Towards general-purpose program obfuscation via local mixing*

Ran Canetti[†] Claudio Chamon[†] Eduardo R. Mucciolo[‡] Andrei E. Ruckenstein[†]

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.

We start by formulating a new (and relatively weak) obfuscation task regarding the ability to obfuscate *random* circuits of *bounded length.* We call such obfuscators *random input & output (RIO)* obfuscators. We then show how to construct indistinguishability obfuscators for all (unbounded length) circuits given only an RIO obfuscator — under a new assumption regarding the pseudorandomness of sufficiently long random reversible circuits with known functionality, which in turn builds on a conjecture made by Gowers (Comb. Prob. Comp. '96) regarding the pseudorandomness of bounded-size random reversible circuits. Furthermore, the constructed obfuscators satisfy a new measure of security - called random output indistinguishability (ROI) obfuscation - which is significantly stronger than IO and may be of independent interest.

We then investigate the possibility of constructing RIO obfuscators using local, functionality preserving perturbations. Our approach is rooted in statistical mechanics and can be thought of as locally "thermalizing" a circuit while preserving its functionality. We provide candidate constructions along with a pathway for analyzing the security of such strategies.

Given the power of program obfuscation, viability of the proposed approach would provide an alternative route to realizing almost all cryptographic tasks under hardness assumptions that are very different from standard ones. Furthermore, our specific candidate obfuscators are relatively efficient: the obfuscated version of an n-wire, m-gate (reversible) circuit with security parameter κ has n wires and poly$(n, \kappa)m$ gates. We hope that our initial exploration will motivate further study of this alternative path to cryptography.

*This work is supported by 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

[†]Boston University

[‡]University of Central Florida

Contents

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$^+$01, BGI$^+$12], *any* polysize representation of a program, even a "perfectly randomized" one, gives, in general, significantly more computational power than black-box access to the function computed by the program. However, the more modest goal of perturbing the program just to the point of making the perturbed versions of any two equal-length, functionally equivalent programs indistinguishable (namely, obtaining *Indistinguishability Obfuscation (IO)* [BGI$^+$01, BGI$^+$12]), is potentially obtainable [GGSW13] and has proven to be immensely powerful. Indeed, while it is rather weak as a stand-alone cryptographic primitive (it cannot even be used to construct one way functions or even imply P$\neq$NP), IO has proven to be an immensely versatile conduit for harnessing and molding unstructured (or, minimally structured) computational hardness to perform almost any cryptographic task. Specifically, combined with one way functions, IO for all circuits 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 or rerandomizable encryption, it gives also fully homomorphic encryption and more [CLTV15a].

The history of attempts at constructing general purpose program obfuscators is intriguing as well. The concept of obfuscating programs as a hedge against copying or modification has been known in the practical security community for decades, in the form of "obfuscation tools" (some of them commercially available) that modify programs by locally perturbing instructions, variable names, and memory access patterns. However, these tools carry no formal security guarantees and have invariably been eventually broken, or "reversed".

Following the introduction of formal notions of "cryptographic grade" program obfuscation [Had00, BGI$^+$01], there has been more than a decade where we knew how to cryptographically obfuscate only very few classes of simple functions such as point functions and related constructs, e.g. [Can97, LPS04, CRV10]. (These works concentrated on obtaining the stronger notion of VBB obfuscation.)

The breakthrough works of [SW14, GGSW13] have opened the floodgate to both applications of IO and candidate constructions thereof, e.g. [BGK$^+$14, AB15, GGH15]. In these "first generation" constructions the obfuscated program typically follows the instruction structure of the the plaintext program without modification, while using the algebraic structure to perform the instructions "homomorphically" while hiding them from an adversary who runs the program and sees its entire execution trace. However, the analyses of these first generation constructions was invariably incomplete, often by way of relying on an idealized version of a core primitive, and indeed explicit attacks have been demonstrated against many proposed instantiations of these candidates (e.g., [CGH$^+$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 "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] 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 as a purely theoretical primitive.

1.1 This work

We propose a new approach to constructing general-purpose program obfuscation. Specifically:

- We formulate two new measures of security for program obfuscation:
  - Random output indistingishability (ROI) obfuscation. This measure strengthens plain IO, providing a natural bridge between IO and VBB obfuscation.
  - Random input & output (RIO) obfuscation. This measure is a significant relaxation of IO: it only requires indistinguishability for *random* programs, and when the distinguisher does not have access to the plaintext program.
- We construct ROI obfuscators for all circuits from RIO obfuscators for bounded-length random circuits. The construction proceeds by way of constructing ROI obfuscators for increasingly long prefixes of the given circuit, and incurs only linear overhead in complexity. Security is proven based on a new intractability assumption regarding random circuits with a given functionality.
- We propose candidate RIO obfuscators. Here we return to the basics: We use very little algebraic structure and instead concentrate on local, functionality preserving randomized perturbations of the circuit structure.

Our constructions and analyses make critical use of the structure of reversible circuits. We thus start the exposition of our results with a brief overview of reversible circuits, followed by an exposition of our intractability assumptions regarding the same.

1.1.1 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, $\mathcal{P}_C = \mathcal{P}_{\gamma_m} \circ \dots \circ \mathcal{P}_{\gamma_1}$, or, in other words, $C(x) = \gamma_m(\dots \gamma_1(x) \dots)$. We restrict our attention to Toffoli gates, namely gates of the form $\gamma_{i,j,k,f}(s_1 \dots s_n) = (s'_1 \dots s'_n)$ where $s_1 \dots s_n$ is the old state, $s'_1 \dots s'_n$ 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].

On the one hand, considering only circuits of the above form does not limit the generality of the treatment: We know that 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 in a way that preserves both the functionality and the complexity of the original circuit.[1]

On the other hand, reversible circuits have some attractive properties which are essential for our treatment. First, 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 rationale and design of block ciphers makes extensive use of this property, though not at the level of 3-wire gates (see e.g. [Fei74, CG75]). Furthermore, as we discuss momentarily, random reversible circuits appear to have some strong pseudorandomness properties even to observers that see the circuits themselves, rather than having only oracle access to their functionality. Lastly, random 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.

Pseudorandomness. 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 $m = \Omega(n^3 t^2 + n^2 t \log(\varepsilon^{-1}))$. We note that Gowers considers all permutations on 3 wires as base permutations (or, gates). In contrast, Brodsky and Hoory consider the above set $\mathbb{B}_n$ of gates.

Furthermore, random circuits in $\mathcal{C}_{n,m}$ appear to have even stronger pseudorandomness properties. Gowers conjectured that there exist some $n^*_\kappa, m^*_\kappa \in \mathrm{poly}(\kappa)$ such that $\mathcal{C}_{n_\kappa,m_\kappa}$ is a cryptographic-grade pseudorandom permutation family (with κ being the security parameter). That is, given time polynomial in κ one can distinguish between oracle access to $C \xleftarrow{\mathrm{R}} \mathcal{C}_{n^*_\kappa,m^*_\kappa}$ and oracle access to a random permutation on $\{0,1\}^{n^*_\kappa}$ only with probability that's negligible in κ.[2] Chamon et al. [CMR22] propose a more structured distribution over circuits and provide quantum-mechanical evidence that a few as $m_\kappa = O(n_\kappa \log n_\kappa)$ may suffice for pseudorandomness whenever $n_\kappa = \Omega(\kappa)$. Their analysis extends also to completely random circuits with $m_\kappa = \tilde{O}(n_\kappa)$ gates.

Still, while these properties may be intriguing, they only relate to the *functionality* of random reversible circuits; that is, they only consider attackers with black-box access to the chosen circuit. 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 any (not too large) segment of a sufficiently long random circuit remains pseudorandom *even given oracle access to the overall*

[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 $x \in \{0,1\}^\alpha, y \in \{0,1\}^\beta$ (see e.g. [Ben73, Tof80]). We also show how to make this transform "obfuscation compatible" by providing the additional guarantee that $C'(x, y, z) = (x, y, z)$ whenever $z \neq 0^\delta$.

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

circuit. For instance, let $C \xleftarrow{R} \mathcal{C}_{n^*_\kappa,m_\kappa}$, for some $m_\kappa > m^*_\kappa$ and let C_i denote the circuit C without the m^*_κ-gate sub-circuit that starts at gate i. It is easy to see that, under Gower's conjecture and for any i, polytime adversaries that are given i, oracle access to C and a challenge $(m_\kappa - m^*_\kappa)$-gate circuit C', cannot tell whether $C' = C_i$, or else $C' \xleftarrow{R} \mathcal{C}_{n^*_\kappa,m_\kappa-m^*_\kappa}$, significantly better than a random guess.[3]

Furthermore, it is only natural that this same property — pseudorandomness of not-too-long 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 $C \xleftarrow{R} \mathcal{E}_{I_{n^*_\kappa},2m^*_\kappa}$, and let $C_{[1,m^*_\kappa]}$ denote the m^*_κ-gate prefix of C. While $C_{[1,m^*_\kappa]}$ is statistically far from a random n^*_κ-wire, m^*_κ-gate circuit, it is plausible that the two distributions are indistinguishable.

By the same token, it seems plausible that

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

namely that a random $2m^*_\kappa$-gate identity circuit is indistinguishable from a random m^*_κ-gate circuit C followed by the inverse of another random m^*_κ-gate circuit C' that's functionally equivalent to C. (Here 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 $m' \geq 2m^*_\kappa m$. Then it is plausible that any $(m' - m^*_\kappa)$-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:

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

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

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

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

[4]One consequence of the correlation is that here m^*_κ 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 $m^*_\kappa = \tilde{\Omega}(|\mathbb{B}_n|)$. 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.

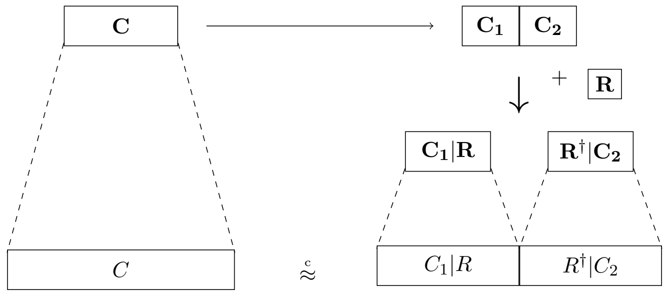

Figure 1: The Split Circuit Pseudorandomness (SCP) assumption. Circuit $\mathbf{C}$ (top left) is an arbitrary n-wire, m-gate reversible circuit. Circuits $\mathbf{C_1}$ and $\mathbf{C_2}$ at the top right are the m_1-gate prefix and m_2-gate suffix of $\mathbf{C}$ (with $m_1 + m_2 = m$), and R is a 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.

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 $n^*_\kappa, m^*_\kappa \in \mathrm{poly}(\kappa)$ such that for any large enough 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.

1.1.2 New notions of security for circuit obfuscation

Next, we sketch the definitions of ROI obfuscation (which strengthens IO) and RIO obfuscation (which relaxes IO). 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 $O = \{O_\kappa\}_{\kappa \in \mathbf{N}}$, $O_\kappa : \mathcal{C}_{n_\kappa} \to \mathcal{C}_{n_\kappa}$, is a **random output indistinguishability (ROI)** obfuscator for a set $\{\mathbb{C}_\kappa \subseteq \mathcal{C}_{n_\kappa}\}_{\kappa \in \mathbf{N}}$ of circuits and inner-stretch function ξ if there exists an efficient "post-processing algorithm" π such that for any m_κ-gate circuit $C_\kappa \in \mathbb{C}_\kappa$ we have:

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

It can be verified that if $\xi(\kappa, n, m) = m$ then ROI obfuscation coincides with standard indistinguishability obfuscation (IO). (In particular, in this case we can set $\pi = O$ without losing generality.) However, when $\xi(\kappa, n, m) \geq m + \Omega(\kappa)$ ROI obfuscation becomes non-trivial to obtain even in situations where IO is trivial (e.g. when the input circuit C is the only one with the same size and functionality). Furthermore, together with the SCP assumption, ROI obfuscation with large inner-stretch (namely, when $\xi(\kappa, n, m) = \Omega(m^*_\kappa m)$) guarantees that both the structure and the functionality of any not-too-large portion of C are essentially lost. This is a strong, VBB-like security guarantee – and is obtained without being susceptible to the broad impossibility results that limit the applicability of VBB obfuscation.

A functionality-preserving transformation $O = \{O_\kappa\}_{\kappa\in\mathbf{N}}$, $O_\kappa : \mathcal{C}_{n_\kappa} \to \mathcal{C}_{n_\kappa}$, is a **random input and output (RIO)** obfuscator with respect to $\mathcal{C}_{n_\kappa,m_\kappa}$ if the following two requirements hold:[6]

1. $\{(O(C), O(C)) : C \xleftarrow{\text{R}} C_{n_\kappa,m_\kappa}\}_{\kappa\in\mathbf{N}} \overset{\text{c}}{\approx} \{(O(C), O(C') : C \xleftarrow{\text{R}} C_{n_\kappa,m_\kappa}; C' \xleftarrow{\text{R}} \mathcal{E}_{C,m_\kappa}\}_{\kappa\in\mathbf{N}}$

2. For any "advice" circuit $Z_\kappa = Z_{1,\kappa}|Z_{2,\kappa}$ we have
$$O(C), \widehat{C}_1, \widehat{C}_2 \overset{\text{c}}{\approx} O(C'), \widehat{C}_1, \widehat{C}_2$$
where $C \xleftarrow{\text{R}} \mathcal{C}_{n_\kappa,m_\kappa}$, $C' \xleftarrow{\text{R}} \mathcal{E}_{C,m_\kappa}$, C_1 and C_2 are the $m_\kappa/2$-gate prefix and suffix of C, respectively, $\widehat{C}_1 \xleftarrow{\text{R}} \mathcal{E}_{(Z_{1,\kappa}|C_1^\dagger),l_{1,\kappa}\lambda_\kappa}$, and $\widehat{C}_2 \xleftarrow{\text{R}} \mathcal{E}_{(C_2^\dagger|Z_{2,\kappa}),l_{2,\kappa}\lambda_\kappa}$. Here $l_{i,\kappa}$ is the number of gates in $Z_{i,\kappa}$, $i = 1, 2$, and λ_κ is a 'leeway parameter' associated with O.

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 advice that is tantamount to oracle access to C_1 and C_2, the first and second halves of C, and their inverses. More specifically, let Z be some a-priori fixed circuit. Then the advice consists of the following "randomized completion" $\widehat{C}_1, \widehat{C}_2$ of C to a (sufficiently long) circuit that is functionally equivalent to Z: Let $Z = Z_1|Z_2$; then $\widehat{C}_1$ and $\widehat{C}_2$ are sufficiently long random circuits that respectively compute $Z_1|C_1^\dagger$ and $C_2^\dagger|Z_2$. (Indeed, while the only "operable information" in $\widehat{C}_1, \widehat{C}_2$ appears to be the ability to evaluate C_1, C_2, this advice has significant additional structure; in particular, both $\widehat{C}_1|C|\widehat{C}_2$ and $\widehat{C}_1|C'|\widehat{C}_2$ are functionally equivalent to Z. Whether this requirement can be further simplified (or relaxed) while preserving its usefulness is an intriguing question.)

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 ROI) where the distinguisher sees both C and $O(C)$, making RIO potentially easier to obtain — not only from IO, but also from IO for random circuits.

1.1.3 From RIO to ROI for all circuits

We show:

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

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

[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.

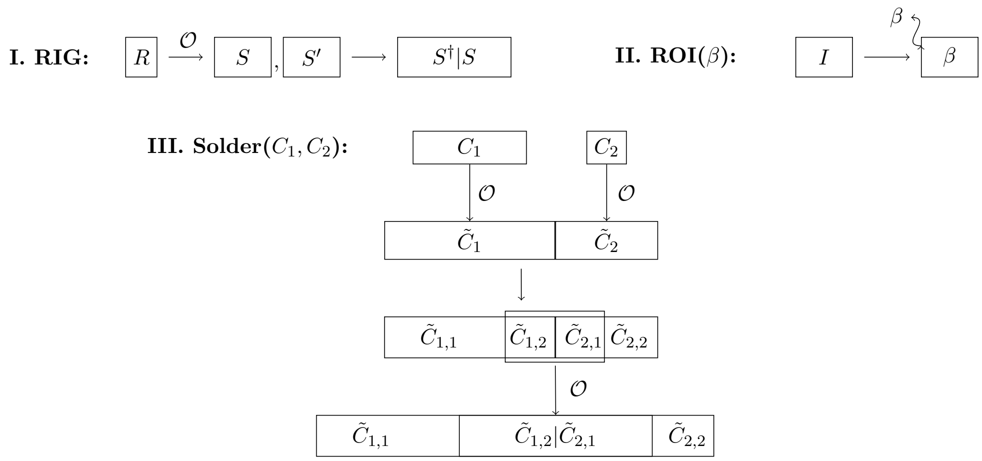

Figure 2: The building blocks for constructing ROI obfuscation for all reversible circuits from RIO obfuscation for bounded length random circuits. The first building block is random identity generators (RIGs), constructed by concatenating two RIO-obfuscated versions of a random circuit, one in reverse. The second building block is ROI obfuscators for single gates, constructed by sampling a RIG with the desired first gate and removing that gate. The third building block is soldering ROI-obfuscated versions of circuits C_1 and C_2 into an ROI-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.

- A random identity generator (RIG), which is an ROI obfuscator for the identity permutation with inner-stretch $2m^*_\kappa$. This is done by choosing $C \xleftarrow{\text{R}} \mathcal{C}_{n^*_\kappa, m^*_\kappa}$, then sampling $C', C'' \xleftarrow{\text{R}} O(C)$ 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 ROI obfuscator for β, per each gate $\beta \in \mathbb{B}_{n^*_\kappa}$. This can be done simply by sampling random itentities 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" ROI-obfuscated circuits, namely combining an ROI obfuscator O_1 for a circuit C_1 and an ROI obfuscator O_2 for a circuit C_2 into an ROI obfuscator for the circuit $C_1|C_2$. The idea is again simple: Let $\tilde{C}_1 \xleftarrow{\text{R}} O_1(C_1)$, $\tilde{C}_2 \xleftarrow{\text{R}} O_2(C_2)$. 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^*_κ-gates each. Now, compute $G \xleftarrow{\text{R}} O(C_{1,2}|C_{2,1})$ 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 $\Gamma_i \xleftarrow{\text{R}} \text{GO}(\gamma_i)$ 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.

The use of ROI 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 ROI obfuscation.

1.1.4 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 mixing (and hence obfuscation) properties.

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 Figure 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 a 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.

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 difference 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 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.

1.2 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, Muccciolo 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 $C = \gamma_1 \ldots \gamma_m$, let $|C| = m$ denote the number of gates in C. For $i, l \in [m]$, let $C_{[i,l]} = \gamma_i \ldots \gamma_{i+l (\bmod m)}$ 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.

3 Hardness assumptions

This section presents and motivates the hardness assumptions used in this work. 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 Section 6.

We start off with a reminder of the standard definition of computational 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 $\mathcal{A} \overset{c}{\approx} \mathcal{B}$, 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

Let $\mathbb{A}_{2^n}$ denote the set of even permutations on $\{0,1\}^n$. 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

[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].

computes P. Similarly, let $\mathrm{CC}(C) = \mathrm{CC}(\mathcal{P}_C)$ 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 $\mathrm{CG}(C) = m - \mathrm{CC}(C)$.

While $\mathrm{CC}(C)$ 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 $b^m > |\mathcal{E}_{P,m}| > b^{\frac{1}{2}(m-\mathrm{CC}(P))}$ for any permutation P, where $b \approx 8n^3$ is the number of base permutations. (For the lower bound, let C be a circuit of length $\mathrm{CC}(P)$. Then for any sequence of base permutations $\beta_1 \dots \beta_l$ where $l = (m - \mathrm{CC}(P))/2$, the circuit $C\beta_1\beta_1 \dots \beta_l\beta_l$ is functionally equivalent to C.) Furthermore, for all but a negligible faction of the circuits C we actually have $b^m > |\mathcal{E}_{C,m}| > b^{(1-o(1))(m-\frac{\mathrm{CC}(C)}{\log b})}$:

Claim 2 *For all but a negligible fraction of the circuits $C \in \mathcal{C}_{n,m}$ we have $|\mathcal{E}_{C,m}| > b^{(1-o(1))(m-\frac{\mathrm{CC}(C)}{\log b})}$.*

Proof: Note that any string $\sigma = \{0,1\}^s$ can be interpreted as a description of a reversible circuit $C_{n,\sigma}$ 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 $\mathrm{CC}(C_{n,\sigma})$ that's functionally equivalent to $C_{n,\sigma}$, plus the ordinal of $C_{n,\sigma}$ among all m-gate circuits that are functionally equivalent to $C_{n,\sigma}$. This means that $K(\sigma) \le \mathrm{CC}(C_{n,\sigma}) + \log(|\mathcal{E}_{C_{n,\sigma},m}|)$, or equivalently that

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

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

In essence, Claim 2 says that, for almost all permutations $P \in \mathbb{A}_{2^n}$, the size of $\mathcal{E}_{P,m}$ grows at almost the same rate (up to lower order terms) as the overall growth in the size of $\mathcal{C}_{n,m}$. In other words the ratio $F_{P,m} = |\mathcal{E}_{P,m}|/|\mathcal{C}_{n,m}|$ is almost constant: $1 \ge F_{P,m} \ge 2^{-\mathrm{CC}(C)(1+\omega(1))}$.

This "almost uninhibited" exponential growth of $\mathcal{E}_{P,m}$ supports the conjecture (formalized later in this section) that relatively short segments of random circuits in $\mathcal{E}_{P,m}$, where $m >> \mathrm{CC}(P)$, are nearly random.

Another conclusion from this state of affairs is that $|\{C \in \mathcal{C}_{n,m} : \mathrm{CC}(C) = m\}| b^{-m} \le \mathrm{negl}(m)$, 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 Section 6.

3.2 Hardness assumptions regarding random reversible circuits

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 $m = \Omega(n^3t^3 \log(\varepsilon^{-1}))$. That is, for any sequence of distinct values $x_1 \dots x_t \in \{0,1\}^n$, and for $C \xleftarrow{\mathrm{R}} \mathcal{C}_{n,m}$, the statistical distance between $C(x_1) \dots C(x_t)$ and a random sequence of distinct values $r_1 \dots r_t$, is at most ε. Hoory et al. [HMMR05] and later Brodsky and Hoory [HB05] improve this bound to $m = \Omega(n^3t^2 + n^2t\log(\varepsilon^{-1}))$. (We note that, while Gowers considered all $8!\binom{n}{3}$ permutations on 3 wires as base permutations, Brodsky and Hoory [HMMR05, HB05] 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 $m = poly(n)$. While to the best of our knowledge this conjecture has so far not been related to other hardness assumptions used in cryptography (beyond, of course, the obvious implication of the existence of pseudorandom functions), we will adopt this conjecture as a starting point for our investigation. We first restate the standard definition of strong PRPs using our notation:

Definition 3 (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*

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

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

Assumption 4 (Polysize random reversible circuits are SPRPs [Gow96]) *There exist n_κ^*, $m_\kappa^* \in poly(\kappa)$ 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 4. 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 $C \stackrel{\mathrm{R}}{\leftarrow} \mathcal{E}_{Q_\kappa, m}$ 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 4. 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 5 (Prefixes of random circuits with fixed functionality are SPRPs) *There exist $n_\kappa^*, m_\kappa^* \in poly(\kappa)$ 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 $m_{Q_\kappa} \in poly(\kappa)$, 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 5 is only a mild generalization of

[8] As a simple example, compare a random n-wire, $2m$-gate circuit R that computes 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 2, for most α the latter probability is proportional to $b^{-\mathrm{CC}(\alpha)}$.)

Assumption 4.

An immediate consequence from Assumption 5 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 $C = C_1|C_2$ 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 5.

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 $C \overset{R}{\leftarrow} F_\kappa$.

That is:

Claim 6 *Let $n^*_\kappa, m^*_\kappa \in poly(\kappa)$ and $\mathbf{Q} = \{\mathbf{Q}_\kappa\}_{\kappa\in\mathbf{K}}$ be as in Assumption 5 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*

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

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

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

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 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 $C \stackrel{\mathrm{R}}{\leftarrow} F_\kappa$ 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 7 (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 $m^\#_\kappa \in poly(\kappa)$ 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 $\ell_{1,\kappa}, \ell_{2,\kappa}$ such that $\ell_{i,\kappa} \geq m_{i,\kappa} m^\#_\kappa$, $i = 1, 2$, we have:*

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

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 4. Still, the more general statement appears to more closely match the intuition for the nature of the hardness.

Finally, we combine Assumptions 4 and 7 to one:

Assumption 8 (Split Circuit Pseudorandomness (SCP):) *There exist n^*_κ, $m^*_\kappa \in poly(\kappa)$ that satisfy Assumption 4, as well as $m^\#_{1,\kappa} \in poly(\kappa)$ that satisfies Assumption 7 with respect to the SPRP in Assumption 4.*

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 $R \stackrel{\mathrm{R}}{\leftarrow} \mathcal{E}_{I_{n_\kappa} m^\#_\kappa}$ to the split version $C'|C''^\dagger : C \stackrel{\mathrm{R}}{\leftarrow} \mathcal{C}_{n_\kappa,m^*_\kappa}; C', C'' \stackrel{\mathrm{R}}{\leftarrow} \mathcal{E}_{C,m^\#_\kappa}$, 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.

[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 2, this means that if $C \stackrel{\mathrm{R}}{\leftarrow} \mathcal{C}_{n,n^4}$ then $|\mathcal{E}_{C,n^4}| > b^{n^2}$.

Assumption 9 (Strong Split Circuit Pseudorandomness (SSCP):) *Assumption 8 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 10 (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*

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

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:

Definition 11 (IO - an 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:*

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

Claim 12 *An obfuscator satisfies Definition 11 for an ensemble* $\mathbb{C}$ *of circuits iff it satisfies Defiition 10 for* $\mathbb{C}$. $\square$

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.

Random Output Obfuscation. 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 turns out to be a bit too ideal, as the local random perturbations technique of Section 6 does not appear to support this property (see discussion there). We thus settle for a relaxation that on the one hand appears to be plausible, and on the other hand suffices for obtaining IO for all circuits. Specifically, we focus on distributions $\mathcal{D}(\kappa, m, P)$ where the output circuit is the result of applying some efficient (i.e., polytime) post-processing algorithm to a circuit $\widehat{C}$ drawn uniformly from $\mathcal{E}_{P,m'}$ for some $m' \geq m$. We further restrict attention to post-processing algorithms that are applied separately to different segments of C, rather than to the entire circuit C. (We later use this property together with Assumption 8 to argue about properties of separate segments of an obfuscated circuit.) Specifically:

Definition 13 (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 (ROI)** *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]

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

We note that ROI obfuscation where $\xi(\kappa, n_\kappa, m_\kappa) - m_\kappa = \Omega(\kappa)$ is a non-trivial strengthening of plain IO. In particular, it seems to be both meaningful and challenging even in situations where plain IO is trivial, e.g. when the input circuit is the only circuit with the same functionality and length. (In particular recall that, by Claim 2, 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.)

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

1. The computational complexity of the m_κ-gate prefix of obfuscated circuits is low: 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 $\pi(C) = \pi_1(C_{[1,m_\kappa]})|\pi_2(C_{[m_\kappa,*]})$.

Obfuscator $\mathcal{O}$ is m_κ**-right-separable** if $\mathcal{O}^\dagger$ is left-separable, where $\mathcal{O}^\dagger(C) = (\mathcal{O}(C^\dagger))^\dagger$. An m_κ-**separable** obfuscator is both m_κ-left-separable and m_κ-right-separable.

For a function $f : \mathcal{C} \to \mathcal{C}$, let $f^\dagger(C) = (f(C^\dagger))^\dagger$. Observe that if $\mathcal{O} = \{O_\kappa\}_{\kappa\in\mathcal{N}}$ is an m_κ-left-separable ROI obfuscator then $\mathcal{O}^\dagger = \{O^\dagger_\kappa\}_{\kappa\in\mathcal{N}}$ is an m'_κ-right-separable ROI obfuscator (and vice versa).

Random Input & 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 the obfuscated circuit plus some limited information on the plaintext circuit. More specifically, we consider two (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 output distribution $\mathcal{D}$

- The observer should not be able to distinguish between two obfuscated versions of C and two circuits drawn from $\mathcal{D}(\kappa, 2m_\kappa, \mathcal{P}_C)$. (In essence, this requirement says that two obfuscated versions of the same random circuit should not look "too much alike" compared to two independent draws from the underlying distribution $\mathcal{D}(\kappa, 2m_\kappa, \mathcal{P}_C)$.)
- 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 given the "halfway functionality" of the original circuit C, namely the permutation computed by the m_κ-gate prefix of C. In fact, we consider only the

[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 $\sigma(\kappa, n_\kappa, m_\kappa) = \tau(\kappa, n_\kappa, \xi(\kappa, n_\kappa, m_\kappa))$.

following partial information on that permutation: the observer is given $\widehat{C}_1 \stackrel{\mathrm{R}}{\leftarrow} \mathcal{E}_{(Z_1|C_{[1,m]}),|Z_1|\lambda_\kappa})$, $\widehat{C}_2 \stackrel{\mathrm{R}}{\leftarrow} \mathcal{E}_{(C_{[m,*]}|Z_2),|Z_2|\lambda_\kappa}$ for some fixed circuits Z_1, Z_2 and sufficiently large "leeway" λ_κ. (Indeed, with the premise that there exists $\lambda_\kappa \in \mathrm{poly}(\kappa)$ such that a random $(\mathrm{CC}(P)\lambda_\kappa)$-gate random circuit for a permutation P provides "no useful computational ability other than the ability to evaluate P", this requirement essentially says that an obfuscated version of C should not be correlatable with the functionality of the m_κ-gate prefix of C.)

Definition 14 (Random Input (RI) obfuscators:) *An obfuscator $\mathcal{O} = \{O_\kappa\}_{\kappa\in\mathcal{N}}$ is* **Random Input (RI)** *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.

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

II. There exists a leeway function $\lambda_\kappa \in$ poly(κ) 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:

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

Definition 15 (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, for simplicity of the treatment we only consider obfuscators that satisfy both requirements. (Indeed, our current candidate RI obfuscators do not distinguish between the two requirements.)

5 From RIO obfuscation to ROI for all circuits

This section presents the construction of ROI obfuscators for all circuits from RIO obfuscators. More specifically, Let $n^*_\kappa, m^*_\kappa, m^\#_\kappa$ be length functions that satisfy Assumption 8. 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 $C \xleftarrow{\text{R}} \mathcal{C}_{n^*_\kappa,m^*_\kappa}$, 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 $C = \pi(C')|\pi^\dagger(C'') : C', C'' \xleftarrow{\text{R}} \mathcal{C}_{n^*_\kappa,m^*_\kappa}$ and leeway $\lambda_\kappa \leq m^\#_\kappa$.

That is, we show:

Theorem 16 *Let $n^*_\kappa, m^*_\kappa, m^\#_\kappa$ be length functions that satisfy Assumption 8. If there exist algorithms O_1, π, O_2 such that:*

- *O_1 satisfies property I of RIO obfuscation for input distribution ensemble $\{C : C \xleftarrow{R} \mathcal{C}_{n^*_\kappa,m^*_\kappa}\}_{\kappa\in\mathbf{N}}$, 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 $C = \pi(C')|\pi^\dagger(C'') : C', C'' \xleftarrow{R} \mathcal{C}_{n^*_\kappa,m^*_\kappa}$ and leeway $\lambda_\kappa \leq m^\#_\kappa$.*

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

We present the construction and its analysis in four steps. First, we show how to construct ROI obfuscators for the identity function, with some specific parameters. (We call such obfuscators *pseudrandom identity generators.*)

Next we use random identity generators to construct ROI 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 ROI obfuscation for all reversible circuits.

Claim 22 in the Appendix demonstrates how to use an indistinguishability obfuscator for all reversible circuits obtain an indistinguishability obfuscator for all Boolean circuits.

5.1 Random Identity Generators

Random identity generators (RIGs) are separable ROI 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 ROI 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 17 (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 ROI 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 8. We construct an $(n^*_\kappa, 2m_\kappa)$-RIG G_κ given an obfuscator O that satisfies property I of RIO obfuscation (see Definition 14) 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 18 *Let $n^*_\kappa, m^*_\kappa, m^\#_\kappa$ be length functions that satisfy Assumption 8, 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 ROI 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:

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

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

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

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 8. Indeed, by Assumption 4, $\{C : C \xleftarrow{R} \mathcal{C}_{n^*_\kappa, m^*_\kappa}\}_{\kappa \in \mathbf{N}}$ is an SPRP. Since $|C'| = |C| = m_\kappa \geq m^\#_\kappa$, we can use Assumption 7 to conclude that:

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

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. □

5.2 ROI obfuscation of single gates

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

Definition 19 (Gate Obfuscators.) *An algorithm* $\mathrm{GO} = \{\mathrm{GO}_k\}_{\kappa \in \mathcal{N}}$ *is an* (n_κ, m_κ)**-gate obfuscator** *if, for any $\beta = \{\beta_\kappa \in \mathbb{B}_{n_\kappa}\}$, we have that* $\mathrm{GO}_\kappa(\beta_\kappa)$ *is an m_κ-separable ROI obfuscator for β_κ, with inner-stretch $\xi(\kappa, n_\kappa, 1) \geq 2m_\kappa$.*

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

Proof: To see that $\mathrm{GO}_\kappa(\beta)$ is an m_κ-separable ROI obfuscator for β, let $\pi = (\pi_1, \pi_2)$ be the post-processing algorithm guaranteed by Definition 17, such that

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

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

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

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 8 and let $\mathbf{C}_1 = \{C_{1,\kappa}\}_{\kappa\in\mathbf{N}}, \mathbf{C}_2 = \{C_{2,\kappa}\}_{\kappa\in\mathbf{N}}$ be circuit ensembles 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 ROI obfuscator RO_1 for ensemble 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 ROI obfuscator RO_2 for ensemble 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:

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

We use RO_1, RO_2, O to construct an ROI obfuscator $\mathrm{RO}_{1|2}$ for $C = \{C_\kappa\}_{\kappa \in \mathbf{N}}$, where $C_\kappa = C_{1,\kappa}|C_{2,\kappa}$. Obfuscator $\mathrm{RO}_{1|2,\kappa}$ proceeds as follows:

1. Sample $C_1 \xleftarrow{\mathrm{R}} \mathrm{RO}_{1,\kappa}(C_{1,\kappa})$ and $C_2 \xleftarrow{\mathrm{R}} \mathrm{RO}_{2,\kappa}(C_{2,\kappa})$.
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 $C_{1,1} = (C_1)_{[1,-t_{1,2}]}, C_{1,2} = (C_1)_{[-t_{1,2},*]}, C_{2,1} = (C_2)_{[1,t_{2,1}]}$, $C_{2,2} = (C_2)_{[t_{2,1},*]}$.

   Sample $G \xleftarrow{\mathrm{R}} O_\kappa(C_{1,2}|C_{2,1})$.
3. Output $C_{1,1}|G|C_{2,2}$.

Claim 21 *Let $n^*_\kappa, m^*_\kappa, m^\#_\kappa$ satisfy Assumption 8, 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 ROI 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 with inner-stretch fuction ξ_3 and with post-processing algorithm π_3, for the input distribution ensemble in (12). Then $\mathrm{RO}_{1|2}$ defined above is an ROI 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

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

Futhermore, 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 $\mathrm{RO}_{1|2}$.

Proof: Conceptually, the proof is a straightforward application of the security properties of the components used by $\mathrm{RO}_{1|2}$. However, since some of the distributions involved are not efficiently generatable, the actual proof requires some care. We show:

$$
\begin{aligned}
&\{C : C \xleftarrow{\mathrm{R}} \mathrm{RO}_{1|2,\kappa}(C_{1,\kappa}|C_{2,\kappa})\}_{\kappa\in\mathbf{N}} && = \\
&\{C_{1,1}|G|C_{2,2} : C_{1,1}|C_{1,2} \xleftarrow{\mathrm{R}} \mathrm{RO}_{1,\kappa}(C_{i,\kappa}); C_{2,1}|C_{2,2} \xleftarrow{\mathrm{R}} \mathrm{RO}_{2,\kappa}(C_{2,\kappa}); G \xleftarrow{\mathrm{R}} O_\kappa(C_{1,2}|C_{2,1})\}_{\kappa\in\mathbf{N}} && \stackrel{\mathrm{c}}{\approx} \quad (13) \\
&\{\pi_{1,1}(\widehat{C}_{1,1})|G|C_{2,2} : && (14) \\
&(\widehat{C}_{1,1}|\widehat{C}_{1,2} \xleftarrow{\mathrm{R}} \mathcal{E}_{C_{1,\kappa},m_{1,\kappa}m_\kappa}; C_{2,1}|C_{2,2} \xleftarrow{\mathrm{R}} \mathrm{RO}_{2,\kappa}(C_{2,\kappa}); G \xleftarrow{\mathrm{R}} O_\kappa(\pi_{1,2}(\widehat{C}_{1,2})|C_{2,1})\}_{\kappa\in\mathbf{N}} && \stackrel{\mathrm{c}}{\approx} \\
&\{\pi_{1,1}(\widehat{C}_{1,1})|G|\pi_{2,2}(\widehat{C}_{2,2}) : \widehat{C}_{1,1}|\widehat{C}_{1,2} \xleftarrow{\mathrm{R}} \mathcal{E}_{C_{1,\kappa},m_{1,\kappa}m_\kappa}; && (15) \\
&\widehat{C}_{2,1}|\widehat{C}_{2,2} \xleftarrow{\mathrm{R}} \mathcal{E}_{C_{2,\kappa},m_{2,\kappa}m_\kappa}; G \xleftarrow{\mathrm{R}} O_\kappa(\pi_{1,2}(\widehat{C}_{1,2})|(\pi_{2,1}(\widehat{C}_{2,1}))\}_{\kappa\in\mathbf{N}} && \stackrel{\mathrm{c}}{\approx} \\
&\{\pi_{1,1}(\widehat{C}_{1,1})|G|\pi_{2,2}(\widehat{C}_{2,2}) : && (16) \\
&C_{1,2}, C_{2,1} \xleftarrow{\mathrm{R}} \mathcal{C}_{n^*_\kappa,m^*_\kappa}; \widehat{C}_{1,2} \xleftarrow{\mathrm{R}} \mathcal{E}_{C_{1,2},m_\kappa}; \widehat{C}_{2,1} \xleftarrow{\mathrm{R}} \mathcal{E}_{C_{2,1},m_\kappa}; \widehat{C}_{1,1} \xleftarrow{\mathrm{R}} \mathcal{E}_{(C_{1,\kappa}|C^\dagger_{1,2}),m_{1,\kappa}m_\kappa}; \\
&\widehat{C}_{2,2} \xleftarrow{\mathrm{R}} \mathcal{E}_{(C^\dagger_{2,1}|C_{2,\kappa}),m_{2,\kappa}m_\kappa}; G \xleftarrow{\mathrm{R}} O_\kappa(\pi_{1,2}(\widehat{C}_{1,2})|(\pi_{2,1}(\widehat{C}_{2,1}))\}_{\kappa\in\mathbf{N}} && \stackrel{\mathrm{c}}{\approx} \\
&\{\pi_{1,1}(\widehat{C}_{1,1})|\pi_3(\widehat{G})|\pi_{2,2}(\widehat{C}_{2,2}) : && (17) \\
&C_{1,2}, C_{2,1} \xleftarrow{\mathrm{R}} \mathcal{C}_{n^*_\kappa,m^*_\kappa}; \widehat{C}_{1,2} \xleftarrow{\mathrm{R}} \mathcal{E}_{C_{1,2},m_\kappa}; \widehat{C}_{2,1} \xleftarrow{\mathrm{R}} \mathcal{E}_{C_{2,1},m_\kappa}; \widehat{C}_{1,1} \xleftarrow{\mathrm{R}} \mathcal{E}_{(C_{1,\kappa}|C^\dagger_{1,2}),m_{1,\kappa}m_\kappa}; \\
&\widehat{C}_{2,2} \xleftarrow{\mathrm{R}} \mathcal{E}_{(C^\dagger_{2,1}|C_{2,\kappa}),m_{2,\kappa}m_\kappa}; \widehat{G} \xleftarrow{\mathrm{R}} \mathcal{E}_{(C_{1,2}|C_{2,1}),\xi_3(\kappa,n^*_\kappa,2m_\kappa)}\}_{\kappa\in\mathbf{N}} && \stackrel{\mathrm{c}}{\approx} \\
&\{\pi_{1,1}((\widehat{C}_{1^+})_{[1,m_{1,\kappa}m_\kappa]})|\pi_3((\widehat{C}_{1^+})_{[m_{1,\kappa}m_\kappa,*]})|\pi_{2,2}(\widehat{C}_{2,2}) : && (18) \\
&C_{2,1} \xleftarrow{\mathrm{R}} \mathcal{C}_{n^*_\kappa,m^*_\kappa}; \widehat{C}_{1^+} \xleftarrow{\mathrm{R}} \mathcal{E}_{(C_{1,\kappa}|C_{2,1}),m_{1,\kappa}m_\kappa+\xi_3(\kappa,n^*_\kappa,2m_\kappa)}; \widehat{C}_{2,2} \xleftarrow{\mathrm{R}} \mathcal{E}_{(C^\dagger_{2,1}|C_{2,\kappa}),m_{2,\kappa}m_\kappa}\}_{\kappa\in\mathbf{N}} && \stackrel{\mathrm{c}}{\approx} \\
&\{\pi_{1,1}(C_{[1,m_{1,\kappa}+m_\kappa]})|\pi_3(C_{[m_{1,\kappa}m_\kappa+1,\xi_3(\kappa,n^*_\kappa,2m_\kappa)]})|\pi_{2,2}(C_{[-(m_{2,\kappa}-m_\kappa),*]}) : && (19) \\
&C \xleftarrow{\mathrm{R}} \mathcal{E}_{(C_{1,\kappa}|C_{2,\kappa}),m_{1,\kappa}m_\kappa+\xi_3(\kappa,n^*_\kappa,2m^*_\kappa)+m_{2,\kappa}m_\kappa}\}_{\kappa\in\mathbf{N}} && = \\
&\{\pi(C) : C \xleftarrow{\mathrm{R}} \mathcal{E}_{(C_{1,\kappa}|C_{2,\kappa}),\xi(\kappa,n^*_\kappa,m_{1,\kappa}m_{2,\kappa})}\}_{\kappa\in\mathbf{N}}.
\end{aligned}
$$

Experiment (14) differs from experiment (13) in that the application of $\mathrm{RO}_{1,\kappa}$ is replaced by a draw from $\mathcal{E}_{C_{1,\kappa},\xi_1(\kappa,n^*_\kappa,m_{1,\kappa})}$ (where $\xi_1(\kappa, n^*_\kappa, m_{1,\kappa}) = m_{1,\kappa}m_\kappa$) and applying $\pi_1 = (\pi_{1,1}, \pi_{1,2})$ to the result. Indistinguishability follows from the security of of RO_1. Indeed, by security of RO_1 we have:

$$\{\mathrm{RO}_{1,\kappa}(C_{1,\kappa})\}_{\kappa\in\mathbf{N}} \stackrel{\mathrm{c}}{\approx} \{\pi_{1,1}(C_{[1,-m_\kappa]})|\pi_{1,2}(C_{[-m_\kappa,*]}) : C \xleftarrow{\mathrm{R}} \mathcal{E}_{C_{1,\kappa},m_{1,\kappa}m_\kappa}\}_{\kappa\in\mathbf{N}}, \tag{20}$$

whereas a distinguisher A_κ between (13) and (14) can be used to distinguish between the two ensembles in (20): Given a circuit X, sample $C_{2,1}|C_{2,2} \xleftarrow{\mathrm{R}} \mathrm{RO}_{2,\kappa}(C_{2,\kappa})$, compute $G = O_\kappa(X_{[-t_{1,2},*]}|C_{2,1})$, and output $A_\kappa(X_{[1,-t_{1,2}]}|G|C_{2,2})$. (Recall that $t_{i,j} = \tau_{i,j}(\kappa, n^*_\kappa, m_\kappa)$ where $\tau_{i,j}$ is the stretch of $\pi_{i,j}$.) If X is drawn from $\mathrm{RO}_{1,\kappa}(C_{1,\kappa})$ then A_κ's input is drawn from (13); in the other case, A_κ's input is drawn from (14).

Experiment (15) differs from experiment (14) in that the application of $\mathrm{RO}_{2,\kappa}$ is replaced by a draw from $\mathcal{E}_{C_{2,\kappa},m_{2,\kappa}m_\kappa}$ and applying $\pi_2 = (\pi_{2,1}, \pi_{2,2})$ to the result. By security of RO_2 we have:

$$\{\mathrm{RO}_{2,\kappa}(C_{2,\kappa})\}_{\kappa\in\mathbf{N}} \stackrel{\mathrm{c}}{\approx} \{\pi_{2,1}(C_{[1,m_\kappa]})|\pi_{2,2}(C_{[m_\kappa,*]}) : C \xleftarrow{\mathrm{R}} \mathcal{E}_{C_{1,\kappa},m_{2,\kappa}m_\kappa}\}_{\kappa\in\mathbf{N}}. \tag{21}$$

As above, a distinguisher A_κ between (14) and (15) can be used to distinguish between the two ensembles in (20). Here however the straightforward reduction would require drawing a sample from

$\mathcal{E}_{C_{1,\kappa},m_{1,\kappa}m_\kappa}$, which is not known to be doable in polynomial time. We thus resort to a non-uniform reduction: by averaging, for any A_κ there must exist a polysize string $z = z_1|z_2$ such that A_κ distinguishes between

$$\{z_1|G|C_{2,2} : C_{2,1}|C_{2,2} \xleftarrow{\mathrm{R}} \mathrm{RO}_{2,\kappa}(C_{2,\kappa}); G \xleftarrow{\mathrm{R}} O_\kappa(z_2|C_{2,1})\}_{\kappa\in\mathbf{N}}\}_{\kappa\in\mathbf{N}} \tag{22}$$

and

$$\{z_1|G|\pi_{2,2}(\widehat{C}_{2,2}) : \widehat{C}_{2,1}|\widehat{C}_{2,2} \xleftarrow{\mathrm{R}} \mathcal{E}_{C_{2,\kappa},m_{2,\kappa}m_\kappa}; G \xleftarrow{\mathrm{R}} O_\kappa(z_2|\pi_{2,1}(\widehat{C}_{2,1})\}_{\kappa\in\mathbf{N}} \tag{23}$$

with the same advantage that it distinguishes between (14) and (15). This means that A_κ can be used to break RO_2 as follows: Given a circuit X, compute $G = O_\kappa(z_2|X_{[1,t_{2,1}]})$, and output $A_\kappa(z_1|G|X_{[t_{2,1},*]})$. Indeed, if X is drawn from $\mathrm{RO}_{2,\kappa}(C_{2,\kappa})$ then A_κ's input is drawn from (21); in the other case, A_κ's input is drawn from (21).

Experiment (16) differs from experiment (15) in the way that the circuits $\widehat{C}_{i,j}$ (for $i, j = 1, 2$) are chosen. Rather than choosing $\widehat{C}_{i,1}|\widehat{C}_{i,2} \xleftarrow{\mathrm{R}} \mathcal{E}_{C_{i,\kappa},m_{i,\kappa}m_\kappa}$ (for $i = 1, 2$), we now choose $C_{1,2}, \xleftarrow{\mathrm{R}} \mathcal{C}_{n^*_\kappa,m^*_\kappa}$, then $\widehat{C}_{1,2} \xleftarrow{\mathrm{R}} \mathcal{E}_{C_{1,2},m_\kappa}$, and then randomly choose $\widehat{C}_{1,1}$ so that $\widehat{C}_{1,1}|\widehat{C}_{1,2}$ computes $\mathcal{P}_{C_{1,\kappa}}$. Similarly, we choose $C_{2,1}, \xleftarrow{\mathrm{R}} \mathcal{C}_{n^*_\kappa,m^*_\kappa}$, then $\widehat{C}_{2,1} \xleftarrow{\mathrm{R}} \mathcal{E}_{C_{2,1},m_\kappa}$ and then randomly choose $\widehat{C}_{2,2}$ so that $\widehat{C}_{2,1}|\widehat{C}_{2,2}$ computes $\mathcal{P}_{C_{2,\kappa}}$.

Indistinguishability follows by applying Assumptions 4 and 7. First, by Assumption 4, each of $C_{1,2}, C_{2,1}$ is, individually, an SPRP. Furthermore, for each $i, j = 1, 2$ the circuit $\widehat{C}_{i,j}$ is longer than the SPRP that defines its functionality by sufficiently many gates for Assumption 7 to hold. (That is, $|\widehat{C}_{1,1}| \geq (|C_{1,\kappa}| + |C_{1,2}|)m^{\#}_\kappa$, $|\widehat{C}_{1,2}| \geq |C_{1,2}|m^{\#}_\kappa$, $|\widehat{C}_{2,1}| \geq |C_{2,1}|m^{\#}_\kappa$, and $|\widehat{C}_{2,2}| \geq (|C_{2,\kappa}| + |C_{2,1}|)m^{\#}_\kappa$.) We thus obtain:

$$\{\widehat{C}_{1,1}|\widehat{C}_{1,2} : \widehat{C}_{1,1}|\widehat{C}_{1,2} \xleftarrow{\mathrm{R}} \mathcal{E}_{C_{1,\kappa},m_{1,\kappa}m_\kappa}\}_{\kappa\in\mathbf{N}} \overset{\mathrm{c}}{\approx}$$
$$\{\widehat{C}_{1,1}, \widehat{C}_{1,2} : C_{1,2} \xleftarrow{\mathrm{R}} \mathcal{C}_{n^*_\kappa,m^*_\kappa}; \widehat{C}_{1,2} \xleftarrow{\mathrm{R}} \mathcal{E}_{C_{1,2},m_\kappa}; \widehat{C}_{1,1} \xleftarrow{\mathrm{R}} \mathcal{E}_{(C_{1,\kappa}|C^\dagger_{1,2}),m_{1,\kappa}m_\kappa}\}_{\kappa\in\mathbf{N}}$$

and similarly:

$$\{\widehat{C}_{2,1}, \widehat{C}_{2,2} : \widehat{C}_{2,1}|\widehat{C}_{2,2} \xleftarrow{\mathrm{R}} \mathcal{E}_{C_{2,\kappa},m_{2,\kappa}m_\kappa}\}_{\kappa\in\mathbf{N}} \overset{\mathrm{c}}{\approx}$$
$$\{\widehat{C}_{2,1}, \widehat{C}_{2,2} : C_{2,1} \xleftarrow{\mathrm{R}} \mathcal{C}_{n^*_\kappa,m^*_\kappa}; \widehat{C}_{2,1} \xleftarrow{\mathrm{R}} \mathcal{E}_{C_{2,1},m_\kappa}; \widehat{C}_{2,2} \xleftarrow{\mathrm{R}} \mathcal{E}_{(C^\dagger_{2,1}|C_{2,\kappa}),m_{2,\kappa}m_\kappa}\}_{\kappa\in\mathbf{N}}.$$

By transitivity it follows that

$$\{\widehat{C}_{1,1}, \widehat{C}_{1,2}, \widehat{C}_{2,1}, \widehat{C}_{2,2} : \tag{24}$$
$$\widehat{C}_{1,1}|\widehat{C}_{1,2} \xleftarrow{\mathrm{R}} \mathcal{E}_{C_{1,\kappa},m_{1,\kappa}m_\kappa}; \widehat{C}_{2,1}|\widehat{C}_{2,2} \xleftarrow{\mathrm{R}} \mathcal{E}_{C_{2,\kappa},m_{2,\kappa}m_\kappa}\}_{\kappa\in\mathbf{N}} \overset{\mathrm{c}}{\approx}$$
$$\{\widehat{C}_{1,1}, \widehat{C}_{1,2}, \widehat{C}_{2,1}, \widehat{C}_{2,2} : C_{1,2}, C_{2,1} \xleftarrow{\mathrm{R}} \mathcal{C}_{n^*_\kappa,m^*_\kappa}; \widehat{C}_{1,2}, \xleftarrow{\mathrm{R}} \mathcal{E}_{C_{1,2},m_\kappa}; \widehat{C}_{2,1}, \xleftarrow{\mathrm{R}} \mathcal{E}_{C_{2,1},m_\kappa}; \tag{25}$$
$$\widehat{C}_{1,1} \xleftarrow{\mathrm{R}} \mathcal{E}_{C_{1,\kappa}|\widehat{C}^\dagger_{1,2},m_{1,\kappa}m_\kappa}; \widehat{C}_{2,2} \xleftarrow{\mathrm{R}} \mathcal{E}_{C^\dagger_{2,1}|C_{2,\kappa},m_{1,\kappa}m_\kappa}\}_{\kappa\in\mathbf{N}}.$$

Now, consider the following distinguisher A'_κ between (24) and (25), given a disinguisher A_κ between (15) and (16): $A'_\kappa(\widehat{C}_{1,1}, \widehat{C}_{1,2}, \widehat{C}_{2,1}, \widehat{C}_{2,2}) = A_\kappa(\pi_{1,1}(\widehat{C}_{1,1})|\mathcal{O}_\kappa(\widehat{C}_{1,2}|\widehat{C}_{2,1})|\pi_{2,2}(\widehat{C}_{2,2}))$. Indeed, if the input of A'_κ is drawn from (24) then the input of A_k is drawn from (15); if the input of A'_κ is drawn from (25) then the input of A_k is drawn from (16).

Experiment (17) differs from experiment (16) in that $O_\kappa(C_{1,2}|C_{2,1})$ is replaced by $\pi_3(\widehat{C})$ where $\widehat{C} \xleftarrow{\mathrm{R}} \mathcal{E}_{C_{1,2}|C_{2,1},\xi_3(\kappa,n^*_\kappa,2m^*_\kappa)}$. Indistinguishability follows from the premise that O is an RIO obfuscator that satisfies property II for the input distribution ensemble $\mathcal{R}_\kappa$ from (12), with inner-stretch ξ_3 and post-processing algorithm π_3, and with respect to auxiliary circuits $Z_{1,\kappa} = C^\dagger_{1,\kappa}$, $\mathcal{Z}_{2,\kappa} = C^\dagger_{2,\kappa}$ and leeway $\lambda = m_\kappa \geq \lambda_\kappa$.

Indeed, let A_κ be an adversary that distinguishes (16) from (17), and construct the following O_κ-adversary A'_κ. Recall that A'_k is given input $(X, \widehat{C}_1, \widehat{C}_2)$, where X is either drawn from $O_k(C)$ for $C \xleftarrow{\text{R}} \mathcal{R}_\kappa$ or $X = \pi_3(\widehat{C})$ for $\widehat{C} \xleftarrow{\text{R}} \mathcal{E}_{C,\xi_3(\kappa,n^*_\kappa,2m_\kappa)}$, and where $\widehat{C}_1 \xleftarrow{\text{R}} \mathcal{E}_{C^\dagger_{1,\kappa}|C_{[1,m_\kappa]},\ell_{1,\kappa}}$ and $\widehat{C}_2 \xleftarrow{\text{R}} \mathcal{E}_{C_{[m_\kappa,*]}|C^\dagger_{2,\kappa},\ell_{2,\kappa}}$ where $\ell_{i,\kappa} = m_{i,\kappa} m_\kappa$. Adversary A'_κ will now output $A_\kappa(\pi_{1,1}(\widehat{C}^\dagger_1)|X|\pi_{2,2}(\widehat{C}^\dagger_2))$. If $X = \mathcal{O}_\kappa(C)$ for $C \xleftarrow{\text{R}} \mathcal{R}_\kappa$ then the input of A_k is drawn from (16); in the other case, the input of A_k is drawn from (17).

Experiment (18) differs from experiment (17) in how the first two components of the overall circuit are chosen, before the application of the post-processing algorithms: in experiment (17) the two components are chosen separately as $\widehat{C}_{1,1}|\widehat{G}$ where $\widehat{G}$ is a random circuit of the appropriate length that computes $C_{1,2}|C_{2,1}$, whereas in experiment (18) we have a single component $\widehat{C}_{1+}$, which is a random circuit of the appropriate length that computes $C_1|C_{2,1}$. Indistinguishability follows from Assumption 8. Indeed, by Assumption 8, for any circuit ensemble $\{Z_\kappa\}_{\kappa\in\mathbf{N}}$ where $\mathcal{Z}_\kappa \in \mathcal{C}_{n^*_\kappa,m^*_\kappa}$ we have:

$$\{\widehat{C}_{1,1}|\widehat{G} : C_{1,2} \xleftarrow{\text{R}} \mathcal{C}_{n^*_\kappa,m^*_\kappa}; \widehat{C}_{1,2} \xleftarrow{\text{R}} \mathcal{E}_{C_{1,2},m_\kappa}; \widehat{C}_{1,1} \xleftarrow{\text{R}} \mathcal{E}_{(C_{1,\kappa}|C^\dagger_{1,2}),m_{1,\kappa}m_\kappa};$$
$$\widehat{G} \xleftarrow{\text{R}} \mathcal{E}_{(C_{1,2}|Z_\kappa),\xi_3(\kappa,n^*_\kappa,2m_\kappa)}\}_{\kappa\in\mathbf{N}} \overset{\text{c}}{\approx} \{\widehat{C}_1 : \widehat{C}_{1+} \xleftarrow{\text{R}} \mathcal{E}_{(C_{1,\kappa}|Z_\kappa),m_{1,\kappa}m_\kappa+\xi_3(\kappa,n^*_\kappa,2m_\kappa)}\}_{\kappa\in\mathbf{N}}. \quad (26)$$

Now any adversary A_κ that distinguishes between (18) and (17) can be used to distinguish between the two distributions in (26) for randomly chosen $\{Z_\kappa \xleftarrow{\text{R}} \mathcal{C}_{n^*_\kappa,m^*_\kappa}\}_{\kappa\in\mathbf{N}}$, using $\widehat{C}_{2,2} \xleftarrow{\text{R}} \mathcal{E}_{C_{2,\kappa},m_{2,\kappa}m_\kappa}$ as non-uniform advice. By averaging, there exists also a singe sequence of circuits $\{Z_\kappa\}_{\kappa\in\mathbf{N}}$ for which the reduction holds.

Experiment (18) again differs from experiment (19) in how the two components of the overall circuit are chosen, before the application of the post-processing algorithms: in experiment (18) we have $\widehat{C} = \widehat{C}_{1+}|\widehat{C}_{2,2}$, whereas in experiment (19) $\widehat{C}$ is chosen uniformly from all circuits of the appropriate length that compute $C_{1,\kappa}|C_{2,\kappa}$. Indistinguishability again follows directly from Assumption 8.

The claim follows by transitivity of computational indistinguishability. □

5.4 ROI for all circuits

The ROI 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 8. 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 $\Gamma_i \xleftarrow{\text{R}} \text{GO}(\gamma_i)$ 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 21 that AS is an m_κ-separable ROI 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 Sections 5.2 and 5.1 above, Theorem 16 follows from Claims 18 and 20.

Furthermore, observe that the stretch of AS grows only linearly in m. Specifically, it follows from Claim 21 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 Sections 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 8.

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 with parameters that make them a viable basis for ROI (and in particular IO) obfuscators for all circuits as in Theorem 16.

Recall that Theorem 16 requires two types of RIO obfuscators: (a) an obfuscator O_1 that satisfies property I with respect to a random n-wire, m-gate input and with inner-stretch m', and (b) an obfuscator O_2 that satisfies property II when its input circuit has the distribution of two concatenated circuits that are an output of O_1. The parameter m is the number of gates needed for Assumption 4 to kick in, while m' is the number of gates needed for Assumption 7 to kick in. While we have evidence that m can be as low as $m = \tilde{O}(n)$ [CMR22], it appears that m' might need to be at least $\Omega(n^3)$. (As discussed in Section 3 following Assumption 8, when $m' = o(n^3)$ there might not be sufficiently many - and sufficiently diverse - m'-gate circuits that are functionally equivalent to a random m-gate circuit.)

For concreteness we thus concentrate on the case where $m = O(n \log^4 n)$ and $m' = n^3$. Furthermore, we first focus on realizing O_1, as this appears to be the more demanding requirement. (As we'll see, the construction scales naturally to diffent values of m and m'. Furthermore, the same construction also appears to satisfy the requirements of O_2.) Recall that, for this setting of the parameters, property I boils down to coming up with a post-processing algorithm π such that:

$$\{O(C), O(C) : C \overset{R}{\leftarrow} \mathcal{C}_{n_\kappa, n_\kappa \log n^4_\kappa}\}_{\kappa\in\mathbf{N}} \overset{c}{\approx} \{\pi(C), \pi(C') : C \overset{R}{\leftarrow} \mathcal{C}_{n_\kappa, n^3_\kappa}; C' \overset{R}{\leftarrow} \mathcal{R}_{C,n^4_\kappa}\}_{\kappa\in\mathbf{N}}.$$

That is, for large enough n it should be infeasible to distinguish between two obfuscated versions of a random n-wire, $n \log n^4$-gate circuit C, and the post-processed versions of two random n^4-gate circuits that are functionally equivalent to C.

6.1 Towards a mixing scheme

The general template. We proceed to describe the general template of the obfuscation process. As sketched in the Introduction, the template is to repeat the following process polynomially many times:

1. Pick an ℓ^{OUT}-gate sub-circuit C^{OUT} of the current circuit C.
2. Pick an ℓ^{IN}-gate circuit C^{IN} that's functionally equivalent to C^{OUT} (but is otherwise independent of C^{OUT}), and update C by replacing C^{IN} for C^{OUT}.

This process clearly rerandomizes the way in which C^{OUT} realizes the permutation it computes, $\mathcal{P}_{C^{\text{OUT}}}$. Furthermore, since the start and end locations of each new randomly chosen sub-circuit C^{OUT} to be replaced will likely be unaligned with the start and end locations of the sub-circuits chosen in previous iterations, the permutation computed by each subsequently replaced C^{OUT} will be randomized as well, as it will depend on the functionality of somewhat random *sub-circuits* of several replacement circuits C^{IN} made in previous iterations. The hope is then that this local randomization of both the functionality and the structure of short circuit segments will then cause the information about each segment of the circuit to be diffused throughout the mixed circuit.

Still, recall that, for most pairs of functionally equivalent circuits C_0, C_1, the corresponding perturbed variants $\tilde{C}_0, \tilde{C}_1$ of are unlikely to be statistically close. Indeed, had this been the case then, as discussed in the Introduction, then NP would equal coNP (at least in a distributional sense). This of course also means that the statistical distance between $(\tilde{C}_0, \tilde{C}_1)$ and two draws from $\tilde{C}_0$ is likely to be significant.

We thus settle for the more modest goal for of trying to get some confidence that two independent draws from $\tilde{C}$ do not bear any "efficiently discernable" resemblance to each other, aside from their overall functionality. Note that resemblance can come in any number of ways that combine structural and functional properties of corresponding sub-sequences of the two circuits, or even the circuits as a whole. At the same time, since our goal is only to make similarities hard to detect (rather than eliminate them altogether), it makes sense to concentrate on creating "limited diffusion" of the structure of relatively short segments of the given circuit, without attempting to obtain more global statistical mixing properties such as approaching some stable distribution over the space of perturbed versions of functionally equivalent circuits. (On the other hand, more modest statistical goals, such as having the random variable describing each gate in the mixed circuit depend on all the gates in the original circuit, may still be within reach.) Here the premise that the permutations computed by medium-length sub-circuits of the original circuit are pseudorandom becomes handy: it means that The mixing process only needs to be effective with respect to sub-circuits of short-to-medium size.

While the above template appears robust and realizable in a variety of ways, we describe a concrete realization method.

The skeleton graph. To facilitate describing the proposed mixing method, we introduce an alternative, somewhat more condensed representation of circuits that will make it easier to describe and visualize the effect of different perturbation methods. The idea is to factor out the ordering of gates that commute with each other. Indeed, while this ordering adds a significant amount of "nominal randomness" to circuits (namely it allows generating many different functionally equivalent variants of a given random circuit), this randomness is useless from the point of view of obfuscation since all variants generated this way can be easily traced back to the original. The alternative representation, called the *skeleton graph* of a circuit, factors out this randomness. Furthermore, it will provide useful structure for directing the choice of the sub-circuits C^{OUT} in the perturbation process.

Recall that each base permutation $\beta \in \mathbb{B}_n$ is defined by way of a control function ϕ, the wire $w_1 \in [n]$ connected to the active pin and the two wires $w_2, w_3 \in [n]$ connected to the non-active pins. Observe that if two base permutations $\beta = (\phi, w_1, w_2, w_3)$ and $\beta' = (\phi', w_1', w_2', w_3')$ *collide* (namely, they do not commute) then either $w_1 \in w_2', w_3'$ or else $w_1' \in w_2, w_3$. For $p \in [3]$, we say that β collides with β' on pin p if $w_p = w'_{p'}$ for some p'.

Let $C = \gamma_1 \dots \gamma_m$, where each $\gamma_i \in \mathbb{B}_n$, be an n-wire, m-gate circuit. The *skeleton graph* of C, denoted $S_C = (V_C, E_C)$, has the vertex set $V_C = g_1 \dots g_m$ g_i is labeled by the gate γ_i. The set of edges E_C contains a directed edge from g_i to g_j iff $i < j$, γ_i and γ_j collide, and no gate γ_k, $i < k < j$, collides both with γ_i and with γ_j. The edge (g_i, g_j) is labeled by the pins that γ_i and γ_j collide with each other on.

We say that circuits $C = \gamma_1 \dots \gamma_m$ and $C' = \gamma_1' \dots \gamma_m'$ *have the same skeleton* if their skeleton graphs are isomorphic, namely if there exists a permutation π on $[m]$ such that $\gamma_{\pi(i)} = \gamma_i'$ for all i, and furthermore $(g_{\pi(i)}, g_{\pi(j)}) \in E_C$ if and only if $(g_i', g_j') \in E_{C'}$. Observe that circuits that have the same skeleton are functionally equivalent. In the same way, any permutation π on $[m]$ that is an isomorphism on S_C results in a graph which is the skeleton graph of a circuit C' that is functionally equivalent to C. Furthermore, sampling randomly from all the circuits that have the same skeleton as a given circuit is easily doable in polynomial time.

Observe that both the in-degree and the out-degree of a vertex in the skeleton graph of an m-gate circuit can in principle be as large as $m - 1$. However, in the skeleton graph of a random circuit the expected in-degree of a vertex that is not a source is at most 4. Similarly, the expected out-degree of a non-sink vertex is at most 4. For all i the expected number of vertices that are at level i (i.e., vertices that are at distance i from the closest source) is roughly $n/4$. Similarly, the expected level of a sink is roughly $4m/n$.

Mixing using skeleton graphs. The skeleton graph provides a useful language both for describing the goal of the sub-circuit replacement process and on its mechanics.

Recall that the goal of the mixing process is to diffuse the structure and functionality of short circuit segments over large portions of the perturbed circuit (or even over the entire circuit). Restated in the language of skeleton graphs, a "circuit segment" translates to a *neighborhood* of vertices in the skeleton graph, namely a weakly connected set of vertices. (Recall that two vertices in a directed graph are weakly connected if they are connected in the underlying undirected graph.) That is, the mixing process is aimed at diffusing the structure and functionality of each small *neighborhood* across larger and larger portions of the mixed circuit, where these larger portions are in fact *neighborhoods* in and of themselves.

Indeed, the fact that the gates in a neighborhood collide (and in particular share wires) makes sure that, in either direction of the computation, the values of some output wires are likely to depend on the values of multiple input wires. Furthermore the overall connectivity within a neighborhood make it hard to find functionally equivalent alternatives to a neighborhood by factoring the neighborhood out to smaller subsets of gates and separately finding functionally equivalent realizations of the subsets. In other words, restricting attention to neighborhoods counters at least naive attempts to "reverse" (or otherwise break) obfuscation in time that is less than doubly exponential in the size of the circuit portion considered.

For the mechanics of the replacement process, observe that any convex subset of vertices in the skeleton graph of a circuit C naturally induces a sub-circuit C^{OUT} of C that's fit for replacement using the general template, in spite of the fact that C^{OUT} may consist of gates that are not consecutive in C.

That is, recall that a subset H of the vertices in a graph G is convex if for any two vertices $u, v \in H$, all vertices on any (directed) path from u to v are also in H. Now, let H be a convex set of vertices in the skeleton graph S of some circuit C. Since H is convex, we can rename (i.e., reorder) the vertices in S to obtain a graph S' that's isomorphic to S and in addition the vertices in H', the reordered version of H, are consecutive. Let $H' = g_1 \dots g_{\ell^{\text{OUT}}}$, let C^{OUT} be the circuit $\gamma_1 \dots \gamma_{\ell^{\text{OUT}}}$ where γ_i is the gate that labels g_i, let circuit $C^{\text{IN}} = \gamma'_1 \dots \gamma'_{\ell^{\text{IN}}}$ be functionally equivalent to circuit C^{OUT}, and let $S_{C^{\text{IN}}}$ denote the skeleton graph of C^{IN}. Then, let $\hat{S}$ be identical to S' except that the subgraph $S_{C^{\text{IN}}}$ is replaced for H'. That is, all the vertices in H' (and the edges connecting them) are removed, and instead the vertices in $S_{C^{\text{IN}}}$ are inserted, in sequence, such that the first vertex in $S_{C^{\text{IN}}}$ is the same ordinal in $\hat{S}$ as the first vertex of H' in S'. The edges in $\hat{S}$ are now determined by the gate labels of the vertices in the usual way for skeleton graphs. (In addition to the edges among the vertices of S' and edges among the vertices of $S_{C^{\text{IN}}}$, $\hat{S}$ may have edges that connect vertices of $S_{C^{\text{IN}}}$ and vertices of S'.) Observe that, at the end of the process, the graph $\hat{S}$ is the skeleton graph of a circuit $\hat{C}$ that has $m - \ell^{\text{OUT}} + \ell^{\text{IN}}$ gates and is functionally equivalent to C.

Furthermore, although the circuits C^{OUT} and C^{IN} are n-wire circuits, the numbers w^{OUT} and w^{IN} of *active wires* (i.e. the number of wires that are used by some gate) in C^{OUT} and C^{IN}, respectively, are bounded; specifically, $w^{\text{OUT}} \leq 3\ell^{\text{OUT}}$ and $w^{\text{IN}} \leq 3\ell^{\text{IN}}$. This means that, as long as $\ell^{\text{OUT}}, \ell^{\text{IN}} = O(\log \log n)$, the computational cost of choosing C^{IN} out of some distribution over ℓ^{IN}-gate circuits that are functionally equivalent to C^{OUT} is polynomial in n (and therefore in κ).[11]

6.2 The proposed scheme

We are now ready to describe the mixing process in more detail. We partition the process to two stages, as follows.

Inflationary stage. This stage provides initial mixing and randomization of the input circuit. The main goal here is to make sure that small neighborhoods are fully randomized, in the sense that both their structure and their functionality will be essentially independent of the input circuit. To do that, we let C^{OUT} be a relatively small random neighborhood in the skeleton graph of the current circuit, and let C^{IN} be a (functionally equivalent) circuit with more gates and more active wires. Specifically, fix ℓ^{OUT} to be some small constant. Then:

1. Pick a random gate γ in the current circuit C.

[11] The most computationally intensive part here is verifying that a candidate C^{IN} is functionally equivalent with C^{OUT}, which can be done in $O(2^{2^{w^{\text{IN}}}})$ time.

2. Pick a random ℓ^{OUT}-gate convex, weakly connected subset C^{OUT} of C that contains γ.[12] Let $P = \mathcal{P}_{C^{\text{OUT}}} \in \mathbb{A}_{2^n}$ be the permutation computed by C^{OUT}.

3. Let $\ell^{\text{IN}} = c\ell^{\text{OUT}}$, where c is another small constant. Let C^{IN} be a random ℓ^{IN}-gate, n-wire circuit that is functionally equivalent to C^{OUT}, and whose skeleton graph is weakly connected.

4. Update $C \leftarrow C \setminus C^{\text{OUT}} \cup C^{\text{IN}}$.

It is stressed that C^{IN} can have a different (and potentially larger) set of active wires than C^{OUT}. Still, since it has only ℓ^{IN} gates it can have at most $2\ell^{\text{IN}}$ active wires. It is thus useful to view the choice of C^{IN} as a two stage process, as follows. Let $\bar{P} \in \mathbb{A}_{2^{2\ell^{\text{IN}}}}$ be the projection of P on $2\ell^{\text{IN}}$ dimensions that include all the active dimensions of P. (A dimension i is active for P if there exist $x_1 \dots x_n$ such that either $P(x_1 \dots x_n)_i \neq x_i$ or else there exists $j \neq i$ such that $P(x_1 \dots x_n)_j \neq P(x_1 \dots x_{i-1}, 1 - x_i, x_{i+1} \dots x_n)_j$. It follows that if i is an active dimension of P then i is an active wire in any circuit that computes P.) To choose C^{IN}, first choose a random $2\ell^{\text{IN}}$-wire, ℓ^{IN}-gate weakly connected circuit $\bar{C}$ that computes $\bar{P}$. Next, extend $\bar{C}$ to an n-wire circuit by mapping each active dimension in P to the appropriate active wires in $\bar{C}$ so as to preserve functionality, and then mapping the remaining active wires in $\bar{C}$ to random unused wires in $[n]$.

Figure 3 depicts a simplified version of the replacement process for the case where $\ell^{\text{OUT}} = 2$ and $\ell^{\text{IN}} = 10$ (which corresponds to $c = 5$). We note that larger values of ℓ^{OUT} and c would naturally yield a larger set of replacement patterns.

As argued above, the restriction of C^{IN} to have a weakly connected skeleton graph (i.e., to consist of a single neighborhood) is aimed at creating larger and more densely connected neighborhoods that will make it harder to "reverse" the mixing process. The restriction of C^{OUT} to be single-neighborhood circuits to start with makes sure that $\mathcal{P}_{C^{\text{OUT}}}$ is realizable by many single-neighborhood circuits that are not much larger than C^{OUT}.

We conjecture that, after $\tilde{O}(m\kappa)$ iterations of the inflationary stage, the permutation $\mathcal{P}_{C_i^{\text{OUT}}}$ computed by each new circuit C_i^{OUT} is distributed almost independently of the m-gate input circuit. Furthermore, we conjecture that the same holds for the joint distribution of any collection of t permutations $\mathcal{P}_{C_{i_1}^{\text{OUT}}} \dots \mathcal{P}_{C_{i_t}^{\text{OUT}}}$, for any $t = o(m)$. (That is, there exists some fixed distribution $D_{n,m,t}$ such that for most input circuits C, the statistical distance between $\mathcal{P}_{C_{i_1}^{\text{OUT}}} \dots \mathcal{P}_{C_{i_t}^{\text{OUT}}}$ and $D_{n,m,t}$ is negligible in κ.) Setting $\kappa = n$ and $m = \tilde{O}(n)$, we conclude that $\tilde{O}(n^2)$ iterations suffice for that property to hold. However, since our goal is to construct an obfuscator with inner stretch of $O(n^3)$, we let the inflationary stage proceed for $\tilde{O}(n^4)$ iterations, resulting in an $\tilde{O}(n^4)$-gate obfuscated circuit.

We note however that, local statistical mixing properties aside, the mixing process as described so far can be effectively undone as follows. Given a mixed circuit C, search for neighborhoods C' of ℓ^{IN} vertices in the skeleton graph of C, such that $\text{CC}(C') \leq \ell^{\text{OUT}}$. Once such a neighborhood C' is found, replace it with the shortest functionally equivalent C'', and iterate. This reversing strategy is likely to be effective: a random $O(1)$-gate convex sub-circuit C' of a random circuit C is likely to have zero complexity gap, so ℓ^{IN}-gate neighborhoods with a significant complexity gap are likely to be the result of a replacement operation.[13] Furthermore, all same-size circuits C'' that are functionally equivalent to C' are very likely to have the same skeleton as C'. This holds even

[12] Here and for the rest of this section we conflate the circuit C and its skeleton graph S_C whenever there is little danger of confusion.

[13] Recall that The complexity gap of an m-gate circuit C is $\text{CG}(C) = m - \text{CC}(C)$.

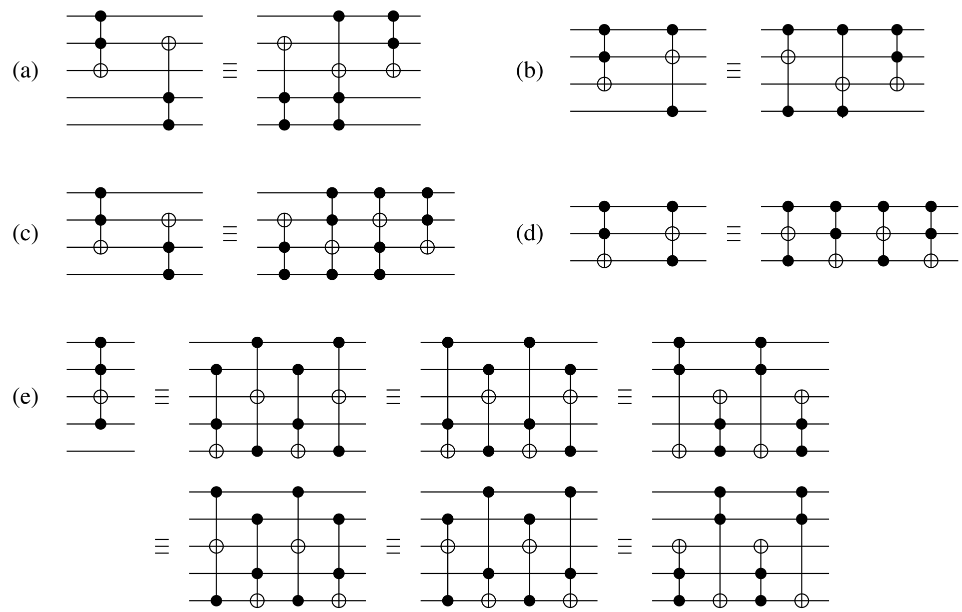

Figure 3: Some possible replacements for the case of $\ell^{\text{OUT}} = 2$ (namely, colliding pairs of gates), for the special case where the control function is $\phi(a, b) = ab$ (namely, logical conjunction). A gate is depicted as a vertical line connecting several wires, where the control wires 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.

if C' is restricted to have a weakly connected skeleton graph. (Note that this attack is inherently non-local, in that it requires access to the entire circuit. Consequently it does not contracdict the above conjectures regarding the staistical local mixing properties of the first stage.)

Kneading stage. As exemplified by the above attack, the "redundancy", or complexity gap introduced in the inflationary stage is still "too chunky:" it is concentrated in small neighborhoods and can thus be easily identified and removed. The second stage of the mixing process is aimed a spreading the complexity gap over increasingly larger neighborhoods so as to make it harder to recognize and remove. In particular, this stage counters the above attack and others like it.

The sub-circuit replacement operation in this stage is the same operation from the inflationary stage, but with (a) larger C^{OUT} and (b) C^{IN} of the same size as C^{OUT}. Specifically, we set a new parameter ℓ^{KND}, that is a significantly larger constant than ℓ^{IN} from the first stage. (In fact, we can have ℓ^{KND} be as large as $\Theta(\log\log n)$.) We then repeat the same process as in the inflationary stage, with the exceptions that C^{OUT} is a random ℓ^{KND}-gate, weakly connected, convex sub-circuit of C that contains the chosen gate γ, and C^{IN} is a random ℓ^{KND}-gate, weakly connected circuit that is functionally equivalent to C^{OUT}.

Recall that at the kneading stage the circuit has $\tilde{O}(n^4)$ gates. This stage proceeds for $\tilde{O}(n^5)$ iterations. Finally, apply a random isomorphism to the nodes of the final skeleton graph, and output the resulting graph (or, equivalently, the resulting circuit).

6.3 Arguments for security

While we keep a more rigorous security analysis of the proposed scheme out of scope for this work, we provide arguments supporting its security. We first describe the overall rationale (building on the observations made so far), and then discuss the specific properties needed for Theorem 16.

The idea underlying the two-stage structure of the scheme is to have the first stage inject initial randomization to the circuit. While randomization (or, complexity gap) is injected everywhere, it is injected in small pieces, where the pieces remain relatively localized and thus identifiable and removable. The second stage does not inject additional complexity gap, but instead mixes the complexity gap across larger and larger neighborhoods, thus making it computationally harder to identify and remove.

The argument proceeds as follows. First observe that the complexity gap is super-additive. That is, say that two convex neighborhoods in some circuit C are *adjoining* if their union is also a convex neighborhood in C. Then, for any adjoining neighborhoods C_1, C_2 in C, we have $\mathrm{CG}(C_1) + \mathrm{CG}(C_2) \leq \mathrm{CG}(C_1 \cup C_2)$.

Next, define the *inclusion set* s_i of iteration i in the mixing process as follows: Initialize $T \leftarrow C_i^{\text{OUT}}$. Now for $j = i-1 \dots 1$ do: if $C_j^{\text{IN}} \subseteq T$ then add j to s_i and update $T \leftarrow T \setminus C_j^{\text{IN}}$. It follows that $\mathrm{CG}(C_i^{\text{OUT}}) \geq \sum_{j \in s_i} \mathrm{CG}(C_j^{\text{IN}})$.

Finally, observe that the partition set of each one of the first few iterations i of the second stage is bound to be relatively large (specifically, its size is expected to be proportional to $\ell^{\text{KND}}/\ell^{\text{IN}}$). This means that, in these iterations, $\mathrm{CG}(C_i^{\text{OUT}})$ is expected to be proportional to $\ell^{\text{KND}}(\ell^{\text{IN}} - \ell^{\text{OUT}})/\ell^{\text{IN}}$, or in other words a significant fraction of ℓ^{KND}.

As the second stage continues, the inclusion set of the chosen circuit C_i^{OUT} is expected to shrink in size, since more and more of the C_j^{IN}'s from the first stage have been incorporated within earlier iterations from the second stage. This means that $\mathrm{CG}(C_i^{\text{OUT}})$ is expected to decrease.[14] Still, we conjecture that: (a) since the inclusion sets of C_i^{OUT} are initially large, $\mathrm{CG}(C_i^{\text{OUT}})$ decreases only gradually, and furthermore any putative variations in the complexity gap are only a function of the randomness in the mixing process itself rather than of the input circuit; (b) after $\tilde{\Theta}(n^5)$ iterations, the complexity gap of random ℓ^{KND}-gate neighborhoods is bound to stabilize below a small threshold; (c) once the complexity gap of random ℓ^{KND}-gate neighborhoods has been consistently small for $\tilde{\Theta}(n^5)$ iterations, the complexity gap of ℓ-gate neighborhoods remains a small fraction of ℓ/ℓ^{KND} even for $\ell \gg \ell^{\text{KND}}$.[15]

[14] Another potential contribution to the complexity gap of C_i^{OUT} can come from neighborhoods of the form $C_i^{\text{OUT}} \cap C_j^{\text{IN}}$ for a previous iteration j in the second stage. However, we don't expect such contribution to make up for the shrinking of the inclusion set. Indeed, if $\mathrm{CG}(C_j^{\text{IN}})$ is significant then for a random partitioning of C_j^{IN} into two neighborhoods C_1, C_2 we expect that $\mathrm{CG}(C_1) + \mathrm{CG}(\mathcal{C}_2) < \mathrm{CG}(C_j^{\text{IN}})$.

[15] Indeed,any ℓ-gate neighborhood for $\ell > \ell^{\text{KND}}$ can be partitioned to neighborhoods $C_1 \dots C_t$ such that each C_i is contained in at most a single C_i^{IN} for some i, which means that $\mathrm{CG}(C_i) \approx 0$. Furthermore, as long as, say, $\ell < |C|/2$, the permutations $\mathcal{P}_{C_1} \dots \mathcal{P}_{C_t}$ are distributed essentially independently from each other. Finally, for any two adjoining neighborhoods C_1, C_2 such that $\mathcal{P}_{C_1}$ and $\mathcal{P}_{C_2}$ are random and independent from each other, and both $\mathrm{CG}(C_1) \approx 0$ and $\mathrm{CG}(C_1) \approx 0$, we expect that $\mathrm{CG}(C_1 \cup\ C_2) \approx 0$ as well (that is, there would no significant new "shortcuts" in computing $\mathcal{P}_{C_1 \cup\ C_2}$).

Conjectured properties (a) and (b) mean that the naive reversing attack described above will no longer work, since there are no "ends of a thread" to start the unwinding process. Property (c) makes the stronger claim that any attack that's based on finding patterns in the complexity gaps of even medium-to-large circuit portions is doomed.

We proceed to consider the two specific properties of RIO obfuscation required in Therem 16. Since we opted to construct an obfuscator with inner stretch n^3 that's larger than the input size $n \log^4 n$, RIO obfuscation requires providing a post-processing algorithm, namely a "simulated obfuscation" algorithm, π, that takes as input a circuit $\widehat{C} \xleftarrow{R} \mathcal{E}_{C,n^3}$ for $C \xleftarrow{R} \mathcal{C}_{n,n\log^4 n}$, and outputs a 'simulated obfuscated circuit' $\pi(\widehat{C})$ that is supposed to mimic $O(C)$ in the security experiments.

The proposed algorithm π is identical to the obfuscator O, except that π runs the inflationary stage for somewhat fewer iterations so as to make sure that the output of π on an n^3-gate circuit will have the same length as the output of O on an $n \log^4 n$-gate circuit. We proceed to consider the two properties:

Property I, for random $n \log^4 n$-gate circuits: It should be infeasible to distinguish, given two functionally equivalent circuits (C_0, C_1), whether (C_0, C_1) are two obfuscated versions of a random n-wire, $n \log^4 n$-gate circuit C, or else the post-processed versions of two random n^3-gate circuits that are functionally equivalent to C.

We first study the distribution of pairs of circuits $\widehat{C}_0, \widehat{C}_1 \xleftarrow{R} \mathcal{E}_{C,m'}$, for $C \xleftarrow{R} \mathcal{C}_{n,m}$, in the case where Assumption 4 holds for n, m and Assumption 8 holds for n, m'. (Recall that here we further let $m = n \log^4 n$ and $m = n^3$.) In this case, it follows from the two assumptions that even large portions of the two circuits look completely independent from each other. In fact, any portion of, say, $\widehat{C}_0$ of size smaller than $m' - m$ gates is indistinguishable from a completely random circuit of the same size, even given the entire other circuit, $\widehat{C}_1$. Consequently, the pair $(\pi(\widehat{C}_0), \pi(\widehat{C}_1)$ exhibits similar independence properties: any portion of $\pi(\widehat{C}_0)$ of size smaller than $m' - m$ gates is indistinguishable from a portion of$\pi(R)$ (where R is a random circuit of the same size as C_0) — even given $\pi(\widehat{C}_1)$ in its entirety.

Conjectured properties (a), (b), (c) indicate that the pair C_1, C_2, where $C_1, C_2 \xleftarrow{R} O(C)$ and $C \xleftarrow{R} \mathcal{C}_{n,n\log^4 n}$, exhibits a similar behavior in face of a computationally bounded distinguisher. Indeed, these conjectures indicate that the only way to find similarities between any large-but-incomplete portion of one sample from $O(C)$ and another complete sample from $O(C)$ necessitates analyzing the functionalities and structures (in particular, the complexity gaps) of neighborhoods that are significantly larger than ℓ^{KND} — however such analysis is infeasible in polynomial time.

Property II, for $\tilde{O}(n^4)$-gate circuits: Essentially, property II requires that distinguishing between $O(C)$ for $C \xleftarrow{R} \mathcal{C}_{n,m}$ and $\pi(\widehat{C})$ for $\widehat{C} \xleftarrow{R} \mathcal{E}_{C,m'}$ be infeasible, even with oracle access to the $m/2$-gate prefix of C and its inverse. (This is a somewhat over-simplified version of the requirement: the actual requirement provides the distinguisher with two circuits, Z', Z'', where Z' is a sufficiently long random circuit that computes $M'|C'^{\dagger}$ and Z'' is a sufficiently long random circuit that computes $C''^{\dagger}|M''$, and M', M'' are arbitrary, known circuits.)

However, while this simplistic version of the requirement is unlikely to formally imply the actual one, for all practical purposes the two appear to be equivalent, since a distinguisher has no way to make use of Z', Z'' other than using them as oracles to the $m/2$-gate prefix and suffix of C, and their inverses.)

We first observe that, without the auxiliary information Z', Z'' or oracle access, indistinguishability essentially follows from Property I: $O(C) : C \xleftarrow{R} \mathcal{C}_{n,m} \overset{c}{\approx} \pi(\widehat{C}) : \widehat{C} \xleftarrow{R} \mathcal{E}_{C,m'}$. (The implication is not immediate, since we need property II to hold with respect to input circuits that are the concatenation of two obfuscated n-wire $n \log^4 n$-gate random circuits, namely $C = (O(C')|O(C'')) : C', C'' \xleftarrow{R} \mathcal{C}_{n,n\log^4 n}$, whereas the above argumentation considered property I with respect to randomly chosen circuits. However: (a) the above justification that the scheme satisfies property I appears to hold whenever most ℓ^{OUT}-gate neighborhoods in the input circuit have zero complexity gap, and (b) the output of the obfuscation scheme is likely to have that property as well.)

It remains to argue that the auxiliary information Z', Z'' does not help in distinguishing $O(C) : C \xleftarrow{R} \mathcal{C}_{n,m}$ from $\pi(\widehat{C}) : \widehat{C} \xleftarrow{R} \mathcal{E}_{C,m'}$. While making such an argument rigorous appears out of reach (in particular, the ability to generate Z', Z'' appears to require the existence of perfect obfuscators), we instead argue that having oracle access to $C', C'', C'^{\dagger}, C''\dagger$, where C' and C'' are the $m/2$-gate prefix and suffix of C, respectively, does not help. In particular, we argue that oracle access to $C', C'', C'^{\dagger}, C''\dagger$ is indistinguishable from oracle access to $R, R^{\dagger}C, R^{\dagger}, C^{\dagger}R$, where $R \xleftarrow{R} \mathbb{A}_{2^n}$ is a random permutation; furthermore, this is the case both when the distinguisher sees $\pi(\widehat{C})$ and when the distinguisher sees $O(C)$.

When the distinguisher sees $\pi(\widehat{C})$, the claim follows from Assumption 4. (In fact, the claim holds even if the distinguisher sees $\widehat{C}$ directly.) For the case where the distinguisher sees $O(C)$, we observe that oracle access to $C', C'', C'^{\dagger}, C''\dagger$ remains indistinguishable from oracle access to $R, R^{\dagger}C, R^{\dagger}, C^{\dagger}R$ even if all but $\tilde{O}(n)$ out of the $\Theta(n^4)$ gates in C' and C'' were known. This means that distinguishing between the two cases given $O(C)$ essentially implies a complete reversal of O, one that would also violate property I.

Unifying the two stages. The separation between the two stages has been made mainly for clarity of exposition of the ingredients of the scheme and the underlying rationale. However, it does not appear to be essential: consider instead the alternative mixing process that performs the following "unified" replacement step for $\tilde{O}(m\kappa)$ times, where m is the number of gates in the input circuit and $\kappa \approx n$ is the security parameter.

1. Pick $\ell^{\text{OUT}} \xleftarrow{R} \{2 \dots \ell^{\text{MAX}}\}$ where $\ell^{\text{MAX}} = O(\log \log n)$.
2. Choose a random gate γ in the current circuit and let C^{OUT} be a random ℓ^{OUT}-gate weakly connected, convex circuit that contains γ.
3. Let ℓ, w be the minimum number of gates and wires such that $|\mathcal{E}_{C^{\text{OUT}},\ell,w}| \geq L(\ell^{\text{OUT}}, w^{\text{OUT}}, w)$, where $\mathcal{E}_{C^{\text{OUT}},\ell,w}$ is the set of ℓ-gate, w-wire circuits that are functionally equivalent to the w-wire version of C^{OUT}, w^{OUT} is the number of active wires in C^{OUT}, and L is a predetermined function.[16] Let $\ell^{\text{IN}} = \min\{\ell, \ell^{\text{MAX}}\}$ and Let C^{IN} be a random single-neighborhood, ℓ^{IN}-gate, circuit with w active wires that's functionally equivalent to C^{OUT}.

This process combines the ingredients of the above two-stage process, providing for a more gradual transition from the first stage to the second. At first, when the complexity gap of small neighborhoods in the circuit is still small, we will have $\ell^{\text{IN}} > \ell^{\text{OUT}}$, allowing for effective functionality-preserving randomization of C^{OUT}. Later on, as the complexity gap grows, ℓ^{IN} will likely be not much

[16]By extension of the example given in Figure 3, we would like to have $L(n, \ell^{\text{OUT}}, w) \geq c\ell^{\text{OUT}}w$ for some constant c.

larger than ℓ^{OUT}, and even potentially smaller at times, allowing for the kneading process (namely, removing the complexity gap from small neighborhoods and shifting it to larger and larger ones) to take place.

7 Open questions and future directions

This work leaves open a number of intriguing open questions and research directions. We briefly mention some of them.

1. An immediate question that arises from this line of research is exploring the power of ROI obfuscation of reversible circuits. Indeed, ROI obfuscation with significant inner stretch, together with the SCP assumption, provide strong VBB-like hiding properties (as exhibited in the proof of Theorem 16). While a similar effect to ROI obfuscation can in principle be obtained using generic IO and one way functions with sub-exponential security (say, using Probabilistic IO [CLTV15b]), the interplay between ROI obfuscation and the SCP assumption appears powerful.

   In particular, can we use ROI obfuscation together with the SCP assumption to realize cryptographic primitives in ways that are simpler than known realizations based on plain IO for general circuits? Can the use of ROI security together with the SCP assumption get around the need for sub-exponential hardness assumptions? Can it avoid structural barriers that apply to generic IO schemes (e.g. [AS16])? Can it enable new applications?

2. Another immediate question is gaining better understanding of the local perturbation method for obtaining RIO obfuscation. Can we put this method on firmer ground? In particular, can we formulate a simple computational hardness assumption that provably suffices for the security of an obfuscation scheme based on local, functionality preserving random perturbations of the given circuit? In particular, can the security of the proposed scheme be based on the hardness of distinguishing between circuits with different computational complexities? Or even on one of the existing hardness assumptions regarding Kolmogorov complexity and MCSP (such as the ones in [LP20, LP21, IRS22, BLMP23, ILW23])?

   Also, can some variant of the local perturbation method be applied directly to arbitrary reversible circuits (as opposed to random circuits, as done in this work) and still provide concrete security guarantees? Alternatively, are there concrete counter examples of circuits for which some variant of this method - say, the one proposed in this work - would necessarily fail?

3. Can we obtain RIO obfuscation for short random circuits, based on more traditional computational hardness assumptions, such as, say, Learning With Errors? Recall that Assumption 8 might well hold for for $m^* = \tilde{O}(n)$ and $m^\# = \tilde{O}(n^3)$, in which case RIO obfuscation for circuits with n wires, $\tilde{O}(n)$ gates, and inner stretch $\tilde{O}(n^3)$ would suffice for obtaining fully fledged ROI obfuscation.

4. Can we improve on the the various constructions proposed in this work? For instance:

   (a) Do there exist efficient Random Identity Generators (RIGs, see Section 5.1) that provide statistical security?

(b) Alternatively, can ROI obfuscation for all circuits be constructed only from RIGs?

(c) Can a weaker version of RIO obfuscation suffice for obtaining ROI for all circuits? For instance, can property I of RIO obfuscation suffice in and of itself?

5. Obtaining better understanding of the SCP assumption, its potential relations to the hardness of Kolmogorov complexity and the MCSP problem, is another intriguing research direction. For instance, is it necessary to have a multiplicative computational gap $m_\kappa^\#$ (see Assumption 7), or can an additive gap suffice? It would also be interesting to find other ways to exploit this assumption in cryptographic applications — even ones that appear unrelated to program obfuscation.

6. Another intriguing direction is whether quantum techniques can be used to obtain more effective (or easier to analyze) obfuscation by way of locally applying functionality-preserving random perturbations. This question can in fact be asked at two different levels:

   (a) Can we have an obfuscator that uses quantum perturbations, but still takes a classical circuit as input and generates an obfuscated classical circuit?

   (b) Alternatively, Can we have an obfuscator that, using quantum perturbations, transforms a classical circuit into an obfuscated quantum circuit that, on any classical input, outputs a quantum state that has high fidelity with the output of the input circuit?

7. Yet another question is whether notions and techniques from this work can be extended to the case of obfuscating quantum circuits (as in, say, [BK21])? Furthermore, can they result in an obfuscation process where both the input and the output are classical descriptions of quantum circuits? Can the obfuscation process itself be classical?

Acknowledgements

We thank Luowen Qian for participating in early stages of this research. R.C. also thanks Nir Bitansky, Shafi Goldwasser and Omer Paneth for very helpful discussions.

References

[AB15] Benny Applebaum and Zvika Brakerski. Obfuscating circuits via composite-order graded encoding. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, volume 9015 of *Lecture Notes in Computer Science*, pages 528–556. Springer, 2015.

[AHP10] Nouraddin Alhagi, Maher Hawash, and Marek A. Perkowski. Synthesis of reversible circuits with no ancilla bits for large reversible functions specified with bit equations. In *40th IEEE International Symposium on Multiple-Valued Logic, ISMVL 2010, Barcelona, Spain, 26-28 May 2010*, pages 39–45. IEEE Computer Society, 2010.

[AJJ14] Gorjan Alagic, Stacey Jeffery, and Stephen P. Jordan. Circuit obfuscation using braids. In Steven T. Flammia and Aram W. Harrow, editors, *9th Conference on the Theory of*

Quantum Computation, Communication and Cryptography, TQC 2014, May 21-23, 2014, Singapore, volume 27 of *LIPIcs*, pages 141–160. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2014.

[AJS15] Prabhanjan Ananth, Abhishek Jain, and Amit Sahai. Achieving compactness generically: Indistinguishability obfuscation from non-compact functional encryption. *IACR Cryptol. ePrint Arch.*, page 730, 2015.

[AS16] Gilad Asharov and Gil Segev. Limits on the power of indistinguishability obfuscation and functional encryption. *SIAM J. Comput.*, 45(6):2117–2176, 2016.

[Bar86] David A. Mix Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in nc^1. In Juris Hartmanis, editor, *Proceedings of the 18th Annual ACM Symposium on Theory of Computing, May 28-30, 1986, Berkeley, California, USA*, pages 1–5. ACM, 1986.

[Bar17] Boaz Barak. The complexity of public-key cryptography. In Yehuda Lindell, editor, *Tutorials on the Foundations of Cryptography*, pages 45–77. Springer International Publishing, 2017.

[Ben73] Charles H. Bennett. Logical reversibility of computation. *Ibm Journal of Research and Development*, 17:525–532, 1973.

[Ben89] Charles H. Bennett. Time/space trade-offs for reversible computation. *SIAM Journal on Computing*, 18(4):766–776, 1989.

[BGI+01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, volume 2139 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2001.

[BGI+12] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *J. ACM*, 59(2):6:1–6:48, 2012.

[BGK+14] Boaz Barak, Sanjam Garg, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Protecting obfuscation against algebraic attacks. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 221–238. Springer, 2014.

[BK21] Anne Broadbent and Raza Ali Kazmi. Constructions for quantum indistinguishability obfuscation. In Patrick Longa and Carla Ràfols, editors, *Progress in Cryptology - LATINCRYPT 2021 - 7th International Conference on Cryptology and Information Security in Latin America, Bogotá, Colombia, October 6-8, 2021, Proceedings*, volume 12912 of *Lecture Notes in Computer Science*, pages 24–43. Springer, 2021.

[BLMP23] Marshall Ball, Yanyi Liu, Noam Mazor, and Rafael Pass. Kolmogorov comes to cryptomania: On interactive kolmogorov complexity and key-agreement. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 458–483, 2023.

[BPW16] Nir Bitansky, Omer Paneth, and Daniel Wichs. Perfect structure on the edge of chaos - trapdoor permutations from indistinguishability obfuscation. In Eyal Kushilevitz and Tal Malkin, editors, *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, volume 9562 of *Lecture Notes in Computer Science*, pages 474–502. Springer, 2016.

[Bro04] Alex Brodsky. Reversible circuit realizations of boolean functions. In Jean-Jacques Lévy, Ernst W. Mayr, and John C. Mitchell, editors, *Exploring New Frontiers of Theoretical Informatics, IFIP 18th World Computer Congress, TC1 3rd International Conference on Theoretical Computer Science (TCS2004), 22-27 August 2004, Toulouse, France*, volume 155 of *IFIP*, pages 67–80. Kluwer/Springer, 2004.

[BV15] Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation from functional encryption. *IACR Cryptol. ePrint Arch.*, page 163, 2015.

[Can97] Ran Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In Burton S. Kaliski Jr., editor, *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, volume 1294 of *Lecture Notes in Computer Science*, pages 455–469. Springer, 1997.

[CG75] Don Coppersmith and Edna Grossman. Generators for certain alternating groups with applications to crytography. *SIAM Journal on Applied Mathematics*, 29(4):624–627, 1975.

[CGH+15] Jean-Sébastien Coron, Craig Gentry, Shai Halevi, Tancrède Lepoint, Hemanta K. Maji, Eric Miles, Mariana Raykova, Amit Sahai, and Mehdi Tibouchi. Zeroizing without low-level zeroes: New MMAP attacks and their limitations. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 247–266. Springer, 2015.

[CHVW19] Yilei Chen, Minki Hhan, Vinod Vaikuntanathan, and Hoeteck Wee. Matrix prfs: Constructions, attacks, and applications to obfuscation. In Dennis Hofheinz and Alon Rosen, editors, *Theory of Cryptography - 17th International Conference, TCC 2019, Nuremberg, Germany, December 1-5, 2019, Proceedings, Part I*, volume 11891 of *Lecture Notes in Computer Science*, pages 55–80. Springer, 2019.

[CJMR22] Claudio Chamon, Jonathan Jakes-Schauer, Eduardo R. Mucciolo, and Andrei E. Ruckenstein. Encrypted operator computing: an alternative to fully homomorphic encryption. *CoRR*, abs/2203.08876, 2022.

[CLTV15a] Ran Canetti, Huijia Lin, Stefano Tessaro, and Vinod Vaikuntanathan. Obfuscation of probabilistic circuits and applications. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, volume 9015 of *Lecture Notes in Computer Science*, pages 468–497. Springer, 2015.

[CLTV15b] Ran Canetti, Huijia Lin, Stefano Tessaro, and Vinod Vaikuntanathan. Obfuscation of probabilistic circuits and applications. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, volume 9015 of *Lecture Notes in Computer Science*, pages 468–497. Springer, 2015.

[CMR22] Claudio Chamon, Eduardo R Mucciolo, and Andrei E Ruckenstein. Quantum statistical mechanics of encryption: Reaching the speed limit of classical block ciphers. *Annals of Physics*, 446:169086, 2022.

[CRMC23] Claudio Chamon, Andrei E Ruckenstein, Eduardo R Mucciolo, and Ran Canetti. Circuit complexity and functionality: a thermodynamic perspective. *arXiv preprint arXiv:2309.05731*, 2023.

[CRV10] Ran Canetti, Guy N. Rothblum, and Mayank Varia. Obfuscation of hyperplane membership. In Daniele Micciancio, editor, *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings*, volume 5978 of *Lecture Notes in Computer Science*, pages 72–89. Springer, 2010.

[CV13] Ran Canetti and Vinod Vaikuntanathan. Obfuscating branching programs using black-box pseudo-free groups. *IACR Cryptol. ePrint Arch.*, page 500, 2013.

[CVW18] Yilei Chen, Vinod Vaikuntanathan, and Hoeteck Wee. GGH15 beyond permutation branching programs: Proofs, attacks, and candidates. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II*, volume 10992 of *Lecture Notes in Computer Science*, pages 577–607. Springer, 2018.

[DQV$^+$21] Lalita Devadas, Willy Quach, Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Succinct LWE sampling, random polynomials, and obfuscation. In Kobbi Nissim and Brent Waters, editors, *Theory of Cryptography - 19th International Conference, TCC 2021, Raleigh, NC, USA, November 8-11, 2021, Proceedings, Part II*, volume 13043 of *Lecture Notes in Computer Science*, pages 256–287. Springer, 2021.

[Fei74] Horst Feistel. Block cipher cryptographic system, May 3 1974. US Patent 3,798,359.

[GGH15] Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, volume 9015 of *Lecture Notes in Computer Science*, pages 498–527. Springer, 2015.

[GGHR14] Sanjam Garg, Craig Gentry, Shai Halevi, and Mariana Raykova. Two-round secure MPC from indistinguishability obfuscation. In Yehuda Lindell, editor, *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014, San Diego, CA, USA, February 24-26, 2014. Proceedings*, volume 8349 of *Lecture Notes in Computer Science*, pages 74–94. Springer, 2014.

[GGSW13] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 467–476, 2013.

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

[GP21] Romain Gay and Rafael Pass. Indistinguishability obfuscation from circular security. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 736–749. ACM, 2021.

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

[Had00] Satoshi Hada. Zero-knowledge and code obfuscation. In Tatsuaki Okamoto, editor, *Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings*, volume 1976 of *Lecture Notes in Computer Science*, pages 443–457. Springer, 2000.

[HB05] Shlomo Hoory and Alex Brodsky. Simple permutations mix even better, 2005.

[HMMR05] Shlomo Hoory, Avner Magen, Steven Myers, and Charles Rackoff. Simple permutations mix well. *Theoretical Computer Science*, 348(2):251–261, 2005. Automata, Languages and Programming: Algorithms and Complexity (ICALP-A 2004).

[ILW23] Rahul Ilango, Jiatu Li, and Ryan Williams. Indistinguishability obfuscation, range avoidance, and bounded arithmetic. *Electron. Colloquium Comput. Complex.*, TR23-038, 2023.

[IRS22] Rahul Ilango, Hanlin Ren, and Rahul Santhanam. Robustness of average-case meta-complexity via pseudorandomness. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1575–1583. ACM, 2022.

[JLS21] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 60–73. ACM, 2021.

[Kil88] Joe Kilian. Founding cryptography on oblivious transfer. In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 20–31. ACM, 1988.

[LP20] Yanyi Liu and Rafael Pass. On one-way functions and kolmogorov complexity. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 1243–1254. IEEE, 2020.

[LP21] Yanyi Liu and Rafael Pass. Cryptography from sublinear-time average-case hardness of time-bounded kolmogorov complexity. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 722–735. ACM, 2021.

[LPS04] Ben Lynn, Manoj Prabhakaran, and Amit Sahai. Positive results and techniques for obfuscation. *IACR Cryptol. ePrint Arch.*, page 60, 2004.

[LPST16] Huijia Lin, Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation with non-trivial efficiency. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *Public-Key Cryptography - PKC 2016 - 19th IACR International Conference on Practice and Theory in Public-Key Cryptography, Taipei, Taiwan, March 6-9, 2016, Proceedings, Part II*, volume 9615 of *Lecture Notes in Computer Science*, pages 447–462. Springer, 2016.

[Sel18] Peter Selinger. A finite alternation result for reversible boolean circuits. *Sci. Comput. Program.*, 151:2–17, 2018.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In David B. Shmoys, editor, *STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 475–484. ACM, 2014.

[Tof80] Tommaso Toffoli. Reversible computing. In J. W. de Bakker and Jan van Leeuwen, editors, *Automata, Languages and Programming, 7th Colloquium, Noordweijkerhout, The Netherlands, July 14-18, 1980, Proceedings*, volume 85 of *Lecture Notes in Computer Science*, pages 632–644. Springer, 1980.

[WW21] Hoeteck Wee and Daniel Wichs. Candidate obfuscation via oblivious LWE sampling. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part III*, volume 12698 of *Lecture Notes in Computer Science*, pages 127–156. Springer, 2021.

[Xu15] Siyao Xu. Reversible logic synthesis with minimal usage of ancilla bits, 2015.

A From IO for reversible circuits to IO for all circuits

We demonstrate how an obfuscation scheme for reversible circuits can be used as a general-purpose obfuscation scheme for all Boolean circuits.

There is a substantial body of literature regarding the synthesis of reversible circuits, and in particular regarding how to represent general computation within a reversible one, see e.g. [Ben73, Tof80, Ben89, Bro04, AHP10, Xu15, Sel18]. This study has a variety of motivations, ranging from basic feasibility results to the optimization of various parameters, mostly either for the purpose of minimizing energy consumption or to enable quantum computation.

However, while it is in principle possible to represent any even permutation on $\{0,1\}^n$ using only base permutations (specifically Toffoli gates with at most two control wires), all methods that we are aware of for embedding general Boolean circuits in reversible ones in an "efficiency preserving way" (namely in a way that preserves the number of gates and wires up to polynomial factors) are only guaranteed to preserve correctness when some input wires are "mantissa wires", i.e. have a fixed value (wlog, 0). Such a guarantee does not suffice in and of itself for the purpose of using an obfuscator for reversible circuits to obfuscate non-reversible circuits, since it leaves the functionality of the reversible circuit unspecified when some of the mantissa wires have other values. Indeed, this under-specification opens the door to situations where even a perfectly obfuscated version of the reversible embedding of a non-reversible circuit C would leak information on the internals of C when the mantissa wires are set to "illegitimate" values.

We describe a simple way around this caveat, by demonstrating how to embed general Boolean circuits within reversible circuits in an efficiency preserving way, so that the embedded versions of any two functionally equivalent Boolean circuits are functionally equivalent as well (as reversible circuits). More specifically, whenever fed with illegitimate mantissa values the embedded circuit's output will be identical to its input.

We first sketch (a somewhat rephrased version of) Toffoli's method for embedding any general Boolean circuit within a reversible circuit [Tof80]. Consider the following transform TO that, given a general Boolean circuit C with α input wires, β output wires, μ NAND gates, and width ω, generates a reversible circuit $C' = \mathrm{TO}(C)$ with $n = 4\omega$ wires and $m = O(\mu)$ gates, and such that $C'(x, 0^{\beta+\delta}) = (x, C(x), 0^{\delta})$ for any $x \in \{0,1\}^{\alpha}$. (Here $\delta = n - \alpha - \beta$.)

Without loss of generality we assume that the original circuit C is layered, namely that the input wires of each NAND gate at layer i connect to output wires of NAND gates at layer $i-1$ (or to input wires of the circuit if $i = 1$); similarly, the output wires of a NAND gate at layer i connect to input wires of NAND gates at layer $i+1$ (or to output wires of the circuit if i is the last layer). In this setting, ω is the maximum (over all i) number of wires from layer i to layer $i+1$.

The construction proceeds as follows. We partition the workspace wires z into two batches of ω wires each. For each NAND gate g at the first layer of C, where g has fanout ϕ, append ϕ Toffoli gates $T_1 \ldots T_{\phi}$ to the reversible circuit C', where the control wires of each T_i correspond to the input wires of g, the control function is NAND, and the active wire of T_i is a free workspace wire from the first batch.

For each gate g in a subsequent even (resp., odd) layer j of C, except for the last one, follow the same process except that the control wires for each T_i are the corresponding workspace wires from the first (resp., second) batch of workspace wires, and the control wires are fresh wires from the second (resp., first) batch of workspace wires. Next, uncompute the Toffoli gates of the previous layer by writing these gates again in reverse order. (This will revert all the active wires in that layer to their original value of 0, ahead of using these wires again as active wires in the next layer).

For the last layer of gates in C, follow a similar process as above except that the active wires are taken to be fresh output wires out of $y_1 \dots y_\beta$.

Observe that, for any Boolean circuit C, TO implicitly defines a function $f_C : \{0,1\}^{\alpha+\delta} \to \{0,1\}^\beta$ such that $\mathrm{TO}(C)(x,y,z) = (x, y \oplus f(x,z), z)$ for any $x \in \{0,1\}^\alpha, y \in \{0,1\}^\beta, z \in \{0,1\}^\delta$. In particular, we have that (a) the values of x, z do not change under C', and (b) $\mathrm{TO}(C)$ is the inverse of itself, i.e. $\mathrm{TO}(C)|\mathrm{TO}(C)$ computes the identity permutation. Finally, f_C satisfies $f_C(x, 0^\delta) = C(x)$ for all $x \in \{0,1\}^\alpha$.

It remains to "harden" TO to make sure that $f(x,z)$ does not "leak unintended information on C" even when $z \neq 0^\delta$. More concretely, we would like to have a transform where the transformed versions of any two functionally equivalent Boolean circuits are functionally equivalent as well (as reversible circuits). This is done via the following "hardened Toffoli" transform, HTO. Given a general Boolean circuit C with the same parameters as above, $C" = \mathrm{HTO}(C)$ is a reversible circuit with $n" = \alpha + \beta + 2\delta$ wires and such that:

$$C"(x,y,z,u) = \begin{cases} (x, y + C(x), z, u) & \text{if } z = 0^\delta \\ (x,y,z,u) & \text{if } z \neq 0^\delta \end{cases}$$

where $u = u_1 \dots u_\delta$, and each $u_i \in \{0,1\}$. In other words, HTO makes sure that $f_C(x,z) = 0^\delta$ whenever $z \neq 0^\delta$ (while still guaranteeing that $f_C(x, 0^\delta) = C(x)$).

Circuit $C" = \mathrm{HTO}(C)$ is composed of two sub-circuits, where each sub-circuit is run twice. That is, $C" = S|T|S|T$, where S is a "controlled $\mathrm{TO}(C)$" circuit where the control wire is the last input wire (i.e u_δ), and T flips u_δ iff $z = 0^\delta$. Neither of these circuits require any mantissa wires. Specifically:

$$S(x,y,z,u) = \begin{cases} (C'(x,y,z), u) & \text{if } u_\delta = 0 \\ (x,y,z,u) & \text{if } u_\delta = 1 \end{cases}$$

$$T(x,y,z,u) = \begin{cases} (x,y,z,u_1 \dots, u_{\delta-1}, u_\delta \oplus 1) & \text{if } z = 0^\delta \\ (x,y,z,u) & \text{if } z \neq 0^\delta \end{cases}$$

Indeed, consider an input value (x,y,z,u). if $z \neq 0^\delta$ then the value of u_δ remains unchanged throughout, which means that $\mathrm{HTO}(C)(x,y,z,u) = S(S(x,y,z,u)) = (x,y,z,u)$. (Here we use the fact that $\mathrm{TO}(C)$ always preserves the value of z and that $\mathcal{P}_{\mathrm{TO}(C)|\mathrm{TO}(C)} = I_{n"}$.) On the other hand, if $z = 0^\delta$ then both applications of T flip the value of u_δ, in which case exactly one out of the two instances of S computes the identity function and so $\mathrm{HTO}(C)(x,y,0^\delta,u) = (\mathrm{TO}(x,y,0^\delta), u)) = (x, y \oplus C(x), 0^\delta, u)$.

It remains to describe how S and T are implemented. To construct S, first modify $\mathrm{TO}(C)$ by adding a control wire to each gate g and connecting that control wire to u_δ - i.e. for each gate $g(a,b,c) = a \oplus f(b,c)$ in $\mathrm{TO}(C)$ add the gate $g'(a,b,c) = a \oplus f(b,c) \cdot u_\delta$ to S. Next, factor each such three-control-wires gate in S to a functionally equivalent sequence of gates with two control wires.[17]

To construct T, we observe that in the special case of computing the OR over i wires it is possible to gradually accumulate partial results using $i-1$ uninitialized "borrowed wires", without relying on any initialized mantissa wires.

[17]For instance, the sequence $[u_\delta = u_\delta \oplus a; a = a \oplus \phi(b,c); u_\delta = u_\delta \oplus a; a = a \oplus u_\delta; u_\delta = u_\delta \oplus \phi(b,c);]$ is functionally equivalent to $[a = a \oplus \phi(b,c) \cdot u_\delta]$.

That is, let $z = z_1 \ldots z_\delta$, and let Z_1 be the circuit $Z_1 = [u_1 = u_1 \oplus (z_1 z_2 \oplus 1)]$. For $i > 1$ let $Z_i = [u_i = u_i \oplus (u_{i-1} z_{i+1} \oplus 1)); Z_{i-1}; u_i = u_i \oplus (u_{i-1} z_{i+1} \oplus 1))]$. It can be seen by induction that Z_i flips the value of u_i if and only if $z_1 = \ldots = z_{i+1} = 0$.

Circuit T now first computes $Z_{\delta-1}$, then transfers the information from wire $u_{\delta-1}$ to wire u_δ, and then uncomputes $Z_{\delta-1}$. More specifically $T = [u_\delta = u_\delta \oplus u_{\delta-1}; Z_{\delta-1}; u_\delta = u_\delta \oplus u_{\delta-1}; Z^\dagger_{\delta-1}]$.

Now, let $\mathcal{O}$ be an obfuscator for all reversible circuits, and let $\mathcal{O}_{\text{HTO}}$ be the following obfuscator for all circuits. Given a (not necessarily reversible) circuit C with α input bits and β output bits, first transform C to a layered circuit C' consisting of NAND gates. Next sample $C" \stackrel{\text{R}}{\leftarrow} \mathcal{O}(\text{HTO}(C'))$ and output the circuit $\hat{C}$ that, given input $x \in \{0,1\}^\alpha$, computes $y = C"(x, 0^{n-\alpha})$ and outputs $y_{\alpha+1} \ldots y_{\alpha+\beta}$. The following claim follows immediately from the fact that $\text{HTO}(C_1)$ and $\text{HTO}(C_2)$ are same size and functionally equivalent whenever C_1, C_2 are same size and functionaly equivalent:

Claim 22 *If $\mathcal{O}$ is an indistinguishability obfuscator for all reversible circuits then $\mathcal{O}_{\text{HTO}}$ is an indistinguishability obfuscator for all circuits.* $\square$