded circuit $\tilde{C}$, 2) we prove that our shuffling gadget from the previous section satisfies this definition, and 3) we show how to get security for the full circuit $\tilde{C}$ by composition. The main benefit of this approach is that 3) only depends on 1), therefore we can later modify the shuffling gadget and still get security for the full circuit, as long as the shuffling gadget satisfies the security definition.

Definition 6 (Shuffling security). *We say that a randomized gadget achieves ℓ-shuffling security if any set of t probes, excluding the input wires of the gadget, can be perfectly simulated from scratch, except with probability at most t/ℓ, where the probability is taken over the randomness used by the gadget.*

In the above definition we exclude the probing of the gadget input wires, because in the composition the probing of the input wires of a gadget can be handled by the probing of the output wires of a previous gadget (except for the input wires of $\tilde{C}$ which we will handle separately).

Lemma 4. *The gadget $\tilde{G}$ as described in Algorithm 1 is ℓ-shuffling secure.*

Proof. We must construct a simulator that can simulate any set of t probes, with failure probability at most t/ℓ. In the simulation the input indices j, j' are fixed, as well as the input arrays $(a_0, \ldots, a_{\ell-1})$ and $(b_0, \ldots, b_{\ell-1})$. From the definition the adversary cannot probe those input arrays; in particular, the adversary cannot probe the signal $a = a_j$ and $b = b_{j'}$.

The proof is based on the fact that the adversary must commit to the position of the probes before the execution of the gadget. Therefore in the simulation the probes have fixed positions while the index j'' is randomly and uniformly distributed in $[0, \ell)$. This implies that for a fixed $i \in [0, \ell)$, the variable a'_i contains the secret value a with probability at most $1/\ell$; the same holds for the variables b'_i and c_i. This implies that the t probes can be perfectly simulated, except with probability at most t/ℓ. □

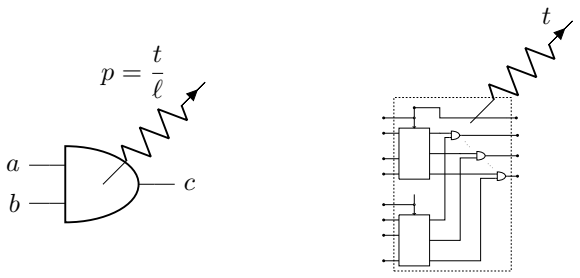

Fig. 5: A set of t probes in a shuffling gadget in $\tilde{C}$ (right) correspond to a gate-leaking probability at most $p = t/\ell$ in C' (left).

Composition. We now prove the worst-case statistical t-privacy of the final circuit $\tilde{C}$, from the p-random probing security of the intermediate circuit C'; see Fig. 5 for an illustration.

Lemma 5. *Suppose that C' is (p, ε)-random gate probing secure. Then, the circuit $\tilde{C}'$ constructed as described above with $\ell := t/p$ achieves ε-statistical security in the worst case against t probes.*

Proof. We must construct a simulator $\tilde{\mathcal{S}}$ for the t probes of the final circuit $\tilde{C}$, using a simulator $\mathcal{S}'$ for the average-case security of the intermediate circuit C'. According to Definition 5, the simulator $\mathcal{S}'$ receives as input a random sampling W of the gates, where each gate leaks with independent probability $p_i \leq p$, and $\mathcal{S}'$ provides a perfect simulation of the gates in W, except with probability at most ε over the sampling W.

Therefore, our simulator $\tilde{\mathcal{S}}$ will proceed as follows. First, we receive all the probes of the adversary on the wires of the final circuit $\tilde{C}$. We then generate all random index positions of the circuit $\tilde{C}$. As illustrated in Figure 5, a set of t_i probes in a shuffling gadget $\tilde{G}_i$ of $\tilde{C}$ corresponds to a leaking probability $p_i = t_i/\ell$ in the corresponding gate G_i of C'. More precisely, according to Lemma 4, each gadget $\tilde{G}_i$ can be perfectly simulated from scratch, except with probability at most t_i/ℓ. Therefore our simulation for the full circuit $\tilde{C}$ will first run the simulators $\mathcal{S}_i$ for all gadgets $\tilde{G}_i$ independently; when any such simulator $\mathcal{S}_i$ fails with probability at most $p_i = t_i/\ell \leq t/\ell = p$, we include the corresponding gate G_i in the sampling W. Moreover, from Definition 6 and Lemma 4, the failure probabilities are independent, because in Definition 6 the failure probability is taken over the gadget randomness, and the randomness in each gadget is generated independently. Therefore the gates G_i are included in W with independent probability $p_i \leq p$ as required in Definition 5. Eventually our simulator $\tilde{\mathcal{S}}$ runs the intermediate circuit simulator $\mathcal{S}'$ with the sampling W, and receives a simulation of the input wires of the corresponding gates $G_i \in W$; it then uses this simulation of the input wires to perfectly simulate the t_i probes within each expanded gadget $\tilde{G}_i$, for the case when $\mathcal{S}_i$ has failed. Since the simulator $\mathcal{S}'$ for the intermediate circuit C' provides a perfect simulation except with probability ε over the sampling W, our final simulator $\tilde{\mathcal{S}}$ provides a perfect simulation of the t probes of $\tilde{C}$ except with probability ε. □

Eventually our construction has better running time $\mathcal{O}(|C| \cdot t)$ but same circuit complexity $\mathcal{O}(|C| \cdot t \cdot \log t)$ as ISW.

Theorem 5. *There exists a statistically t-private stateless transformer $(\widetilde{T}, \widetilde{I}, \widetilde{O})$, such that $\widetilde{T}(C, k)$ transforms a circuit C into a circuit $\tilde{C}$ of running time $\mathcal{O}(|C| \cdot k^3 \cdot t)$ and size $\mathcal{O}(|C| \cdot k^3 \cdot t \cdot (\log k + \log t))$.*

Proof. From Lemma 5, the circuit $\tilde{C}$ achieves statistical privacy in the stateless worst-case model. The intermediate circuit C has size $\mathcal{O}(|C|k^2)$, while the final circuit has running time $\mathcal{O}(|C|k^2\ell)$

and size $\mathcal{O}(|C|k^2\ell\log\ell)$. With $p = t/\ell$ and $p = \Omega(1/k)$ to achieve average-case privacy for C', we get running time $\mathcal{O}(|C| \cdot k^3 \cdot t)$. and size $\mathcal{O}(|C| \cdot k^3 \cdot t \cdot (\log k + \log t))$. □

3.3 Improved time complexity

From the proof of Lemma 5 the previous circuit $\tilde{C}$ is actually secure in the region probing model where the adversary can put t probes per gadget in $\tilde{C}$. We can however further optimize the circuit complexity if we only require security against a total of t probes in the full circuit $\tilde{C}$. Namely in that case we can consider that each gadget of $\tilde{C}$ has t_i probes with the condition $\sum_i t_i \leq t$, instead of $t_i \leq t$ for all i in the proof of Lemma 5. This means that for each corresponding gate in the intermediate circuit C', we can consider a leakage probability $p_i = t_i/\ell$ such that $\sum_i p_i \leq \mu$ over the full circuit C', with $\mu = t/\ell$. Note that this is a much looser condition than in Definition 5, where we required $p_i \leq p = t/\ell$ for all gates. In particular, if we take $t/\ell > 1$, we can tolerate a leakage probability $p_i = 1$ for a fraction of the gates in C', as long as $\sum_i p_i \leq t/\ell$ over all gates of C'; see Fig. 6 for an illustration. To handle this looser condition, we modify the definition of random gate probing security as follows.

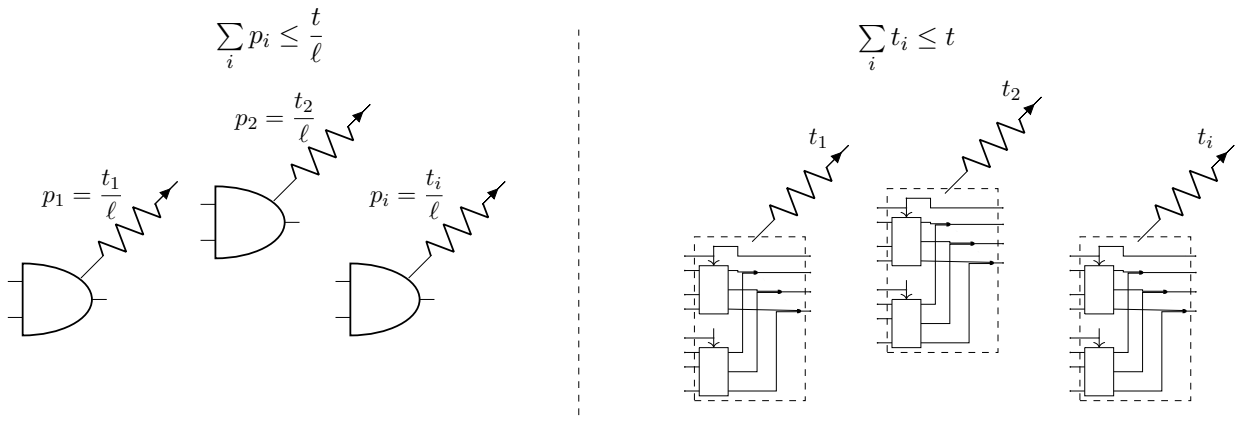

Fig. 6: A total of t probes in the final circuit $\tilde{C}$ (right) corresponds to a total of leaking probabilities at most t/ℓ in the intermediate circuit C' (left).

Definition 7 (Random Σ-gate-probing security). *Consider a randomized circuit C' and a random sampling W of its internal wires, where each gate G_i of C' leaks with independent probability p_i. The circuit C' is said (μ, ε)-random Σ-gate-probing secure if for any $(p_i)_i$ with $\sum_i p_i \leq \mu$, there exists a simulator $\mathcal{S}_{C'}$ such that $\mathcal{S}_{C'}(W) \overset{id}{=} C'_W(\mathsf{Encode}(\vec{x}))$ for every plain input $\vec{x}$, except with probability at most ε over the sampling of W.*

Note that here $\sum_i p_i$ is the average number of leaking gates in the circuit C'. Thanks to the looser condition $\sum_i p_i \leq \mu$, when applying Chernoff's bound on the intermediate circuit $C' = T(C, k)$ secure against k probes, we can prove random Σ-gate-probing security with $\mu = \Omega(k)$ instead of $\Omega(1/k)$. Since we are interested in a practical implementation of our countermeasure (see Section 6), we now provide concrete values for $\mu(k)$ and $\varepsilon(k)$ for the intermediate circuit C' based on the masking countermeasure; we provide the proof in Appendix C.3.

Lemma 6. *There exists a circuit transformer $T(C,k)$ producing a circuit C' of size $\mathcal{O}(k^2|C|)$, such that T achieves (μ,ε)-random Σ-gate-probing security for $\mu=\Omega(k)$ and ε a negligible function of the security parameter k. In particular, one can take $\mu=k/4$ and $\varepsilon=2^{-k/(12\log 2)}$.*

Note that the above circuit C' does not need to be secure in the region probing model with k probes per region; namely in the proof of Lemma 6 only the total number of probes matters. Therefore we can use $n=k+1$ shares with appropriate mask refreshing as in [BBD+16] (instead of $n=2k+1$). Eventually, we obtain a statistically t-private stateless transformer with complexity $\mathcal{O}(|C|\cdot k\cdot t)$, instead of $\mathcal{O}(|C|\cdot k^3\cdot t)$ in Theorem 5; as previously, we proceed by first proving the worst-case security of $\tilde{C}$ from the average-case security of C'.

Lemma 7. *Suppose that C' is (μ,ε)-random Σ-gate-probing secure. Then, the circuit $\tilde{C}$ constructed as described above with $\ell:=t/\mu$ achieves ε-statistical security in the worst case against t probes.*

Proof. The proof is essentially the same as the proof of Lemma 5. Instead of having each gadget $\tilde{G}_i$ simulator $\mathcal{S}_i$ fail with probability $p_i\le p$, the failure probabilities are still independent but with the looser condition $\sum_i p_i\le\mu=t/\ell$. This gives a sampling W of the gates G_i in C' with the same condition $\sum_i p_i\le\mu$. Since C' is (μ,ε)-random Σ-gate-probing secure, we obtain that $\tilde{C}$ achieves ε-statistical security in the worst case against t probes. □

Theorem 6. *There exists a statistically t-private stateless transformer $(\widetilde{T},\widetilde{I},\widetilde{O})$, such that $\widetilde{T}(C,k)$ transforms a circuit C into a circuit $\tilde{C}$ of running time $\mathcal{O}(|C|\cdot k\cdot t)$.*

Proof. From Lemma 7, the circuit $\tilde{C}$ achieves worst-case statistical t privacy in the stateless model. Its running time is $\mathcal{O}(|C|\cdot k^2\cdot\ell)$. With $\ell=t/\mu$ and $\mu=k/4$, the running time is $\mathcal{O}(|C|\cdot k\cdot t)$. □

Note that with running time $\mathcal{O}(|C|\cdot k\cdot t)$ instead of $\mathcal{O}(|C|\cdot k^{10}\cdot t\cdot(\log k+\log t))$ for ISW (see Theorem 3), our construction has an improved complexity also with respect to the security parameter k. In Section 6 we describe an implementation for AES that is practical for large t compared to the masking countermeasure, while the original ISW would be completely unpractical.

3.4 Pure circuit description

The construction described in Section 3.1 can be implemented using table look-ups and is secure in the software probing model, where the adversary can only probe the input address and input/output value of a RAM cell, but not the content of the internal wires of the circuit implementation of the RAM (see Section 1). However it is easy to obtain a pure circuit implementation of the construction, using a circuit implementation of a cyclic shift, with complexity $\mathcal{O}(\ell\cdot\log\ell)$. The construction achieves worst-case statistical privacy with complexity $\mathcal{O}(t\log t)$, as the original ISW construction. We refer to Appendix D for the description and security proof.

4 Statistical security in the stateful model

In this section we consider the more useful stateful model, in which the adversary can move its probes between successive executions of the circuit. We first recall the ISW construction for worst-case statistical security in the stateful model; we will describe our improved construction in Section 5.

A *stateful* circuit is a circuit augmented with memory cells. A memory cell is a stateful gate with fan-in 1: on any invocation the gate outputs its previous input, and stores the current input for the next invocation. We denote by $C[s_0]$ the circuit C with memory cells initialized with the

initial state s_0. A stateful circuit can also have external input and output wires. For example, for a block-cipher, the secret key is stored in the memory cells, while the input wires receive the plaintext, and the output wires produce the ciphertext.

4.1 Perfect privacy for stateful circuits

We recall the perfect privacy definition from [ISW03]. In the stateful case, we consider the circuit inputs and outputs as public; only the internal state is kept private. The adversary can now access the transformed circuit and invoke it multiple times, choosing freely the new invocation inputs; the adversary may choose the next input based on what it has observed in the previous execution.

Definition 8 (Perfect privacy for stateful circuits.). *Let T be an efficiently computable randomized algorithm mapping a stateful circuit C along with an initial state s_0 to a stateful circuit C' along with an initial state s'_0. We say that T is a t-private stateful transformer if it satisfies:*

1. ***Soundness.*** *The input-output functionality of C initialized with s_0 is indistinguishable from that of C' initialized with s'_0. This should hold for any sequence of invocations on an arbitrary sequence of inputs. In other words, $C[s_0]$ and $C'[s'_0]$ are indistinguishable to an interactive distinguisher.*
2. ***Privacy.*** *We require that C' be private against a t-limited interactive adversary. Specifically, the adversary is given access to C' initialized with s'_0 as its internal state. Then, the adversary may invoke C' multiple times, adaptively choosing the inputs based on the observed outputs. Prior to each invocation, the adversary may fix an arbitrary set of t internal wires to which it will gain access in that invocation. To define privacy against such a t-limited adversary, we require the existence of a simulator which can simulate the adversary's view using only a black-box access to C', i.e., without having access to any internal wires.*

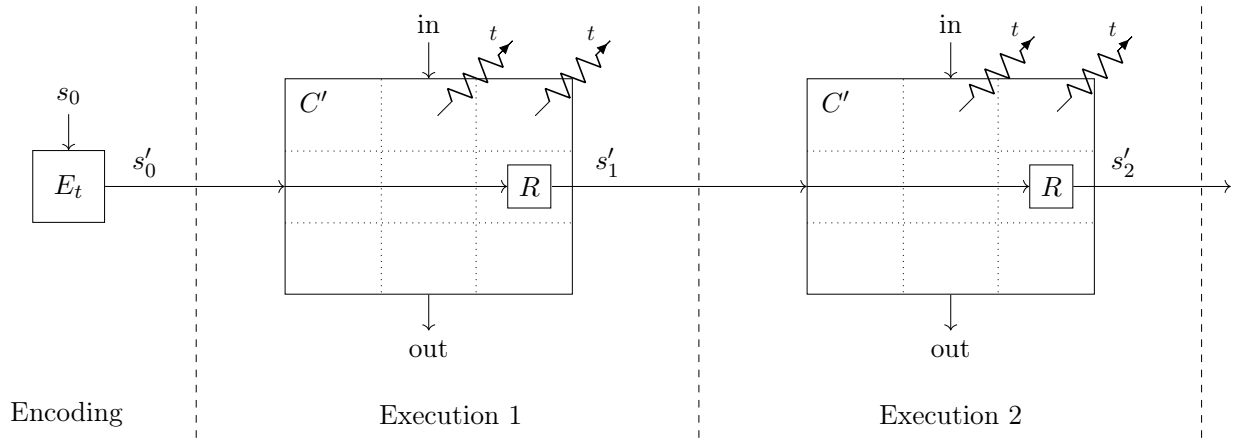

Fig. 7: Illustration of the stateful model. The initial encoding s'_0 used in the first execution gets refreshed into s'_1 before getting passed to the next execution. The adversary can put t probes per region within each execution, where the position of the probes can be changed between executions.

The ISW construction for perfect privacy in the stateful model proceeds as follows; see Figure 7 for an illustration. We use the stateless transformer $T(C, t)$ secure in the region probing model, with t probes per region. Let denote by $E_t(x)$ the encoding used by the stateless transformer, where x is

the input being encoded. The initial state s_0 of C is encoded as $s'_0 = E_t(s_0)$.[3] At the i-th invocation, the circuit $C' = T(C, t)$ takes as input an encoded state s'_i and outputs an encoded state s'_{i+1} that is passed to the next circuit execution. Note that the encoded state s'_i must be refreshed after each execution; otherwise the adversary could recover the internal state by probing the encoded state t probes at a time; in the circuit C' this is done by using a $(n-1)$-SNI mask refreshing R as output. For a block-cipher, the internal state corresponds to the key whose encoding must be refreshed after each execution. The regular input in of C is unprotected, and need not be encoded before getting fed into C'; for each execution of C', this input is first encoded using E_t and the output out is decoded using the corresponding D_t. This implies that these inputs and outputs are known to the adversary, so that they can be given for free to the simulator.

Perfect privacy in the stateful model follows from perfect privacy in the stateless case, thanks to the region probing model. Namely, a sequence of invocations of the stateful circuit C' can be unwound into a larger stateless circuit C''; in the unwound circuit, the adversary can corrupt up to t wires in each region of each circuit produced by the stateless transformation. This means that every new circuit execution corresponds to adding more regions in the unwound circuit (see Figure 7). However the adversary can move its probes between circuit executions; therefore in the unwound circuit C'', the probes corresponding to the i-th execution must be simulated without knowing the position of the probes from the $(i+1)$-th execution. To perform these successive simulations we must consider a slightly stronger definition than t-SNI security for mask refreshing between successive executions, where in a given gadget the set of input variables I that must be known for the simulation, does not depend on the set of output variables O to be simulated; we refer to Appendix E.1 for the definition, and a proof that the AND-based mask refreshing algorithm satisfies this stronger definition. Finally we prove in Appendix E.2 the following theorem, similar to [ISW03, Theorem 2], except that we can tolerate t probes per region for each circuit execution (instead of t probes per execution).

Theorem 7 (Perfect privacy, region stateful model). *There exists a perfectly t-private stateful circuit transformer secure in the region probing model which maps any stateful circuit C of size $|C|$ and depth d to a randomized stateful circuit of size $\mathcal{O}(|C| \cdot t^2)$ and depth $\mathcal{O}(d \log t)$.*

4.2 Worst-case statistical privacy in the stateful model and the ISW construction

The definition of worst-case statistical privacy in the stateful model is the same as for perfect privacy, except that the simulator can now fail with negligible probability ε; we recall the definition in Appendix E.3. We now recall the ISW construction. As for the stateless case, one proceeds in two steps, with each wire in the intermediate circuit C' being expanded into ℓ wires in the final circuit $\tilde{C}$, such that only one of the ℓ wires contains the original signal v_i from C', while the other wires contain only the dummy value \$.

However for the stateful construction we must add some additional countermeasure, because the adversary can move its probes between executions, and therefore could accumulate knowledge about the locations of the signal in $\tilde{C}$. This is easy to see in our stateless construction from Section 3.1: since the adversary can probe the index location j of a signal v at the end of an execution, he could directly probe v at the beginning of the next execution; this holds for memory cells that must be transmitted from one execution to the next; see Figure 8 for an illustration. In ISW this is prevented by using a perfectly t-private encoding of a random cyclic shift for each pack of ℓ wires, for each signal v_i from C' that must be transmitted from one execution to the other. Such t-private

[3] Here we can use E_t instead of E_{2t} in [ISW03] because we consider a circuit C' already secure in the region probing model.

encoding has complexity $\mathcal{O}(t^2)$, and since for every memory cell this cyclic shift requires a circuit of size $\mathcal{O}(\ell \log \ell)$, the additional complexity is $\mathcal{O}(st^2 \ell \log \ell)$, which gives a complexity $\tilde{\mathcal{O}}(st^3)$ for s memory cells. We recall below the theorem from [ISW03].

Theorem 8 (Worst-case statistical privacy, stateful model). *There exists a statistically t-private stateful transformer $\widetilde{T}$, such that $\widetilde{T}(C, k)$ maps a circuit C with s memory cells to a circuit $\tilde{C}$ of size $\mathcal{O}(|C| \cdot t \log t + s \cdot t^3 \log t)$. The depth of $\tilde{C}$ is the same as that of C, up to* polylog *factors.*

5 Our construction in the statistical stateful model

In this section we describe two constructions in the worst-case statistical stateful model that achieve a better complexity bound than the ISW construction recalled in the previous section. Recall that in our stateless construction from Section 3.1, the position $j \in [0, \ell - 1]$ of the signal v_i among the ℓ wires is explicitly computed; we can therefore assume that it is known to the adversary at the end of a given execution. For the stateful model, this means that without any additional countermeasure, the adversary could directly probe the signal v_i at the beginning of the next execution; this holds for the hidden state that must be transmitted from one execution to the other (see Fig. 8 for an illustration).

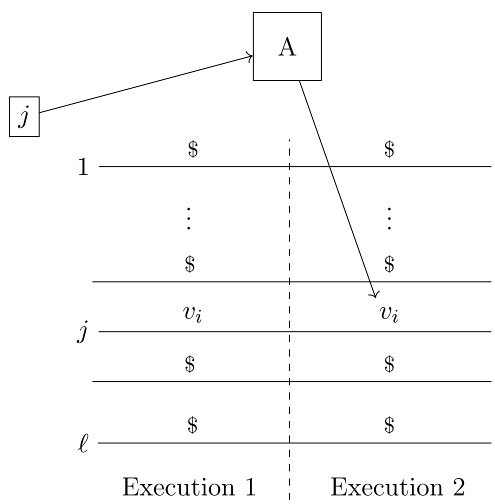

Fig. 8: Without additional countermeasure, the adversary learning the index position j at the end of a given execution can directly probe the signal v_i at the beginning of the next.

To handle the adaptive case of the stateful model, we extend Definition 6 by requiring that even after having observed t probes in the gadget, any set O of output probes can be simulated from scratch, except with probability at most $2t/\ell$. Note that the construction of Fig. 4 from the stateless case cannot satisfy this definition, since the adversary could directly probe the output position j'' of the signal.

Definition 9 (Strong ℓ-shuffling security). *We say that a randomized gate G achieves strong ℓ-shuffling security if any set S of t probes (excluding the input wires) and any set O of output probes, can be perfectly simulated from scratch, except with probability at most $2t/\ell$, where the set O is chosen adaptively from the value of the probes in S.*

As recalled in the previous section the authors of [ISW03] used a perfectly t-private random cyclic shift; this construction satisfies Definition 9, since from the t-privacy the adversary gets no information about the position of the output signal, and therefore for a total of $2t$ probes the probability to recover the signal is at most $2t/\ell$; the complexity of the ISW construction for a single gadget is $\tilde{O}(\ell t^2)$.

In the following we describe two improved constructions achieving the strong ℓ-shuffling security defined above. We then show that any construction satisfying Definition 9 enables to obtain worst-case statistical security in the stateful model.

5.1 First construction: iterated cyclic shifts

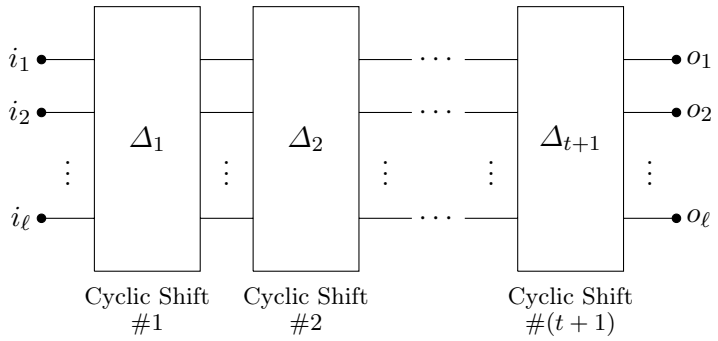

Fig. 9: First construction: sequence of $t+1$ random cyclic shifts.

Our first construction consists of a sequence of $t+1$ random cyclic shifts with uniformly and independently distributed shifts $\Delta_1, \ldots, \Delta_{t+1} \leftarrow [0, \ell - 1]$; see Figure 9 for an illustration. As in [ISW03], the construction is used as output for every hidden state bit that must be transmitted from one execution to the next. As opposed to our stateless construction described in Fig. 4, the index position of the signal is not explicitly computed in the cyclic shifts; the position of the signal is only implicitly determined by the value of the wires in $\{0, 1, \$\}$, where $\$$ is the dummy value. At the end of a given execution, we must therefore convert from a representation with explicit signal index j to a representation with wire values in $\{0, 1, \$\}$; this can be done with complexity $\mathcal{O}(\ell)$. The index position of the signal is computed again explicitly as the beginning of the next execution.

The construction satisfies the strong ℓ-shuffling security (Definition 9), since with at most t internal probes, one of the $t+1$ random cyclic shifts is not probed, and therefore the adversary does not get information about the position of the output signal. We refer to Appendix E.4 for the proof of the following lemma. The cost of this first construction is $\tilde{O}(\ell t)$, instead of $\tilde{O}(\ell t^2)$ for the ISW construction with the perfectly t-private random cyclic shift.

Lemma 8. *The gadget described above is strong ℓ-shuffling secure.*

5.2 Second construction: randomizing network

Our second construction consists of a network of $\log_2 \ell$ layers, where in the m-th layer for $0 \leq m < \log_2 \ell$ the information in all wires of index i and $i + 2^m$ is swapped with independent probability $1/2$; see Fig. 10 for an illustration; note that for simplicity we assume that ℓ is a power of 2. Letting

$j' \in \{0, \ldots, \ell - 1\}$ be the index position of the signal before the randomizing network, at layer m the m-th bit of the signal position is therefore randomly flipped. Since this is done for all layers $0 \leq m < \log_2 \ell$, at the end the output index of the signal is randomly distributed in $\{0, \ldots, \ell - 1\}$.

As in the previous construction, the index position j' is only known as input and not computed explicitly during the swaps: the position of the signal is only implicitly determined by the value of the wires in $\{0, 1, \$\}$; the index position of the signal is computed again explicitly as the beginning of the next execution; see Fig. 11 for an illustration. For a single gadget, our second construction has complexity $\tilde{O}(\ell)$, instead of $\tilde{O}(\ell t)$ in our first construction and $\tilde{O}(\ell \cdot t^2)$ in [ISW03]. Moreover, our second construction has depth polylogarithmic in t, instead of linear in t in our first construction.

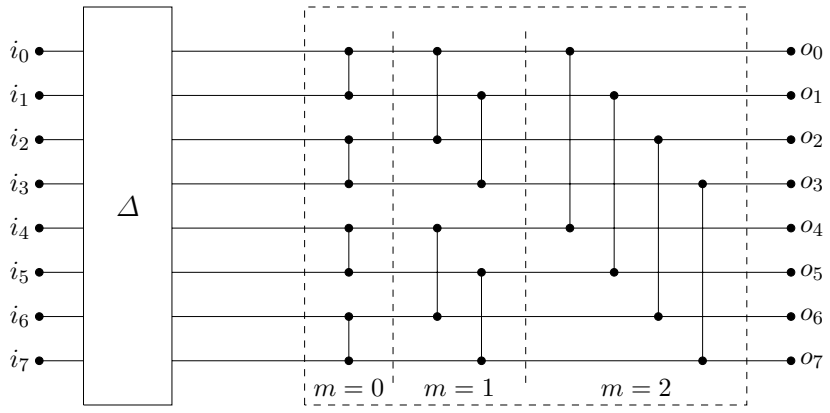

Fig. 10: Second construction: random cyclic shift and randomizing network for $\ell = 8$.

Finally, to satisfy the strong ℓ-shuffling security (Definition 9), we must prepend a random cyclic shift; otherwise, since in Definition 9 the input index j is fixed, the adversary could directly probe the j-th wire after the first layer, and learn the signal with probability 1/2. We provide the proof of Lemma 9 in Appendix E.5.

Lemma 9. *The gadget described above is strong ℓ-shuffling secure, with circuit complexity $\mathcal{O}(\ell \cdot \log \ell)$.*

5.3 Composition in the statistical stateful model

As in the stateless case, we show that the worst-case statistical privacy of $\tilde{C}$ in the stateful model follows from the p-random gate-probing security of C', based on the ℓ-shuffling security (Definition 6) and strong ℓ-shuffling security (Definition 9) of the gadgets. We provide the proof in Appendix E.6.

Lemma 10. *Suppose that C' is (p, ε)-random gate probing secure. Then, the circuit $\tilde{C}$ constructed as described above with $\ell := 3t/(2p)$ achieves stateful ε-statistical security in the worst case against t probes per execution.*

Eventually, thanks to the randomizing network construction with complexity $\mathcal{O}(\ell \log \ell)$, the complexity of the final circuit $\tilde{C}$ is $\mathcal{O}(|C| \cdot t \log t)$ instead of $\mathcal{O}(|C| t \log t + s \cdot t^3)$ in [ISW03]. Therefore as opposed to ISW our construction has quasi-linear complexity even for a large number s of memory cells. Moreover the construction applies without the RAM model as well, see Appendix D.

Theorem 9. *There exists a statistically t-private stateful transformer $\widetilde{T}$, such that $\widetilde{T}(C, k)$ maps a circuit C with s memory cells to a circuit $\tilde{C}$ with complexity $\mathcal{O}(|C| \cdot t \cdot \log t)$. The depth of $\tilde{C}$ is the same as that of C, up to* polylog *factors.*

Proof. From Lemma 10, the circuit $\tilde{C}$ achieves statistical privacy in the stateful worst-case model, with circuit complexity $\mathcal{O}(|C| \cdot \ell \log \ell)$. With $\ell = \mathcal{O}(t)$, the circuit complexity is finally $\mathcal{O}(|C| \cdot t \cdot \log t)$. □

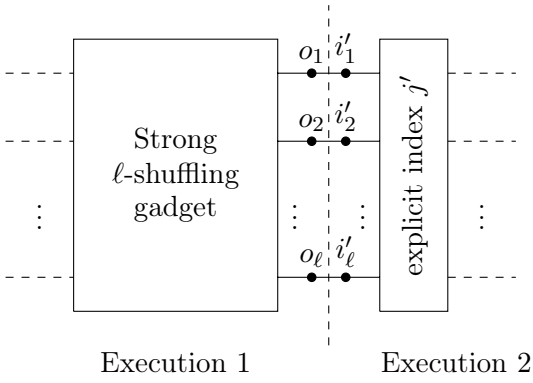

Fig. 11: Thanks to a strong ℓ-shuffling gadget, the adversary does not get information about the index position of the signal at the end of an execution.

6 Implementation

Security parameters. We consider the implementation of our stateless construction from Section 3.1 under the model from Section 3.3, that is worst-case statistical security against a total of t probes in the circuit. Recall that the construction proceeds in two steps. Starting from the original circuit C, we first construct an intermediate circuit C' based on the classical masking countermeasure with perfect security against k probes, where k is the security parameter. From Chernoff bound, the intermediate circuit C' is also secure in the random probing model; more precisely, according to Lemma 6, the circuit C' achieves the (μ, ε)-random Σ-gate-probing security with parameters $\mu(k) = k/4$ and $\varepsilon(k) = 2^{-k/(12 \log 2)}$. Here $\mu(k) = k/4$ denotes the number of gates that are probed on average, and $\varepsilon(k) = 2^{-k/(12 \log 2)}$ the probability of simulation failure (for the unlucky case when the number of leaking gates in C' is too large). Therefore, to get $\varepsilon = 2^{-80}$ security, we must fix $k = 668$. For the intermediate circuit C' we use $n = k + 1$ shares with appropriate mask refreshing, as in [BBD+16].

In the second step, every wire from the intermediate circuit C' must be expanded into ℓ wires in the final circuit $\tilde{C}$, where according to Lemma 7 we must take $\ell = \lceil t/\mu \rceil = \lceil 4t/k \rceil$ to get security against t probes. This implies that we get security against $t = k \cdot \ell/4$ probes as a function of ℓ. For $\varepsilon = 2^{-80}$ and $k = 668$, this gives security against $t = 167 \cdot \ell$ probes as a function of the parameter ℓ. [4] Since the running time of our construction is $\mathcal{O}(\ell)$, the running time is $\mathcal{O}(t)$ for security against t probes, instead of $\mathcal{O}(t^2)$ for the masking countermeasure.

[4] We see that it would not make sense to use $\ell \leq 4$, since the intermediate circuit C' already provides perfect security against $k = 668$ probes.

Number of operations. We compare the concrete number of operations between the masking countermeasure and our construction. For simplicity we consider a single AND gadget. From the gadget description in Appendix B.1, the AND gadget in the intermediate circuit C' performs a total of $n \cdot (7n-5)/2$ operations, with $n = t+1$ shares for perfect security against t probes. This includes $n \cdot (n-1)/2$ random generations, and $n \cdot (3n-2)$ boolean operations. This gives $N_m = (t+1) \cdot (7t+2)/2 \simeq 7t^2/2$ operations as a function of the maximum number of probes t.

We now consider the circuit $\tilde{C}$ corresponding to the expansion of the AND gadget in C'. Every random generation in the intermediate circuit C' requires $\ell+1$ operations in $\tilde{C}$. From Algorithm 1, every boolean operation in C' requires $5\ell+3$ operations in $\tilde{C}$. The total number of operations is therefore:

$$N_s = n \cdot (n-1)/2 \cdot (\ell+1) + n \cdot (3n-2) \cdot (5\ell+3) \simeq \frac{31}{2} \cdot n^2 \cdot \ell$$

With $n = k+1$ and $\ell = 4t/k$, we get $N_s \simeq 62 \cdot k \cdot t$. Finally, with $k = 668$, the number of operations is therefore $N_s \simeq 41\,416 \cdot t$ for worst-case security against t probes. We refer to Table 2 for a summary of the operation count.

Since the masking countermeasure has complexity $7t^2/2$ and our shuffling countermeasure has complexity $41\,416 \cdot t$, the two countermeasures have equal complexity for $7t^2/2 = 41\,416 \cdot t$, which gives $t \simeq 12 \cdot 10^3$. Therefore we expect our shuffling countermeasure to beat the masking countermeasure for a number of probes $t \geq 12 \cdot 10^3$.

	Masking countermeasure	Shuffling countermeasure
#rand	$n^2/2$	$\frac{1}{2} \cdot n^2 \cdot \ell$
#bool	$3n^2$	$15n^2 \cdot \ell$
#op	$7n^2/2$	$\frac{31}{2} \cdot n^2 \cdot \ell$
#op	$\mathbf{7t^2/2}$	$\mathbf{41\,416 \cdot t}$

Table 2: Number of operations for worst-case security against t probes, where $n = t+1$ for the masking countermeasure, and $n = k+1$ and $\ell = 4t/k$ for the shuffling countermeasure, with $k = 668$; we only keep the high-order terms.

AES implementation. We have performed an AES implementation of our shuffling countermeasure, which we compare with an AES implementation of the masking countermeasure, using the same parameters as above. For our shuffling construction we use the following optimization: in the implementation of SecMult (see Algorithm 2 in Appendix B.1), when accumulating $c_i \leftarrow c_i \oplus r$ (lines 7 and 9), for a given index i we always use the same signal position j_i among the ℓ wires. This is because the SecMult algorithm would remain secure in an extended model of security where the adversary could obtain all successive values of the c_i variables with a single probe.We summarize the timings in Table 3; see also Fig. 12. We see that our shuffling construction outperforms the masking countermeasure for a number of probes $t \geq 6\,000$, with a running time of approximately 2 minutes for $t \simeq 6\,000$. We provide the source code in [Cor21].

t	668	2004	3340	4676	6012	7348	8684	10020
Masking (s)	1.4	12	34	70	111	187	235	310
Shuffling (s)	52	63	78	91	102	119	134	141

Table 3: Running time of AES implementation, as a function of the number of probes t. We use $n = t+1$ for the masking countermeasure, and $\ell = 4t/k$ for the shuffling countermeasure. Implementation on a 3,2 GHz Intel processor, running on a single core.

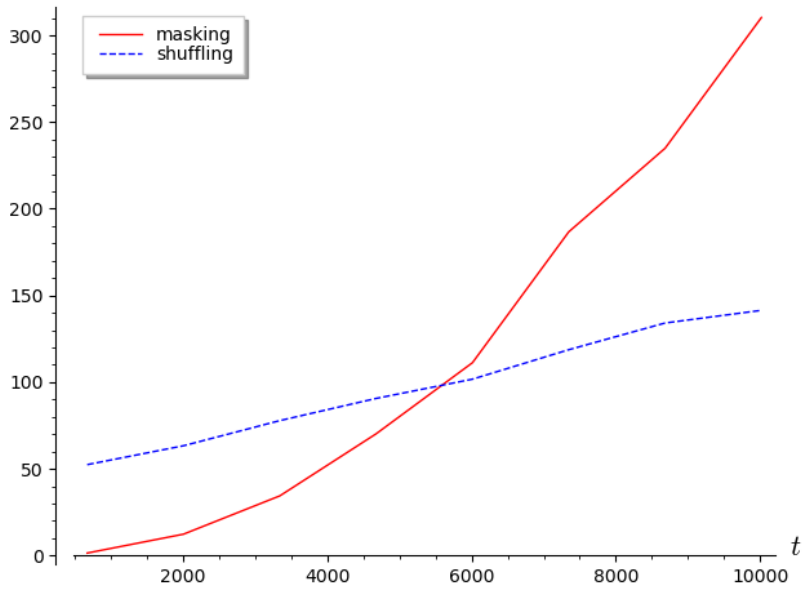

Fig. 12: Running time (in seconds) of the masking and shuffling countermeasure for AES, to get security against t probes. Implementation on a 3,2 GHz Intel processor, running on a single core.

7 Conclusion

We have described the first improvement of the wire shuffling countermeasure against side-channel attacks described by Ishai, Sahai and Wagner at Crypto 2003, with running time $\mathcal{O}(t)$ instead of $\mathcal{O}(t \log t)$ for worst-case security against t probes, and $\mathcal{O}(t^2)$ for the classical masking countermeasure. Our construction is somehow practical in that for an AES implementation we can beat the classical masking countermeasure for a reasonable running time. However the crossover point occurs for $t \simeq 6\,000$, so our countermeasure is probably unpractical for embedded implementations.

Acknowledgements

We thank the reviewers for insightful comments. The two authors were supported by the ERC Advanced Grant no. 787390.

References

ADF16. Marcin Andrychowicz, Stefan Dziembowski, and Sebastian Faust. Circuit compilers with $O(1/\log n)$ leakage rate. In *Advances in Cryptology - EUROCRYPT 2016 - Proceedings, Part II*, pages 586–615, 2016.

AKS83. Miklós Ajtai, János Komlós, and Endre Szemerédi. An $O(n \log n)$ sorting network. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, pages 1–9, 1983.

BBD+16. 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 *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 116–129, 2016.

BDD+04. T. Biedl, E.D. Demaine, C.A. Duncanc, R. Fleischerd, and S.G. Kobourove. Tight bounds on maximal and maximum matchings. *Discrete Mathematics*, 285:7–15, 2004.

CCD00. Christophe Clavier, Jean-Sébastien Coron, and Nora Dabbous. Differential power analysis in the presence of hardware countermeasures. In *Cryptographic Hardware and Embedded Systems - CHES 2000, Proceedings*, pages 252–263, 2000.

CGP+12. Claude Carlet, Louis Goubin, Emmanuel Prouff, Michaël Quisquater, and Matthieu Rivain. Higher-order masking schemes for s-boxes. In *FSE*, pages 366–384, 2012.

CGPZ16. Jean-Sébastien Coron, Aurélien Greuet, Emmanuel Prouff, and Rina Zeitoun. Faster evaluation of sboxes via common shares. In *Cryptographic Hardware and Embedded Systems - CHES 2016 - Proceedings*, pages 498–514, 2016.

Cor14. Jean-Sébastien Coron. Higher order masking of look-up tables. In *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, pages 441–458, 2014.

Cor21. Jean-Sébastien Coron. Implementation of higher-order countermeasures, 2021. Publicly available at https://github.com/coron/htable/.

FPS17. Sebastian Faust, Clara Paglialonga, and Tobias Schneider. Amortizing randomness complexity in private circuits. In *Advances in Cryptology - ASIACRYPT 2017 - Proceedings, Part I*, pages 781–810, 2017.

GJRS18. Dahmun Goudarzi, Anthony Journault, Matthieu Rivain, and François-Xavier Standaert. Secure multiplication for bitslice higher-order masking: Optimisation and comparison. In *Constructive Side-Channel Analysis and Secure Design - 9th International Workshop, COSADE 2018, Singapore, April 23-24, 2018, Proceedings*, pages 3–22, 2018.

HOM06. Christoph Herbst, Elisabeth Oswald, and Stefan Mangard. An AES smart card implementation resistant to power analysis attacks. In *Applied Cryptography and Network Security, ACNS 2006, Proceedings*, pages 239–252, 2006.

ISW03. Yuval Ishai, Amit Sahai, and David A. Wagner. Private circuits: Securing hardware against probing attacks. In *Advances in Cryptology - CRYPTO 2003, Proceedings*, pages 463–481, 2003.

JS17. Anthony Journault and François-Xavier Standaert. Very high order masking: Efficient implementation and security evaluation. In *Cryptographic Hardware and Embedded Systems - CHES 2017. Proceedings*, pages 623–643, 2017.

MS08. Kurt Mehlhorn and Peter Sanders. *Algorithms and Data Structures: The Basic Toolbox*. Springer, 2008.

Pap18. Kostas Papagiannopoulos. Low randomness masking and shuffling: An evaluation using mutual information. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(3):524–546, 2018.

RP10. Matthieu Rivain and Emmanuel Prouff. Provably secure higher-order masking of AES. In *Cryptographic Hardware and Embedded Systems, CHES 2010. Proceedings*, pages 413–427, 2010.

RPD09. Matthieu Rivain, Emmanuel Prouff, and Julien Doget. Higher-order masking and shuffling for software implementations of block ciphers. In *Cryptographic Hardware and Embedded Systems - CHES 2009, Proceedings*, pages 171–188, 2009.

THM07. Stefan Tillich, Christoph Herbst, and Stefan Mangard. Protecting AES software implementations on 32-bit processors against power analysis. In *Applied Cryptography and Network Security, ACNS 2007, Proceedings*, pages 141–157, 2007.

VMKS12. Nicolas Veyrat-Charvillon, Marcel Medwed, Stéphanie Kerckhof, and François-Xavier Standaert. Shuffling against side-channel attacks: A comprehensive study with cautionary note. In *Advances in Cryptology - ASIACRYPT 2012 - Proceedings*, pages 740–757, 2012.

VML16. Nikita Veshchikov, Stephane Fernandes Medeiros, and Liran Lerman. Variety of scalable shuffling countermeasures against side channel attacks. *J. Cyber Secur. Mobil.*, 5(3):195–232, 2016.

A Re-randomization and region probing model: a counterexample

In [ISW03], the authors write: "for each original output bit [of the AND gadget with $m+1$ shares], the encoded outputs are m-wise independent even given the entire encoding of the inputs. This can be used to prove that the construction is in fact secure against a stronger type of adversary who may observe at most t' wires in each gadget, where $t' = \Omega(t)$." While the re-randomization property indeed holds for the encoded output of the AND gadget, we explain that such property is not enough to prove security in the region probing model, by providing a simple counterexample.

Namely we consider a mask refreshing algorithm taking as input $x_1, \ldots, x_n$ and outputting $y_1 = x_1 \oplus r_1, \ldots, y_{n-1} = x_{n-1} \oplus r_{n-1}$ and $y_n = x_n \oplus r_1 \oplus \cdots \oplus r_{n-1}$, for randoms $r_1, \ldots, r_{n-1}$; here y_n is computed from left to right. Such mask refreshing clearly achieves the re-randomization property as the output shares $y_1, \ldots, y_n$ are $(n-1)$-wise independent, even given all the input shares $x_1, \ldots, x_n$. However such mask refreshing does not achieve region probing security. Namely if we iterate such mask refreshing multiple times, one can see that the adversary can accumulate knowledge and eventually recover the original secret, by probing only a constant fraction of the wires in each mask refreshing gadget.

More precisely, assume that we can probe $n/4+2$ internal variables at each iteration. At the first iteration, the adversary probes the variables $x_1, \ldots, x_{n/4}$, x_n and $x_n \oplus r_1 \oplus \cdots \oplus r_{n/4}$. Therefore the adversary can compute:

$$\begin{aligned} y_1 \oplus \cdots \oplus y_{n/4} &= (x_1 \oplus r_1) \oplus \cdots \oplus (y_{n/4} \oplus r_{n/4}) \\ &= x_1 \oplus \cdots \oplus x_{n/4} \oplus x_n \oplus (x_n \oplus r_1 \oplus \cdots \oplus r_{n/4}) \end{aligned}$$

Therefore from the knowledge of $x_1 \oplus \cdots \oplus x_{n/4}$, the adversary can keep the knowledge of $y_1 \oplus \cdots \oplus y_{n/4}$ by spending only 2 additional probes within the mask refreshing.

Let $z_1 = y_1 \oplus r'_1, \ldots, z_{n-1} = y_{n-1} \oplus r'_{n-1}$ and $z_n = y_n \oplus r'_1 \oplus \cdots \oplus r'_{n-1}$ be the second iteration of the mask refreshing algorithm. The adversary can now probe $y_{n/4+1}, \ldots, y_{n/2}$, y_n and $y_n \oplus r'_1 \oplus \cdots \oplus r'_{n/2}$. He can then compute as previously:

$$z_1 \oplus \cdots \oplus z_{n/2} = (y_1 \oplus \cdots \oplus y_{n/4}) \oplus y_{n/4+1} \oplus \cdots \oplus y_{n/2} \oplus y_n \oplus (y_n \oplus r'_1 \oplus \cdots \oplus r'_{n/2})$$

Similarly, at the 3rd iteration, the adversary can compute the xor of 3/4 of the output shares, and after the 4th iteration, he can eventually recover the xor of the n shares. The attack can be adapted when probing any constant fraction of the n shares. Therefore the above mask refreshing algorithm is not secure in the region probing model, despite achieving the re-randomization property.

B AND gadget, mask refreshing and composition

B.1 The AND gadget

We recall in Algorithm 2 the AND gadget from the ISW construction (SecMult). It was extended to $\mathbb{F}_{2^k}$ in [RP10] for protecting AES against t-th order attacks. The SecMult gadget was proven $(n-1)$-SNI in [BBD+16].

Lemma 11. *The* AND *gadget from Algorithm 2 achieves* $(n-1)$*-*SNI *security.*

Algorithm 2 SecMult

Input: shares a_i satisfying $\bigoplus_{i=1}^{n} a_i = a$, shares b_i satisfying $\bigoplus_{i=1}^{n} b_i = b$
Output: shares c_i satisfying $\bigoplus_{i=1}^{n} c_i = a \cdot b$

for $i = 1$ **to** n **do**
  $c_i \leftarrow a_i \cdot b_i$
end for
for $i = 1$ **to** n **do**
  for $j = i + 1$ **to** n **do**
    $r \leftarrow \mathbb{F}_2$
    $c_i \leftarrow c_i \oplus r$
    $r \leftarrow (a_i \cdot b_j \oplus r) \oplus a_j \cdot b_i$
    $c_j \leftarrow c_j \oplus r$
  end for
end for
return $(c_1, \ldots, c_n)$

B.2 Mask refreshing

Since the above AND gadget achieves the $(n-1)$-SNI security property [BBD+16], we can use it to obtain an $(n-1)$-SNI mask refreshing algorithm, by using a n-sharing of 1 as one of its inputs. This gives the following RefreshMasks algorithm below.

Lemma 12 ($(n-1)$-SNI of RefreshMasks). *Let* $(a_i)_{1 \le i \le n}$ *be the input shares of* RefreshMasks *algorithm, and* $(c_i)_{1 \le i \le n}$ *be the output shares. For any set of* t *intermediate variables and any subset* $|O| \le t_O$ *of output shares such that* $t + t_O < n$*, there exists a subset* I *of indexes with* $|I| \le t$*, such that the distribution if those* t *intermediate variables as well as the output shares* $c_{|O}$ *can be perfectly simulated from* $a_{|I}$*.*

B.3 Security in the region probing model with $n = 3t + 1$ shares

We first prove the region probing security of a slightly simpler construction where we only apply an $(n-1)$-SNI mask refreshing algorithm as output of each XOR gadget. In that case, we show that for a circuit C of fan-out 2, we can get security against t probes per region, with $n = 3t + 1$ shares. We show in the next section that the number of shares can actually be decreased to $n = 2t + 1$, moreover without the bounded fan-out condition, by adding more mask refreshing.

Algorithm 3 RefreshMasks

Input: $a_1, \ldots, a_n$
Output: $c_1, \ldots, c_n$ such that $\bigoplus_{i=1}^{n} c_i = \bigoplus_{i=1}^{n} a_i$

```
For i = 1 to n do c_i ← a_i
for i = 1 to n do
    for j = i + 1 to n do
        r ← 𝔽
        c_i ← c_i ⊕ r
        c_j ← c_j ⊕ r
    end for
end for
return c_1, ..., c_n
```

Theorem 10. *Let C be a circuit with fan-out at most 2. Let (I, O, T) be the transformer from Section 2.2 with $n = 3t + 1$ shares, where a $(n-1)$-SNI mask refreshing is applied after every* XOR *gadget. The transformed circuit is t-private secure where the adversary can put at most t probes per region, each of size $\mathcal{O}(t^2)$.*

To prove Theorem 10 we first show that the XOR gadget followed by a mask refreshing achieves the $(n-1)$-SNI property; see Fig. 13 for an illustration.

Lemma 13. *Let G be the* XOR *gadget from Section 2.2, where a $(n-1)$-SNI mask refreshing is applied after it. Such gadget achieves $(n-1)$-SNI security.*

Proof. Let $(a_i)_{1 \le i \le n}$ and $(b_i)_{1 \le i \le n}$ be the input shares of the XOR gadget, and $(c_i)_{1 \le i \le n}$ be the corresponding output shares; let $(c'_i)_{1 \le i \le n}$ the output shares of Refresh. Let t_1 and t_2 be the number of probes within the XOR and Refresh gadgets respectively, with $t_1 + t_2 + |\mathcal{O}| < n$, where $\mathcal{O}$ is the set of output variables to be simulated.

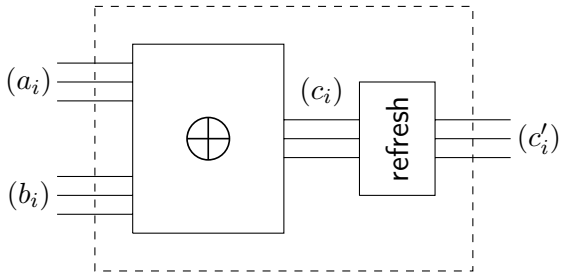

Fig. 13: XOR gadget composed with RefreshMasks.

Since by assumption the Refresh gadget is $(n-1)$-SNI and $t_2 + |\mathcal{O}| < n$, the t_2 probes within Refresh and the output variables $c'_{|\mathcal{O}}$ can be simulated from $c_{|R}$, with $|R| \le t_2$; furthermore the variables $c_{|R}$ can be simulated from $a_{|R}$ and $b_{|R}$. Moreover the t_1 probes within the XOR gadget can be perfectly simulated from $a_{|S}$ and $b_{|S}$, with $|S| \le t_1$. Therefore the output variables $c'_{|\mathcal{O}}$ and the $t = t_1 + t_2$ probes within the full gadget can be perfectly simulated from $a_{|I}$ and $b_{|I}$ with $I = R \cup S$ and $|I| \le |R| + |S| \le t_1 + t_2 = t$ as required. □

Proof (of Theorem 10). We consider the circuit as a set of q gadgets $G_1, \ldots, G_q$ that we order as a direct acyclic graph from output to input in a reverse topological sort order; see Figure 14 for an illustration. We assume that each gadget G_i achieves the $(n-1)$-SNI property, with $n = 3t+1$ shares. Let denote by $(x_i^{(j)})_{1 \le i \le n}$ the input shares of the composed circuit, for $1 \le j \le k$, where k is the number of inputs in the original circuit corresponding to the q gadgets $G_1, \ldots, G_q$. We prove by recurrence on q that the composition of the q gadgets achieves the following property: any set of t probes in each gadget can be perfectly simulated from the knowledge of the input shares $x^{(j)}_{|I_j}$, for subsets I_j with $|I_j| \le t$ for all $1 \le j \le k$.

The base case $q = 1$ is straightforward from the $(n-1)$-SNI property of the gadget G_1. We now consider q gadgets $G_1, \ldots, G_q$ with input shares $(x_i^{(j)})_{1 \le j \le n}$ for $1 \le j \le k$, and an additional gadget G_{q+1}. By assumption any set of t probes in each gadget $G_1, \ldots, G_q$ can be perfectly simulated from the knowledge of the input shares $x^{(j)}_{|I_j}$, with $|I_j| \le t$ for all $1 \le j \le k$. Let $(y_i)_{1 \le i \le n}$ be the output shares of G_{q+1}, and let $(x_i^{(k+1)})_{1 \le i \le n}$, $(x_i^{(k+2)})_{1 \le i \le n}$ be the corresponding input shares. Since the original circuit C has fan-out 2, the output shares of G_{q+1} are used by at most two gadgets G_{j_1} and G_{j_2}. Let $O = I_{j_1} \cup I_{j_2}$ be the output shares of G_{q+1} that must be simulated; we have $|O| \le |I_{j_1}| + |I_{j_2}| \le 2t$. Since G_{q+1} is $(n-1)$-SNI and $|O| + t \le 3t < n$, the t probes within G_{q+1} and the output shares $y_{|O}$ can be perfectly simulated from $x^{(k+1)}_{|I_{k+1}}$ and $x^{(k+2)}_{|I_{k+2}}$, with $|I_{k+1}|, |I_{k+2}| \le t$. This proves the inductive property for the $q+1$ gadgets $G_1, \ldots, G_{q+1}$. Eventually, this proves that the transformed circuit is t-private secure where the adversary can put at most t probes per region. □

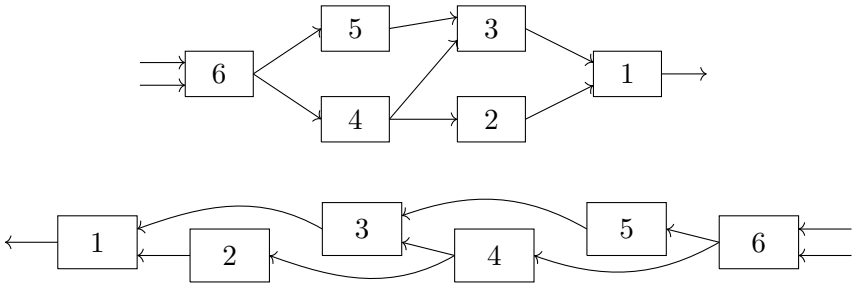

Fig. 14: Example of a circuit (top), and same circuit displayed in reverse topological sort order (bottom)

B.4 Security in the region probing model with $n = 2t+1$ shares: proof of Theorem 2

In our construction, a $(n-1)$-SNI mask refreshing is applied as input of each XOR and AND gadgets. We define each region as comprising an AND or XOR gadget, and the mask refreshing of the corresponding output variable, so that each output $z^{(j)}$ is used only once in the next region; see Fig. 2 for an illustration. It is easy to see that the corresponding gadget achieves an extended $(n-1)$-SNI property, where for any subsets $O_1, \ldots, O_f$ with $|O_j| + t < n$ for all $1 \le j \le f$, the output shares $z^{(j)}_{|O_j}$ for $1 \le j \le f$ and any set of t probes can be perfectly simulated from the input shares $x_{|I}$ and $y_{|J}$, for $|I|, |J| \le t$.

The proof is then essentially the same as the proof of Theorem 10 in the previous section. The difference is that since each output variables $z^{(j)}$ is used only once (instead of at most twice), we can assume recursively that $|O_i| \le t$ for all $1 \le i \le f$ in each gadget (instead of $|O| \le 2t$), and therefore we can use $n = 2t+1$ shares instead of $n = 3t+1$.

C Security in the random probing model

C.1 Chernoff Bound

The Chernoff bound is obtained by applying Markov's inequality to e^{tX}. For every $t > 0$:

$$\Pr[X \geq a] = \Pr[e^X \geq e^a] \leq \frac{\mathbb{E}[e^{t \cdot X}]}{e^{t \cdot a}}$$

Assume $X = X_1 + \cdots + X_n$ where $X_1, \ldots, X_n$ are independent variables. We get for any $t > 0$:

$$\Pr[X \geq a] \leq e^{-t \cdot a} \prod_{i=1}^{n} \mathbb{E}[e^{tX_i}]$$

If the variables X_i have their value in $\{0, 1\}$, letting $p_i = \Pr[X_i = 1]$, we get:

$$\mathbb{E}[e^{tX_i}] = e^0 \cdot (1 - p_i) + e^t \cdot p_i = 1 + p_i \cdot (e^t - 1) \leq e^{p_i \cdot (e^t - 1)}$$

This gives:

$$\Pr[X \geq a] \leq e^{-t \cdot a} \cdot e^{\sum_{i=1}^{n} p_i \cdot (e^t - 1)} = e^{-t \cdot a + \mu \cdot (e^t - 1)}$$

where we let $\mu = \mathbb{E}[X] = \sum_{i=1}^{n} p_i$. We take $a = (1 + \delta) \cdot \mu$ for $\delta > 0$. We take $t = \log(1 + \delta)$. This gives the Chernoff bound:

$$\Pr[X \geq (1 + \delta)\mu] \leq \left(\frac{e^\delta}{(1 + \delta)^{1+\delta}} \right)^\mu$$

Using the inequality $2\delta/(2 + \delta) \leq \log(1 + \delta)$, we obtain:

$$\Pr[X \geq (1 + \delta)\mu] \leq \exp\left(\frac{-\delta^2 \mu}{2 + \delta} \right)$$

For $\delta \geq 1$, we have:

$$\Pr[X \geq (1 + \delta)\mu] \leq \exp\left(\frac{-\delta \mu}{3} \right)$$

C.2 Security in the random probing model: proof of Lemma 1

We consider the ISW construction from Theorem 2 with $n = 2k + 1$ shares and secure against the probing of at most k probes per gadget, where each gadget has a number of wires at most $c \cdot k^2$ for some constant c. We first consider a single gadget with $\ell \leq c \cdot k^2$ wires, where each wire is leaking with probability p. Let $X_1, \ldots, X_\ell$ be independent Bernoulli random variables, where $X_i = 1$ if the i-th wire is probed, and $X_i = 0$ otherwise; we have $\Pr[X_i = 1] = p$ for all $1 \leq i \leq \ell$. Let $X = X_1 + \ldots + X_\ell$ be the total number of leaking wires in the gadget. From Theorem 2, we can provide a perfect simulation of the probes if $X \leq k$ in each gadget.

We apply Chernoff bound on a single gadget with $\delta \geq 1$ (see Appendix C.1):

$$\Pr[X \geq (1 + \delta)\mathbb{E}[X]] \leq \exp\left(-\frac{\delta}{3} \mathbb{E}[X] \right) \tag{1}$$

where $\mathbb{E}[X] = p \cdot \ell$. We fix $p = 1/(2c \cdot k)$. Since we must upper-bound $\Pr[X \geq k]$, we fix δ such that:

$$k = (1 + \delta)\mathbb{E}[X]$$

This gives using $\ell \leq c \cdot k^2$:

$$1 + \delta = \frac{k}{\mathbb{E}[X]} = \frac{k}{p \cdot \ell} = \frac{2c \cdot k^2}{\ell} \geq \frac{2c \cdot k^2}{c \cdot k^2} \geq 2$$

and therefore $\delta \geq 1$ as required. Moreover we have:

$$\frac{\delta}{3}\mathbb{E}[X] = \frac{\delta}{3} \cdot \frac{k}{1+\delta} \geq \frac{k}{6}$$

Therefore we obtain from (1):

$$\Pr[X \geq k] \leq \exp(-k/6)$$

The simulation failure probability over all gadgets is therefore at most $|C| \cdot \exp(-k/6)$.

C.3 Proof of Lemma 6

Let $X_1, \ldots, X_m$ be independent Bernoulli random variables, where $X_i = 1$ if the i-th gate of C' is probed, and $X_i = 0$ otherwise. We assume that $\sum_i \Pr[X_i] \leq \mu$ with $\mu = k/4$. Let $X = X_1 + \ldots + X_m$ be the total number of leaking gates in the circuit C'. The circuit $C' = T(C, k)$ is perfectly secure against an adversary probing at most k variables; hence it is perfectly secure against the leakage of $k/2$ gates; namely a gate can be perfectly simulated from the knowledge of its two inputs. Therefore we can provide a perfect simulation of the probes if $X \leq k/2$.

To compute the simulation failure probability we apply the Chernoff bound for $\delta \geq 1$ (see Appendix C.1):

$$\Pr[X \geq (1+\delta) \cdot \mathbb{E}[X]] \leq \exp\left(-\frac{\delta}{3}\mathbb{E}[X]\right)$$

We fix δ such that $(1+\delta)\mathbb{E}[X] = 2\mu$. Since $\mathbb{E}[X] = \sum_i \Pr[X_i] \leq \mu$, we have $\delta \geq 1$ as required. We obtain:

$$\frac{\delta}{3}\mathbb{E}[X] = \frac{\delta}{3} \cdot \frac{2\mu}{1+\delta} \geq \mu/3$$

Finally, using $\mu = k/4$, we obtain:

$$\Pr[X \geq k/2] \leq \exp(-k/12)$$

The simulation failure probability is therefore upper-bounded by $\varepsilon = \exp(-k/12) = 2^{-k/(12 \log 2)}$.

D Worst-case statistical security: pure circuit description

In this section we consider a pure circuit construction of a worst-case statistically secure transformer, without using the RAM model. The construction is the same as in Section 3.1, except that the cyclic shift is implemented by a circuit of complexity $\mathcal{O}(\ell \log \ell)$, instead of a table look-up with running time $\mathcal{O}(\ell)$. The main difference with Section 3.1 is that the adversary can now probe the wires within the cyclic shift. Therefore we can use the same approach as in [ISW03, Lemma 2], with $\ell = \mathcal{O}(t/p^7)$ instead of $\ell = \mathcal{O}(t/p^4)$; see Section 2.6. Since our expanded gate has complexity $\mathcal{O}(\ell \log \ell)$ instead of $\mathcal{O}(k \cdot \ell \log \ell)$ in [ISW03], the final complexity is $\mathcal{O}(|C| \cdot k^9 \cdot t \cdot (\log k + \log t))$ instead of $\mathcal{O}(|C| \cdot k^{10} \cdot t \cdot (\log k + \log t))$. This proves the following theorem.

Theorem 11. *There exists a statistically t-private stateless transformer $(\widetilde{T}, \widetilde{I}, \widetilde{O})$, such that $\widetilde{T}(C, k)$ transforms a circuit C to a circuit $\tilde{C}$ of size $\mathcal{O}(|C| \cdot t \cdot \log t)$. The depth of $\tilde{C}$ is the same as that of C, up to* polylog *factors.*

Similarly, we can obtain the same result in the stateful model using the randomizing network construction from Section 5.2.

Theorem 12. *There exists a statistically t-private stateful transformer $\widetilde{T}$, such that $\widetilde{T}(C,k)$ maps a circuit C with s memory cells to a circuit $\tilde{C}$ with complexity $\mathcal{O}(|C| \cdot t \cdot \log t)$. The depth of $\tilde{C}$ is the same as that of C, up to* polylog *factors.*

E Security in the stateful model

E.1 Free t-SNI security

We introduce a slightly stronger notion of security than t-SNI, called free-t-SNI security. Under this notion, the output variables $c_{|O}$ must be uniformly and independently distributed for any $O \subsetneq [1,n] \setminus I$, and this must be true even conditioned on the probed variables and $c_{|I}$. This means that even after the simulation of the probed variables has been provided to the adversary, we can still simulate the variables in $c_{|O}$, simply by generating uniformly and independently random values (this was not necessarily the case with the original t-SNI notion).

Definition 10 (Free-t-SNI security). *Let G be a gadget taking as input n shares $(a_i)_{1\leq i\leq n}$ and outputting n shares $(c_i)_{1\leq i\leq n}$. The gadget G is said to be free t-SNI secure if for any set of t_1 probed intermediate variables, there exists a subset I of input indices with $|I| \leq t_1$, such that the t_1 intermediate variables and the output variables $c_{|I}$ can be perfectly simulated, while for any $O \subsetneq [1,n] \setminus I$ the output variables in $c_{|O}$ are uniformly and independently distributed, conditioned on the probed variables and $c_{|I}$.*

The RefreshMasks algorithm (Algorithm 3 from Appendix B.2) satisfies the above free t-SNI notion. The proof is essentially the same as the proof of the ISW multiplication algorithm from [CGPZ16, Appendix B.1]; namely the proof constructs a subset $I = U$ of indices such that the t_1 probes and $c_{|I}$ can be perfectly simulated from $a_{|I}$, and in the proof the other output variables $c_{|O}$ from any $O \subsetneq [1,n] \setminus I$ are simulated by generating a random value. Therefore the output variables in $c_{|O}$ are uniformly and independently distributed, conditioned on the probed variables and $c_{|I}$.

Lemma 14 (Free-$(n-1)$-SNI of RefreshMasks). *The* RefreshMasks *algorithm is free-$(n-1)$-SNI.*

E.2 Perfect privacy in the stateful model: proof of Theorem 7

The proof is essentially the same as the proof of Theorem 2 in Appendix B.4. The difference is that in the unwound circuit C'', the adversary can move its probe between regions corresponding to different executions of the circuit C'. This implies that the simulation for the i-th execution must be performed before knowing the position of the probes for the $(i+1)$-th execution. This is not a problem thanks to the free-$(n-1)$-SNI property of RefreshMasks (Lemma 14). Namely the input shares of the next gadget (that must be known for the next simulation), either belong to the set $c_{|I}$ of outputs shares from the previous gadget and are already simulated, or to a set $c_{|O}$ of output shares that can be simulated by generating uniformly distributed values. Therefore the simulation for the stateful model can be performed by running a sequence of simulations from the stateless model.

E.3 Statistical privacy in the stateful model

We recall the definition of statistical privacy for stateful circuits from [ISW03].

Definition 11 (Statistical privacy for stateful circuits.). *Let T be an efficiently computable randomized algorithm mapping a stateful circuit C along with an initial state s_0 to a stateful circuit $\tilde{C}$ along with an initial state s'_0. We say that T is a statistical t-private stateful transformer if it satisfies:*

1. ***Soundness.*** *The input-output functionality of C initialized with s_0 is indistinguishable from that of $\tilde{C}$ initialized with s'_0. This should hold for any sequence of invocations on an arbitrary sequence of inputs.*
2. ***Privacy.*** *We require that $\tilde{C}$ be private against a t-limited interactive adversary. Specifically, the adversary is given access to $\tilde{C}$ initialized with s'_0 as its internal state. Then, the adversary may invoke $\tilde{C}$ multiple times, adaptively choosing the inputs based on the observed outputs. Prior to each invocation, the adversary may fix an arbitrary set of t internal wires to which it will gain access in that invocation. To define privacy against such a t-limited adversary, we require the existence of a simulator which can simulate the adversary's view, using only a black-box access to $\tilde{C}$, except with negligible probability.*

E.4 Iterated cyclic shifts: proof of Lemma 8

We consider the following sequence of two security games.

Game_1: we follow Definition 9, in which the adversary first chooses t probes (excluding the input wires), obtain their value, and then chooses another t probes among the output wires of the gadget. The adversary wins the game if he observes the signal value x. We denote by S_1 the event that the value x is revealed by the first t probes, and by T_1 the event that the value x is revealed by the t output wires chosen by the adversary. We have $\Pr[S_1] \leq t/\ell$.

Game_2: we proceed as in Game_1 above, but now we modify the way the output of the first t probes are generated, in that we never reveal the value x; instead we always leak the dummy value \$. Let T_2 be the event in Game_2 that the value x is revealed by the t output wires. We see that events T_1 and T_2 are identical, unless one of the first t probes would reveal x. This means $T_1 \wedge \neg S_1 = T_2 \wedge \neg S_1$.

We claim that in Game_2 the adversary gets no information about the index position of the signal in the output variables. Namely from the first set of t probes the adversary can only receive the dummy value \$; moreover the adversary can only probe at most t of the $t+1$ shifting index Δ_i of the cyclic shifts. Since at least one of those $t+1$ random cyclic shifts has not been probed, the distribution of the signal position after this cyclic shift is independent from the adversary's view and uniform in $[0, \ell)$; this also holds for the signal position in the output variables. This implies $\Pr[T_2] \leq t/\ell$. Eventually we obtain:

$$\Pr[S_1 \vee T_1] = \Pr[S_1] + \Pr[T_1 \wedge \neg S_1] = \Pr[S_1] + \Pr[T_2 \wedge \neg S_1] \leq \Pr[S_1] + \Pr[T_2] \leq 2t/\ell$$

Therefore, the $2t$ probes can be simulated except with probability at most $2t/\ell$.

E.5 Randomizing network: proof of Lemma 9

The proof is essentially the same as the proof of Lemma 8 from the previous section. We assume that when the adversary probes a swap, he learns the two input wires of the swap, and additionally a bit b corresponding to whether a swap of the wire values occurred or not. We consider two security

games Game_1 and Game_2. In Game_1 we follow Definition 9; as previously, in Game_2 we always leak the dummy value \$; moreover, when a swap is probed, we replace the bit b by an independently generated random bit. As previously, we claim that in Game_2 the adversary learns nothing about the position of the signal among the output wires. Denoting by S_1 the event in Game_1 that the value x is revealed by the first t probes, we have $\Pr[S_1] \leq 2t/\ell$, since the adversary can now learn 2 wires at a time. Denoting by T_1 and T_2 the same events as in the proof of Lemma 8, we obtain as previously:

$$\Pr[S_1 \vee T_1] \leq \Pr[S_1] + \Pr[T_2] \leq 2t/\ell + t/\ell = 3t/\ell$$

Therefore the $2t$ probes can be simulated except with probability at most $3t/\ell$. To satisfy Definition 9 we can therefore use $3\ell/2$ wires instead of ℓ.

E.6 Composition: proof of Lemma 10

The proof is essentially the same as the proof for the stateless case (see Lemma 5 in Section 3.2). For the stateful case we first obtain a p-random secure intermediate circuit C' from Theorem 7. The proof then proceeds as in the stateless case. In a given execution the adversary probes in $\tilde{C}$ generate a random sampling W of the gates in C', where from Definition 6 each gate leaks with probability at most p. Moreover Definition 9 implies that the input wires for the next execution of C' also leak with probability at most p. From the p-random probing security of C', this implies worst-case t-privacy in the stateful model of the final circuit $\tilde{C}$.