Towards General-Purpose Program Obfuscation via Local Mixing

Ran Canetti[1,2(✉)], Claudio Chamon[1,2], Eduardo R. Mucciolo[1,2], and Andrei E. Ruckenstein[1,2]

[1] Boston University, Boston, USA
canetti@bu.edu
[2] University of Central Florida, Orlando, USA

Abstract. We explore the possibility of obtaining general-purpose obfuscation for all circuits by way of making only simple, local, functionality preserving random perturbations in the circuit structure. Towards this goal, we use the additional structure provided by reversible circuits, but no additional algebraic structure. Our approach is rooted in statistical mechanics and can be thought of as locally "thermalizing" a circuit while preserving its functionality.

We analyze the security of this approach in two steps. First, we provide arguments towards its security for a relatively simple task: obfuscating random circuits of bounded length. Next we show how to construct indistinguishability obfuscators for all (unbounded length) circuits given an obfuscator for random reversible circuits of bounded length. Here security is proven under a new assumption regarding the pseudorandomness of sufficiently-long random reversible circuits.

Our specific candidate obfuscators are very simple and relatively efficient: the obfuscated version of an n-wire, m-gate (reversible) circuit with security parameter κ has n wires and $O(\kappa m)$ gates. We hope that our initial exploration will motivate further study of this alternative path to program obfuscation (and, consequently, to cryptography in general).

1 Introduction

Program obfuscation [Had00, BGI+01, BGI+12], namely the ability to efficiently purturb a program in a way that preserves its functionality but hides "all other information" about the program, is an intriguing beast. At first, perturbing - or randomizing - the internal structure of a program may appear to be rather mundane and inconsequential. However, with the right formalization of "sufficiently perturbed", program obfuscation has proven to be immensely powerful.

As shown in [BGI+01, BGI+12], in general *any* polysize representation of a program, even a "perfectly randomized" one, gives significantly more com-

This work is supported by NSF grants 2428487 and 2428488, DARPA grants HR00112020021 and HR00112020023 (R.C.), DOE Grant DE-FG02-06ER46316 (C.C.), and a Grant from the Mass Tech Collaborative Innovation Institute (A.E.R.). R.C., C.C., and A.E.R. also acknowledge the Quantum Convergence Focused Research Program, funded by the Rafik B. Hariri Institute at Boston University.

E. Boyle and M. Mahmoody (Eds.): TCC 2024, LNCS 15367, pp. 37–70, 2025.
https://doi.org/10.1007/978-3-031-78023-3_2

putational power than black-box access to the function computed by the program. However, the more modest goal of perturbing the program just to the point of making the perturbed versions of any two equal-length, functionally equivalent programs indistinguishable is potentially obtainable and has proven to be immensely powerful. Indeed, while the ability to obfuscate general programs to that level (namely obtaining *Indistinguishability Obfuscation (IO)* [BGI+01, BGI+12]) does not imply any computational hardness in and of itself (Indeed, if P=NP then IO exists), IO for all circuits combined with the mere assumption that P≠NP implies public key encryption, trapdoor permutations, general secure multiparty computation, non-interactive zero knowledge, succinct non-interactive arguments, and deniable encryption to name only very few, see e.g. [SW14, GGHR14, BPW16]. When combined with lossy one way functions, it gives also fully homomorphic encryption, collision resistant hashing, and more [CLTV15].

The history of attempts at constructing general purpose program obfuscators, starting from the breakthrough works of [SW14, GGSW13], is intriguing as well.

In the "first generation" constructions such as [GGSW13, BGK+14, AB15] [GGH15] the obfuscated program typically follows the instruction structure of the the plaintext program without modification, while using algebraic structures to perform the instructions "homomorphically" while hiding them from an adversary who runs the program and sees the entire execution trace. However, the analyses of these first generation constructions was invariably incomplete, often by way of relying on an idealized version of a core primitive, and indeed explicit attacks have been demonstrated against many proposed instantiations of these candidates (e.g., [CGH+15, CVW18, CHVW19]).

The "second generation" constructions (starting from [BV15, AJS15, LPST16]) take a different approach: Rather than directly follow the steps of the input program, the obfuscated program is treated as a "compressed store" of "garbled programs", namely, obfuscated programs that are valid only for a single input. Given an input, the overall obfuscated program first gradually "uncompresses" the garbled program for that input, and then runs this garbled program to obtain the desired output. A number of more recent IO candidate constructions, including the breakthrough works of Jain, Lin and Sahai [JLS21] that provide the first IO schemes whose security is proven based on relatively well understood assumptions, as well as [GP21, WW21, DQV+21, KNT22, RVV24] and others, use that structure.

This two-stage structure is, however, a bit roundabout and results in prohibitively high space and time overhead relative to the complexity of the plaintext program, rendering general program obfuscation a purely theoretical primitive.

1.1 This Work

We explore a new approach to constructing general-purpose program obfuscation. The idea is very simple: Repeatedly perform random local perturbations of the given program, while guaranteeing that each perturbation preserves the overall functionality of the program. The overarching hope is that, while individual

perturbations can be easily undone, the aggregate effect of the perturbation process will be that of converging to a distribution over programs that hides (or even completely destroys) the structure of the original program—all while preserving functionality.

Of course, significantly more detail and structure are needed in order to turn this very high-level idea ideal into a concrete proposal. Here one must also keep in mind the long list of failed attempts at using such techniques to provide "white box security" for programs that are accessible to an adversary (see e.g. [Wik24]).

The structure we employ is that of reversible computation, where the number of state variables remains fixed throughout the computation, and each individual computational step can be reversed (namely, undone) in a single step. Specifically, we concentrate on reversible circuits, where state variables correspond to wires, and a computational step corresponds to a gate that applies a permutation to the current state.

It is stressed that, while reversible circuits are often studied because of their physical properties (say for energy efficiency or quantum computation [Ben73, Tof80]), here the motivation to consider reversible circuits is purely cryptographic. Specifically, we use the algebraic structure provided by the fact that reversible circuits consist of sequences of permutations to argue that appropriately chosen local perturbations of the circuit structure are likely to have a global effect that is hard to reverse and is likely to make the obfuscated versions of any two same-size, functionally equivalent programs indistinguishable. More specifically, our construction and analysis proceeds in two main steps:

- The first step describes a candidate scheme (or, rather, a meta-scheme) for obfuscating *bounded-length random circuits.* These are n-wire circuits that consist of m gates (where m is some fixed polynomial in the security parameter) and each gate is chosen independently at random from a fixed set of gates. While we only provide informal arguments for the security of this scheme, we do rigorously formulate a notion of security, Random Input and Output (RIO) obfuscation, that we conjecture to be satisfied by our scheme.
- The second step constructs an IO scheme for all circuits (not necessarily reversible), given any RIO obfuscator. We prove security of this construction under a new intractability assumption on the distribution of random reversible circuits.

We start the exposition of our results with a brief overview of reversible circuits, followed by an exposition of our intractability assumptions regarding the same. Next we review our definition of RIO obfuscation and how we use it to construct an IO scheme for all circuits. Finally, we sketch our candidate RIO obfuscator and the arguments for its security.

1.2 Reversible Circuits and Their Pseudorandomness Properties

Reversible circuits. Recall that reversible circuits have a fixed number, n, of wires (or, binary state variables), and each gate γ computes a permutation on the n-bit state. The permutation $\mathcal{P}_C$ computed by $C = \gamma_1 \dots \gamma_m$ is the composition of

the individual permutations, $\mathcal{P}_C = \mathcal{P}_{\gamma_m} \circ \ldots \circ \mathcal{P}_{\gamma_1}$, or, in other words, $C(x) = \gamma_m(\ldots\gamma_1(x)\ldots)$. We restrict our attention to Toffoli gates, namely gates of the form $\gamma_{i,j,k,f}(s_1 \ldots s_n) = (s'_1 \ldots s'_n)$ where $s_1 \ldots s_n$ is the old state, $s'_1 \ldots s'_n$ is the new state, i, j, k are distinct indices in $[n]$, $f : \{0,1\}^2 \to \{0,1\}$, $s'_i = s_i + f(s_j, s_k)$, and $s'_{i'} = s_{i'}$ for all $i' \neq i$ [Tof80].

We first argue that restricting attention to obfuscation of reversible circuits of the above form does not limit the generality of the treatment. Indeed, the set $\mathbb{B}_n$ of gates of the above form generates the alternating group $\mathbb{A}_{2^n}$ of even permutations over $\{0,1\}^n$ (see e.g. [CG75, Bro04]). Furthermore, any (non-reversible) circuit can be embedded in a reversible circuit while preserving both the functionality and the complexity of the original circuit (see e.g. [Tof80]).[1]

On the other hand, reversible circuits have some attractive properties which are essential for our treatment:

Limited Independence and Pseudorandomness. The model enables for a natural notion of *random circuits* of certain dimensions (say, numbers of wires and gates), which is efficiently samplable. Furthermore, the fact that all gates compute permutations makes it plausible that the permutation computed by a *random n-wire, m-gate circuit* has some randomness properties, and that the "level of randomness" increases monotonically with m. (Natural distributions over general Boolean circuits do not appear to exhibit such properties.) Indeed, the pseudorandomness of random reversible circuits has been the focus of much study over the past decades, with some very new and exciting progress.

Gowers [Gow96] shows that $\mathcal{C}_{n,m}$, the family of n-wire, m-gate circuits is ε-close to being strongly t-wise independent whenever $m = \Omega(n^3 t^3 \log(\varepsilon^{-1}))$. Hoory et al. [HMMR05] and later Brodsky and Hoory [HB05] improve this bound to $m = \Omega(n^3 t^2 + n^2 t \log(\varepsilon^{-1}))$. Very recently, He and O'Donnell [HO24] and Gretta, He and Pelecanos [GHP24] have further improved the bound to $m = \tilde{O}(nt \log(\varepsilon^{-1}))$. (We note that, while Gowers considered all $8!\binom{n}{3}$ permutations on 3 wires as base permutations, all other works mentioned above consider the same set $\mathbb{B}_n$ of base permutations considered here.)

Gowers conjectured that the family of permutations defined by m-gate reversible circuits on n wires might be pseudorandom (in the cryptographic sense) for some $m = poly(n)$.[2] In fact, his construction can be viewed as the "quintessential block cipher" where each base permutation is an independently

[1] More specifically, any circuit C with α input wires, β output wires, μ NAND gates and width ω can be transformed to a reversible circuit C' on $n = \alpha + \beta + \delta$ wires and m gates, where $n = O(\omega)$ and $m = O(\mu)$, and where $C'(x, y, 0^\delta) = (x, C(x) + y, 0^\delta)$ for any $x \in \{0,1\}^\alpha, y \in \{0,1\}^\beta$. While known constructions are only guaranteed to preserve functionality when some of the input wires are set to 0 and may thus not be sufficient for the purpose of program obfuscation. To address this issue, we give an "obfuscation compatible" transform with the additional guarantee that $C'(x, y, z) = (x, y, z)$ whenever $z \neq 0^\delta$ (see [CCMR24]).

[2] The conjecture is actually only implicit in [Gow96]. It is made explicit in Barak's survey [Bar17].

chosen "S-Box" and the key essentially specifies the schedule and ordering of S-boxes to be applied. Indeed, the main conceptual difference between the Gowers construction and modern block ciphers such as AES is the use of a key schedule that significantly reduces the overall key size. (AES and other block ciphers contain additional linear operations over the entire state; however as evidenced by the t-wise independence results mentioned above, the Gowers construction effectively approximates such operations as well.) The t-wise independence of AES and the pseudorandomness of the Gowers construction have also been studied in [LTV21,LPTV23,HO24].

Rerandomiability. Reversible circuits appear to be readily amenable to functionality-preserving rerandomization via local perturbations. We discuss this property at length later on, and only note at this point that all base permutations $\beta \in \mathbb{B}_n$ are inverses of themselves, namely $\beta\beta = I_n$, where I_n denotes the identity permutation on $\{0,1\}^n$. This also means that, for any circuit C on n wires, the circuit $C|C^\dagger$ computes I_n, where $C^\dagger$ has the gates of C in reverse order and $|$ denotes circuit concatenation. In fact, the set of n-gate reversible circuits with set $\mathbb{B}$ of base gates can be viewed as the Free Group $F_{\mathbb{B}}$ over alphabet $\mathbb{B}$. Furthermore, the operation of evaluating a circuit can be viewed as a group action of $F_{\mathbb{B}}$ on the Alternating group $\mathbb{A}_{2^n}$ of even permutations on $\{0,1\}^n$. The kernel of this action is the set of identity circuits and the cosets are the sets of functionally equivalent circuits. This algebraic structure provides a basis for our obfuscation scheme based on local perturbations, described in Sects. 1.5 and 6.

White-box Pseudorandomness. while the pseudorandomness properties of the permutations computed by random reversible circuits may be intriguing, they do not suffice for our needs. Here instead we are concerned with adversaries that have full access to the circuit description, and can mount attacks that combine the circuit's functionality and structure.

The good news about random reversible circuits is that their "internal structure" appears to be largely uncorrelated with their functionality, in the sense that even very large portions of a sufficiently long random circuit remain pseudorandom *even given oracle access to the overall circuit.* For instance, let m^* denote the number of gates that suffices for Gower's conjecture to hold with respect to some number of wires n and security parameter κ. (For notational simplicity we assume $n = \kappa$). Now, let $C \xleftarrow{R} \mathcal{C}_{n,m}$ for some $m > 2m^*$, and let C_i denote the circuit C without the m^*-gate sub-circuit that starts at gate i. It is easy to see that the following is implied by Gower's conjecture, for any i: Polytime adversaries that are given i, oracle access to C and a challenge $(m - m^*)$-gate circuit C', cannot tell significantly better than a random guess whether $C' = C_i$, or else C' is an independently chosen random circuit, i.e. $C' \xleftarrow{R} \mathcal{C}_{n,m-m^*}$.[3]

[3] Indeed, an algorithm A that guesses correctly for some i can be used to break Gower's conjecture: Given oracle access to an unknown function F, choose $P_0, P_1 \xleftarrow{R} \mathcal{C}_{n,i}, S_0, S_1 \xleftarrow{R} \mathcal{C}_{n,m-m^*-i}$ and $b \xleftarrow{R} \{0,1\}$, run A on input (i, P_0, S_0), and answer each oracle query x of A with $S_b(F(P_b(x))$. If $\mathcal{A}$ guesses b correctly then guess that F is taken from Gower's PRF, else guess that F is a random permutation.

Furthermore, it is only natural that this same property—pseudorandomness of large circuit segments—would extend also to sufficiently long random circuits with some *fixed functionality.* For instance, let $\mathcal{E}_{P,m}$ denote the set of all m-gate circuits that compute permutation P, let $C \xleftarrow{R} \mathcal{E}_{I_n,2m^*}$, and let $C_{[1,m^*]}$ denote the m^*-gate prefix of C. While $C_{[1,m^*]}$ is statistically far from a random n-wire, m^*-gate circuit, it appears plausible that the two distributions are indistinguishable.

By the same token, it seems plausible that

$$C : C \xleftarrow{R} \mathcal{E}_{I_n,2m^*} \overset{c}{\approx} C|C' : C \xleftarrow{R} \mathcal{C}_{n,m^*}; C' \xleftarrow{R} \mathcal{E}_{C^\dagger,m^*},$$

namely that a random $2m^*$-gate identity circuit is indistinguishable from a random m^*-gate circuit C followed by the inverse of another random m^*-gate circuit C' that's functionally equivalent to C. (We use $\mathcal{E}_{C,m}$ as a shorthand for $\mathcal{E}_{\mathcal{P}_C,m}$, where $\mathcal{P}_C$ is the permutation computed by circuit C.) Indeed, here we have two instances of the previous distribution, where the instances are correlated only via the permutation $\mathcal{P}_C$.[4]

Taking this logic a step further, let $\mathbf{C}$ be an arbitrary, potentially highly structured m-gate circuit, and let $C \xleftarrow{R} \mathcal{E}_{\mathbf{C},m'}$ be a random m'-gate circuit that is functionally equivalent to $\mathbf{C}$, where $m' \geq 2m^*m$. Then it is plausible that any $(m' - m^*)$-gate portion of C is indistinguishable from a random circuit of the same length. Furthermore, let $\mathbf{C}_1, \mathbf{C}_2$ be m_1-gate prefix and m_2-gate suffix of $\mathbf{C}$, $m_1 + m_2 = m$. Then it seems plausible that:

$$C : C \xleftarrow{R} \mathcal{E}_{\mathbf{C},2m^*m} \overset{c}{\approx} C_1|C_2 : \\ C_1 \xleftarrow{R} \mathcal{E}_{(\mathbf{C}_1|R),2m^*m_1}; C_2 \xleftarrow{R} \mathcal{E}_{(R^\dagger|\mathbf{C}_2),2m^*m_2}; R \xleftarrow{R} \mathcal{C}_{n,m^*},$$

namely that a random $2m^*m$-gate circuit that's functionally equivalent to $\mathbf{C}$ is indistinguishable from a random $2m^*m_1$-gate circuit C_1 that computes the permutation $\mathbf{C}_1|R$ for a random m^*-gate circuit R, followed by a random $2m^*m_2$-gate circuit C_2 that computes $R^\dagger|\mathbf{C}_2$. We call this assumption the **Split-Circuit Pseudorandomness (SCP)** assumption (see also Fig. 1).[5]

Discussion. We stress that the SCP assumption may not be efficiently falsifiable even if false. This is so since it considers indistinguishability of distributions

[4] One consequence of the correlation is that here m^* needs to be large enough not only to make Gower's conjecture work, but also to make sure that two random instantiations of the same permutation look sufficiently different from each other. However, this distinction appears to become moot when $m^* = \tilde{\Omega}(|\mathbb{B}_n|)$. See more discussion within.

[5] The constant 2 above is clearly arbitrary and was only used to underline the progression of the logic underlying the assumption. Also, the above formulation actually corresponds to a strong version of the SCP assumption, whereas a somewhat weaker version suffices for our treatment. See more details within.

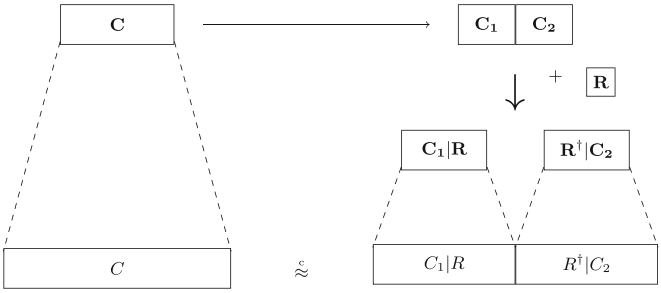

Fig. 1. The Split Circuit Pseudorandomness (SCP) assumption. Circuit $\mathbf{C}$ (top left) is an arbitrary n-wire, m-gate reversible circuit. Circuits $\mathbf{C_1}$ and $\mathbf{C_2}$ at the top right are the m_1-gate prefix and m_2-gate suffix of $\mathbf{C}$ (with $m_1 + m_2 = m$), and R is a random $m^{\#}$-gate circuit, where $m^{\#}$ depends only on n and the security parameter, while m is an arbitrarily large polynomial. Circuits $C_1|R$ and $R^{\dagger}|C_2$ at the bottom right are random $m^{\#}m_1$-gate and $m^{\#}m_2$-gate circuits that are functionally equivalent to $\mathbf{C_1}|R$ and $R^{\dagger}|\mathbf{C_2}$, respectively. The assumption states that the concatenation of these two circuits is computationally indistinguishable from a random $m^{\#}m$-gate circuit that's functionally equivalent to $\mathbf{C}$ (bottom left), in spite of the fact that each one of $C_1|R$ and $R^{\dagger}|C_2$, taken separately, computes a pseudorandom permutation.

which are not known to be efficiently samplable. In fact, many of these distributions are not even efficiently recognizable - e.g. we don't have a feasible way to know for sure that a given circuit computes even the identity permutation.

At the same time, this assumption is a fairly minimal instantiation of a more general intuition regarding the pseudorandomness of sufficiently long random circuits with fixed functionality. This intuition essentially states that there exist $n^*_\kappa, m^*_\kappa \in \text{poly}(\kappa)$ such that for any large enough value of the security parameter κ, any $m \geq m^*_\kappa$, and any fixed circuit $\mathbf{C} \in \mathcal{C}_{n^*_\kappa, m}$, a random $O(m^*_\kappa m)$-gate circuit C that is functionally equivalent to $\mathbf{C}$ essentially renders "all information on both the structure and functionality of short and medium range segments of $\mathbf{C}$" inaccessible to polytime observers, while keeping the overall functionality intact.

We note that the SCP assumption appears closely related - at least in spirit - to assumptions regarding the hardness of distinguishing between random strings with different Kolmogorov (respectively, MCSP) complexities (see e.g. [LP20, LP21, IRS22, BLMP23, ILW23]). While some initial connections are made within, further exploration and exploitation of these apparent connections may be of independent interest.

1.3 New Notions of Security for Circuit Obfuscation

Next, we sketch the definition of RIO obfuscation (which relaxes IO). We also define another variant, called random output (RO) obfuscation, which will be useful for presenting and analyzing our constructions. (As we'll see, RO obfuscation

for some circuit classes will provide stronger guarantees than IO for these classes; still, IO for all circuits and RO for all circuits will end up being equivalent.)

Let $\mathcal{C}_n$ denote the set of all n-wire reversible circuits. A transformation $O : \mathcal{C}_n \to \mathcal{C}_n$ is functionality-preserving if $O(C)$ and C are functionally equivalent for any $C \in \mathcal{C}_n$.

A functionality-preserving transformation $O : \mathcal{C}_n \to \mathcal{C}_n$, is a **random output indistinguishability (RO)** obfuscator for a set $\mathbb{C} \subseteq \mathcal{C}_n$ of circuits and inner-stretch function ξ if there exists an efficient "post-processing algorithm" π such that for any m-gate circuit $C \in \mathbb{C}$ we have:

$$O(C) \stackrel{c}{\approx} \pi(\widehat{C}) : \widehat{C} \stackrel{R}{\leftarrow} \mathcal{E}_{C,\xi(n,m)}.$$

It can be verified that if $\xi(n, m) = m$ then RO obfuscation coincides with standard indistinguishability obfuscation (IO). (In particular, in this case we can set $\pi = O$ without losing generality.) When $\xi(n, m) > m$, RO obfuscation for some classes of circuits becomes non-trivial to obtain even in situations where IO for these classes is trivial (e.g. when the input circuit C is the only one with the same size and functionality in that class). Furthermore, together with the SCP assumption, RO obfuscation with large inner-stretch (namely, when $\xi(n, m) = \Omega(m^* m)$) guarantees that both the structure and the functionality of any not-too-large portion of C are essentially lost. Still, observe that RO obfuscation for any class of circuits can be constructed from IO for all circuits, by appropriately padding the input circuit before obfuscating it.

A functionality-preserving transformation $O : \mathcal{C}_n \to \mathcal{C}_n$, is a **random input and output (RIO)** obfuscator with respect to $\mathcal{C}_{n,m}$ if the following two requirements hold:[6]

1. $(O(C), O(C)) : C \stackrel{R}{\leftarrow} C_{n,m} \stackrel{c}{\approx} (O(C), O(C')) : C \stackrel{R}{\leftarrow} C_{n,m}; C' \stackrel{R}{\leftarrow} \mathcal{E}_{C,m}$
2. For any "advice" function Z with poly-length output we have

$$O(C), Z(\mathcal{P}(C_1, C_2)) \stackrel{c}{\approx} O(C'), Z(\mathcal{P}(C_1, C_2))$$

where $C \stackrel{R}{\leftarrow} \mathcal{C}_{n,m}$, $C' \stackrel{R}{\leftarrow} \mathcal{E}_{C,m}$, $\mathcal{P}(C)$ denotes the permutation computed by circuit C, and C_1 and C_2 are the $m/2$-gate prefix and suffix of C, respectively.

The two requirements from an RIO obfuscator are incomparable and capture different security aspects: The first requirement makes sure that two obfuscated versions of the same random circuit C do not look "too much alike" relative to the obfuscated versions of two random circuits C, C' with the same functionality and length.

The second requirement makes sure that $O(C)$ remains indistinguishable from $O(C')$ even when given arbitrary polysize advice that's computed given the permutations computed by C_1 and C_2, the first and second halves of C.

[6] For simplicity we present here the definition only for the special case where there is no inner-stretch requirement and the input is uniform. A more general formulation appears within.

Note that in the left hand side distribution, the permutation computed by C_1 is the same as the permutation computed by the first half of the obfuscated circuit C. In contrast, in the right hand side experiment the permutation computed by the first half of the obfuscated circuit C' is different than the permutation computed by C_1.

It is stressed that neither of the two RIO requirements considers a distinguisher that has access to the input circuit C. This stands in sharp contrast to the case of IO (and RO) where the distinguisher sees both C and $O(C)$, making RIO potentially easier to obtain—not only than IO, but also than IO for random circuits.

1.4 From RIO to RO for All Circuits

We show:

Theorem 1 (Informal). *If there exist RIO obfuscators for $\mathcal{C}_{n,m^*}$, where n, m^* satisfy the SCP assumption, then there exists an RO obfuscator O with large inner-stretch for all circuits in $\mathcal{C}_n$. Furthermore, if C has m gates then $O(C)$ has poly$(m^*)m$ gates.*

For the construction, we first construct the following building blocks (See Fig. 2):

- A random identity generator (RIG), which is an RO obfuscator for the identity permutation with inner-stretch $2m^*$. This is done by choosing $C \xleftarrow{R} \mathcal{C}_{n,m^*}$, then sampling $C', C'' \xleftarrow{R} O(C)$ where O is an RIO obfuscator, and finally outputting $C'|C''^{\dagger}$. Security is proven using the RIO security of O and the SCP assumption.
- A gate obfuscator GO, namely an RO obfuscator for β, per each gate $\beta \in \mathbb{B}_n$. This can be done simply by sampling random identities using the previous step, until an identity circuit that starts with β is sampled. Then, remove the leading β gate (or alternatively replace it with an identity gate) and output the result.
- A procedure for "soldering" RO-obfuscated circuits, namely combining an RO obfuscator O_1 for a circuit C_1 and an RO obfuscator O_2 for a circuit C_2 into an RO obfuscator for the circuit $C_1|C_2$. The idea is again simple: Let $\tilde{C}_1 \xleftarrow{R} O_1(C_1)$, $\tilde{C}_2 \xleftarrow{R} O_2(C_2)$. Now, let $\tilde{C}_1 = C_{1,1}|C_{1,2}$ and $\tilde{C}_2 = C_{2,1}|C_{2,2}$, where $C_{1,2}$ and $C_{2,1}$ have m^*-gates each. Now, compute $G \xleftarrow{R} O(C_{1,2}|C_{2,1})$ where O is an RIO obfuscator, and output the circuit $C_{1,1}|G|C_{2,2}$. Security is proven based on the security properties of the building blocks, using the SCP assumption. (While the proof is conceptually straightforward, care has to be taken to the fact that several of the intermediate distributions are not efficiently samplable.)

Now, to obfuscate a circuit $C = \gamma_1 \dots \gamma_m$, first sample $\Gamma_i \xleftarrow{R} \mathrm{GO}(\gamma_i)$ for $i = 1..m$, and then solder the circuit pieces one by one: Let $C_1 = \Gamma_1$, and for $i = 2..m$ let C_i be the result of soldering C_{i-1} and Γ_i. Finally output C_m.

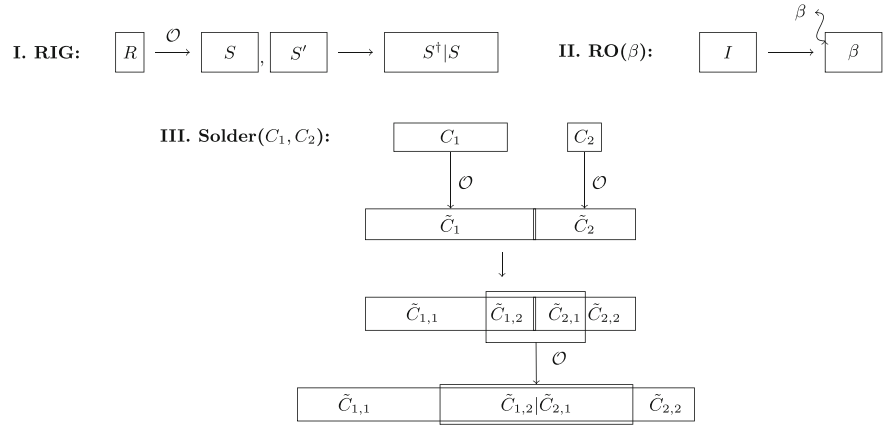

Fig. 2. The building blocks for constructing RO obfuscation for all reversible circuits from RIO obfuscation for bounded length random circuits. The first building block is random identity generators (RIGs), constructed by concatenating two RIO-obfuscated versions of a random circuit, one in reverse. The second building block is RO obfuscators for single gates, constructed by sampling a RIG with the desired first gate and removing that gate. The third building block is soldering RO-obfuscated versions of circuits C_1 and C_2 into an RO-obfuscated version of $C_1|C_2$ by concatenating the individual obfuscations and re-obfuscating the circuit segment around the seam. These basic building blocks are then iterated to solder obfuscated versions or arbitrarily long circuits.

The use of RO obfuscation with large inner-stretch for the intermediate steps in the obfuscation process (rather than, say, plain IO) is critical for this approach to work. In particular, we critically use the fact that, after each step, the intermediate circuit C_i has essentially lost "all polynomially accessible information" on its structure (i.e. on $\gamma_1 \dots \gamma_i$) *other than the overall functionality of* $\gamma_1 \dots \gamma_i$. This may be viewed as evidence for the power of RO obfuscation.

1.5 Constructing RIO Obfuscators

Reversible circuits admit a wide variety of *functionality preserving local perturbations*. For instance, given a circuit $C = \gamma_1 \dots \gamma_i \dots \gamma_{i+\ell} \dots \gamma_m$ one can replace a circuit segment $\gamma_i \dots \gamma_{i+\ell}$ with any circuit $C' = \gamma'_1 \dots \gamma'_{\ell'}$ that is functionally equivalent to $\gamma_i \dots \gamma_{i+\ell}$ (i.e. $\mathcal{P}_{C'} = \mathcal{P}_{\gamma_i \dots \gamma_{i+\ell}}$), obtaining a perturbed circuit $C'' = \gamma_1 \dots \gamma_{i-1}|C'|\gamma_{i+\ell+1} \dots \gamma_m$ that is functionally equivalent to C (i.e. $\mathcal{P}_{C''} = \mathcal{P}_C$). When ℓ, ℓ' are small enough (say, constants), it is possible to sample uniformly from all - or sufficiently many - ℓ'-gate circuits that are functionally equivalent to any given ℓ-gate circuit so as to make for effective randomization of that particular segment. It is thus tempting to explore the possibility that the space of functionally equivalent circuits within a given length is ergodic—namely that iterative replacements of randomly chosen small circuit segments with random functionally equivalent alternative segments may provide more global mix-

ing (and hence obfuscation) properties. Note that this mixing approach can be viewed as a recipe for generating random elements in a group presentation where the underlying alphabet is the set of base gates, and the set of generating words consist of the initial circuit, plus a sufficiently large set of identity circuits.

One drawback of a literal implementation of this idea is that much of the randomness in a random circuit can be effectively "factored out", say via efficiently computable cannonical representations of circuits. For instance, note that many pairs of gates $\beta, \beta' \in \mathbb{B}_n$ commute, namely $\mathcal{P}_{\beta\beta'} = \mathcal{P}_{\beta'\beta}$. (In fact, all but $O(1/n)$ of them do.) Thus applying the above process with segments of size up to $o(\sqrt{n})$ and $\ell' = \ell$ will end up only re-ordering commuting gates, almost always. However, such re-randomization is easily factored out by using a cannonical representation that fixes the order for each pair of commuting gates (say, starting from the left and using some lexicographic ordering of the gates).

A natural approach to get around the above "attack" is to consider circuit segments that are not consecutive: for instance, pick a random gate γ_i in the circuit and a random direction (left/right), and let γ_j be the nearest gate in that direction that "collides" (i.e., does not commutes) with γ_i. Then remove γ_i and γ_j, and replace them by a functionally equivalent sequence of gates (say, as in Fig. 3), placed anywhere between locations i and j. Such a strategy may appear harder to reverse, but it is again ultimately reversible (at least in and of itself) since it leaves behind clusters of "collision debris" gates that are relatively easy to identify.

A more general issue with naive realizations of local rerandomization of circuit segments is that, for most ℓ-gate circuits C, the set $\mathcal{E}_{C,\ell}$ is relatively small. (As we demonstrate within, this is in fact a general property that holds for all values of ℓ; but it is perhaps most prominent when ℓ is small.) This means that, when $\ell = \ell'$ the above process may again not provide sufficient randomization. On the other hand, when $\ell < \ell'$, the circuit would continually grow in size, which means that there is little hope to reach any stationary distribution—or to even to guarantee more basic mixing properties such as having each segment in the final circuit depend on all gates in the original circuit.

Furthermore, it is unlikely to be the case any two functionally equivalent circuits of the same size are connected via a "path", or sequence of polynomially many local transformations that are quaranteed to be functionality preserving. Indeed, if this were the case, then we would have a polysize witness for the fact that two circuits are functionally equivalent, implying NP=coNP. (Note that this rules out the very existence of such a sequence, not just the feasibility of finding one. This observation is a slight variant of a more general result by Goldwasser and Rothblum [GR14], which demonstrates, in a similar way, that perfect IO for all circuits implies NP = coNP.)

Still, these arguments leave open the possibility that a somewhat more nuanced or structured local perturbation process could actually provide sufficient "confusion and diffusion" so as to satisfy the relatively weak requirements of RIO obfuscation for random circuits that have sufficiently many gates so as to make the Gowers conjecture hold.

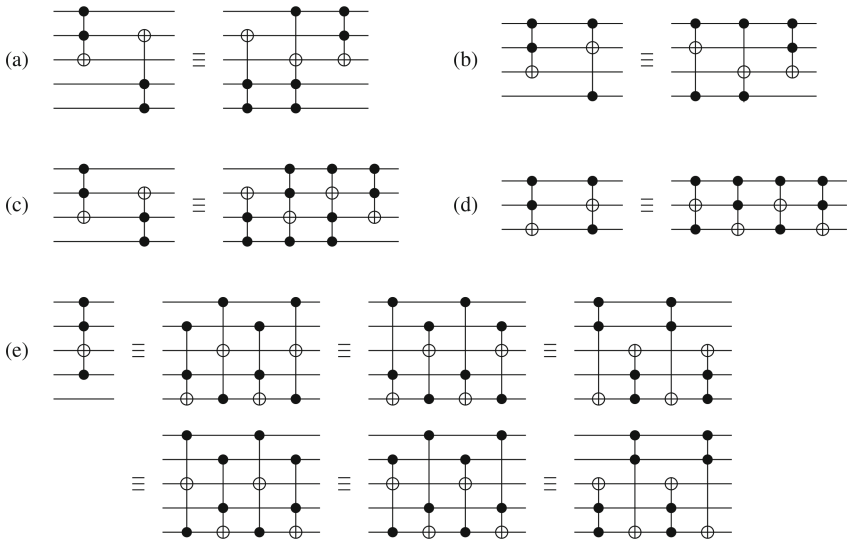

Fig. 3. Some possible replacements for the case of $\ell^{\text{OUT}} = 2$ (namely, colliding pairs of gates), for the special case where the control function is $\phi(a, b) = ab$ (namely, logical conjunction). A gate is depicted as a vertical line connecting several wires, where the control wires are identified by black dots and the active wire is identified via a circle. Panels (a) and (b) show possible replacements for one-headed collision, i.e. for the case where the active wire of one gate is also a control wire of the other gate. Panels (c) and (d) correspond to a two-head collision, when the active wires of both gates are also control wires of the other gate. Notice that in panels (a) and (c) the circuit on the right includes a 3-control gate. As shown in panel (e), this 3-control gate can be decomposed into four base gates, while using an additional wire (to be chosen out of the $n-4$ remaining wires in the circuit). Overall, in case (c) the figure depicts $6^2\binom{n-4}{2}$ replacement circuits.

We heuristically propose such a process. First, we formulate a representation of circuits that facilitates generalizing the above "colliding gates" method to identifying sets of nearby (albeit not necessarily consecutive) gates that form structured sub-circuits that are amenable to rerandomization.

Second, we split the mixing process into two stages. In the first, "inflationary" stage, the size ℓ^{OUT} of the sub-circuits to be replaced is a relatively small constant, and the size ℓ^{IN} of the replacement circuit is only slightly larger - just enough for effective re-randomization of the structure of the replaced sub-circuit while preserving its functionality. In the second, "kneading" stage, the size ℓ^{KND} of the replacement circuit is set to be identical to the size of the circuit to be replaced, and both are set to be significantly larger than ℓ^{IN}—say $\ell^{\text{KND}} = \Theta(\log \log n)$, where n is the number of wires.

In a nutshell, the rationale here is the following. The inflationary stage adds a significant amount of "random redundancy" to the circuit. (We measure the "level of redundancy" in a circuit by way of the "complexity gap", or the differ-

ence between the number of gates in the circuit and the number of gates in the smallest functionally equivalent circuit.) As noted above, this stage alone does not suffice since the complexity gap is concentrated in small sub-circuits of the overall circuit and may thus still be identifiable and removable with feasible computational overhead. Still, the structure of the replaced sub-circuits enables the kneading stage to spread the already-existing complexity gap over successively larger sub-circuits, thus making it computationally hard to localize and remove.

We provide more detailed rationale within. It is stressed however that the analysis is far from rigorous, and that the proposed process is merely an exploration meant to demonstrate the viability of the approach rather than well-analyzed candidate circuit obfuscator. We leave further analysis to future work.

1.6 Related Work

The randomizing power of permutation groups is not new to cryptography, with a prominent examples being the seminal work of Kilian that shows how to use Barrington's $\mathbb{S}_5$ representation of branching programs to randomize general NC^1 computations [Bar86, Kil88]. Kilian's randomization technique has been widely used, including in early candidate obfuscation schemes [CV13].

Alagic, Jeffery and Jordan [AJJ14] use the randomizing power of permutation groups (in the more restricted context of Braid permutations) to show unconditional "partial inditinguishability obfuscation" mechanisms for programs that are within the same equivalence class of a certain normal-form representation.

Chamon, Mucciolo and Ruckenstein [CMR22] study pseudorandomness properties of random reversible circuits, and provide evidence that as little as $m = O(n \log n)$ gates suffice for the family $\mathcal{C}_{n,m}$ to be an SPRP, when n is taken to be the security parameter.

Chamon et al. [CJMR22] use local perturbation techniques of a different flavor of the ones proposed here to construct a candidate "homomorphic pseudorandom permutation family" and use it as a basis for a symmetric homomorphic encryption scheme. It is stressed though that the security requirements needed in that application are significantly weaker than the ones needed for general program obfuscation, or even RIO obfuscation.

Finally, [CRMC23] takes a thermodynamic approach to circuit complexity, and in particular studies mixing of polynomial-sized reversible circuits of a given functionality through the iterative equilibration of concatenated short subcircuits described via local equilibrium distributions of reversible gates. In particular, that work uses the thermodynamics framework to argue that the set of functionally equivalent reversible circuits of some size is partitioned to sectors where each sector is ergodic with mixing time that's polynomial in the circuit size. In other words, that work suggests that viability of the local mixing approach as an obfuscation method reduces to the indistinguishability of random circuits from different sectors.

2 Reversible Boolean Circuits

This section recalls the model of reversible Boolean circuits and its relationship with standard Boolean circuits.

A reversible Boolean circuit C on n wires consists of a sequence of permutations $C = \gamma_1 \dots \gamma_m$ where each γ_i is a permutation on $\{0,1\}^n$, taken from a predetermined set B of *base permutations.* The permutation $\mathcal{P}_C$ computed by C is the composition of the individual permutations, $\mathcal{P}_C = \gamma_m \circ \dots \circ \gamma_1$, or in other words $C(x) = \gamma_m(\dots \gamma_1(x) \dots)$.

We concentrate on circuits where the base permutations consist of applying a Toffoli gate to three chosen wires, where a Toffoli gate is a permutation on $\{0,1\}^3$ of the form $\tau_\phi(a_1, a_2, a_3) = (a_1 + \phi(a_2, a_3), a_2, a_3)$ where $\phi : \{0,1\}^2 \to \{0,1\}$ is the *control function* of the gate. (We often refer to the three wires of a Toffoli gate as *pins*, where the first pin is *active* and the second and third pins are *non-active.*) That is, we consider the set of base permutations defined by

$$\mathbb{B}_n = \{\beta_{w_1,w_2,w_3,\phi} : w_1, w_2, w_3 \in [n]^3, w_2 \neq w_1 \neq w_3, w_2 \neq w_3, \phi : \{0,1\}^2 \to \{0,1\}\}$$

where $\beta_{w_1,w_2,w_3,\phi}(x_1 \dots x_n) = y_1 \dots y_n$ such that $(y_{w_1}, y_{w_2}, y_{w_3}) = \tau_\phi(x_{w_1}, x_{w_2}, x_{w_3})$, and $y_j = x_j$ for each $j \in [n] \setminus \{w_1, w_2, w_3\}$. (We note that, as defined above, $\mathbb{B}_n$ is actually a multi-set since $\beta_{w_1,w_2,w_3,\phi}$ and $\beta_{w'_1,w'_2,w'_3,\phi'}$ may well describe the same permutation. In fact, while there are 16 different control functions ϕ, there are roughly $8n^3$ different base permutations overall. For convenience we use the convention where only a single representative of each base permutation is used, i.e. $b_n \stackrel{\text{def}}{=} |\mathbb{B}_n| \approx 8n^3$. However this convention does not appear essential for the treatment.)

The natural evaluation of $C = \gamma_1 \dots \gamma_m$, where each $\gamma_i = \beta_{w_{1,i},w_{2,i},w_{3,i},\phi_i}$, on input $x = x_1, ..., x_n \in \{0,1\}^n$ is described iteratively as follows. For $j = 1..n$ we have $x_j^{(0)} = x_j$, and for each $i = 1..m$ we have $(x_1^{(i)} \dots x_n^{(i)} = \gamma_i(x_1^{(i-1)} \dots x_n^{(i-1)})$. The *value of wire j after gate i* is defined as $x_j^{(i)}$. It may be useful to envision reversible circuits as a sequence of n horizontal parallel wires, where each gate connects three wires, and where the computation proceeds from left to right.

Since all base permutations (or, gates) are even, reversible circuits can only compute even permutations on $\{0,1\}^n$. Still, considering only circuits of the above form does not limit the generality of the treatment. Indeed, the set $\mathbb{B}_n$ of gates generates all even permutations over $\{0,1\}^n$, namely the alternating group $\mathbb{A}_{2^n}$ (see e.g. [CG75, Bro04]).

Furthermore, any circuit C with α input wires, β output wires, μ NAND gates and width ω can be transformed to a reversible circuit C' on $n = \alpha + \beta + \delta$ wires and m gates, where $n = O(\omega)$ and $m = O(\mu)$, and where $C'(x, y, 0^\delta) = (x, C(x) + y, 0^\delta)$ for any $x \in \{0,1\}^\alpha, y \in \{0,1\}^\beta$ (see e.g. [Ben73, Tof80, Ben89, Bro04]). In the Appendix we show how to "harden" the standard transformation so as to guarantee that $C'(x, y, z) = (x, y, z)$ for $z \neq 0^\delta$, and how to use the hardened

transform to show that obfuscation of reversible circuits suffices for general-purpose obfuscation of all circuits[7].

Let $C^\dagger$ denote the natural inverse (or, "reverse") of circuit C. That is, if $C = \gamma_1, ..., \gamma_m$ then $C^\dagger = \gamma_m, ..., \gamma_1$. Indeed, note that $\mathcal{P}_{C|C^\dagger} = \mathcal{P}_{C^\dagger|C} = I_n$, where I_n denotes the identity permutation on $\{0,1\}^n$. This is so since the base permutations are the inverses of themselves, i.e. $\mathcal{P}_{\beta|\beta} = I_n$ for all base permutations β. (Here '|' denotes the natural concatenation, or composition, of gates or circuits.) Let $\mathcal{C}_{n,m}$ denote the set of all m-gate circuits on n wires, and let $\mathcal{C}_n = \bigcup_{m>0} \mathcal{C}_{n,m}$.

For a circuit $C = \gamma_1 \dots \gamma_m$, let $|C| = m$ denote the number of gates in C. For $i, l \in [m]$, let $C_{[i,l]} = \gamma_i \dots \gamma_{i+l(\mathrm{mod}\ m)}$ denote the l-gate segment of C that starts at the ith gate, taken modularily; in particular, $i < 0$ refers to $m - i$. We also use $C_{[i,*]}$ as a shorthand for $C_{[i+1,m-i]}$.

A note about asymptotics. Throughout we treat n, the number of wires, m, the number of gates, and the runtimes of adversaries as functions of (specifically, polynomials in) the security parameter κ. We will also be mostly interested in the regime where m is polynomial in n. While our treatment is mostly asymptotic in κ, a non-asymptotic treatment with concrete values can be naturally derived.

2.1 Reversible Circuits as a Free Group

The set $\mathcal{C}_n$ of circuits with the set $\mathbb{B}_n$ of base gates can be viewed as the free group $F_{\mathbb{B}_n}$ of reduced strings (namely, strings where any two consecutive identical characters are eliminated) over the alphabet $\mathbb{B}_n$, with the group operation being the standard concatenation followed by reduction. One can then define the following natural action of $F_{\mathbb{B}_n}$ on the alternating group $\mathbb{A}_{2^n}$ (namely the group of all even permutations on $\{0,1\}^n$): for a circuit $C \in F_{\mathbb{B}_n}$ and permutation $\pi \in \mathbb{A}_{2^n}$, let $Eval(C, \pi) = \pi' \circ \pi$ where $\pi' = \mathcal{P}_C$. That is, $Eval(C, \pi)$ returns the permutation that first computes π and then evaluates C on the result. It is easy to see that $Eval$ is a group action whose kernel is the set of all identity circuits, and where each coset consists of all functionally equivalent circuits. Viewed in this way, program obfuscation is the problem of efficiently generating a pseudorandom sample from the coset of a given circuit (restricted to some given length).

3 Hardness Assumptions

This section presents and motivates the hardness assumptions used in this work. We start off with a reminder of the standard definition of computational

[7] Note that not all 16 control functions are needed for completeness to hold. In fact, the functions $\phi(x, y) = xy$, $\phi(x, y) = x$, $\phi(x, y) = 1$ suffice. However, considering all 16 control functions will be convenient for our treatment. In particular, this way the value of the active wire of τ_ϕ for a random control function is uniformly distributed regardless of the values of the input wires. Furthermore, having the identity as a base permutation (with $\phi(x, y) = 0$) will be convenient as well. This set of permutations is also the one considered by Brodsky and Hoory [HMMR05, HB05].

indistinguishability, and a natural extension thereof. Let $\mathcal{A} = \{A_\kappa\}_{\kappa\in\mathbf{N}}$ and $\mathcal{B} = \{B_\kappa\}_{\kappa\in\mathbf{N}}$ be distribution ensembles. (More precisely, we think of each A_κ (resp., B_κ) as a sampling algorithm. The distribution is defined via the probability of obtaining each possible output value when running the algorithm on an input which is drawn uniformly from $\{0,1\}^\dagger$.) $\mathcal{A}$ and $\mathcal{B}$ are said to be computationally indistinguishable, denoted $\mathcal{A} \stackrel{c}{\approx} \mathcal{B}$, if there exists a negligible function $\varepsilon(\kappa)$ such that for any polysize family of distinguishing algorithms $\mathcal{D} = \{D_\kappa\}_{\kappa\in\mathbf{N}}$ and all large enough values of κ it holds that $\text{Prob}[D_\kappa(A_\kappa) = 1] - \text{Prob}[D_\kappa(B_\kappa) = 1] < \varepsilon(\kappa)$.

3.1 On the Distribution of Functionally Equivalent Reversible Circuits

We first take a moment to define a measure of complexity for reversible circuits and then use it to estimate the sizes and makeup of the clusters of functionally equivalent reversible circuits of a given length. This detour will be useful both as a basis for our hardness assumptions, and as a basis for the local perturbation mechanisms developed in Sect. 6.

For a permutation $P \in \mathbb{A}_{2^n}$, let $\mathcal{E}_{P,m}$ denote the set of all m-gate circuits that compute P, namely $\mathcal{E}_{P,m} = \{C \in \mathcal{C}_{n,m} : \mathcal{P}_C = P\}$. Slightly abusing notation, for a circuit C we let $\mathcal{E}_{C,m} = \mathcal{E}_{\mathcal{P}(C),m}$.

We would like to estimate the size of $\mathcal{E}_{C,m}$. Towards this, we define the **Computational Complexity** $\text{CC}(P)$ of a permutation P as the number of gates in the smallest circuit that computes P. Similarly, let $\text{CC}(C) = \text{CC}(\mathcal{P}_C)$ denote the number of gates in the smallest circuit that computes $\mathcal{P}_C$. The **complexity gap** of an m-gate circuit C is defined to be $\text{CG}(C) = m - \text{CC}(C)$.

While $\text{CC}(C)$ is clearly distinct from the Kolmogorov complexities of string representations of a circuit C, these notions have many similarities. For one, it is easy to see that $b^m > |\mathcal{E}_{P,m}| > b^{\frac{1}{2}(m-\text{CC}(P))}$ for any permutation P, where $b \approx 8n^3$ is the number of base permutations:

Claim. For any circuit $C \in \mathcal{C}_n$ and any m we have $b^m > |\mathcal{E}_{C,m}| > b^{\frac{1}{2}(\text{CG}(C))}$.

Proof. The upper bound is immediate. For the lower bound, observe that for any sequence of base permutations $\beta_1 \dots \beta_l$ where $l = (\text{CG}(C))/2$, the circuit $C\beta_1\beta_1 \dots \beta_l\beta_l$ is functionally equivalent to C.

Furthermore, for almost all circuits in $\mathcal{C}_{n,m}$ we have $b^m > |\mathcal{E}_{C,m}| > b^{m-\frac{\text{CC}(C)}{\log b}-o(1)}$:

Claim. For all but an ε-fraction of the circuits $C \in \mathcal{C}_{n,m}$ we have $|\mathcal{E}_{C,m}| > b^{m-\frac{\text{CC}(C)}{\log b}-\log(\varepsilon m \log b)}$.

Proof. Note that any string $\sigma = \{0,1\}^s$ can be interpreted as a description of a reversible circuit $\hat{\sigma}$ on n wires and m gates where $s = m \log b$. (Recall that $b \approx 8n^3$ is the number of base permutations.) Furthermore, for any such n, m,

the string σ is fully determined via a circuit δ of size $\mathrm{CC}(\hat{\sigma})$ that's functionally equivalent to $\hat{\sigma}$, plus the ordinal of $\hat{\sigma}$ among all m-gate circuits that are functionally equivalent to $\hat{\sigma}$. This means that $K(\sigma) \leq \mathrm{CC}(\hat{\sigma}) + \log(|\mathcal{E}_{\hat{\sigma},m}|)$, or equivalently that

$$|\mathcal{E}_{\hat{\sigma},m}| \geq 2^{K(\sigma)-\mathrm{CC}(\hat{\sigma})} = b^{\frac{1}{\log b}(K(\sigma)-\mathrm{CC}(\hat{\sigma}))} = b^{\frac{m}{s}(K(\sigma)-\frac{\mathrm{CC}(\hat{\sigma})}{\log b})}$$

where $K(\sigma)$ denotes the Kolmogorov complexity of σ. The bound follows by noting that $K(\sigma) > s - \log(\varepsilon s)$ for all but εs of the strings $\sigma \in \{0,1\}^s$.

Put together, Claims 3.1 and 3.1 says that, while for a random m-gate circuit C, the size of $\mathcal{E}_{C,m}$ is only moderate, the size of $\mathcal{E}_{C,m'}$ grows exponentially in m', for any given C.

Another conclusion from this state of affairs is that $|\{C \in \mathcal{C}_{n,m} : \mathrm{CC}(C) = m\}|b^{-m} \leq \mathrm{negl}(m)$, namely that the fraction of m-gate circuits whose computational complexity is m, out of all m-gate circuits, tends to zero rather quickly as m grows (see more discussion in [CRMC23].) This fact becomes handy in Sect. 6.

3.2 Hardness Assumptions Regarding Random Reversible Circuits

We present the hardness assumptions used in this work. The presentation builds on the motivation given in the Introduction. Further motivation is provided by presenting a sequence of gradually stronger assumptions culminating in the assumptions we actually use later on. This presentation will hopefully provide additional evidence for the viability of the main assumption (Assumption 4).

Limited Independence. We start by recalling the works that serve as the mathematical and intuitive basis for our analysis. Intrigued by the potential pseudorandomness of random reversible circuits, Gowers [Gow96] showed that $\mathcal{C}_{n,m}$, the family of m-gate permutations on n wires, is ε-close to being strongly t-wise independent for any $t < 2^n$ and $m = O(n^3 t^3 \log(\varepsilon^{-1}))$. That is, for any sequence of distinct values $x_1 \ldots x_t \in \{0,1\}^n$, and for $C \xleftarrow{R} \mathcal{C}_{n,m}$, the statistical distance between $C(x_1) \ldots C(x_t)$ and a random sequence of distinct values $r_1 \ldots r_t$, is at most ε. Hoory et al. [HMMR05] and later Brodsky and Hoory [HB05] improve this bound to $m = O(n^3 t^2 + n^2 t \log(\varepsilon^{-1}))$. Very recently, He and O'Donnell [HO24] and Gretta, He and Pelecanos [GHP24] have further improved the bound to $m = \tilde{O}(nt \log(\varepsilon^{-1}))$. (We note that, while Gowers considered all $8!\binom{n}{3}$ permutations on 3 wires as base permutations, all other works mentioned above consider the same set $\mathbb{B}_n$ of base permutations considered here.)

Pseudorandomness. Gowers conjectured that the family of permutations defined by m-gate reversible circuits on n wires might be pseudorandom (in the cryptographic sense) for some $m = poly(n)$. This construction can be viewed as the "quintessential block cipher" where each base permutation is an independently chosen "S-Box" and the key essentially specifies the schedule of which S-boxes to use. Indeed, the main difference between the Gowers construction and modern block ciphers such as AES is the key schedule that significantly reduces the

key size. (AES and other block ciphers contain additional linear operations over the entire state; however as evidenced by the k-wise independence results mentioned above, the Gowers construction effectively approximates such operations as well.) The k-wise independence of AES and the pseudorandomness of the Gowers construction have been studied in [LTV21,LPTV23,HO24]. We adopt this conjecture as a starting point for our investigation:

Definition 1 (Strong Pseudorandom Permutations (SPRPs)). *An ensemble $F = \{F_\kappa\}_{\kappa\in\mathbf{N}}$ of circuit families, where the family $F_\kappa \subset \mathcal{C}_{n_\kappa}$ consists of circuits on n_κ wires, is a* **strong pseudorandom permutation family** *if there exists a negligible function $\nu(\kappa)$ such that for any family of polynomial-size adversaries $\mathcal{A} = \{A_\kappa\}_{\kappa\in\mathbf{N}}$, and all large enough value of κ we have*

$$\text{Prob}[A_\kappa^{C,C^\dagger} = 1 : C \xleftarrow{R} F_\kappa] - \text{Prob}[A_\kappa^{P,P^{-1}} = 1 : P \xleftarrow{R} \mathbb{A}_{2^{n_\kappa}}] < \nu(\kappa). \quad (1)$$

Here $\mathbb{A}_{2^n}$ denotes the set of even permutations on the set $\{0,1\}^n$ and $\text{poly}(\kappa)$ denotes the set of polynomials in κ. Next we state Gowers' conjecture from the Introduction):

Assumption 1 (Polysize random reversible circuits are SPRPs [Gow96]). There exist $n^*_\kappa, m^*_\kappa \in \text{poly}(\kappa)$ such that the ensemble $F = \{F_\kappa\}_{\kappa\in\mathbf{N}}$ where $F_\kappa = \mathcal{C}_{n^*_\kappa,m^*_\kappa}$ is an SPRP.

We note that, while our analysis remains valid for any polynomial values of n^*_κ, m^*_κ, the assumption does not appear to be easy to refute even for relatively shallow circuits with $n^*_\kappa = \Theta(\kappa)$ and $m^*_\kappa = \tilde{\Theta}(\kappa)$. Additional argumentation for the viability of this assumption for the case where $m^*_\kappa = \tilde{\Theta}(n^*_\kappa)$ appears in [CMR22].

Pseudorandomness of correlated SPRPs. As a first step towards presenting our main assumption regarding pseudorandomness of split random circuits with fixed functionality, we demonstrate that a milder form of that assumption actually follows from a mild extension of Assumption 1. Rather than considering only the family of all circuits of a given lenth, the extension considers circuits that are sufficiently long prefixes of a sufficiently long random circuit that computes some fixed permutation. Specifically, let $\mathbf{Q} = \{\mathbf{Q}_\kappa\}_{\kappa\in\mathbf{N}}$ with $Q_\kappa \in \mathcal{C}_{n^*_\kappa,m_\kappa}$ be an ensemble of circuits, and let $C \xleftarrow{R} \mathcal{E}_{Q_\kappa,m}$ be a random m-gate circuit that computes Q_κ, where $m \geq m_\kappa m^*_\kappa$ for a "long enough cushion" m^*_κ, akin to the number of gates needed to obtain pseudorandomness in Assumption 1. We assume that, for any ℓ such that $m^*_\kappa \leq \ell \leq (m_\kappa - 1)m^*_\kappa$, the ℓ-gate prefix of C is an SPRP:

Assumption 2 (Prefixes of Random Circuits with Fixed Functionality are SPRPs). There exist $n^*_\kappa, m^*_\kappa \in \text{poly}(\kappa)$ such that for any ensemble $\mathbf{Q} = \{\mathbf{Q}_\kappa\}_{\kappa\in\mathbf{N}}$ of circuits with $Q_\kappa \in \mathcal{C}_{n^*_\kappa,m_{Q_\kappa}}$ for $m_{Q_\kappa} \in \text{poly}(\kappa)$, and any m_κ, ℓ_κ such that $m_\kappa \geq m_{Q_\kappa} m^*_\kappa$ and $m^*_\kappa \leq \ell_\kappa \leq m_\kappa - m^*_\kappa$, the ensemble $\{G_\kappa\}_{\kappa\in\mathbf{N}}$ where $G_\kappa = \{C_{[1,\ell_\kappa]} : C \in \mathcal{E}_{Q_\kappa,m_\kappa}\}$ is an SPRP.

We note that circuits drawn from G_κ (for some fixed Q_κ) are in general statistically far from random ℓ_κ-gate circuits.[8] Still, it appears that Assumption 2 is only a mild generalization of Assumption 1.

An immediate consequence from Assumption 2 is that, for any ensemble of fixed circuits $\{Q_\kappa\}_{\kappa\in\mathbf{N}}$ where $Q_\kappa \in \mathcal{C}_{n^*_\kappa, m_{Q_\kappa}}$, polysize adversaries can distinguish between the following two cases only with negligible advantage.

Oracle Access to a Prefix and Remainder of a Random Circuit for Q_κ: The adversary has oracle access to C_1, C_2 (and their inverses), where $C = C_1|\mathcal{C}_2$ is a random m_κ-gate circuit for $Q_\kappa \in \mathbb{A}_{2^{n^*(\kappa)}}$, where $|C_1| = \ell_\kappa$, and where $n^*_\kappa, m^*_\kappa, m_\kappa, \ell_\kappa$ satisfy the length requirements of Assumption 2.

Oracle Access to two SPRPs that Jointly Compute Q_κ: Let $\{Q_{1,\kappa}, Q_{2,\kappa}\}_{\kappa\in\mathbf{N}}$ be an ensemble of pairs of circuits where $Q_\kappa = Q_{1,\kappa}|Q_{2,\kappa}$, and let $F = \{F_\kappa\}_{\kappa\in\mathbf{N}}$ be an SPRP ensemble where $\mathcal{F}_\kappa \subseteq \mathcal{C}_{n^*}$. The adversary has oracle access to P_1, P_2 (and their inverses), where $P_1 = Q_{1,\kappa}|C$ and $P_2 = C^\dagger|Q_{2,\kappa}$, and $C \xleftarrow{R} F_\kappa$.

That is:

Claim. Let $n^*_\kappa, m^*_\kappa \in \text{poly}(\kappa)$ and $\mathbf{Q} = \{\mathbf{Q}_\kappa\}_{\kappa\in\mathbf{K}}$ be as in Assumption 2 with $Q_\kappa = Q_{1,\kappa}|Q_{2,\kappa}$, and let $F = \{F_\kappa\}_{\kappa\in\mathbf{N}}$ where $F_\kappa \subset \mathcal{C}_{n^*_\kappa}$ be an SPRP. Then for any m_κ, ℓ_κ s.t. $m_{Q_\kappa} m^*_\kappa \le m_\kappa$ and $m^*_\kappa \le \ell_\kappa \le m_\kappa - m^*_\kappa$ there exists a negligible function $\nu(\kappa)$ such that for any family of polynomial-size adversaries $\mathcal{A} = \{A_\kappa\}_{\kappa\in\mathbf{N}}$, and all large enough value of κ we have

$$\begin{aligned}\text{Prob}[A_\kappa^{C_1, C_1^\dagger, C_2, C_2^\dagger} = 1 : C \in \mathcal{E}_{Q_\kappa, m_\kappa}; C_1 = C_{[1,\ell_\kappa]}, C_2 = C_{[\ell_\kappa, *]}] - \\ \text{Prob}[A_\kappa^{P_1, P_1^{-1}, P_2, P_2^{-1}} = 1 : C \xleftarrow{R} F_\kappa; P_1 = Q_{1,\kappa}|C; P_2 = C^\dagger|Q_{2,\kappa}] < \nu(\kappa).\end{aligned} \tag{2}$$

Proof. Since $\{F_\kappa\}_{\kappa\in\mathbf{N}}$ is an SPRP ensemble then so is the ensembles $\{P_{1,\kappa}|C : C \xleftarrow{R} F_\kappa\}_{\kappa\in\mathbf{N}}$. It follows that:

$$\text{Prob}[A_\kappa^{C_1, C_1^\dagger} = 1 : C \in \mathcal{E}_{Q_\kappa, m_\kappa}; C_1 = C_{[1,\ell_\kappa]}] - \text{Prob}[A_\kappa^{P_1, P_1^\dagger} = 1 : P_1 \xleftarrow{R} F_\kappa] < \nu(\kappa).$$

The claim follows by observing that oracle access to the last two oracles in (2), namely either $C_2, C_2^\dagger$ in the left hand side experiment or P_2, P_2^{-1} in the

[8] As a simple example, compare a random n-wire, $2m$-gate circuit R that computes the identity permutation I_n to a circuit $C_1|C_2$ where C_1 is a random m-gate circuit and C_2 is a random m-gate circuit such that $C_1|C_2$ computes I_n. Observe that R_1, the m-gate prefix of R, is more likely to compute a permutation that's computed by many m-gate circuits, or in other words a permutation with smaller circuit complexity than C_1. (Indeed, let $\alpha \in \mathcal{C}_{n,m}$. Then $\Pr[C_1 = \alpha] = b^{-m}$ (where b is the number of gates on n wires), whereas $\Pr[R_1 = \alpha]$ is the number of m-gate circuits R_2 such that $\mathcal{P}_{\alpha|R_2} = I_n$ divided by the number of $2m$-gate identity circuits, namely $|\mathcal{E}_{\alpha,m}|/|\mathcal{E}_{I_n,2m}|$. By Claim 3.1, for most α the latter probability is proportional to $b^{-\text{CC}(\alpha)}$.).

right hand side experiment, can be emulated given oracle access to the first two oracles in that experiment and advice in the form of a polysize circuit C_{Q_κ} that computes Q_κ. (Specifically, let O_1, O_2, O_3, O_4 denote the four oracles. Then, $O_3(x) = C_{Q_\kappa}(y)$ where $y = O_2(x)$. Similarly, $O_4(x) = O_1(y)$ where $y = C^\dagger_{Q_\kappa}(x)$.)

Pseudorandomness of Split Random Ciruits with Fixed Functionality. We now turn to considering observers that, rather than only having oracle access to the permutations in (2), have access to a *random circuit* (of a certain size) that computes each permutation. Clearly, having access to a polysize circuit that computes a permutation provides significantly more "computational power" than oracle access to the permutation (for one, the permutation is now easily distinguishable from a random permutation). Still, intuitively, the added power provided by *sufficiently long* random circuits that compute the two permutations in question (either $\mathcal{P}_{C_1}, \mathcal{P}_{C_2}$ or alternatively P_1, P_2) should not be of any help in distinguishing (2). This intuition is formalized in the next assumption, which states that for any ensemble of fixed permutations $\{Q_\kappa\}_{\kappa\in\mathbf{N}}$, which are defined by way of an ensemble of pairs of polysize circuits $\{P_{1,\kappa}, P_{1,\kappa}\}_{\kappa\in\mathbf{N}}$ where $P_{i,\kappa} \in C_{n_\kappa, m_{i,\kappa}}$, $i = 1, 2$, and $\mathcal{P}_{P_{1,\kappa}|P_{1,\kappa}} = Q_\kappa$, polysize adversaries can distinguish between the following distributions only with negligible advantage.

- A circuit of the form $\widehat{C}_1|\widehat{C}_2$ where $\widehat{C}_1$ is a random $\ell_{1,\kappa}$-gate circuit that computes $\mathcal{P}_{P_{1,\kappa}|C}$, where $C \xleftarrow{R} F_\kappa$ for an SPRP ensemble $\{F_\kappa\}_{\kappa\in\mathbf{N}}$, where $\ell_{1,\kappa}$ is larger than $(m_{1,\kappa} + |C|)$ by a sufficiently large margin, $\widehat{C}_2$ is a random $\ell_{2,\kappa}$-gate circuit that computes $\mathcal{P}_{C^\dagger|P_{2,\kappa}}$ and $\ell_{2,\kappa}$ is larger than $(m_{2,\kappa} + |C|)$ by a sufficiently large margin.
- A random $(\ell_{1,\kappa} + \ell_{2,\kappa})$-gate circuit $\widehat{C}$ that computes Q_κ.

A bit more formally:

Assumption 3 (Split Pseudorandom Circuits are Pseudorandom (SPCP)). For any SPRP ensemble $F = \{F_\kappa\}_{\kappa\in\mathbf{N}}$ where $F_\kappa \subset \mathcal{C}_{n_\kappa, m_\kappa}$ there exist $m^\#_\kappa \in \text{poly}(\kappa)$ such that for any ensemble of pairs of circuits $\mathbf{Q} = \{\mathbf{P_1}_{,\kappa}, \mathbf{P_2}_{,\kappa}\}_{\kappa\in\mathbf{N}}$ where $P_{i,\kappa} \in C_{n_\kappa, m_{i,\kappa}}$, and any $\ell_{1,\kappa}, \ell_{2,\kappa}$ where $\ell_{i,\kappa} \geq m_{i,\kappa} m^\#_\kappa$, $i = 1, 2$, we have:

$$\{\widehat{C}_1|\widehat{C}_2 : C \xleftarrow{R} F_\kappa; \widehat{C}_1 \xleftarrow{R} \mathcal{E}_{(P_{1,\kappa}|C),\ell_{1,\kappa}}; \widehat{C}_2 \xleftarrow{R} \mathcal{E}_{(C^\dagger|P_{2,\kappa}),\ell_{2,\kappa}}\}_{\kappa\in\mathbf{N}} \overset{c}{\approx} \{\widehat{C} : \widehat{C} \xleftarrow{R} \mathcal{E}_{(P_{1,\kappa}|P_{2,\kappa}),\ell_{1,\kappa}+\ell_{2,\kappa}}\}_{\kappa\in\mathbf{N}}. \quad (3)$$

In the present work we only need a restricted variant of this assumption, where F is the family of all m^*_κ gate circuits from Assumption 1. Still, the more general statement appears to more closely match the intuition for the nature of the hardness.

Finally, we combine Assumptions 1 and 3 to one:

Assumption 4 (Split Circuit Pseudorandomness (SCP):). There exist n^*_κ, $m^*_\kappa \in \text{poly}(\kappa)$ that satisfy Assumption 1, as well as $m^\#_\kappa \in poly(\kappa)$ that satisfies Assumption 3 with respect to the SPRP in Assumption 1.

We also consider a somewhat stronger variant of the SCP assumption, where $m^*_\kappa = m^\#_\kappa$. To see why this variant is stronger, consider again the case of comparing a random n_κ-wire, $m^\#_\kappa$ gate identity circuit $R \xleftarrow{R} \mathcal{E}_{I_{n_\kappa} m^\#_\kappa}$ to the split version $C'|C''^\dagger : C \xleftarrow{R} \mathcal{C}_{n_\kappa, m^*_\kappa}; C', C'' \xleftarrow{R} \mathcal{E}_{C, m^\#_\kappa}$, and recall that, when $m^*_\kappa = m^\#_\kappa$, the split version tends to be skewed towards circuits C whose computational complexity is higher than that of R_1, the m^*_κ-gate prefix of R, or in other words $|\mathcal{E}_{C,m^*_\kappa}| < |\mathcal{E}_{R_1,m^*_\kappa}|$. (See the exposition in Footnote 8.) This also means that C'' is likely to be "more similar" to C' than R_2 to R_1, making distinguishing R from $C'|C''^\dagger$ potentially easier than distinguishing R_1 from C' alone. When $m^\#_\kappa$ grows relative to m^*_κ, this discrepancy tapers off and $\mathrm{CC}(C)$ (which is at most m^*_κ) eventually drops below $\mathrm{CC}(R_1)$ (which keep growing with $m^\#_\kappa$). Furthermore, the discrepancy between $|\mathcal{E}_{C,m^*_\kappa}|$ and $|\mathcal{E}_{R_1,m^*_\kappa}|$ is prominent only when $m^*_\kappa < b$. When $m^*_\kappa \gg b$, e.g. $m^*_\kappa = \Omega(n^4)$, we have that $|\mathcal{E}_{C,m^*_\kappa}|$ is sufficiently large so as to make the discrepancy moot.[9]

On the other hand, we note that this somewhat stronger assumption enables demonstrating that a weaker variant of RIO obfuscation suffices for obtaining full-fledged obfuscation for all circuits.

Assumption 5 (Strong Split Circuit Pseudorandomness (SSCP):). Assumption 4 holds with $m^*_\kappa = m^\#_\kappa$.

4 Notions of Obfuscation for Reversible Circuits

A (randomized) transformation $O : \mathcal{C}_n \to \mathcal{C}_n$ on reversible circuits has **stretch** σ if for any $C \in \mathcal{C}_{n,m}$ we have $O(C) \in \mathcal{C}_{n,m+\sigma(n,m)}$. O is said to be **functionality preserving** on a set $\mathbb{C}$ of circuits if $\mathcal{P}_{O(C)} = \mathcal{P}_C$ for any $C \in \mathbb{C}$. An **obfuscator** $\mathcal{O} = \{O_\kappa\}_{\kappa\in\mathcal{N}}$ for $\mathbb{C} = \{\mathbb{C}_\kappa\}_{\kappa\in\mathbf{N}}$ is an ensemble of transformations on reversible circuits where O_κ is functionality preserving on $\mathbb{C}_\kappa$. We start by recalling the standard definition of Indistinguishability obfuscation (IO):

Definition 2 (Indistingiushability Obfuscation (IO):). *An obfuscator $\mathcal{O} = \{O_\kappa\}_{\kappa\in\mathcal{N}}$ is an* **indistingusihability obfuscator (IO)** *for $\mathbb{C} = \{\mathbb{C}_\kappa\}_{\kappa\in\mathbf{N}}$ if for any ensemble of pairs of circuits $\{C_{0,\kappa}, C_{1,\kappa}\}_{\kappa\in\mathbf{N}}$ that are equal size (i.e., $C_{0,\kappa}, C_{1,\kappa} \in \mathbb{C}_\kappa \cap \mathcal{C}_{n_\kappa,m_\kappa}$ for some n_κ, m_κ) and functionally equivalent (i.e. $\mathcal{P}_{C_{0,\kappa}} = \mathcal{P}_{C_{1,\kappa}}$ for all κ), we have*

$$\{\mathcal{O}_\kappa(C_{0,\kappa})\}_{\kappa\in\mathbf{N}} \overset{c}{\approx} \{\mathcal{O}_\kappa(C_{1,\kappa})\}_{\kappa\in\mathbf{N}}.$$

An alternative and equivalent formulation of this definition requires that $\mathcal{O}_\kappa(C) \approx_c R_C$ for any circuit $C \in \mathbb{C}_\kappa$, where R_C is a circuit drawn from some (not necessarily efficiently computable) **reference distribution** that depends only on $\mathcal{P}_C$ and the size of C:

[9] Observe that the computational complexity of a random n=wire, m=gate circuit C is at most $\tilde{\Theta}(m/n^2)$. Indeed, it is easy to verify that a each gate γ_i cancels out with an earlier identical gate $\gamma_j = \gamma_i$ for some $j < i$ with probability $\Theta(n^{-2})$. By Claim 3.1, this means that if $C \xleftarrow{R} \mathcal{C}_{n,n^4}$ then $|\mathcal{E}_{C,n^4}| > b^{n^2}$.

Definition 3 (IO - alternative formulation:). *An obfuscator* $\mathcal{O} = \{O_\kappa\}_{\kappa\in\mathcal{N}}$ *is an* **indistingusihability obfuscator (IO)** *for* $\mathbb{C} = \{\mathbb{C}_\kappa\}_{\kappa\in\mathbf{N}}$ *if there exists a (not necessarily polytime) sampling algorithm* $\mathcal{D}$ *such that for any ensemble* $C = \{C_\kappa\}_{\kappa\in\mathbf{N}}$ *of circuits such that* $C_\kappa \in \mathbb{C}_\kappa \cap \mathcal{C}_{n_\kappa,m_\kappa}$ *we have:*

$$\{O_\kappa(C_\kappa)\}_{\kappa\in\mathbf{N}} \overset{c}{\approx} \{R : R \overset{R}{\leftarrow} \mathcal{D}(\kappa, m_\kappa, \mathcal{P}_{C_\kappa})\}_{\kappa\in\mathbf{N}}.$$

Claim. An obfuscator satisfies Definition 3 for an ensemble $\mathbb{C}$ of circuits iff it satisfies Definition 2 for $\mathbb{C}$. □

This alternative formulation provides a stepping stone towards presenting two new notions of obfuscation that will be key to our construction and analysis: Random Output (RO) and Random Input (RI) obfuscation.

4.1 Random Output Obfuscators

In the rest of this work we will be mostly interested in obfuscators where the output distribution $\mathcal{D}$ is of a particular form. Ideally, we would have liked to require that the distribution $\mathcal{D}(n, m, P)$ be the uniform distribution over $\mathcal{E}_{P,m'}$ for some $m' \geq m$. However, this may be over-restrictive, as it may set an unnecessarily high bar for obfuscation schemes. (For instance, the local random perturbations technique of Sect. 6 may well be a secure obfuscation scheme for random circuits even if it outputs circuits that are distinguishable from random ones.) We thus settle for the following relaxation: We allow distributions $\mathcal{D}(\kappa, m, P)$ where the output circuit is the result of applying some polytime post-processing algorithm to a circuit $\widehat{C}$ drawn uniformly from $\mathcal{E}_{P,m'}$ for some $m' \geq m$. Indeed, on the one hand this relaxation allows obfuscation mechanisms that fall short of generating circuits that match a specific distribution, and on the other hand it still guarantees that $\pi(\widehat{C})$ is independent of the original circuit C, other than having access to $P = \mathcal{P}(C)$.

In some cases we will make the additional requirement that the post-processing algorithm be applied separately to different segments of $\widehat{C}$, e.g. $\pi = (\pi_1, \pi_2)$ where $\pi(\widehat{C}_1|\widehat{C}_2) = \pi_1(\widehat{C}_1)|\pi_2(\widehat{C}_2)$. This additional requirement will be used, together with Assumption 4, to argue that that a certain segment of an obfuscated circuit is "computationally independent" even from the overall functionality of the circuit.

Definition 4 (Random Output Obfuscators). *An IO obfuscator* $\mathcal{O} = \{O_\kappa\}_{\kappa\in\mathcal{N}}$ *for* $\mathbb{C} = \{\mathbb{C}_\kappa\}_{\kappa\in\mathbf{N}}$ *is a* **Random Output Indistinguishability (RO)** *obfuscator with inner-stretch function* $\xi : \mathbf{N}^3 \to \mathbf{N}$ *and post-processing algorithm* $\pi : \mathcal{C}_{n_\kappa} \to \mathcal{C}_{n_\kappa}$ *if for any ensemble* $\{C_\kappa\}_{\kappa\in\mathbf{N}}$ *of circuits where* $C_\kappa \in \mathbb{C}_\kappa \cap \mathcal{C}_{n_\kappa}$ *we have:*[10]

$$\{O_\kappa(C_\kappa)\}_{\kappa\in\mathbf{N}} \overset{c}{\approx} \{\pi(\widehat{C}) : \widehat{C} \overset{R}{\leftarrow} \mathcal{E}_{C_\kappa,\xi(\kappa,n_\kappa,|C_\kappa|)}\}_{\kappa\in\mathbf{N}}.$$

[10] Note that the overall stretch of $\mathcal{O}$ is the composition of the inner-stretch function ξ and the stretch of the post-processing algorithm π. That is, if $\pi : \mathcal{C}_{n_\kappa,m'_\kappa} \to \mathcal{C}_{n_\kappa,\tau(\kappa,n,m'_\kappa)}$ then the stretch of $\mathcal{O}$ is $\sigma(\kappa, n_\kappa, m_\kappa) = \tau(\kappa, n_\kappa, \xi(\kappa, n_\kappa, m_\kappa))$.

The inner-stretch function ξ captures the "effective stretch" of the obfuscator. That is, if $\mathcal{O}$ has inner stretch ξ and $\tilde{C} = O_\kappa(C)$, where $C \in \mathcal{C}_{n,m}$, then $\tilde{C}$ provides "effectively the same obfuscation guarantees" as would a random circuit in $\mathcal{E}_{C,\xi(\kappa,n,m)}$. This is so in spite of the fact that $\mathcal{O}$ might have longer stretch, and $\tilde{C}$ might not look random at all. In particular, note that RO obfuscation where $\xi(\kappa, n_\kappa, m_\kappa) - m_\kappa = \Omega(\kappa)$ provides a meaningful security guarantee (and may be challenging to obtain) even when the input circuit is the only circuit with the same functionality and length in the class $\mathbb{C}$. (In particular recall that, by Claim 3.1, for each $C_\kappa \in \mathbb{C}_\kappa \cap \mathcal{C}_{n_\kappa,m_\kappa}$ we have that the size of $\mathcal{C}_{C_\kappa,\xi(\kappa,n_\kappa,m_\kappa)}$ is exponential in κ.) In contrast, plain IO is meaningless in such cases.

Separable RO obfuscators. The following variant of RO obfuscators will be useful for our soldering-based construction. An RO obfuscator $\mathcal{O}$ is called m_κ**-left-separable** if:

1. The computational complexity of the m_κ-gate prefix of obfuscated circuits is not too high: for any C, $\mathrm{CC}((O_\kappa(C))_{[1,m_\kappa]}) \leq m_\kappa/2$.
2. The post-processing algorithm is of the form $\pi = (\pi_1, \pi_2)$ where $\pi(C) = \pi_1(C_{[1,m_\kappa]})|\pi_2(C_{[m_\kappa,*]})$.

Right-separable obfuscators are defined analogously. (Formally, obfuscator $\mathcal{O}$ is m_κ**-right-separable** if $\mathcal{O}^\dagger$ is left-separable, where $f^\dagger(C) = (f(C^\dagger))^\dagger$ for a function $f : \mathcal{C} \to \mathcal{C}$.) An m_κ**-separable** obfuscator is both m_κ-left-separable and m_κ-right-separable.

Observe that if $\mathcal{O} = \{O_\kappa\}_{\kappa\in\mathcal{N}}$ is an m_κ-left-separable RO obfuscator then $\mathcal{O}^\dagger = \{O^\dagger_\kappa\}_{\kappa\in\mathcal{N}}$ is an m'_κ-right-separable RO obfuscator (and vice versa).

4.2 Random Input and Output Obfuscators

Here we consider obfuscators (namely, functionality preserving transformations on circuits) where security is required only with respect to circuits drawn from a specific distribution. Furthermore, in contrast with IO where security must hold against an observer who sees both the plaintext circuit and the obfuscated circuit, here the observer sees only one or more obfuscated circuits, plus some limited auxiliary information on the plaintext circuit. More specifically, we consider two alternative (and incomparable) security requirements, made with respect to a circuit C chosen from some base distribution $\mathcal{R}_\kappa$ over $\mathcal{C}_{n_\kappa,2m_\kappa}$, and an output distribution $\mathcal{D}(\kappa, 2m_\kappa, \mathcal{P}_C)$:

1. Two obfuscated versions C should not look "too much alike" compared to two independent draws from the underlying distribution $\mathcal{D}(\kappa, 2m_\kappa, \mathcal{P}_C)$. In other words, the observer should not be able to distinguish between two obfuscated versions of C and two draws from $\mathcal{D}(\kappa, 2m_\kappa, \mathcal{P}_C)$.
2. An obfuscated version of C should hide the "midway functionality" of C, namely the permutation computed by the first m_κ-gate block of of C. More specifically, the observer should not be able to distinguish between an obfuscated version of C and a circuit drawn from $\mathcal{D}(\kappa, 2m_\kappa, \mathcal{P}_C)$, even when given

circuits $\widehat{C}_1, \widehat{C}_2$ computed as follows. Let Z_1, Z_2 be two fixed circuits (which are tantamount to an "auxiliary input"), let $C = C_1|C_2$ where $|C_1| = m_\kappa$, and let $\widehat{C}_1 \stackrel{R}{\leftarrow} \mathcal{E}_{(Z_1|C_{[1,m]}),\lambda_\kappa|Z_1|)}$, $\widehat{C}_2 \stackrel{R}{\leftarrow} \mathcal{E}_{(C_{[m,*]}|Z_2),\lambda_\kappa|Z_2|}$ and sufficiently large "leeway" λ_κ. (That is, $\widehat{C}_1$ is a sufficiently long random circuit that's functionally equivalent to $Z_1|C_{[1,m]}$. Similarly, $\widehat{C}_2$ is a sufficiently long random circuit that's functionally equivalent to $C_{[m,*]}|Z_2$.) The rationale here is that $\widehat{C}_1$ and $\widehat{C}_2$ essentially give the observer only the ability to evaluate $Z_1|C_{[1,m]}$ and $C_{[m,*]}|Z_2$ (and their inverses) on inputs of its choice.

More formally:

Definition 5 (Random Input (RI) Obfuscators). *$\mathcal{O} = \{O_\kappa\}_{\kappa\in\mathcal{N}}$ is* **Random Input (RI)** *obfuscator for $n_\kappa, 2m_\kappa$, input distribution ensemble $\mathcal{R} = \{\mathcal{R}_\kappa\}_{\kappa\in\mathbf{N}}$ where $\mathcal{R}_\kappa \subseteq \mathcal{C}_{n_\kappa,2m_\kappa}$, and output distribution $\mathcal{D}$, if:*

I.

$$\left\{ \begin{array}{l} (C_1, C_2) : C \stackrel{R}{\leftarrow} \mathcal{R}_{n_\kappa,2m_\kappa}; \\ C_1, C_2 \stackrel{R}{\leftarrow} \mathcal{O}_\kappa(C) \end{array} \right\}_{\kappa\in\mathbf{N}} \stackrel{c}{\approx} \left\{ \begin{array}{l} (\widehat{C}_1, \widehat{C}_2) : C \stackrel{R}{\leftarrow} \mathcal{R}_\kappa; \\ \widehat{C}_1, \widehat{C}_2 \stackrel{R}{\leftarrow} \mathcal{D}(\kappa, n_\kappa, 2m_\kappa, \mathcal{P}_C) \end{array} \right\}_{\kappa\in\mathbf{N}}.$$

II. There exists a leeway function $\lambda_\kappa \in poly(\kappa)$ such that for any two circuit ensembles $\mathbf{Z_1} = \{\mathbf{Z_{1,\kappa}}\}_{\kappa\in\mathbf{N}}, \mathbf{Z_2} = \{\mathbf{Z_{2,\kappa}}\}_{\kappa\in\mathbf{N}}$ with $Z_i \in \mathcal{C}_{n_\kappa,m_{i,\kappa}}$ for some $m_{i,\kappa}$, $i = 1, 2$, and any $\lambda \geq \lambda_\kappa$ we have:

$$\left\{ \begin{array}{l} (O_\kappa(C), \widehat{C}_1, \widehat{C}_2) : \\ C \stackrel{R}{\leftarrow} \mathcal{R}_\kappa; \\ \widehat{C}_1 \stackrel{R}{\leftarrow} \mathcal{E}_{\mathcal{P}_{(C_{[1,m_\kappa]}|Z_{1,\kappa})},m_{1,\kappa}\lambda}; \\ \widehat{C}_2 \stackrel{R}{\leftarrow} \mathcal{E}_{(\mathcal{P}_{Z_{2,\kappa}|C_{[m_\kappa,*]}}),m_{2,\kappa}\lambda} \end{array} \right\}_{\kappa\in\mathbf{N}} \stackrel{c}{\approx} \left\{ \begin{array}{l} (\widehat{C}, \widehat{C}_1, \widehat{C}_2) : \\ C \stackrel{R}{\leftarrow} \mathcal{R}_\kappa; \widehat{C} \stackrel{R}{\leftarrow} \mathcal{D}(\kappa, n_\kappa, 2m_\kappa, \mathcal{P}_C); \\ \widehat{C}_1 \stackrel{R}{\leftarrow} \mathcal{E}_{\mathcal{P}_{(C_{[1,m_\kappa]}|Z_{1,\kappa})},m_{1,\kappa}\lambda}; \\ \widehat{C}_2 \stackrel{R}{\leftarrow} \mathcal{E}_{(\mathcal{P}_{Z_{2,\kappa}|C_{[m_\kappa,*]}}),m_{2,\kappa}\lambda} \end{array} \right\}_{\kappa\in\mathbf{N}}.$$

Definition 6 (Random Input & Output (RIO) Obfuscators). *An RI obfuscator $\mathcal{O} = \{O_\kappa\}_{\kappa\in\mathcal{N}}$ for n_κ, m_κ is* **Random Input Output (RIO)** *with inner-stretch function $\xi : \mathbf{N^3} \to \mathbf{N}$ and post-processing algorithm $\pi : \mathcal{C}_{n_\kappa} \to \mathcal{C}_{n_\kappa}$ if its output distribution $\mathcal{D}$ is of the form $\mathcal{D}(\kappa, n_\kappa, m_\kappa, P) = \pi(C)$ for $C \stackrel{R}{\leftarrow} \mathcal{E}_{P,\xi(\kappa,n_\kappa,m_\kappa)}$.*

Requirements (I) and (II) appear to be incomparable. Furthermore, each use of RIO obfuscators within our construction needs only one of the two requirements, with respect to a specific input distribution. This means that in principle one could have two different constructions of RIO obfuscation, where each construction is geared towards realizing only one of the two requirements. Still, the rationale for the validity of the obfuscation algorithm described in Sect. 6 applies in the same way to both properties (see discussion there).

5 From RIO Obfuscation to RO for All Circuits

This section presents the construction of RO obfuscators for all circuits from RIO obfuscators. More specifically, Let $n^*_\kappa, m^*_\kappa, m^\#_\kappa$ be length functions that satisfy Assumption 4. Our starting point is two obfuscators, O_1 and O_2, such that:

- O_1 is an RIO obfuscator that satisfies property I with respect to the uniform input distribution $C \overset{\text{R}}{\leftarrow} \mathcal{C}_{n^*_\kappa, m^*_\kappa}$, with inner-stretch $\xi(\kappa, n^*_\kappa, m^*_\kappa) = m^\#_\kappa$ and with post-processing algorithm π.
- O_2 is an an RIO obfuscator that satisfies property II with respect to the input distribution $C = \pi(C')|\pi^\dagger(C'') : C', C'' \overset{\text{R}}{\leftarrow} \mathcal{C}_{n^*_\kappa, m^*_\kappa}$ and leeway $\lambda_\kappa \leq m^\#_\kappa$.

That is, we show:

Theorem 2. *Let $n^*_\kappa, m^*_\kappa, m^\#_\kappa$ be length functions that satisfy Assumption 4. If there exist algorithms O_1, π, O_2 such that:*

- *O_1 satisfies property I of RIO obfuscation for input distribution ensemble $\{C : C \overset{R}{\leftarrow} \mathcal{C}_{n^*_\kappa, m^*_\kappa}\}_{\kappa \in \mathbf{N}}$, with inner-stretch $\xi(\kappa, n^*_\kappa, m^*_\kappa) = m_\kappa \geq m^\#_\kappa$ and post-processing algorithm π,*
- *O_2 is an an RIO obfuscator that satisfies property II with respect to the input distribution $C = \pi(C')|\pi^\dagger(C'') : C', C'' \overset{R}{\leftarrow} \mathcal{C}_{n^*_\kappa, m^*_\kappa}$ and leeway $\lambda_\kappa \leq m^\#_\kappa$.*

then there exists an RO obfuscator $\mathcal{O}$ for all reversible circuits. Furthermore, the inner-stretch of $\mathcal{O}$ for m-gate circuits is $\Omega(m^\#_\kappa m)$.

An overview of the obfuscation algorithm appears in the Introduction. We present the construction and its analysis in four steps. First, we show how to construct RO obfuscators for the identity function, with some specific parameters. (We call such obfuscators *pseudrandom identity generators.*)

Next we use random identity generators to construct RO obfuscators for single gate circuits.

Nest we show how to use RIO obfuscators with the above parameters to combine, or "solder" obfuscated circuits to obtain obfuscated versions of the concatenation of these circuits.

Next we combine the last two steps to construct full-fledged RO obfuscation for all reversible circuits.

The Appendix of [CCMR24] demonstrates how indistinguishability obfuscator for all Boolean circuits can be obtained using an indistinguishability obfuscator for all reversible circuits.

5.1 Random Identity Generators

Random identity generators (RIGs) are separable RO obfuscators for the identity permutation with specific parameters: Let I_{n_κ} denote the identity permutation on n_κ wires. An (n_κ, m_κ)-RIG is an m_κ-separable RO obfuscator for I_{n_κ} with inner-stretch $\xi(\kappa, n_\kappa, 1) \geq 2m_\kappa$.

In other words, an RIG is a sampling algorithm that, given κ, generates circuits that are indistinguishable from $\pi(C)$, where C is a random circuit with n_κ wires and $2m_\kappa$ gates that computes the identity permutation, and π is a post-processing algorithm. Furthermore, π is of the form $\pi = (\pi_1, \pi_2)$ where π_1 is applied to $C_{[1,m_\kappa]}$ and π_2 is applied to $C_{[m_\kappa,*]}$, and the computational complexities of both $C_{[1,m_\kappa]}$ and $C_{[m_\kappa,*]}$ are less than $m_\kappa/2$,

Definition 7 (Random Identity Generators). *An algorithm $\{G_k\}_{\kappa\in\mathcal{N}}$ is an (n_κ, m_κ)-**RIG** if it is an m_κ-separable RO obfuscator for $\{I_{n_\kappa}\}_{\kappa\in\mathbf{N}}$, with inner-stretch $\xi(\kappa, n_\kappa, 1) \geq 2m_\kappa$.*

Let $n^*_\kappa, m^*_\kappa, m^\#_\kappa$ be length functions that satisfy Assumption 4. We construct an $(n^*_\kappa, 2m_\kappa)$-RIG G_κ given an obfuscator O that satisfies property I of RIO obfuscation (see Definition 5) for uniformly chosen inputs in $\mathcal{C}_{n^*_\kappa,m^*_\kappa}$, with inner-stretch ξ such that $\xi(\kappa, n^*_\kappa, m^*_\kappa) = m_\kappa$ where $m_\kappa \geq m^\#_\kappa$. The construction is straightforward:

1. Sample $C \xleftarrow{R} \mathcal{C}_{n^*_\kappa,m^*_\kappa}$
2. Sample $C', C'' \xleftarrow{R} O_\kappa(C)$
3. Output $C'|C''^\dagger$.

We show:

Claim. Let $n^*_\kappa, m^*_\kappa, m^\#_\kappa$ be length functions that satisfy Assumption 4, and let $O = \{O_\kappa\}_{\kappa\in\mathcal{N}}$ satisfy property I of RIO obfuscation for input distribution $\mathcal{R}_\kappa = \mathcal{C}_{n^*_\kappa,m^*_\kappa}$, and with inner-stretch $\xi(\kappa, n^*_\kappa, m^*_\kappa) = m_\kappa$ where $m_\kappa \geq m^\#_\kappa$. Then $G = \{G_k\}_{\kappa\in\mathcal{N}}$ described above is an (n^*_κ, m_κ)-RIG.

Proof. We show that G_κ is an m_κ-separable RO obfuscator for the identity function $\{I_{n^*_\kappa}\}_{\kappa\in\mathbf{N}}$, with inner-stretch $\xi(\kappa, n^*_\kappa, 1) = 2m_\kappa$, and with post-processing algorithm $\pi' = (\pi, \pi^\dagger)$. That is, we show:

$$\{C : C \xleftarrow{R} G_\kappa\}_{\kappa\in\mathbf{N}} = \{C'|C''^\dagger : C \xleftarrow{R} \mathcal{C}_{n^*_\kappa,m^*_\kappa}; C', C'' \xleftarrow{R} O_\kappa(C),\}_{\kappa\in\mathcal{N}} \overset{c}{\approx} \quad (6)$$

$$\{\pi(C')|\pi(C'')^\dagger : C \xleftarrow{R} \mathcal{C}_{n^*_\kappa,m^*_\kappa}; C', C'' \xleftarrow{R} \mathcal{E}_{C,m_\kappa}\}_{\kappa\in\mathcal{N}} \overset{c}{\approx} \quad (7)$$

$$\{\pi(\hat{I}_{[1,m_\kappa]})|(\pi(\hat{I}_{[m_\kappa,*]}))^\dagger : \hat{I} \xleftarrow{R} \mathcal{E}_{I_{n^*_\kappa},2m_\kappa}\}_{\kappa\in\mathcal{N}}. \quad (8)$$

Indistinguishability of experiment (6) and experiment (7) follows directly from the RIO security of O (property I). Indistinguishability of experiment (7) and experiment (8) follows from Assumption 4. Indeed, by Assumption 1, $\{C : C \xleftarrow{R} \mathcal{C}_{n^*_\kappa,m^*_\kappa}\}_{\kappa\in\mathbf{N}}$ is an SPRP. Since $|C'| = |C| = m_\kappa \geq m^\#_\kappa$, we can use Assumption 3 to conclude that:

$$\{C'|C'' : C \xleftarrow{R} \mathcal{C}_{n^*_\kappa,m^*_\kappa}; C', C'' \xleftarrow{R} \mathcal{E}_{C,m_\kappa}\}_{\kappa\in\mathcal{N}} \overset{c}{\approx} \{\widehat{J}_{[1,m_\kappa]}|(\widehat{J}_{[m_\kappa,*]})^\dagger : \widehat{J} \xleftarrow{R} \mathcal{E}_{I_{n^*_\kappa},2m_\kappa}\}_{\kappa\in\mathcal{N}}. \quad (9)$$

Now, an algorithm A_κ that distinguishes between experiments (7) and (8) can be used to distinguish between the two distributions in (9): Given a circuit

$C \in \mathcal{C}_{n,2m_\kappa}$, output $A_\kappa(\pi(C_{[1,m_\kappa]})|\pi^\dagger(C_{[m_\kappa,*]}))$. Observe that if C was drawn from the l.h.s. distribution in (9) then A_κ's input is drawn from (7) and if C was drawn from the r.h.s. distribution then A_κ's input is drawn from (8). The claim follows by transitivity of computational indistinguishability, along with verifying that G_κ is indeed both m_κ-right-separable and m_κ-left-separable.

Directly Generating Random Identities? We note that there may well be other ways to construct RIGs, other than using an RIO obfuscator that satisfies property I. Indeed, functionality-reserving obfuscation may not be needed at all; instead one might opt to "jointly generate" two circuits that are functionally equivalent and look sufficiently random otherwise. In fact we are not aware of any "barrier" to having statistically secure RIGs.

5.2 RO Obfuscation of Single Gates

Next we show how to use a random identity generator G to construct RO obfuscators of single gates, namely RO obfuscators $\mathrm{GO} = \{\mathrm{GO}_\kappa\}_{\kappa\in\mathbf{N}}$ for the set $\mathbb{C} = \{\mathbb{B}_\kappa\}_{\kappa\in\mathbf{N}}$, where $\mathbb{B}_\kappa$ is the set of base permutations on κ wires. That is, given any base permutation $\beta \in \mathbb{B}_\kappa$, algorithm $\mathrm{GO}_\kappa(\beta)$ samples circuits that are indistinguishable from $\pi(C)$ for a random circuit $C \xleftarrow{R} \mathcal{E}_{\beta,m_\kappa}$ for some $m_\kappa \in \mathrm{poly}(\kappa)$ and post-processing algorithm π with length and separability requirements that are similar to those of random identity generators (RIGs): GO_κ should have inner-stretch ξ where $\xi(\kappa,\kappa,m_\kappa) = 2m_\kappa$ with $m_\kappa \geq m^*_\kappa$; furthermore, it should be m_κ-separable.

Definition 8 (Gate Obfuscators.). *An algorithm* $\mathrm{GO} = \{\mathrm{GO}_k\}_{\kappa\in\mathcal{N}}$ *is an* m_κ**-gate obfuscator** *if, for any* $\beta \in \mathbb{B}_\kappa\}$*, we have that* $\mathrm{GO}_\kappa(\beta_\kappa)$ *is an* m_κ*-separable RO obfuscator for* β_κ*, with inner-stretch* $\xi(\kappa,\kappa,1) \geq 2m_\kappa$*.*

The construction is simple: $\mathrm{GO}_\kappa(\beta)$ keeps sampling identity circuits using G_κ until the first gate in the generated circuit is β. Once this happens, GO replaces that first gate with the identity gate β_I and outputs the resulting circuit. Note that in order for $\mathrm{GO}_\kappa(\beta)$ to terminate in polynomial time we need to further assume that the circuits generated by G_κ start with β with polynomial probability. The random identity generators constructed in this work satisfy this property unconditionally.

Claim. Let $\{G_\kappa\}_{\kappa\in\mathbf{N}}$ be an (κ, m_κ)-random identity generator such that $\mathrm{Prob}[C_{[1,1]} = \beta : C \xleftarrow{R} G_k] \in poly(\kappa)$ for all $\beta \in \mathbb{B}_\kappa$. Then GO is an m_κ-gate-obfuscator.

Proof. To see that $\mathrm{GO}_\kappa(\beta)$ is an m_κ-separable RO obfuscator for β, let $\pi = (\pi_1, \pi_2)$ be the post-processing algorithm guaranteed by Definition 7, such that

$$\{C \xleftarrow{R} G_\kappa\}_{\kappa\in\mathbf{N}} \overset{c}{\approx} \{\pi_1(C_{[1,m_\kappa]})|\pi_2(C_{[m_\kappa,*]}) : C \xleftarrow{R} \mathcal{E}_{I_\kappa,2m_\kappa}\}_{\kappa\in\mathbf{N}}. \tag{10}$$

Consider the post-processing algorithm $\pi = (\pi'_1, \pi_2)$ where $\pi'_1(C) = \beta_I | \pi_1(\beta | C)$. We argue that

$$\{\beta_I | C_{[1,*]} : C \overset{R}{\leftarrow} G_\kappa \text{ s.t. } C_{[1,1]} = \beta\}_{\kappa \in \mathbf{N}} \overset{c}{\approx} \tag{11}$$
$$\{\pi'_1(C_{[1,m_\kappa]}) | \pi_2(C_{[m_\kappa,*]}) : C \overset{R}{\leftarrow} \mathcal{E}_{\beta,2m_\kappa} \text{s.t. } C_{[1,1]} = \beta\}_{\kappa \in \mathbf{N}}.$$

Indeed, an algorithm A_κ that distinguishes between the two distributions in (11) can be used to distinguish between the two distributions in (10): given a circuit C, if $C_{[1,1]} = \beta$, output $A_\kappa(\beta_I | C_{[1,*]})$; else, output a random bit. Observe that if C was drawn from the l.h.s. distribution in (10) then, whenever $C_{[1,1]} = \beta$, we have that $C_{[1,*]}$ is drawn from the l.h.s. distribution in (11). If C was drawn from the r.h.s. distribution in (10) then, whenever $C_{[1,1]} = \beta$, we have that $C_{[1,*]}$ is drawn from the r.h.s. distribution in (11).

We note that both the efficiency and security of GO can be significantly improved with little effort: Once the first base permutation $\beta' = (w'_1, w'_2, w'_3, \phi)$ in the sampled circuit has the same control function ϕ as the given $\beta = (w_1, w_2, w_3, \phi)$, can remove β' and then"rotate" the remaining circuit so that the wires w'_1, w'_2, w'_3 will become w_1, w_2, w_3. That is, if the sampled circuit is of the form $\beta' | C$ then output the circuit C' that is the result of renaming the wires in C via the permutation $\sigma = (w_1, w'_1)(w_2, w'_2)(w_3, w'_3)$ on $[n]$. This way, the random identity generator needs to be run at most 16 times in expectation (assuming that the control function of the first gate is distributed uniformly). The expected number of samples needed can be further reduced (for "nice" post-processing functions) by noting that any circular shift of an identity circuit is an identity circuit.

5.3 Soldering Obfuscated Circuits

Next we show how to combine (or, "solder") obfuscated circuits to obtain obfuscated versions of the concatenation of these circuits. Specifically, let $n^*_\kappa, m^*_\kappa, m^\#_\kappa$ satisfy Assumption 4 and let $\mathbb{C}_1 = \{\mathbb{C}_{1,\kappa}\}_{\kappa \in \mathbf{N}}, \mathbb{C}_2 = \{\mathbb{C}_{2,\kappa}\}_{\kappa \in \mathbf{N}}$ be ensembles of sets of circuits such that $C_{i,k} \in \mathcal{C}_{n^*_\kappa, m_{i,\kappa}}$ for $i = 1, 2$. Consider the following building blocks, with respect to some $m_\kappa \geq \max(m^*_\kappa, m^\#_\kappa)$:

- an m_κ-right-separable RO obfuscator RO_1 for ensemble $\mathbb{C}_1$, with post-processing algorithm $\pi_1 = (\pi_{1,1}, \pi_{1,2})$ and inner-stretch ξ_1 such that $\xi_1(\kappa, n^*_\kappa, m) \geq m_\kappa m$,
- an m_κ-left-separable RO obfuscator RO_2 for ensemble $\mathbb{C}_2$, with post-processing algorithm $\pi_2 = (\pi_{2,1}, \pi_{2,2})$ and inner-stretch ξ_2 such that $\xi_2(\kappa, n^*_\kappa, m) \geq m_\kappa m$,
- an RIO obfuscator O that satisfies Property II with leeway function $\lambda_\kappa \leq m_\kappa$, for auxiliary circuits $C^\dagger_{1,\kappa}, C^\dagger_{2,\kappa}$, and for input distribution ensemble:

$$\{(\pi_{1,2}(\widehat{C}_{1,2}) | \pi_{2,1}(\widehat{C}_{2,1})) : \tag{12}$$
$$C_{1,2}, C_{2,1} \overset{R}{\leftarrow} \mathcal{C}_{n^*_\kappa, m^*_\kappa}; \widehat{C}_{1,2} \overset{R}{\leftarrow} \mathcal{E}_{C_{1,2}, m_\kappa}; \widehat{C}_{2,1} \overset{R}{\leftarrow} \mathcal{E}_{C_{2,1}, m_\kappa}\}_{\kappa \in \mathbf{N}}. \tag{13}$$

We use RO_1, RO_2, O to construct an RO obfuscator $\mathrm{RO}_{1|2}$ for the ensemble $\mathbb{C} = \{\mathbb{C}_\kappa\}_{\kappa\in\mathbf{N}}$, where each circuit $C \in \mathbb{C}_\kappa$ is of the form $C = C_1|C_2$ where $C_1 \in \mathbb{C}_{1,\kappa}$ and $C_2 \in \mathbb{C}_{2,\kappa}$. Given a circuit $C = C_1|C_2 \in \mathbb{C}_\kappa$, obfuscator $\mathrm{RO}_{1|2,\kappa}$ proceeds as follows (see also Fig. 4):

1. Sample $\tilde{C}_1 \xleftarrow{R} \mathrm{RO}_{1,\kappa}(C_1)$ and $\tilde{C}_2 \xleftarrow{R} \mathrm{RO}_{2,\kappa}(C_2)$.
2. Let $\tau_{i,j}$ denote the stretch of the post-processing algorithm $\pi_{i,j}$, let $t_{i,j} = \tau_{i,j}(\kappa, n^*_\kappa, m_\kappa)$, and let $C_{1,1} = (\tilde{C}_1)_{[1,-t_{1,2}]}, C_{1,2} = (\tilde{C}_1)_{[-t_{1,2},*]}, C_{2,1} = (\tilde{C}_2)_{[1,t_{2,1}]}$, $C_{2,2} = (\tilde{C}_2)_{[t_{2,1},*]}$.
   Sample $G \xleftarrow{R} O_\kappa(C_{1,2}|C_{2,1})$.
3. Output $C_{1,1}|G|C_{2,2}$.

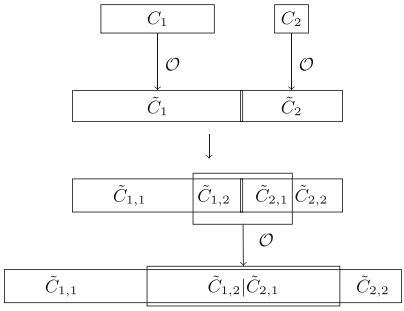

Fig. 4. Soldering RO-obfuscated circuits: The operation of obfuscator $\mathrm{RO}_{1|2}$ given circuits C_1 and C_2.

Claim. Let $n^*_\kappa, m^*_\kappa, m^\#_\kappa$ satisfy Assumption 4, and let $m_\kappa \geq m^\#_\kappa$. For $i = 1, 2$, let $\mathbf{C}_i = \{C_{i,\kappa}\}_{\kappa\in\mathbf{N}}$ be a circuit ensemble where $C_{i,k} \in \mathcal{C}_{n^*_\kappa, m_{i,\kappa}}$, and let $\mathrm{RO}_i = \{\mathrm{RO}_{i,\kappa}\}_{\kappa\in\mathbf{N}}$ be an RO obfuscator for $\mathbf{C}_i$ with inner-stretch ξ_i such that $\xi_i(\kappa, n^*_\kappa, m) = m_\kappa m$ and with post-processing algorithm $\pi_i = (\pi_{i,1}, \pi_{i,2})$; furthermore, RO_1 is m_κ-right-separable and RO_2 is m_κ-left-separable. Let O be an RIO obfuscator functionξ_3 and with post-processing algorithm π_3, for the input distribution ensemble in (12). Then $\mathrm{RO}_{1|2}$ defined above is an RO obfuscator for the circuit ensemble $\{C_{1,\kappa}|C_{2,\kappa}\}_{\kappa\in\mathbf{N}}$, with inner-stretch function $\xi(\kappa, n^*_\kappa, m) = m_\kappa(m-2) + \xi_3(\kappa, n^*_\kappa, 2m_\kappa)$ and post-processing algorithm

$$\pi(C) = \pi_1(C_{[1,\xi_1(\kappa,n^*_\kappa,m_{1,\kappa})-m_\kappa]})|$$
$$\pi_3(C_{[\xi_1(\kappa,n^*_\kappa,m_{1,\kappa})-m_\kappa+1,\xi_3(\kappa,n^*_\kappa,2\,m^*_\kappa)]})|\pi_2(C_{[-(\xi_2(\kappa,n^*_\kappa,m_{2,\kappa})-m_\kappa),*]}).$$

Furthermore, if RO_1 is m_κ-left-separable then so is $\mathrm{RO}_{1|2}$. If RO_2 is m_κ-right-separable then so is $\mathrm{RO}_{1|2}$.

See proof in [CCMR24].

5.4 RO for All Circuits

The RO obfuscator for all circuits combines a single gate obfuscator GO with the soldering process in the natural way. Specifically, consider the append-and-solder obfuscator AS that, to obfuscate an n-wire, m-gate circuit $C = \gamma_1 \dots \gamma_m$ with security parameter κ, proceeds as follows:

1. Let $n^*_\kappa, m^*_\kappa, m^\#_\kappa$ satisfy Assumption 4. Without loss of generality assume that $n = n^*_\kappa$. (If $n < n^*_\kappa$ then embed the circuit in n^*_κ wires. If $n > n^*_\kappa$ then proceed with the smallest $\kappa' > \kappa$ such that $n \leq n^*(\kappa')$.)
2. Let GO be a (n^*_κ, m_κ)-gate obfuscator for $m_\kappa \geq \max(m^*_\kappa, m^\#_\kappa)$. For each gate γ_i, $i = 1 \dots m$, let $\Gamma_i \stackrel{R}{\leftarrow} \mathrm{GO}(\gamma_i)$ be a $2m_\kappa$-gate circuit such that $\mathcal{P}_{\Gamma_i} = \gamma_i$.
3. Solder the circuits $\Gamma_1 \dots \Gamma_m$ one by one, using an RIO obfuscator O for the input distribution ensemble in (12). That is:
   (a) Let $C_1 = \Gamma_1$.
   (b) For $i = 2..m$, let $C_i = (C_{i-1})_{[1,-t_{1,\kappa}]} | O_\kappa((C_{i-1})_{[-t_{1,\kappa},*]} | (\Gamma_i)_{[1,t_{2,\kappa}]}) | (\Gamma_i)_{[t_{2,\kappa},*]}$ be the result of soldering C_{i-1} and Γ_i, where $t_{1,\kappa}$ and $t_{2,\kappa}$ are the lengths of the left and right margins for soldering, namely $\pi_1 : \mathcal{C}_{n^*,m_\kappa} \to \mathcal{C}_{n^*_\kappa,t_{1,\kappa}}$ and $\pi_2 : \mathcal{C}_{n^*,m_\kappa} \to \mathcal{C}_{n^*_\kappa,t_{2,\kappa}}$, where $\pi = (\pi_1, \pi_2)$ is the post-processing algorithm of GO_κ.
4. Output C_m.

It follows from Claim 5.3 that AS is an m_κ-separable RO obfuscator for all reversible circuits, with inner-stretch $\xi(\kappa, n, m) \geq m_\kappa m$. When GO is instantiated via the RIG and RIO described in Sects. 5.2 and 5.1 above, Theorem 2 follows from Claims 5.1 and 5.2.

Furthermore, observe that the stretch of AS grows only linearly in m. Specifically, it follows from Claim 5.3 that $|C_i| = |C_{i-1}| + \sigma_2(\kappa, n^*_\kappa, t_{1,\kappa} + t_{2,\kappa})$, where $\sigma_2(\kappa, n^*_\kappa, m^*_\kappa)$ is the overall stretch of the RIO obfuscator used in the soldering operation. When instantiating the construction with the single-gate obfuscator and random identity generator described in Sects. 5.2 and 5.1, based on an RIO obfuscator with stretch $\sigma_1(\kappa, n^*, m^*_\kappa)$, we obtain $|C_m| \leq m\sigma_2((\kappa, n_\kappa, 2\sigma_1(\kappa, n_\kappa, m^*_\kappa)))$, where n^*_κ, m^*_κ are length functions that satisfy Assumption 4.

Finally, straightforward hybrids argument demonstrates that the security level of AS decreases only linearly in the number of gates. That is, to guarantee distinguishing probability of at most ϵ between an obfuscated m-gate circuit C and a circuit drawn from $D_{\mathcal{P}_C,|C|}$, it suffices to use building blocks (RIO and GO obfuscators) with security $\Omega(\epsilon/m)$.

6 Constructing RIO Obfuscators

This section presents a general approach for constructing RIO obfuscators, along with a family of candidate RIO obfuscators that may be a viable basis for RO (and in particular IO) obfuscators for all circuits as in Theorem 2.

This section, as well as the open problems section and the appendix have removed from this version due to page limits. These sections appear in [CCMR24].

Acknowledgements. We thank Luowen Qian for participating in early stages of this research, and the TCC'24 reviewers for their insightful comments. R.C. also thanks Nir Bitansky, Shafi Goldwasser and Omer Paneth for very helpful discussions.

References

[AB15] Applebaum, B., Brakerski, Z.: Obfuscating circuits via composite-order graded encoding. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015. LNCS, vol. 9015, pp. 528–556. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46497-7_21

[AJJ14] Alagic, G., Jeffery, S. and Jordan, S.P.: Circuit obfuscation using braids. In: Flammia, S.T., Harrow, A.W. (eds.) 9th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2014, 21-23 May 2014, Singapore, LIPIcs, vol. 27 , pp. 141–160. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2014)

[AJS15] Ananth, P., Jain, A., Sahai, A.: Achieving compactness generically: indistinguishability obfuscation from non-compact functional encryption. IACR Cryptol. ePrint Arch., pp. 730 (2015)

[Bar86] Barrington, D.A.: Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. In: Hartmanis, J. (ed.) Proceedings of the 18th Annual ACM Symposium on Theory of Computing, 28-30 May 1986, Berkeley, California, USA, pp. 1–5. ACM (1986)

[Bar17] Barak, B.: The complexity of public-key cryptography. In: Tutorials on the Foundations of Cryptography. ISC, pp. 45–77. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-57048-8_2

[Ben73] Bennett, C.H.: Logical reversibility of computation. IBM J. Res. Dev. **17**, 525–532 (1973)

[Ben89] Bennett, C.H.: Time/space trade-offs for reversible computation. SIAM J. Comput. **18**(4), 766–776 (1989)

[BGI+01] Barak, B., et al.: On the (Im)possibility of obfuscating programs. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 1–18. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_1

[BGI+12] Boaz et al.: On the (Im)possibility of obfuscating programs. J. ACM, **59**(2), 6:1–6:48 (2012)

[BGK+14] Barak, B., Garg, S., Kalai, Y.T., Paneth, O., Sahai, A.: Protecting obfuscation against algebraic attacks. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 221–238. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55220-5_13

[BLMP23] Ball, M., Liu, Y., Mazor, N., Pass, R.: Kolmogorov comes to cryptomania: on interactive Kolmogorov complexity and key-agreement. In: 2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS), pp. 458–483 (2023)

[BPW16] Bitansky, N., Paneth, O., Wichs, D.: Perfect structure on the edge of chaos. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016. LNCS, vol. 9562, pp. 474–502. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49096-9_20

[Bro04] Brodsky, A.: Reversible circuit realizations of Boolean functions. In: Levy, J.-J., Mayr, E.W., Mitchell, J.C. (eds.) TCS 2004. IIFIP, vol. 155, pp. 67–80. Springer, Boston, MA (2004). https://doi.org/10.1007/1-4020-8141-3_8

[BV15] Bitansky, N., Vaikuntanathan, V.: Indistinguishability obfuscation from functional encryption. IACR Cryptol. ePrint Arch., pp. 163 (2015)

[CCMR24] Canetti, R., Chamon, C., Mucciolo, E., Ruckenstein, A.: Towards general-purpose program obfuscation via local mixing. Cryptology ePrint Archive, Paper 2024/006 (2024)

[CG75] Coppersmith, D., Grossman, E.: Generators for certain alternating groups with applications to crytography. SIAM J. Appl. Math. **29**(4), 624–627 (1975)

[CGH+15] Coron, J.-S., et al.: Zeroizing without low-level zeroes: new MMAP attacks and their limitations. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9215, pp. 247–266. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-47989-6_12

[CHVW19] Chen, Y., Hhan, M., Vaikuntanathan, V., Wee, H.: Matrix PRFs: constructions, attacks, and applications to obfuscation. In: Hofheinz, D., Rosen, A. (eds.) TCC 2019. LNCS, vol. 11891, pp. 55–80. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-36030-6_3

[CJMR22] Chamon, C., Jakes-Schauer, J., Mucciolo, E.R., Ruckenstein, A.E.: Encrypted operator computing: an alternative to fully homomorphic encryption. CoRR, abs/2203.08876 (2022)

[CLTV15] Canetti, R., Lin, H., Tessaro, S., Vaikuntanathan, V.: Obfuscation of probabilistic circuits and applications. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015. LNCS, vol. 9015, pp. 468–497. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46497-7_19

[CMR22] Chamon, C., Mucciolo, E.R., Ruckenstein, A.E.: Quantum statistical mechanics of encryption: reaching the speed limit of classical block ciphers. Ann. Phys. **446**, 169086 (2022)

[CRMC23] Chamon, C., Ruckenstein, A.E., Mucciolo, E.R., Canetti, R.: Circuit complexity and functionality: a thermodynamic perspective. arXiv preprint arXiv:2309.05731, 2023

[CV13] Canetti, R., Vaikuntanathan, V.:D Obfuscating branching programs using black-box pseudo-free groups. IACR Cryptol. ePrint Arch., pp. 500 (2013)

[CVW18] Chen, Y., Vaikuntanathan, V., Wee, H.: GGH15 beyond permutation branching programs: proofs, attacks, and candidates. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018. LNCS, vol. 10992, pp. 577–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96881-0_20

[DQV+21] Devadas, L., Quach, W., Vaikuntanathan, V., Wee, H., Wichs, D.: Succinct LWE sampling, random polynomials, and obfuscation. In: Nissim, K., Waters, B. (eds.) TCC 2021. LNCS, vol. 13043, pp. 256–287. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-90453-1_9

[GGH15] Gentry, C., Gorbunov, S., Halevi, S.: Graph-induced multilinear maps from lattices. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015. LNCS, vol. 9015, pp. 498–527. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46497-7_20

[GGHR14] Garg, S., Gentry, C., Halevi, S., Raykova, M.: Two-round secure MPC from indistinguishability obfuscation. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 74–94. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54242-8_4

[GGSW13] Garg, S., Gentry, C., Sahai, A., Waters, B.: Witness encryption and its applications. In: Symposium on Theory of Computing Conference, STOC 2013, Palo Alto, CA, USA, 1-4 June 2013, pp. 467–476 (2013)

[GHP24] Gretta, L., He, W., Pelecanos, A.: More efficient k-wise independent permutations from random reversible circuits via log-sobolev inequalitie, manuscript (2024)

[Gow96] Gowers, W.T.: An almost m-wise independed random permutation of the cube. Comb. Probab. Comput. **5**, 119–130 (1996)

[GP21] Gay, R., Pass, R.: Indistinguishability obfuscation from circular security. In: Khuller, S., Williams, V.V. (eds.) STOC 2021: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, 21-25 June 2021, pp. 736–749. ACM (2021)

[GR14] Goldwasser, S., Rothblum, G.N.: On best-possible obfuscation. J. Cryptol. **27**(3), 480–505 (2014)

[Had00] Hada, S.: Zero-knowledge and code obfuscation. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 443–457. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44448-3_34

[HB05] Brodsky, A., Hoory, S.: Simple permutations mix even better (2005)

[HMMR05] Hoory, S., Magen, A., Myers, S., Rackoff, C.: Simple permutations mix well. Theor. Comput. Sci. **348**(2):251–261 (2005). Automata, Languages and Programming: Algorithms and Complexity (ICALP-A 2004)

[HO24] He, W., O'Donnell, R.: Pseudorandom permutations from random reversible circuits (2024)

[ILW23] Ilango, R., Li, J., Williams, R.R.: Indistinguishability obfuscation, range avoidance, and bounded arithmetic. Electron. Colloquium Comput. Complex. TR23-038 (2023)

[IRS22] Ilango, R., Ren, H., Santhanam, R.: Robustness of average-case meta-complexity via pseudorandomness. In: Leonardi, S., Gupta, A. (eds.) STOC 2022: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, 20-24 June 2022, pp. 1575–1583. ACM (2022)

[JLS21] Jain, A., Lin, H., Sahai, A.: Indistinguishability obfuscation from well-founded assumptions. In: Khuller, S., Williams, V.V. (eds.) STOC 2021: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, 21-25 June 2021, pp. 60–73. ACM (2021)

[Kil88] Kilian, J.: Founding cryptography on oblivious transfer. In: Simon, J. (ed.) Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA, pp. 20–31. ACM (1988)

[KNT22] Kitagawa, F., Nishimaki, R., Tanaka, K.: Obfustopia built on secret-key functional encryption. J. Cryptol. **35**(3), 19 (2022)

[LP20] Liu, Y., Pass, R.: On one-way functions and Kolmogorov complexity. In: Irani, S. (ed.) 61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, 16-19 November 2020, pp. 1243–1254. IEEE (2020)

[LP21] Liu, Y., Pass. R.: Cryptography from sublinear-time average-case hardness of time-bounded Kolmogorov complexity. In: Khuller, S., Williams, V.V. (eds.) STOC 2021: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, 21-25 June 2021, pp. 722–735. ACM (2021)

[LPST16] Lin, H., Pass, R., Seth, K., Telang, S.: Indistinguishability obfuscation with non-trivial efficiency. In: Cheng, C.-M., Chung, K.-M., Persiano, G., Yang, B.-Y. (eds.) PKC 2016. LNCS, vol. 9615, pp. 447–462. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49387-8_17

[LPTV23] Liu, T., Pelecanos, A., Tessaro, S., Vaikuntanathan, V.: Layout graphs, random walks and the t-wise independence of SPN block ciphers. In: Handschuh, H., Lysyanskaya, A. (eds.) Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part III, LNCS, vol. 14083, pp. 694–726. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-38548-3_23

[LTV21] Liu, T., Tessaro, S., Vaikuntanathan, V.: The t-wise independence of substitution-permutation networks. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021. LNCS, vol. 12828, pp. 454–483. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-84259-8_16

[RVV24] Ragavan, S., Vafa, N. and Vaikuntanathan, V.: Indistinguishability obfuscation from bilinear maps and LPN variants. Theory of Cryptography Conference (TCC) (2024)

[SW14] Sahai, A., Waters, B.: How to use indistinguishability obfuscation: deniable encryption, and more. In: Shmoys, D.B. (ed.) STOC 2014, New York, NY, USA, May 31 - June 03, 2014, pp. 475–484. ACM (2014)

[Tof80] Toffoli, T.: Reversible computing. In: de Bakker, J., van Leeuwen, J. (eds.) ICALP 1980. LNCS, vol. 85, pp. 632–644. Springer, Heidelberg (1980). https://doi.org/10.1007/3-540-10003-2_104

[Wik24] Wikipedia contributors: White-box cryptography — Wikipedia, the free encyclopedia (2024). [Online; accessed 24-September-2024]

[WW21] Wee, H., Wichs, D.: Candidate obfuscation via oblivious LWE sampling. In: Canteaut, A., Standaert, F.-X. (eds.) EUROCRYPT 2021. LNCS, vol. 12698, pp. 127–156. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77883-5_5