Protecting Circuits from Computationally-Bounded Leakage

Sebastian Faust[1], Leonid Reyzin[2] and Eran Tromer[3]

July 31, 2009

[1] K.U. Leuven ESAT-COSIC, Kasteelpark Arenberg 10, bus 2446
B-3001 Leuven-Heverlee, Belgium
sebastian.faust@esat.kuleuven.be
[2] Boston University, Department of Computer Science
111 Cummington St Boston, MA 02215, USA
reyzin@cs.bu.edu
[3] Massachusetts Institute of Technology
Computer Science and Artificial Intelligence Laboratory
32 Vassar St., Cambridge, MA 02139, USA
tromer@csail.mit.edu

Abstract. Physical computational devices leak side-channel information that may, and often does, reveal secret internal states. We present a general transformation that compiles any circuit into a device that maintains secrecy even in the presence of well-defined classes of side-channel leakage. Our construction requires only a minimal leak-proof component: one that draws random elements from a simple distribution. We thus reduce the problem of shielding arbitrary complex circuits to the problem of shielding a single simple component.
Our approach is based on modeling the adversary as a powerful observer that inspects the device via a "limited" measurement apparatus. We capture the notion of "limited" measurements using computational complexity classes, and our proofs of security rely on the hardness of certain functions for these classes. Thus, for example, AC^0 lower bounds yield a construction that is resilient to any leakage that can be computed by constant-depth circuits. More generally, we give a generic composition theorem that shows how to build a provably secure devices of arbitrary complexity out of components that satisfy a simulatability condition. Several applications are shown.
In contrast to previous works, we allow the side-channel leakage to depend on the whole state and on all the wires in the device, and to grow unbounded over time.

1 Introduction

The best of cryptographic algorithms are insecure when their implementations inadvertently reveal secrets to an eavesdropping adversary. Even when the software is flawless, practical computational devices leak information via numerous side channels, including electromagnetic radiation (visible and otherwise) [33][24], timing [9], power consumption [23], acoustic emanations [38], and numerous effects at the system architecture levels (e.g., cache attacks [5][29][30]). These leakages are particularly accessible when the computational device is at the hands of an adversary, as is often the case for many modern devices such as smart-cards, TPM chips and (potentially stolen) mobile phones and laptops. Reducing these leakages has proven excruciatingly difficult and costly, and their complete elimination is nowhere in sight.

If computational device leaks abundantly, then why are many side channel attacks hard to carry out, and why do some devices remain unbroken? It is because *useful* measurements can be practically difficult to realize. Physical measurement apparatuses typically produce a "shallow" observation about the measured object, by combining some of its salient physical properties in a simple way. The observation consists of a limited amount of information, obtained as a simple function of physical state of the device; any in-depth analysis happens only as post-processing.

Following Micali and Reyzin [25], we thus think of the measurement apparatus as performing *computation* on the physical state of the device, on behalf of the adversarial observer. While the observer is powerful (e.g., polynomial-time or even unbounded), it is constrained to learning the output of a computationally-bounded *leakage function* f applied to the state of the device. The function f is adaptively chosen by the observer from a class $\mathcal{L}$, which models the practical limitations of the physical experimental setup available to the adversary. For example, $\mathcal{L}$ may consist of all functions computable by circuits of small depth.

To protect against such computationally-bounded leakages, one may try to encode the computation in a way that is too complicated for the class $\mathcal{L}$ to decode. We show that, indeed, for certain classes of leakages, *any* computation can be so encoded: namely, we give a method for transforming arbitrary circuits into new circuits, which are still leaky but whose leakage is useless to the attacker (in the sense of offering no advantage over black-box access to the original circuit's functionality).

Our model and results generalize those of Ishai, Sahai and Wagner [18], who considered leakage of at most t wires (in our terms, this means $\mathcal{L}$ consists of all functions which output verbatim some t of their inputs). In contrast, we consider classes of leakage functions which have simultaneous access to the whole state of the circuit and can be queried repeatedly in an adaptive manner. The leakage functions are constrained in just two ways: to reside in a low complexity class, and to have a bounded output size per invocation (the *aggregate* output over multiple adaptive invocation is unbounded). Note that these constraints are necessary when we allow observing of the whole state: if $f \in \mathcal{L}$ can output the whole state in one shot then there is no hope of security, and if f is computationally powerful enough to predict the future state of the device then the observer can recover the full state at some point in the future by the "precomputation" attack of [12] and [22, Section 5].

1.1 Our Results

After defining the model, we give a number of positive results, of varying generality, on the existence of circuit transformations that protect against computationally-bounded leakage. We also discuss complementary impossibility results justifying some of our requirements.

Leakage resilience from linear secret sharing. Given any linear secret sharing scheme Π and a leakage class $\mathcal{L}$ which cannot decode Π, we show an explicit construction that transforms any circuit C into a circuit $\widehat{C}$ that is resilient against leakages in $\mathcal{L}$.

The gist of the construction is to encode every wire of C into a bundle of wires in $\widehat{C}$ using Π, where each wire carries a single share. Similarly to Ishai et al. [18], we transform each gate in C into a gadget in $\widehat{C}$ which operates on encoded bundles. The gadgets are carefully constructed to use Π internally in a way that looks "essentially random" to leakages in $\mathcal{L}$, and we show that this implies that the whole content of the transformed circuit remains "essentially random" to a leakage in L. Hence, the adversary gets no advantage from his observation of the leakage; formally, this is captured by a simulation-based definition.

Our construction makes an extra requirement: the gadgets require the use of a small leak-free component $\mathcal{O}$, which merely outputs samples from a fixed distribution, namely the encoding of 0 under Π. Thus, following the approach of Micali and Reyzin [25], who proposed reducing the physical security of complex cryptographic constructions to the physical security of simple components that are used in those constructions, we reduce the security of *arbitrary* circuits to the security of one simple component. This simple "opaque" component is minimal in many respects: it has no secrets, no states and no outputs; moreover, it can be computed by a small shallow circuit, or even computed in advance and read from a leak-free sequential-access storage. Furthermore, we show how the leak-free requirement can be relaxed.

Resilience against AC^0 and $\mathsf{ACC}^0[p]$ leakage. As a concrete example, we invoke known circuit lower bounds to obtain an unconditionally secure transformation. For the case where the scheme Π is given by the parity function (i.e., a bit b is shared into random bits whose parity is b), and the leakage class AC^0 (Boolean circuits of constant depth, polynomial size and unbounded fan-in), the lower bound of Hastad [17] implies that functions in L cannot decode Π. As a further example we show that "sum $\mod p$ encoding" can be used to instantiate our construction to result in resilience of $\mathsf{ACC}^0[q]$ leakage, for distinct primes p and q.

Security proof via general composition. We show a general technique for proving security of leakage-resilient circuit transformations. Namely, we capture a strong notion of leakage-resilience for circuits or parts thereof, by saying that they are *reconstructible* if there exist certain efficient simulators for their internal wires that fool the leakage class. We then show a composition result: if all parts of a circuit are reconstructible then so is the whole circuit. This implies security of the transformation. Thus, security of the overall transformation is reduced to reconstructibility of the individual gadgets used. Our specific results using linear secret-sharing schemes follow this route, and other transformations can be built by devising different gate gadgets and merely showing that each is reconstructible by itself.

Leakage-resilience from opaque public-key encryption. We demonstrate the usefulness of the aforementioned general composition, by using it to concisely prove the security of another (very different) circuit transformation that is secure against all polynomial-time measurement. This transformation relies on more complicated leak-free gates, which compute public-key encryption and decryption.

Necessity of leak-free gates. We argue that the use of leak-free components (as done in our constructions) is actually necessary, at least for "natural" constructions whose security is proven by (or implies) reconstructibility. This is done by showing that if such a transformation uses only leak-free components of fixed size (or even components which can be merely *verified* by circuits of small depth), then hard functions have shallow circuits; for certain parameter regimes this is unconditionally false, and for others it implies an unlikely collapse of complexity class hierarchies, e.g., $\mathsf{AC} = \mathsf{P/poly}$.

1.2 Models and Assumptions

Leakage from computational devices is more than an artifact of practical constraints on engineering and manufacturing: it appear to reflect fundamental physical reality. Indeed, the holographic bound conjecture in physics asserts that all information (entropy) in a region of space could be transcribed on its boundary surface, and moreover, the holographic principle conjecture asserts an isomorphism between the observable properties of the region's interior and those of its boundary surface [44][43][7].[4] Consequently, if two states of a computational device are (statistically) indistinguishable under all physical measurements of the *surface* of the device, then their internal states, and thus their future input/output relation, are (statistically) indistinguishable. Put otherwise, perfect useful containment of (usable) secrets is physically impossible.

Despite these harsh realities, we wish to obtain meaningful security functionality, which typically necessitates storing and computing on secrets. Thus, cryptographers are asked to play poker using a deck of transparent cards. What would be a sound way to proceed? Clearly, it is necessary to posit some limits on the adversary's observational powers, otherwise all secrets might be directly observed. Several recent works (for both particular functionalities and general ones), as well as this work, make specific assumptions about the nature of the leakage. We review these assumptions below.

1.2.1 Leak-free components

A natural restriction on the adversary's power is to posit that some parts of the circuit do not leak (i.e., are not provided as inputs to the leakage function).

The model of Micali and Reyzin [25] (and subsequently Dziembowski and Pietrzak [12], Pietrzak [32] and Faust et al. [13]) assumes the presence of leak-free memory. This is captured by the statement that "only computation leaks information" (axiom 1 in [25]), i.e., memory not accessed during a computation step does not affect the observable leakage from that stage.

The "Oblivious RAM" model of Goldreich and Ostrovsky [15,16] reverses the roles: while memory is leaky, the computation is assumed to be on a leak-free secure processor.

[4] Of the many variants of these conjectures, we refer to the spacelike projection theorem of the covariant entropy bound as defined by Bousso [7], and to the strong holographic principle as defined by Smolin [40]. These apply to closed, smooth surfaces in the absence of gravitational and relativistic effects. Of course, in practice most side channels are rather more prosaic; the gap is analogous to that of energy consumption in current VLSI technology vs. physical lower bounds on reversible computation.

In this model, they show a generic transformation that makes random-access machines resilient to polynomial-time leakage (with polylogarithmic blowup in memory size and running time).

Both the leak-free memory and leak-free processor assumption seem most applicable when the adversary resides within the system (e.g., code executing on a chip) and is restricted by the system's communication channels. They seem more difficult to realize when circuits may be physically probed in ways that do not respect the architecture's designated channels, and may even be in the hands of the adversary. Popular storage technologies leak physical information: SRAM and flip-flops have distinct observable current flows according to their state; RAM is frequently refreshed by (potentially leaky) circuitry; hard disks induce a magnetic field that is, in principle, measurable in aggregate; and the ease of global measurement on optical media is readily observable by looking at the surface of a partially-recorded CD-R disc. The leak-free processor of [15,16] is fairly complex: in particular, it contains a pseudorandom function and its key, a number of registers, and assorted logic. Protecting such complex circuits is, in fact, our goal.

Our constructions, too, rely on a leak-free component (whose necessity is discussed in Section 7). This component is simple, small and stateless, and can be used for protecting arbitrary circuits. Notably, one can compose our construction with that of Goldreich and Ostrovsky [15,16], by applying their transformation to protect the large memory, and then applying our transformation to protect the secure processor.

Given the physical realities, assuming any leak-free components generally means assuming that an adversary is simply not able to capture the information that is leaking from the component. If the adversary is able to capture the information, then the assumption is violated, and thus the proof of security no longer applies, even if the adversary can't put the information to good use for specific schemes. We therefore provide a relaxation of this assumption in Section 6.2. The relaxation requires merely that the internal wires of the component be efficiently simulatable in a way that is indistinguishable for the leakage function.

1.2.2 Spatial locality

Several works build security on the assumption that the leakage measurement is *spatially local*. That is, the model assumes that the leakage observed by the adversary is a function of just a part of the device's state (e.g., a few wires, or one component, or partial memory), independently of the rest of the device's state.

For example, Ishai et al. [18] consider the case of an adversary that can read out a small number of wires but gets no information at all about the rest of the wires. For the case of leakage of a single wire ("first-order power analysis"), practical schemes have been proposed and implemented using XOR-based masking; these trivially fail as soon as the spatial locality assumption is even slightly violated, e.g., by an observer that can simultaneously measure an XOR of two wires (such "high-order power analysis" attacks have indeed been demonstrated in practice (e.g., [45]). More generally, functionality-specific masking schemes have proposed against spatially-local leakage for any fixed number of wires (e.g., [36] specifically for the AES cipher).

Alas, global measurements are typically *easier* to conduct than localized measurements that focus on specific wires or memory cells; in many side-channel attacks, the main practical difficulty for the attacker lies precisely in obtaining high spatial resolution and accuracy. Thus, many attacks do use global measurements. For example, several classical attacks use a global power consumption to learn a global property, namely the total Hamming distance of a state transition, from which it covers the cipher's secret keys.

Note that several aforementioned constructions [12][32][13], which are defined in terms of leak-proof memory (i.e., "only computation leaks information") actually remain secure in a more lax but still spatially-local model. As observed in [12, Footnote 2], the circuit's state consists (essentially) of two halves, and the schemes remain secure if the observer can measure both halves simultaneous but independently, i.e., the leakage function is of the form $f(S_L, S_R) = (f_L(S_L), f_R(S_R))$ where S_L and S_R are the two halves of the state. This relaxation of spatial locality still forbids global measurement of non-associative functions.

In contrast to most previous work, we allow the leakage function to see everything, and assume, instead, that it is limited in what it can compute and output. The price we pay for this generalization is in computational assumptions: we must also assume (unless a complexity lowerbound is readily available, as in the case of AC^0) that some encoding scheme is hard for the leakage function to decode.

1.2.3 Other related approaches

Recently, various constructions [2,3,20,27] have been presented that achieve security against adversaries that can learn arbitrary functions of the secret key without relying on leak free components or the spatial locality assumption. All these constructions are stateless and thus must assume that total leakage does not exceed the size of the secret key. In [2] Akavia et al. show that certain lattice-based public-key encryption schemes remain remarkably secure in this model. Naor and Segev [27] show how to achieve CCA1 and even CCA2 security using hash-proof systems. Provably secure signature schemes have been proposed by Alwen et al. [3] and independently by Katz [20].

Dodis et al. [10] study the problem of "cryptography with auxiliary information." In this model the range of the leakage function f is not necessarily bounded. Instead, they assume that given $f(sk)$ it is exponentially hard to compute sk. This is similar, in spirit, to our assumption that functions in $\mathcal{L}$ cannot decode.

Standaert et al. [41] consider a restricted version of the model in [25] by assuming a limited class of leakage functions, such as ones that are currently used in practice to break systems (such as Hamming weight attacks). In this model Petit et al. [31] analyze a block-cipher based construction for a PRNG.

In [42], Standaert et al. work in the random oracle model and assume that the leakage functions are unable to query the random oracle (and are also non-adaptive). They show that standard PRF constructions are leakage-resilient in the random oracle model.

Recent work of Rabin and Vaikuntanathan [34] considers the case of "noisy leakage," assuming that the observer sees a corrupted copy of the state subject to some noise. For the case where the noise independently flips each observed wire value with some probability, [34] shows how arbitrary circuits can be compiled for resilience against noisy leakage. This model can be recast as a special case of ours, and we provide an alternative security proof for the construction of [34] (see Section 6.4).

1.3 Organization of this Paper

Section 2 defines our model for leakage-resilient circuit transformations. Section 3 describes our main construction of circuit transformers from linear secret sharing schemes. Security of this construction is proved in Section 4 (which defines the notion of "reconstructibile" stateless circuits and proves that it holds for our construction) and Section 5 (which derives security for the general stateful case). Section 6 explores a number of special cases and generalizations of our construction and proof techniques. Lastly, Section 7 investigates whether leak-free circuits can be built without large leak-free component.

2 Definitions

We generalize the notion of a private transformation from Ishai, Sahai, and Wagner [18][19]. For readers familiar with the model of [18], we quickly summarize the generalization here (a more detailed description of the model is provided below). First, whereas [18] speak of a "t-private transformation" that is secure against observers who can access at most t wires, we generalize it to an "$\mathcal{L}$-secure transformation" that is secure against observer who is able evaluate any leakage function f in the class $\mathcal{L}$. At each clock cycle, the observer gets to pick a function $f \in \mathcal{L}$ and obtains f computed on the wires of the circuit (similarly to the model of Micali and Reyzin [25]). Further, whereas the transformers of [18] take boolean circuits into circuits that allow random gates in addition to boolean gates, we consider different sets of of allowable gates, and explicitly specify what circuits are being transformed into what circuits.

2.1 Notation

We consider circuits whose wires carry elements of an arbitrary finite field $\mathcal{K}$. Circuits may use randomness gates, and thus their output is not may not be determined solely by the inputs. For a circuit C containing w wires, a *wire assignment to C* is a string in $\mathcal{K}^w$, where each element represents a value on a wire of C. By $\mathcal{W}_C(X)$ we denote a distribution of wire assignments that is induced when a circuit C is being evaluated on an input X (in particular, if C is deterministic, then $\mathcal{W}_C(X)$ has only one element in its support). We use $\mathcal{W}_C(X|Y)$ to denote the same distribution conditioned on the fact that the output of $C(X)$ was Y. For a circuit C let k_{I} be the number of inputs, k_{O} the number of outputs and k_{S} the size of the stateful memory (if any); the size of a circuit is the number of gates in it. For brevity we let $C \in \mathcal{C}$ mean that the function computed by the circuit C is in the function class $\mathcal{C}$.

If D is a distribution, then $y \leftarrow \mathsf{D}$ means a random variable y is drawn from D. (If D is a set with no distribution specified, then by default we assume the uniform distribution.) If D is an algorithm, then $y \leftarrow \mathsf{D}(x)$ denotes the output of D on input x; in particular, if D is randomized, then y is a random variable. $\mathsf{D} \equiv \mathsf{D}'$ means the distributions D and D' are identical. For brevity, we often identify random variables and their distribution.

For $n \in \mathbb{N}$, let $[1, n]$ denote the range of integers $\{1, \ldots, n\}$. Function composition is denoted by $f \circ g : x \mapsto f(g(x))$. If $\mathcal{L}_1$ and $\mathcal{L}_2$ are two sets of functions, then $\mathcal{L}_2 \circ \mathcal{L}_1$ is a set of functions $\{f \circ g \mid f \in \mathcal{L}_2, g \in \mathcal{L}_1\}$. Also, for integer n and function class $\mathcal{L}$, let $(n \times \mathcal{L})$ denote the class of functions of the form $(x_1, \ldots, x_n) \mapsto (f_1(x_1), \ldots, f_n(x_n))$ where $f_i \in \mathcal{L}$ $(i \in [1, n])$.

Vectors, denoted $\vec{v} = (v_1, \ldots, v_n)$, are column vectors.

2.2 Defining Circuit Transformation

In order to understand our definition, it helps to keep the following scenario in mind. Imagine a circuit that has a secret stored within it and uses the secret together with an input to come up with an output; the secret itself may get modified during the computation. For example, the circuit may implement a pseudorandom generator, a stream cipher, or a block cipher, where the keys are secret. The observer gets to interact with the circuit by giving it inputs, observing some physical leakage from the computation, and viewing the outputs. We want to make sure that the ability to observe physical leakage does not help the observer: that is, the observer learns nothing more about the state of the circuit from the leakage than it could learn from just the inputs and outputs. To this end, we show how to convert arbitrary circuits into transformed circuits that satisfy this goal (i.e., leaks no useful information), yet are functionally equivalent.

Circuits. A *circuit* is a directed graph with gates as nodes and wires as edges. Wires carry values, which, for this paper, will be from an (arbitrary) field $\mathcal{K}$; in particular, we may set $\mathcal{K} = \mathsf{GF}(2)$ to speak of a Boolean circuit. Gates a specified (randomized) function of the values on their input wires and send the result along their output wires. We consider the following gates operating on elements of $\mathcal{K}$ (in addition to the input, output, and memory gates): $\oplus, \ominus$, and $\odot$ (which compute, respectively, the sum, difference, and product in $\mathcal{K}$, of their two inputs), the "coin flip" gate \$ (which has no inputs and produces a random independently chosen element of $\mathcal{K}$), and for every $\alpha \in \mathcal{K}$, the constant gate $\mathtt{const}_\alpha$ (which has no inputs and simply outputs α). Fanout is handled by a special `copy` gate that takes as input a single value and outputs two copys. If we use one output of a gate k times, then it is passed through a subcircuit of $k-1$ `copy` gadgets arranged in a tree (the structure of the tree may be chosen arbitrarily). Notice that `copy` gates are just the identity (pass-through wires) and are present mainly for notational convenience.

$\mathsf{SHALLOW}(d, s)$ denotes the set of all deterministic circuits (i.e., ones without \$ gates) that have at most s $\oplus, \ominus$, and $\odot$ gates that are arranged at most d deep (i.e., the longest path in the circuit has at most d such gates on it). Note that `copy` and $\texttt{const}_\alpha$ gates are allowed in the circuit and do not count towards d or s.

A *stateful* circuit additionally contains memory gates, which have a single incoming edge and any number of outgoing edges.[5] Memory gates maintain state: at any clock cycle, a memory gate sends its current state down its outgoing edges and updates it according to the value of its incoming edge. Any cycle in the circuit must contain at least one memory gate.

The state of all memory gates at clock cycle i is denoted by M_i, with M_0 denoting the initial state. Inputs to and outputs from clock cycle i are denoted, respectively, by x_i and y_i. When a circuit is run in state in M_{i-1} on input x_i, the computation will result in a wire assignment $\mathcal{W}_i$; the circuit will output y_i and the memory gates will be in a new state M_i. We will denote this by $(y_i, M_i, \mathcal{W}_i) \Lleftarrow C[M_{i-1}](x_i)$.

Transformer. A circuit transformer TR takes as input a security parameter t, a circuit C, and an initial state M_0 and produces a new circuit $\widehat{C}$ and new initial state $\widehat{M_0}$.[6] Note that the set of allowable gates of $\widehat{C}$ may be different from the set of allowable gates of C (we will explicitly name those sets when constructing concrete transformers). We require the transformer to be *sound*: for all C and M_0, $C[M_0]$ should behave identically to $\widehat{C}[\widehat{M_0}]$. By "behave identically" we mean that for any number of clock cycles q and any set of inputs $x_1, x_2, \ldots, x_q$ (one for each clock cycle) the distribution of the outputs $y_1, y_2, \ldots, y_q$ is the same for C starting at state M_0 and $\widehat{C}$ starting at state $\widehat{M_0}$.

Class of leakage functions $\boldsymbol{\mathcal{L}}$. The attacker (observer) of our transformed circuit will be able to choose a function f in some class of functions $\mathcal{L}$ that we will specify. The function f will take the circuit's wire assignment as input and output a result in some range λ. In order for the observer to be able to specify f, we assume a fixed (but arbitrary) representation of $\mathcal{L}$ and, for brevity, identify functions in $\mathcal{L}$ with their representation.

Security. We want to make sure that the transformed circuit leaks no useful information to an observer. We use the term $(\mathcal{L}, \tau)$-*observer* to denote an observer OBS with physical observations limited to functions in class $\mathcal{L}$ computed on the wires of the circuit and running time (not including the computation by the leakage function itself) limited to τ. To formalize that such an observer learns nothing useful, we the existence of a simulator SIM: anything the observer learns can also be learned by SIM which does not observe any leakage.

If the observer OBS gets to query the circuit q times, each time choosing a fresh function from $\mathcal{L}$, we call it a q-*adaptive* $(\mathcal{L}, \tau)$-observer. The number of observations q, the observer's running time τ, and various other running times and success probabilities are all parameterized by a security parameter t, which is given as input to the transformation TR. For readability, we will omit t from most of our discussion.

[5] Formally, our notion of a stateful circuit is essentially the same as the one in [18].

[6] Throughout this paper, we use the hat notation $\widehat{\square}$ (reminiscent of the proverbial "tinfoil hat") to designate circuit or components that are transformed for leakage-resilience.

Consider the following two experiments that start with some circuit C in state M_0, and allow it to run for q iterations. In both experiments, we assume that OBS and SIM are stateful, i.e., remember their state from one invocation to the next.

```
Experiment $\mathsf{Exp}^{\mathrm{real}}_{\mathsf{TR}}(\mathsf{OBS}, \mathcal{L}, q, C, M_0, t)$
$(\widehat{C}, \widehat{M_0}) \leftarrow \mathsf{TR}(C, M_0)$
$(x_1, f_1) \leftarrow \mathsf{OBS}(\widehat{C})$, with $f_1 \in \mathcal{L}$
For $i = 1$ to $q - 1$
    $(y_i, \widehat{M_i}, \mathcal{W}_i) \Lleftarrow \widehat{C}[\widehat{M_{i-1}}](x_i)$;
    $(x_{i+1}, f_{i+1}) \leftarrow \mathsf{OBS}(y_i, f_i(\mathcal{W}_i))$
$(y_q, M_q, \mathcal{W}_q) \Lleftarrow \widehat{C}[\widehat{M_{q-1}}](x_q)$;
Return output of $\mathsf{OBS}(y_q, f_q(\mathcal{W}_q))$.
```

```
Experiment $\mathsf{Exp}^{\mathrm{sim}}_{\mathsf{TR}}(\mathsf{SIM}, q, C, M_0, t)$
$x_1 \leftarrow \mathsf{SIM}(C)$, with $x_1$ being an input
For $i = 1$ to $q - 1$
    $(y_i, M_i) \leftarrow C[M_{i-1}](x_i)$
    $x_{i+1} \leftarrow \mathsf{SIM}(y_i)$
$(y_q, M_q) \leftarrow C[M_{q-1}](x_q)$
Return output of $\mathsf{SIM}(y_q)$.
```

We will say that the transformed circuit is secure if the outputs of the two experiments are indistinguishable. In fact, for ease of notation, we will consider only the case when the two experiments output 0 or 1 (this is without loss of generality: if the two experiments have more complex outputs, then we can incorporate the distinguisher, which would get those outputs and produce 0 or 1, into OBS and SIM). We are now ready to state our definition precisely.

Definition 1 (Security of Circuit Transformation). *A circuit transformer* TR *is* $(\mathcal{L}, \tau, \tau', q, \epsilon)$*-secure if for every q-adaptive* $(\mathcal{L}, \tau)$*-observer* OBS *there is a simulator* SIM *running in time* τ' *such that for all circuits* C *and initial states* M_0

$$|\Pr[\mathsf{Exp}^{\mathrm{real}}_{\mathsf{TR}}(\mathsf{OBS}, \mathcal{L}, q, C, M_0, t) = 1] - \Pr[\mathsf{Exp}^{\mathrm{sim}}_{\mathsf{TR}}(\mathsf{SIM}, q, C, M_0, t) = 1]| \leq \epsilon,$$

where the probabilities are taken over all the coin tosses involved in the experiments.[7]

To help explain the meaning of the parameters, we note that a stronger result is obtained when $\mathcal{L}$, τ, and q are larger (because it allows for more leakage functions and stronger observers), τ' is as close as possible to τ, and the distinguishing advantage ϵ is as small as possible (because it indicates tighter simulation). The definition is a generalization of the definition of Ishai, Sahai, and Wagner [18, Section B.3] (for the reader familiar with that definition, we note that our security parameter is denoted t rather than σ; our class of leakage functions is arbitrary $\mathcal{L}$ rather than observations of t wires; and the running time τ'' of the distinguisher is not present in our definition because we incorporate the distinguisher into the observer and simulator).

Leakage-indistinguishability. Our proof will, naturally, involve having the simulator produce simulated wire distributions that are indistinguishable from real ones by the observer. The following definition captures what such indistinguishability means.

[7] Recall that TR itself and the above parameters are functions of a security parameter t.

Definition 2 (Leakage-Indistinguishability). *Two distributions X, X' are said to be p-adaptive $(\mathcal{L}, \tau, \epsilon)$-*leakage-indistinguishable*, if for any $(\mathcal{L}, \tau)$ observer OBS making at most p queries to its oracle,*

$$|\Pr[\mathsf{OBS}^{\mathsf{Eval}(X,\cdot)} = 1] - \Pr[\mathsf{OBS}^{\mathsf{Eval}(X',\cdot)} = 1| \leq \epsilon, \tag{1}$$

where $\mathsf{Eval}(X, f)$ can be queried once with a leakage function $f \in \mathcal{L}$ and evaluates to $f(X)$. The probabilities above are taken over the relevant distributions X, X' and the internal coin tosses of OBS.

If $p = 1$, we will omit the words "p-adaptive" in the definition above.

3 Main Result: Circuit Transformation from Linear Secret Sharing Schemes

3.1 Theorem Statement

Our main result states that if there exists a linear encoding scheme for elements of $\mathcal{K}$ (taking a single element to t elements) for which encodings of any two values are is 2-adaptive leakage-indistinguishable, then there exists a secure circuit transformation, where the loss in leakage class is only $\mathsf{SHALLOW}(3, O(t^2))$, and the loss in the time and success probability is linear in circuit size and the number of queries q_{TR}. We now describe it more formally.

Our main construction, using linear secret sharing schemes, uses the following elements.

Gates in the original circuit. The original circuit C is allowed the already defined gates $\oplus, \ominus, \odot, \$, \mathtt{copy}, \mathtt{const}_\alpha$, as well as memory gates. Note that if $\mathcal{K} = \mathsf{GF}(2)$ then $\odot$ is the $\mathtt{AND}$ gate and $\mathtt{const}_1 \oplus a$ is the $\mathtt{NOT}$ gate, so any boolean circuit can be easily transformed into one in C.

Encoding for the wires. Our transformation can be based on any *linear encoding scheme Π* $= (\mathsf{Enc}, \mathsf{Dec})$, which maps a single element of $\mathcal{K}$ to a vector in $\mathcal{K}^t$ and back. It is defined as follows. In the simplest case of $\mathcal{K} = \mathsf{GF}(2)$, an encoding of a bit x is a random string of t bits whose exclusive-or is x. More generally, for security parameter t, a linear encoding scheme Π is defined by a *decoding vector* $\vec{r} = (r_1, \ldots, r_t)$ (viewed as a column vector for the purposes of linear algebra), with each r_i a nonzero element of $\mathcal{K}$, as follows. $\mathsf{Dec} : (y_1, \ldots, y_t) \mapsto \sum_i y_i r_i = \vec{r}^{\mathsf{T}} \vec{y} = \vec{y}^{\mathsf{T}} \vec{r}$ (these operations are over $\mathcal{K}$), and Enc is a (probabilistic) algorithm that, on input x, chooses uniformly at random an element of $\mathsf{Dec}^{-1}(x)$. For $x \in \mathcal{K}$, we let $\mathsf{Enc}(x)$ denote the distribution of encodings of x, use to $\vec{x}$ to denote a particular encoding from this distribution. For elements $x_1, \ldots, x_n \in \mathcal{K}$, denote $\mathsf{Enc}(x_1, \ldots, x_n) = (\mathsf{Enc}(x_1), \ldots, \mathsf{Enc}(x_n))$.

Beside the aforementioned parity encoding, other examples of linear encodings schemes include threshold linear secret sharing schemes (e.g., [37,6]): the reconstruction function of a perfectly secret linear secret sharing scheme with threshold t may be used as Dec, and the sharing procedure as Enc.

Opaque gates. In our scheme, the transformed circuit $\widehat{C}$ is built of the same gate types as the original circuit, with the addition of a new *opaque* gate denoted $\mathcal{O}$. The $\mathcal{O}$ gate has no inputs, and outputs an encoding sampled from the distribution $\mathsf{Enc}(0)$. Crucially, while the wires coming out of this gate can be observed by the leakage function, we assume that the gate itself (just like every other gate) does not leak information.

One may think of $\mathcal{O}$ as implemented in small subcircuits that are completely free of observable leakage (e.g., for the case of $\mathcal{K} = \mathsf{GF}(2)$, such a subcrcuit can be quite simple: generate t random bits $b_0, \ldots, b_{t-1}$ and output $b_i \oplus b_{i+1 \bmod t}$ for $0 \le i \le t-1$).[8].

The requirement of leak-free component is a strong one. As argued in Section 7, it is actually necessary, in a certain sense (which, admittedly, leaves some loopholes). Note, however, that this leak-free component is minimal in many senses:

1. It is a fixed standardized functionality which can be designed and validated once and added to one's VLSI "cell library" — which is far better than having devise separate protection mechanisms for every circuit of interest.
2. It has no secret keys, no inputs and no internal state — it merely samples from a distribution.
3. It can be realized by a leak-free circuit that is small and shallow, as shown above.
4. It can be realized using just *polylog*(t) random bits, as discussed in Section 6.2.
5. Alternatively, because we only need samples from a distribution, we can have the opaque "gate" simply read them one by one from a precomputed list. Thus, it suffices to have leak-proof one-time storage (a consumable "tape roll") instead of leak-proof computation.
6. It suffices that each instance of $\mathcal{O}$ is leakage-resistant in a weaker sense., as discussed in Section 6.2.

The only sense in which our leak-free component is not minimal is the size of its output, which (in the case of our unconditional results invoking circuit lower bounds, in Section 6.1) turns out to be rather large. Improving this parameter is left as an important open problem.

Encoding leakage-indistinguishability. Before we state our main result, we need to define what it means for functions in $\mathcal{L}$ to be unable to distinguish an encoding of x from an encoding of x'.

Definition 3 (Encoding Leakage-Indistinguishability). *An encoding scheme Π is p-adaptive $(\mathcal{L}, \tau, \epsilon)$-*leakage-indistinguishable*, if for any two elements $x, x' \in \mathcal{K}$ the distributions* $\mathsf{Enc}(x)$ *and* $\mathsf{Enc}(x')$ *are p-adaptive $(\mathcal{L}, \tau, \epsilon)$-leakage-indistinguishable.*

If $p = 1$, we will omit the words "p-adaptive" in the definition above.

As a simple example, the aforementioned parity encoding scheme is ∞-adaptive $(\mathcal{L}, \infty, 0)$-leakage-indistinguishable (i.e., information-theoretically leakage-indistinguishable) against

[8] This method of sampling from the distribution of parity-0 strings was brought to our attention by Vinod Vaikutanathan, and used in [34]

the class $\mathcal{L}$ of leakage function that can access at most $t-1$ wires, because the value being encoded is independent of the observed leakage. This, indeed, is the special case given in [18].

We can now state our main theorem:[9]

Theorem 1. *Let $\mathcal{L}_{\mathsf{TR}}$ be some class of leakage functions and let $q_{\mathsf{TR}}, \epsilon_{\Pi}, \tau_{\Pi} \geq 0$. If there exists a linear encoding scheme Π that is 2-adaptive $(\mathcal{L}_{\Pi}, \tau_{\Pi}, \epsilon_{\Pi})$-leakage-indistinguishable, then there exists a circuit transformation TR that is $(\mathcal{L}_{\mathsf{TR}}, \tau_{\mathsf{TR}}, \tau'_{\mathsf{TR}}, q_{\mathsf{TR}}, \epsilon_{\mathsf{TR}})$-secure for*

- *any $\tau_{\mathsf{TR}} \leq \tau_{\Pi} - q_{\mathsf{TR}} O(st^2)$, where s is the number of gates plus the number of input wires in C,*
- *some $\tau'_{\mathsf{TR}} \leq \tau_{\mathsf{TR}} + q_{\mathsf{TR}} O(st^2)$,*
- *some $\epsilon_{\mathsf{TR}} \leq \epsilon_{\Pi}(q_{\mathsf{TR}}+2)(s(t+2)+k)$, where k is the number of memory gates in C,*
- *$\mathcal{L}_{\Pi} = \mathcal{L}_{\mathsf{TR}} \circ \mathsf{SHALLOW}(3, O(t^2))$ (for $\mathcal{K} = \mathsf{GF}(2)$, $\mathcal{L}_{\Pi} = \mathcal{L}_{\mathsf{TR}} \circ \mathsf{SHALLOW}(2, O(t^2))$).*

The rest of this section is dedicated to describing the transformation; the next two sections contain the proof of its security. Special cases of this theorem, as well, as generalization beyond linear encoding schemes, are discussed in Section 6.

3.2 The Transformation for Stateless Circuits

We will first describe our transformation for circuits without any memory gates, which we call *stateless circuits.* We should note that, unlike in [18], inputs and outputs for our stateless circuits do not come already encoded. Encoding the inputs and decoding the outputs is explicitly the job of our stateless transformation.

We extend the transformation to general (i.e., stateful) circuits in Section 3.3.

Given a stateless circuit C, our transformation TR produces the transformed circuit $\widehat{C}$ as follows (see Figure 1 for an example). Each wire w in C is replaced by a *wire bundle* in $\widehat{C}$, consisting of t wires $\vec{w} = (w_1, \ldots, w_t)$, that carry an encoding of w. Each gate is transformed into a *gadget*, built out of gates, which takes encodings and outputs encodings. Each $\oplus$, $\ominus$, $\odot$, $\$$, $\mathtt{copy}$ and $\mathtt{const}_\alpha$ gate is replaced by a $\widehat{\oplus}$, $\widehat{\ominus}$, $\widehat{\odot}$, $\widehat{\$}$, $\widehat{\mathtt{copy}}$ and $\widehat{\mathtt{const}}_\alpha$ gadget, respectively. Crucially, note that the internals of these gadgets may leak. The gadgets themselves are described in Figure 3 and a graphical presentation of the transformation for the $\odot$ gate is shown in Figure 2.

Because our gadgets operate on encoded values, $\widehat{C}$ needs to have a subcircuit at the beginning of that encodes the inputs and another subcircuit at the end that decodes the outputs. However, in our proofs, we want to be able to also reason about transformed circuits without encoding and decoding. Thus, we do not require that every transformed circuit $\widehat{C}$ should have such encoding and decoding. Instead, we introduce artificial input

[9] The theorem's statement, as well as its proof, involves some careful tracking of parameters. This is necessary since our setting is that of shallow circuits and low complexity classes, where reductions must be tight to be meaningful.

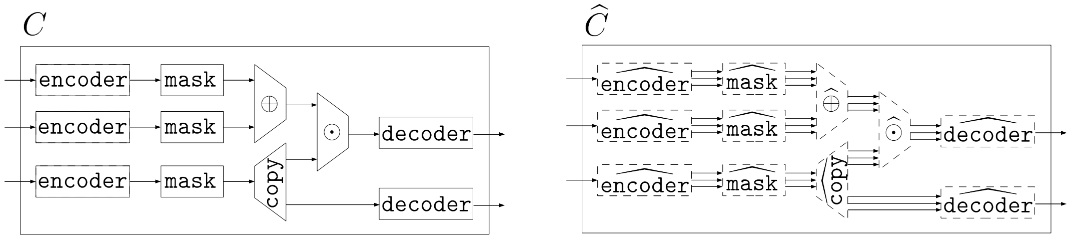
C
encoder
mask
encoder
mask
encoder
mask
$\oplus$
$\odot$
copy
decoder
decoder
$\widehat{C}$
$\widehat{\texttt{encoder}}$
$\widehat{\texttt{mask}}$
$\widehat{\texttt{encoder}}$
$\widehat{\texttt{mask}}$
$\widehat{\texttt{encoder}}$
$\widehat{\texttt{mask}}$
$\widehat{\oplus}$
$\widehat{\odot}$
$\widehat{\texttt{copy}}$
$\widehat{\texttt{decoder}}$
$\widehat{\texttt{decoder}}$

Fig. 1. Example of a circuit C for the function $(a, b, c) \mapsto ((a \odot b) \oplus c, c)$, and the corresponding transformed circuit $\widehat{C}$. Three parallel lines denote encoding (t wires). Dashed borders indicate a gadgets, whose internal wires leak. Note that in C, the special gates `encoder`, `decoder`, `mask` and `copy` are just the identity (pass-through wires) and are present merely for notational convenience.

and output gates that can be part of C for syntactic purposes. If such gates are present (as they would be on any "complete" circuit that one would actually wish to transform), then $\widehat{C}$ will include input encoding and output decoding. If they are not, then $\widehat{C}$ will operate on already encoded inputs and produce encoded outputs.

More precisely, if we wish for $\widehat{C}$ to include input encoding and output decoding, then the circuit C given to TR must have two special gates in sequence on every input wire: an `encoder` gate followed by a `mask` gate, both of which are simply the identity. Also, on every output wire there must be a special `decoder` gate, which is also the identity. These special gates must not appear anywhere else in C. In $\widehat{C}$ each `encoder` gate is replaced by a an $\widehat{\texttt{encoder}}$ gadget which performs encoding (see below), each `decoder` gate is replaced by an $\widehat{\texttt{decoder}}$ gadget that performs decoding (see below), and each `mask` gate is replaced by a $\widehat{\texttt{mask}}$ gadget (that is needed for security and is described in Figure 3).

The $\widehat{\texttt{encoder}}$ gadget takes an input $a \in \mathcal{K}$ and outputs an encoding (i.e., a wire bundle) $\vec{a} \in \mathcal{K}^t$ of a. The encoding can be chosen arbitrarily from the support of $\mathsf{Enc}(a)$ — the choice does not affect security or correctness. This can be implemented using just $\texttt{const}_\alpha$ and $\odot$ gates: $\vec{a} = (r_1^{-1}a, 0, \ldots, 0)$.

The $\widehat{\texttt{decoder}}$ gadget takes an encoding (i.e., a wire bundle) $\vec{a} \in \mathcal{K}^t$ of a and outputs $a \leftarrow \mathsf{Dec}(\vec{a})$. This is computed by a decoding circuit constructed out of $\texttt{const}_\alpha$, $\oplus$, and $\odot$ gates.

Incidentally, observe that because every gadget other than $\widehat{\texttt{encoder}}$ or $\widehat{\texttt{decoder}}$ ends with a masking by an output of $\mathcal{O}$,[10] and wire bundles do not fan-out (instead, they go through the $\widehat{\texttt{copy}}$ gadget), each connecting wire bundle carries encoding of its value that is chosen *uniformly and independently of all the wires in the transformed circuit.* This fact, together with the construction of the gadgets, is what enables the simulation.

[10] One can instead define the basic gadgets as not including this masking with $\mathcal{O}$, and instead place a `mask` gate on every wire. The resulting transformation is similar. However, this doesn't cleanly generalize to the case of transformations not necessarily based on linear encodings — see Section 6.3.

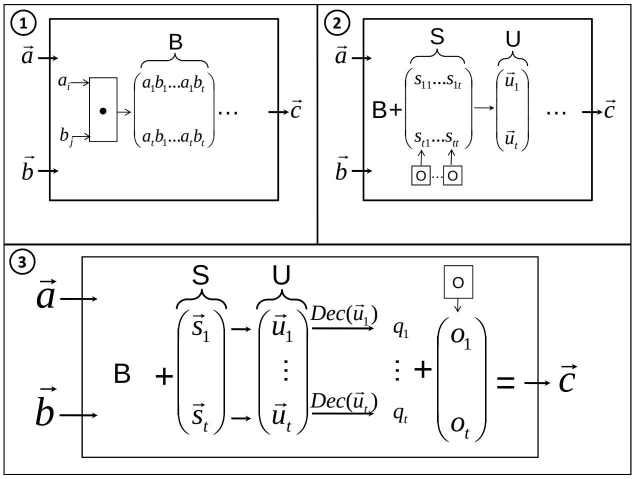

Fig. 2. A step-by-step illustration of the $\widehat{\odot}$ gadget. Steps (1-3) are all part of the transformed gadget $\widehat{\odot}$.

```
Transformation $c \leftarrow a \odot b \;\Rightarrow\; \vec{c} \leftarrow \vec{a}\widehat{\odot}\vec{b}$:
    Compute the $t \times t$ matrix
        $B \leftarrow \vec{a}\vec{b}^{\top} = (a_i b_j)_{1 \le i,j \le t}$ using $t^2$ $\odot$ gates
    Compute the $t \times t$ matrix $S$
        where each column of $S$ is output by $\mathcal{O}$
    $U \leftarrow B + S$ (using $t^2$ $\oplus$ gates)
    Decode each row of $U$ using $t-1$ $\oplus$ gates,
        $t$ $\odot$ gates, and $t$ $\texttt{const}_\alpha$ gates
        to obtain $\vec{q} \leftarrow U\vec{r}$,
        where $\vec{r}$ is the decoding vector
        (it does not matter how this decoding is
        performed as long as there are $O(t)$ wires
        in the decoding subcircuit and each one
        carries some linear combination of the
        wires being decoded, plus possibly a
        constant)
    $\vec{o} \leftarrow \mathcal{O}$
    $\vec{c} \leftarrow \vec{q} + \vec{o}$ (using $t$ $\oplus$ gates)
```

```
Transformation $c \leftarrow \$ \;\Rightarrow\; \vec{c} \leftarrow \widehat{\$}$:
    $c_i \leftarrow \$$    for $i \in [1, t]$
    Output $\vec{c}$
```

```
Transformation $c \leftarrow a \oplus b \;\Rightarrow\; \vec{c} \leftarrow \vec{a}\widehat{\oplus}\vec{b}$
            (or $c \leftarrow a \ominus b \;\Rightarrow\; \vec{c} \leftarrow \vec{a}\widehat{\ominus}\vec{b}$):
    $\vec{q} \leftarrow \vec{a} + \vec{b}$ (or $\vec{q} \leftarrow \vec{a} - \vec{b}$)
        using $t$ $\oplus$ (or $\ominus$) gates
    $\vec{o} \leftarrow \mathcal{O}$
    $\vec{c} \leftarrow \vec{q} + \vec{o}$ (using $t$ $\oplus$ gates)
```

```
Transformation $b \leftarrow \texttt{mask}(a) \;\Rightarrow\; \vec{b} \leftarrow \widehat{\texttt{mask}}(\vec{a})$
    $\vec{o} \leftarrow \mathcal{O}$
    $\vec{b} \leftarrow \vec{a} + \vec{o}$ (using $t$ $\oplus$ gates)
```

```
Transformation $a \leftarrow \texttt{const}_\alpha \;\Rightarrow\; \vec{a} \leftarrow \widehat{\texttt{const}}_\alpha$,
        for any $\alpha \in \mathcal{K}$
    Let $\vec{\alpha}$ be a fixed arbitrary encoding of $\alpha$.
    $\vec{o} \leftarrow \mathcal{O}$
    $\vec{a} \leftarrow \vec{\alpha} + \vec{o}$ (using $t$ $\oplus$ gates)
```

```
Gadget $(\vec{b}, \vec{c}) \leftarrow \widehat{\texttt{copy}}(\vec{a})$
    $\vec{o_1} \leftarrow \mathcal{O}$, $\vec{o_2} \leftarrow \mathcal{O}$
    $\vec{b} \leftarrow \vec{a} + \vec{o_1}$ (using $t$ $\oplus$ gates)
    $\vec{c} \leftarrow \vec{a} + \vec{o_2}$ (using $t$ $\oplus$ gates)
```

Fig. 3. Gadgets used in the stateless circuit transformation TR.

Before we get to the proof of security, however, let us demonstrate that the transformed circuit is functionally the same as C.

Lemma 1 (Soundness of TR). *The stateless circuit transformation* TR *is sound.*

Proof. Since we encode the input, do a gate-by-gate transformation, and then decode the output, it suffices to prove that our gate gadgets work correctly on encoded values:

$\widehat{\oplus}$**:** For $\vec{c} = \vec{a} \oplus \vec{b} \oplus \vec{o}$, with $\vec{o}$ being an encoding of 0, we get by linearity that $\mathsf{Dec}(\vec{c}) = a \oplus b$.
$\widehat{\odot}$: $\mathsf{Dec}(\vec{c}) = \vec{r}^{\mathsf{T}}(\vec{q} + \vec{o}) = \vec{r}^{\mathsf{T}}((B+S)\vec{r} + \vec{o}) = \vec{r}^{\mathsf{T}}((\vec{a}\vec{b}^{\mathsf{T}} + S)\vec{r} + \vec{o}) = (\vec{r}^{\mathsf{T}}\vec{a})(\vec{b}^{\mathsf{T}}\vec{r}) + (\vec{r}^{\mathsf{T}}S)\vec{r} + \vec{r}^{\mathsf{T}}\vec{o} = ab + \vec{0}^{\mathsf{T}}\vec{r} + 0 = ab$
$\widehat{\ominus}, \widehat{\texttt{copy}}, \widehat{\texttt{const}}_{\alpha}, \widehat{\texttt{mask}}, \widehat{\$}$**:** Similar to $\widehat{\oplus}$, by linearity

□

3.3 Full Circuit Transformation

To augment the above stateless circuit transformation to a full circuit transformation, we have to explain how to transform the initial state M_0 and what to do with each memory gate. This is quite simple, given what we have already done. The initial state is replaced by a randomly chosen encoding $\mathsf{Enc}(M_0)$. Each memory gate is replaced by a gadget that consists of t memory gates to store the encoding followed by a $\widehat{\texttt{mask}}$ gadget to guarantee re-randomization of the state.[11] Soundness of this transformation is straightforward, and its security is proved in the next two sections.

4 Reconstructibility of Stateless Circuits

4.1 High-Level Overview

In order to show the security of our transformation, we have to build a simulator. Our simulator will be quite simple, giving random values to internal wires and simulating gadgets to be consistent with those random values (note that this will imply that the simulated outputs of $\mathcal{O}$ used within gadgets will no longer be encodings of 0). The wires that are used to encode the inputs of $\widehat{C}$ (in the $\widehat{\texttt{encoder}}$ gadget) to and decode the outputs (in the $\widehat{\texttt{decoder}}$ gadget) will be simulated honestly, because the simulator knows the inputs and the outputs. The simulator will run the adversary OBS and apply the adversarially-supplied leakage functions the simulated wire values. The difficult part is showing that OBS cannot distinguish true wire values from simulated ones when its access to the wire values is limited by functions available in the class $\mathcal{L}_{\mathsf{TR}}$.

This is done by a hybrid argument, in which encodings of real values are replaced by encodings of random values, one encoding at a time. At each hybrid step, we will prove indistinguishability by a reduction to the security of the encoding scheme. In other words, we will show by reduction that if OBS equipped with functions from $\mathcal{L}_{\mathsf{TR}}$ can distinguish the wire distributions, then some adversary OBS_{Π}, equipped with functions from a slightly

[11] Masking the output of the memory gadget has two reasons: first, we want to allow the total leakage to be much larger than the size of the state, and second, we want to allow adversary to choose leakage functions adaptively.

larger class $\mathcal{L}_{\Pi}$, can distinguish two encodings. Given an encoding, our reduction will need to fake the remaining wires of the circuit and give them as input to the function from $\mathcal{L}_{\mathsf{TR}}$.

Efficiency of such a reduction is particularly important. If OBS specifies a leakage function f for $\widehat{C}$, then OBS_{Π} will specify its own leakage function f_{Π} for the encoding and return its result to OBS. This leakage function f_{Π} has to fake (in a way that will look real to f and OBS) all the wires of $\widehat{C}$ before it can invoke f. At the same time, f_{Π} should not be much more complex than f, because our result is more meaningful when difference between the power of $\mathcal{L}_{\Pi}$ and the power of $\mathcal{L}_{\mathsf{TR}}$ is smaller. The main trick is for OBS_{Π} to hardwire as much as possible into f_{Π}, so that when f_{Π} observes the encoding, it has to do very little work before it can invoke f.

An important observation is that during the hybrid argument, OBS_{Π} and f_{Π} are essentially simulating the circuit in a particularly efficient manner. This very efficient simulator will be called *reconstructor*, and is the main technical tool in our proof.

4.2 Reconstructors

A reconstructor simulates the internal wires of a transformed circuit $\widehat{C}$ given its encoded inputs and outputs in a way that is $\mathcal{L}$-leakage-indistinguishable. We show the existence of reconstructors for stateless circuits without $\widehat{\texttt{encoder}}$ and $\widehat{\texttt{decoder}}$ gadgets.

Reconstructors actually consist of two parts: first, as much as possible is precomputed before the inputs and outputs of $\widehat{C}$ are known. Then, once the inputs and outputs of $\widehat{C}$ are given, all of the remaining (connecting) wires in $\widehat{C}$ are computed. We can think of the precomputed part as sampling from a distribution of functions that map encoded input and output of $\widehat{C}$ into full wire assignments to $\widehat{C}$. The transformed circuit $\widehat{C}$ is randomized, so the simulated wires must be randomized; we let the precomputed part (which, in our reduction, is OBS_{Π}) draw this randomness. This allows us to "hard-wire" the randomness into the on-line part. Thus, crucially, this lets the on-line part of the reconstructor be computed a shallow circuit (as opposed to $\widehat{C}$, which can be very deep).

Intuitively, the existence of a reconstructor shows that functions in $\mathcal{L}$ cannot gain much from looking at the innards of $\widehat{C}$; and since they cannot gain much from looking at encoded inputs and outputs of $\widehat{C}$ either (because these encodings are leakage-indistinguishable), security will follow for any reconstructible circuit.

We will show the existence of reconstructors for the single-gate gadgets, and then give a composition lemma that shows that whole stateless circuits consisting of gadgets connected by wire bundles (i.e., all except $\widehat{\texttt{encoder}}$ and $\widehat{\texttt{decoder}}$) are reconstructible too.

Definition 4 (Reconstructor). *Let $\widehat{C}$ be a (transformed) circuit. We say that a pair of strings (X, Y) is* plausible *for $\widehat{C}$ if $\widehat{C}$ might output Y on input X, i.e., if $\Pr[\widehat{C}(X) = Y] > 0$.*

Consider a distribution $\mathsf{REC}_{\widehat{C}}$ over the functions whose input is a pair of strings, and whose output is an assignment to the wires of $\widehat{C}$. Define $\mathsf{REC}_{\widehat{C}}(X, Y)$ as the distribution obtained by sampling $R_{\widehat{C}} \leftarrow \mathsf{REC}_{\widehat{C}}$ and computing $R_{\widehat{C}}(X, Y)$. Such a distribution is called a

$(\mathcal{L}, \tau, \epsilon)$-reconstructor for $\widehat{C}$ *if for any plausible* (X, Y)*, the following two wire assignment distribution are* $(\mathcal{L}, \tau, \epsilon)$*-leakage-indistinguishable:*

- $\mathcal{W}_{\widehat{C}}(X|Y)$,
- $\mathsf{REC}_{\widehat{C}}(X, Y)$.

If the support of the distribution $\mathsf{REC}_{\widehat{C}}$ *is in some class of functions* $\mathcal{R}$*, we say that* $\widehat{C}$ *is* $(\mathcal{L}, \tau, \epsilon)$*-reconstructible by* $\mathcal{R}$*.*

We shall also use the following property of our gadgets:

Definition 5 (Rerandomizing). *Let* C *be a circuit with* k_{I} *inputs and* k_{O} *outputs, and no* `encoder` *or* `decoder` *gates. Let* $\widehat{C}$ *be the corresponding transformed circuit. We say that* $\widehat{C}$ *is* rerandomizing *if, for any fixed input* $(x_1, x_2, \ldots, x_{k_{\mathrm{I}}})$ *and its encoded input* $X \in \mathsf{Enc}(x_1, x_2, \ldots, x_{k_{\mathrm{I}}})$*, the encoded output* $\widehat{C}(X)$ *is distributed like* $\mathsf{Enc}(C(x_1, x_2, \ldots, x_{k_{\mathrm{I}}}))$*, i.e., independently of the particular encoding* X*.*

Note that the definition of reconstructors speaks only of reconstructing the internal wires when all external wires (i.e., input and output encodings) are *known* and *plausible.* When we invoke this definition, we will see that it implies the stronger notion that the internal wires can be reconstructed even if some external wires are not known (e.g., those corresponding to a circuit's secret inputs) and thus a plausible (X, Y) is not readily available. Intuitively, these proofs will proceed substituting random encodings for the missing external wires (using the rerandomizing property too), and arguing that these cannot be distinguished from the correct (plausible) encodings.

4.3 Single Gadget Reconstructors

Let us show that all single-gate gadgets except $\widehat{\texttt{encoder}}$ and $\widehat{\texttt{decoder}}$ have reconstructors and are rerandomizing. The rerandomizing property follows immediately from the fact that every gadget's output is, as the last step of the gadget, masked by the output of $\mathcal{O}$. Therefore, we focus on the existence of reconstructors.

For the "coin flip" gadget $\widehat{\$}$, this is trivial:

Lemma 2 ($\widehat{\$}$ is reconstructible). *The* $\widehat{\$}$ *gadget is* $(\mathcal{L}, \infty, 0)$*-reconstructible by* $\mathsf{SHALLOW}(0, O(t))$ *for any* $\mathcal{L}$*.*

Proof. The reconstructor $\mathsf{REC}_{\widehat{\$}}$ is the distribution whose only support is the following circuit $R_{\widehat{\$}}$. Given an empty X (i.e., the desired input of $\widehat{\$}$) and a $Y = (\vec{y})$ (i.e., the desired output of $\widehat{\$}$), $R_{\widehat{\$}}(X, Y)$ outputs a wire assignment that simply lets the output of $\widehat{\$}$ carry the only consistent value, namely $\vec{y}$. This is distributed identically to the honest case. □

In the $\widehat{\oplus}$ and $\widehat{\ominus}$ gadgets, the reconstructor will need to "connect" the inputs and outputs:

Lemma 3 **($\widehat{\oplus}$ and $\widehat{\ominus}$ gadgets are reconstructible).** *The $\widehat{\oplus}$ and $\widehat{\ominus}$ gadgets are $(\mathcal{L}, \infty, 0)$-reconstructible by $\mathsf{SHALLOW}(2, O(t))$ for any $\mathcal{L}$.*

Proof. We will do the proof for $\widehat{\oplus}$; the proof for $\widehat{\ominus}$ is similar. The reconstructor $\mathsf{REC}_{\widehat{\oplus}}$ is the distribution whose only support is the following circuit $R_{\widehat{\oplus}}$. On inputs (X, Y) where $X = (\vec{a}, \vec{b})$ (i.e., the desired input of the $\widehat{\oplus}$ gate), and $Y = (\vec{c})$ (i.e., its desired output), $R_{\widehat{\oplus}}$ assigns the wires of $\widehat{\oplus}$ in the only consistent way: $\vec{q} \leftarrow \vec{a} \oplus \vec{b}$ and $\vec{o} \leftarrow \vec{c} \ominus \vec{q}$.

If $\vec{a}, \vec{b}, \vec{c}$ are chosen as in the definition of a reconstructor, then the resulting output of $R_{\widehat{\oplus}}$ is identically distributed to the wire distribution $\mathcal{W}_{\widehat{\oplus}}(X|Y)$, since in both cases $\vec{o}$ takes the only possible consistent value $\vec{o} \leftarrow \vec{c} \ominus \vec{q}$. Notice that $R_{\widehat{\oplus}}$ can be computed by a circuit of depth 2 because on inputs $\vec{a}, \vec{b}, \vec{c}$ it first will compute $\vec{q} \leftarrow \vec{a} \oplus \vec{b}$ and based on that $\vec{o} \leftarrow \vec{c} \ominus \vec{q}$. The $\ominus$ and $\oplus$ gates above operate only on single field elements, so $R_{\widehat{\oplus}}$ requires $O(t)$ size. □

Lemma 4 **($\widehat{\texttt{copy}}, \widehat{\texttt{mask}}$, and $\widehat{\texttt{const}}_\alpha$ are reconstructible).** *The $\widehat{\texttt{copy}}$ gadget, the $\widehat{\texttt{mask}}$ gadget, and, for every $\alpha \in \mathcal{K}$, the $\texttt{const}_\alpha$ gadget are $(\mathcal{L}, \infty, 0)$-reconstructible by $\mathsf{SHALLOW}(1, O(t))$, for any $\mathcal{L}$.*

Proof. We will do the proof for the $\widehat{\texttt{copy}}$ gadget; the other two are similar. The reconstructor $\mathsf{REC}_{\widehat{\texttt{copy}}}$ is the distribution whose only support is a circuit $R_{\widehat{\texttt{copy}}}$ that on inputs (X, Y) where $X = (\vec{a})$ (i.e., the desired input of the $\widehat{\texttt{copy}}$ gate), and $Y = (\vec{b}, \vec{c})$ (i.e., its desired output), assigns the wires of $\widehat{\texttt{copy}}$ in the only consistent way: $\vec{o_b} = \vec{b} \ominus \vec{a}$ and $\vec{o_c} = \vec{c} \ominus \vec{a}$.

If $\vec{a}, \vec{b}, \vec{c}$ are chosen as in the definition of a reconstructor, then the resulting output of $R_{\widehat{\texttt{copy}}}$ is identically distributed to the wire distribution $\mathcal{W}_{\widehat{\texttt{copy}}}(X|Y)$, since in both cases $\vec{o_b}$ and $\vec{o_c}$ take the only possible consistent value $\vec{o_b} \leftarrow \vec{b} \ominus \vec{a}$ and $\vec{o_c} \leftarrow \vec{c} \ominus \vec{a}$. Notice that $R_{\widehat{\texttt{copy}}}$ can be computed by a circuit of depth 1 because on inputs $\vec{a}, \vec{b}, \vec{c}$ it needs only to compute $\vec{o_b}, \vec{o_c}$, both requiring a $\ominus$ operation. The size of $\mathsf{REC}_{\widehat{\texttt{copy}}}$ is $O(t)$ for computing the $2t$ $\ominus$ operations. □

Before we move on to the most interesting case, which is the $\widehat{\odot}$ gadget, we give technical lemma which will let us relate two leakage-indistinguishability statements using a shallow wire simulator f_S.

Lemma 5. *Let $\mathcal{W}_0, \mathcal{W}_0'$ be distributions over $\mathcal{K}^k$ for some $k > 0$.*[12] *Let F_S be a distribution over k-input functions in some class $\mathcal{L}'$. Define the following distributions:*

$$\mathcal{W}_1 \equiv f_S(\mathcal{W}_0) \quad \text{where } f_S \leftarrow \mathsf{F}_S \tag{2}$$

$$\mathcal{W}_1' \equiv f_S(\mathcal{W}_0') \quad \text{where } f_S \leftarrow \mathsf{F}_S \ . \tag{3}$$

Let $\mathcal{L}_1$ be a class of leakage functions and let $\epsilon_0 > 0$, $\tau_0 > 0$. If $\mathcal{W}_0$ and $\mathcal{W}_0'$ are $(\mathcal{L}_0, \tau_0, \epsilon_0)$-leakage-indistinguishable, then $\mathcal{W}_1$ and $\mathcal{W}_1'$ are $(\mathcal{L}_1, \tau_1, \epsilon_1)$-leakage-indistinguishable. Here, $\mathcal{L}_0 = \mathcal{L}_1 \circ \mathcal{L}'$, $\epsilon_0 = \epsilon_1$, and $\tau_0 - \tau_1$ is the time needed to sample from F_S.

[12] In our case, these will be wire assignments to a circuit with k wires. Notice that this can also just be a single encoding.

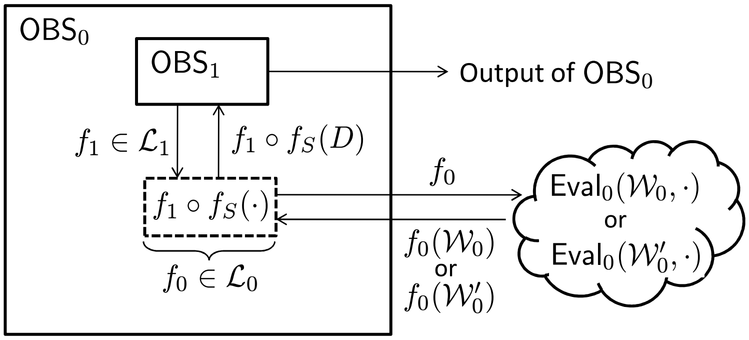

Fig. 4. Outline of the reduction in Lemma 5

Proof. We show by contradiction that for all observers OBS_1 running in time at most τ_1

$$|\Pr[\mathsf{OBS}_1^{\mathsf{Eval}_1(\mathcal{W}_1,\cdot)} = 1] - \Pr[\mathsf{OBS}_1^{\mathsf{Eval}_1(\mathcal{W}'_1,\cdot)} = 1| \leq \epsilon_1, \tag{4}$$

where Eval_1 can be queried once by OBS_1 with a leakage function $f_1 \in \mathcal{L}_1$.

Suppose for contradiction that (4) is violated for some $(\mathcal{L}_1, \tau_1)$-observer OBS_1, then we will construct an $(\mathcal{L}_0, \tau_0)$-observer OBS_0 that breaks the leakage-indistinguishability of the distributions $\mathcal{W}_0$ and $\mathcal{W}'_0$. The observer OBS_0 will invoke OBS_1 as a subroutine, answering OBS_1's leakage query and eventually outputting whatever OBS_1 outputs (see Figure 4). To answer the leakage query $f_1 \in \mathcal{L}_1$ of OBS_1, the observer OBS_0 will use its own oracle Eval_0. The difficulty is that Eval_0 evaluates a leakage function $f_0 \in \mathcal{L}_0$ on a sample either from $\mathcal{W}_0$ or $\mathcal{W}'_0$, whereas OBS_1 produces a query f_1 to be evaluated on a (possibly much larger) wire assignment sampled from $\mathcal{W}_1$ or $\mathcal{W}'_1$.

We address this by using a function f_S, drawn from the distribution F_S, that takes as input a single "challenge" that is either sampled from $\mathcal{W}_0$ or $\mathcal{W}'_0$ and outputs a full wire assignment from either $\mathcal{W}_1$ or $\mathcal{W}'_1$, respectively. To recap, OBS_0 lets OBS_1 choose $f_1 \in \mathcal{L}_1$, and draws a function f_S from F_S. It then queries Eval_0 on $f_0 = f_1 \circ f_S$ and forwards the answer back to OBS_1. Finally, if OBS_1 returns a bit b, then OBS_0 outputs b as its own guess.

To analyze the distinguishing advantage of OBS_0, consider the following two cases, where $x \leftarrow \mathcal{K}$:

$$\Pr[\mathsf{OBS}_0^{\mathsf{Eval}_0(\mathcal{W}_0,\cdot)} = 1] \ = \ \Pr[\mathsf{OBS}_1^{\mathsf{Eval}_1(f_S(\mathcal{W}_0),\cdot)} = 1] \overset{(2)}{=} \Pr[\mathsf{OBS}_1^{\mathsf{Eval}_1(\mathcal{W}_1,\cdot)} = 1]$$
$$\Pr[\mathsf{OBS}_0^{\mathsf{Eval}_0(\mathcal{W}'_0,\cdot)} = 1] \ = \ Pr[\mathsf{OBS}_1^{\mathsf{Eval}_1(f_S(\mathcal{W}'_0),\cdot)} = 1] \overset{(3)}{=} \Pr[\mathsf{OBS}_1^{\mathsf{Eval}_1(\mathcal{W}'_1,\cdot)} = 1]$$

By taking the difference, we see that if (4) then

$$|\Pr[\mathsf{OBS}_0^{\mathsf{Eval}_0(\mathcal{W}_0,\cdot)} = 1] - \Pr[\mathsf{OBS}_0^{\mathsf{Eval}_0(\mathcal{W}'_0,\cdot)} = 1| \leq \epsilon_1.$$

Thus, we get that $\epsilon_0 = \epsilon_1$. Observe also that $f_0 \in \mathcal{L}_0$ (i.e., the reduction doesn't lose much in the leakage function's power): since $f_S \in \mathcal{L}'$ indeed we have that $f_0 = f_1 \circ f_S \in \mathcal{L}_0 = \mathcal{L}_1 \circ \mathcal{L}'$. Finally, note that the only extra time OBS_0 spends (i.e., $\tau_0 - \tau_1$) is the time required to sample from the distribution F_S. □

To show reconstructibility of the $\widehat{\odot}$ gadget, we first consider a reduced variant, denoted $\widehat{\circledast}$. It is the same as $\widehat{\odot}$ but directly outputs $\vec{q}$, i.e. without adding the output of $\mathcal{O}$. We will show that its wire assignment distribution can be replaced by a "fake" one in a leakage-indistinguishable way:

Lemma 6 (Randomization of $\widehat{\circledast}$). *Let $\mathcal{L}_{\widehat{\circledast}}$ be a class of leakage functions and let $\epsilon > 0, \tau > 0$. If the encoding scheme Π is $(\mathcal{L}_\Pi, \tau, \epsilon)$-leakage-indistinguishable, then for any valid encodings $X = (\vec{a}, \vec{b})$ the following wire assignment distributions are $(\mathcal{L}_{\widehat{\circledast}}, \tau - O(t^2), t\epsilon)$-leakage-indistinguishable:*

- $\mathcal{W}_{\widehat{\circledast}}(X)$
- $\mathcal{W}_{\widehat{\circledast}}^{\mathtt{rand}}(X)$*: as* $\mathcal{W}_{\widehat{\circledast}}(X)$ *except that S is drawn independently-and-uniformly from $\mathcal{K}^{t \times t}$*

Here, $\mathcal{L}_\Pi = \mathcal{L}_{\widehat{\circledast}} \circ \mathsf{SHALLOW}(2, O(t^2))$, and in the special case of $\mathcal{K} = \mathsf{GF}(2)$, $\mathcal{L}_\Pi = \mathcal{L}_{\mathsf{REC}_{\widehat{\odot}}} \circ \mathsf{SHALLOW}(2, O(t^2))$).

Proof. We prove this statement by a hybrid argument. We define $t+1$ wire assignment distributions $\mathcal{W}_{\widehat{\circledast}}^{\ell}(X)$ ($\ell \in [0, t]$) as follows:

- $\mathcal{W}_{\widehat{\circledast}}^{\ell}(X)$: The distribution is as $\mathcal{W}_{\widehat{\circledast}}(X)$, except that for the first ℓ columns of S the elements are drawn uniformly-and-independently from $\mathcal{K}$ instead of using $\mathcal{O}$.

Note that the 0th and tth hybrid distributions are the same as the distributions in the claim. We will show that for all $\ell \in [1, t]$ and all X, $\mathcal{W}_{\widehat{\circledast}}^{\ell-1}(X)$ and $\mathcal{W}_{\widehat{\circledast}}^{\ell}(X)$ are $(\mathcal{L}_{\widehat{\circledast}}, \tau - O(t^2), \epsilon)$-leakage-indistinguishable, which will conclude the proof of the lemma.

In Claim 4.3 we show for any $\ell \in [1, t]$ and any X the existence of a distribution F_S^{ℓ} of functions in $\mathsf{SHALLOW}(2, O(t^2))$ samplable in time $O(t^2)$ that take as input a single encoding and map it either to $\mathcal{W}_{\widehat{\circledast}}^{\ell-1}(X)$ or $\mathcal{W}_{\widehat{\circledast}}^{\ell}(X)$, depending on whether the given encoding was an encoding of 0 or of a random value. By applying Lemma 5 to Claim 4.3 (setting $\mathcal{W}_0 = \mathsf{Enc}(0), \mathcal{W}_0' = \mathsf{Enc}(z)$ for a random $z \in \mathcal{K}$) we get that $\mathcal{W}_{\widehat{\circledast}}^{\ell-1}(X)$ and $\mathcal{W}_{\widehat{\circledast}}^{\ell}(X)$ are $(\mathcal{L}_{\widehat{\circledast}}, \tau - O(t^2), \epsilon)$-leakage-indistinguishable, where $\mathcal{L}_\Pi = \mathcal{L}_{\widehat{\circledast}} \circ \mathsf{SHALLOW}(2, O(t^2))$. □

The following technical claim proves the existence of the distribution F_S^{ℓ} used above in Lemma 6 and may be skipped by the reader.

Claim. For any $\ell \in [1, t]$ and any encoding $X = (\vec{a}, \vec{b})$, there exists a distribution F_S^{ℓ} over functions in $\mathsf{SHALLOW}(2, O(t^2))$ that take as input a single encoding and output a wire assignment for $\widehat{\circledast}$, such that for $f_S \leftarrow \mathsf{F}_S^{\ell}$ and $x \leftarrow \mathcal{K}$:

$$\mathcal{W}_{\widehat{\circledast}}^{\ell-1}(X) \equiv f_S(\mathsf{Enc}(0)), \tag{5}$$

$$\mathcal{W}_{\widehat{\circledast}}^{\ell}(X) \equiv f_S(\mathsf{Enc}(x)). \tag{6}$$

Proof. f_S on input $\vec{e}$ shall output a full wire assignment of $\widehat{\circledast}$, with $\vec{e}$ embedded into the ℓth column of S, and with the correct distribution on the remaining wire values. This guarantees that if the target encoding $\vec{e}$ is drawn uniformly-and-independently from $\mathsf{Enc}(0)$ then $f_S(\vec{e})$ is distributed identically to the hybrid wire distribution $\mathcal{W}^{\ell-1}_{\widehat{\circledast}}(X)$. On the other hand, if $\vec{e}$ is drawn uniformly-and-independently from $\mathsf{Enc}(x)$, with $x \leftarrow \mathcal{K}$, then $f_S(\vec{e})$ is distributed identically to $\mathcal{W}^{\ell}_{\widehat{\circledast}}(X)$.

The difficulty is that f_S must have small (constant) depth, but needs to output a wire assignment for the deep circuit $\widehat{\circledast}$. We solve this problem by hard-wiring most of the resulting wire assignment directly into f_S. The only parts of the wire assignment that cannot be hard-wired are those that depend on the input $\vec{e}$, but fortunately they can be easily computed (indeed, this was exactly the goal in designing the $\widehat{\circledast}$ gadget).

Concretely, the distribution F^{ℓ}_{S} is defined by drawing f_S as follows:

1. From given $X = (\vec{a}, \vec{b})$ compute consistently the matrix $B = (a_i b_j)_{i,j \in [1,t]}$ and hard-wire $\vec{a}, \vec{b}, B$ into f_S.
   Hard-wired into f_S***:*** $\vec{a}, \vec{b}$ and $B = (a_i b_j)_{i,j \in [1,t]}$
2. Most columns of S are hard-wired into f_S: left of the ℓth column they are drawn at random, and right of the ℓth column they are drawn from $\mathsf{Enc}(0)$. The ℓth column is filled with the challenge encoding $\vec{e}$.
   Hard-wired into f_S***:*** For $i \in [1, \ell-1]$ $\vec{S_i} \leftarrow \mathcal{K}^t$ and for $i \in [\ell+1, t]$ $\vec{S_i} \leftarrow \mathsf{Enc}(0)$
   Computed by f_S ***on input*** $\vec{e}$***:*** $\vec{S_\ell} = \vec{e}$
3. Using B and S hard-wire all elements of $U = B + S$ into f_S except for the ℓth column. For the ℓth column, f_S computes on input $\vec{e}$, for each $i \in [1,t]$, the value $U_{i,\ell} \leftarrow B_{i,\ell} + e_i$.
   Hard-wired into f_S***:*** For $i \in [1,t], j \in [1,t]_\ell$: $U_{i,j} = B_{i,j} + S_{i,j}$
   Computed by f_S ***on input*** $\vec{e}$***:*** For $i \in [1,t]$: $U_{i,\ell} = B_{i,\ell} + e_i$
4. Consider, for $i \in [1,t]$, the decoding subcircuit in $\widehat{\circledast}$ that computes q_i with values from the row $\vec{U_i}$. As defined in Figure 3, each wire in this subcircuit carries some linear combination of $\{U_{i,j}\}_j$, plus possibly a constant. If this linear combination does not depend on $U_{i,\ell}$ (i.e., the input to f_S), then pre-compute this wire and hard-wire the result into f_S. On the other hand, if it does depend on $U_{i,\ell} = B_{i,\ell} + e_i$, then pre-compute the partial linear combination except the term that depends on e_i and hard-wire the result into the description of f_S. On input $\vec{e}$, f_S computes the missing outputs by $\oplus$-ing the partial linear combination with the missing term (which is e_i times a constant).
   Hard-wired into f_S***:*** Values for wires that do not depend on $U_{i,\ell}$, and partial linear combinations for wires that depend on $U_{i,\ell}$.
   Computed by f_S ***on input*** $\vec{e}$***:*** For wires that depend on $U_{i,\ell}$ compute the output of f_S by $\oplus$-ing the precomputed partial linear combination with e_i times the appropriate constant.

Let us first consider the outputs of f_S that are independent of $\vec{e}$. In $\mathcal{W}^{\ell-1}_{\widehat{\circledast}}(X)$ and $\mathcal{W}^{\ell}_{\widehat{\circledast}}(X)$ the first $\ell - 1$ columns in S are independently-and-uniformly drawn from $\mathcal{K}^t$, whereas the last $t - \ell - 1$ columns are sampled from $\mathsf{Enc}(0)$. The other hard-wired outputs that do not depend on $\vec{e}$, are computed honestly from X and S, thus with respect to only these values, $\mathcal{W}^{\ell-1}_{\widehat{\circledast}}(X)$, $\mathcal{W}^{\ell}_{\widehat{\circledast}}(X)$ and the outputs of f_S are identically distributed. If on the other hand an output of f_S depends on $\vec{e}$ we distinguish two cases:

1. $\vec{e} \leftarrow \mathsf{Enc}(0)$: This means the ℓth column of S is assigned an encoding drawn from $\mathsf{Enc}(0)$. Together with the observation that all remaining wires are computed honestly using S and B, we get that $f_S(\mathsf{Enc}(0))$ and $\mathcal{W}^{\ell-1}_{\widehat{\circledast}}(X)$ are distributed identically.
2. $\vec{e} \leftarrow \mathsf{Enc}(x)$: Here, the ℓth column of S is assigned an encoding drawn from $\mathsf{Enc}(x)$. With the same observation as above we get that $f_S(\mathsf{Enc}(x))$ and $\mathcal{W}^{\ell}_{\widehat{\circledast}}(X)$ are distributed identically.

It is clear that the circuits from F^{ℓ}_S can be sampled in time $O(t^2)$. It remains to show that they are indeed shallow. The input to f_S is used to adjust the ℓth column of U, which requires a circuit of depth 1 and size t. Additionally, adjusting the values in the subcircuits for the computation of q_i requires computation of depth 2 (for the computation of e_i times a constant and $\oplus$-ing it) and $O(t)$ size. Overall, we get circuits of size $O(t^2)$ and depth 2. In the case of $\mathsf{GF}(2)$, there is no need to multiply e_i by a constant, so depth is only 1. □

Now we can prove the existence of a shallow reconstructor circuit for the $\widehat{\odot}$ gadget that is leakage-indistinguishable from the real $\widehat{\odot}$ gadget, even though the real $\widehat{\odot}$ gadget is deep.

Lemma 7 ($\widehat{\odot}$ is reconstructible). *Let $\mathcal{L}_{\mathsf{REC}_{\widehat{\odot}}}$ be a class of leakage functions and let $\tau > 0, \epsilon > 0$. If Π is $(\mathcal{L}_{\Pi}, \tau, \epsilon)$-leakage-indistinguishable, then the $\widehat{\odot}$ gadget is $(\mathcal{L}_{\mathsf{REC}_{\widehat{\odot}}}, \tau - O(t^2), t\epsilon)$-reconstructible by $\mathsf{SHALLOW}(2, O(t^2))$, where $\mathcal{L}_{\Pi} = \mathcal{L}_{\mathsf{REC}_{\widehat{\odot}}} \circ \mathsf{SHALLOW}(3, O(t^2))$ (and if $\mathcal{K} = \mathsf{GF}(2)$, then $\mathcal{L}_{\Pi} = \mathcal{L}_{\mathsf{REC}_{\widehat{\odot}}} \circ \mathsf{SHALLOW}(2, O(t^2))$).*

Proof. The reconstructor $\mathsf{REC}_{\widehat{\odot}}$ is a distribution over circuits $\mathcal{R}_{\widehat{\odot}}$ with inputs (X, Y), where $X = (\vec{a}, \vec{b})$ (i.e., the desired input of the $\widehat{\odot}$ gate), and $Y = (\vec{c})$ (i.e., its desired output). Let $R_{\widehat{\odot}} \leftarrow \mathsf{REC}_{\widehat{\odot}}$, then we define the distribution $\mathsf{REC}_{\widehat{\odot}}$ as follows:

1. Sample independently-and-uniformly from $\mathcal{K}^{t \times t}$ the matrix U and compute the values on the wires in the subsequent subcircuits for the computation of $\vec{q}$ (including $\vec{q}$). All the values are hard-wired as one of $R_{\widehat{\odot}}$'s outputs.
2. On input X, $R_{\widehat{\odot}}$ computes the matrix $B \leftarrow (a_i \odot b_j)_{i,j}, i, j \in [1, t]$. It outputs the result as part of the wire assignment.
3. $R_{\widehat{\odot}}$ computes online $S \leftarrow B \ominus U$ and $\vec{o} \leftarrow \vec{c} \ominus \vec{q}$ (i.e. once using B that depends on input X and once using the input $Y = \vec{c}$).

Circuits sampled from $\mathsf{REC}_{\widehat{\odot}}$ have size $O(t^2)$ (because they need to compute matrices B and S) and depth 2, because S is computed from B, that in turn has been computed from the inputs.

We now show that if X, Y are chosen as in the definition of reconstructors, the wire distribution $\mathsf{REC}_{\widehat{\odot}}(X, Y)$ is $(\mathcal{L}_{\mathsf{REC}_{\widehat{\odot}}}, \tau, t\epsilon)$-leakage-indistinguishable from the wire distribution $\mathcal{W}_{\widehat{\odot}}(X|Y)$. We define the distribution $\mathcal{W}^{\mathtt{rand}}_{\widehat{\odot}}(X|Y)$ based on $\mathcal{W}^{\mathtt{rand}}_{\widehat{\circledast}}(X)$ from Lemma 6 with $\vec{o} \leftarrow Y - \vec{q}$ connecting the output of $\mathcal{W}^{\mathtt{rand}}_{\widehat{\circledast}}(X)$ and Y. In both $\mathsf{REC}_{\widehat{\odot}}(X,Y)$ and $\mathcal{W}^{\mathtt{rand}}_{\widehat{\odot}}(X|Y)$ the matrix U is sampled uniformly-and-independently from $\mathcal{K}^{t\times t}$, since in the latter $U \leftarrow S \oplus M$, where S is sampled uniformly-and-independently from $\mathcal{K}^{t\times t}$. Further, in both cases we have $\vec{o} \leftarrow \vec{c} - \vec{q}$, where $\vec{q}$ is computed honestly and consistently from U. Therefore, the distributions are identical: $\mathsf{REC}_{\widehat{\odot}}(X,Y) \equiv \mathcal{W}^{\mathtt{rand}}_{\widehat{\odot}} X|Y$.

Note that $\mathcal{W}_{\widehat{\odot}}(X|Y)$ can be obtained from $\mathcal{W}_{\widehat{\circledast}}(X)$ by computing $\vec{o} \leftarrow Y - \vec{q}$; similarly, $\mathcal{W}^{\mathtt{rand}}_{\widehat{\odot}}(X|Y)$ can be obtained from $\mathcal{W}^{\mathtt{rand}}_{\widehat{\circledast}}(X)$ by the same computation. And Lemma 6 tells us that $\mathcal{W}_{\widehat{\circledast}}(X)$ and $\mathcal{W}^{\mathtt{rand}}_{\widehat{\circledast}}(X)$ are $(\mathcal{L}_{\mathsf{REC}_{\widehat{\circledast}}}, \tau - O(t^2), t\epsilon)$-leakage-indistinguishable. Therefore, letting f_S be the circuit that performs such a computation (note that it is in $\mathsf{SHALLOW}(1, O(t))$ and F_S be the singleton distribution that has only f_S in it, we can apply Lemma 5 and obtain that $\mathcal{W}^{\mathtt{rand}}_{\widehat{\odot}}(X|Y) \equiv \mathsf{REC}_{\widehat{\odot}}(X,Y)$ and $\mathcal{W}_{\widehat{\odot}}(X|Y)$ are $(\mathcal{L}_{\mathsf{REC}_{\widehat{\odot}}}, \tau - O(t^2), t\epsilon)$-leakage-indistinguishable. □

4.4 Multi-Gadget Circuit Reconstructors

We now proceed to prove the central lemma showing how gadget reconstructors compose together to yield a reconstructor for the whole circuit.

Lemma 8 (Multi-Gadget Circuit Reconstructor). *Let $\mathcal{L}_{\widehat{C}}$ be some set of leakage functions and $\epsilon_\Pi > 0, \tau_\Pi > 0, t > 0$. Let Π be $(\mathcal{L}_\Pi, \tau_\Pi, \epsilon_\Pi)$-leakage-indistinguishable. Let C be a stateless circuit of size s, without* `encoder` *or* `decoder` *gates with k_I inputs and k_O outputs. Then the transformed circuit $\widehat{C}$ is rerandomizing and $(\mathcal{L}_{\widehat{C}}, \tau_{\widehat{C}}, \epsilon_{\widehat{C}})$-reconstructible by $\mathsf{SHALLOW}(2, (k_\mathrm{I} + k_\mathrm{O})O(t^2))$. Here, we have $\epsilon_{\widehat{C}} = \epsilon_\Pi s(t+2)$, $\tau_{\widehat{C}} = \tau_\Pi - O(st^2)$, and $\mathcal{L}_\Pi = \mathcal{L}_{\widehat{C}} \circ \mathsf{SHALLOW}(3, O(t^2))$ (for $\mathcal{K} = \mathsf{GF}(2)$, $\mathcal{L}_\Pi = \mathcal{L}_{\widehat{C}} \circ \mathsf{SHALLOW}(2, O(t^2))$).*

Proof. Let $\widehat{C}$ be the transformed circuit, with inputs denoted $X = (\vec{x}_1, \dots, \vec{x}_{k_\mathrm{I}})$ and outputs denoted $Y = (\vec{y}_1, \dots, \vec{y}_{k_\mathrm{O}})$. Let *first gadgets* denote the set of topologically-first gadgets in $\widehat{C}$, and let *last gadgets* denote the set of topologically-last gadgets in $\widehat{C}$. The wires that go between gadgets (i.e., not directly connected to X or Y, and not part of the innards of some gadget) are called *connecting wires*.

The fact that $\widehat{C}$ is rerandomizing follows immediately from the fact that the last gadget are rerandomizing, and the randomness used in each gadget is independent.

The reconstructor $\mathsf{REC}_{\widehat{C}}$ is a distribution over circuits $\mathcal{R}_{\widehat{C}}$ with inputs (X,Y). We define $\mathsf{REC}_{\widehat{C}}$, with $R_{\widehat{C}} \leftarrow \mathsf{REC}_{\widehat{C}}$, for input (X,Y) that is plausible for $\widehat{C}$, as follows:

1. For each g gadget in $\widehat{C}$, sample $R_{\widehat{g}} \leftarrow \mathsf{REC}_{\widehat{g}}$.
2. For each connecting wire, sample a random encodings, i.e., $\vec{v} \leftarrow \mathsf{Enc}(v)$ with $v \leftarrow \mathcal{K}$.
3. For each gadget g in $\widehat{C}$ except for the first gadgets and last gadgets, pre-compute $R_{\widehat{g}}(U,V)$ and hard-wire the result into $R_{\widehat{C}}$. Here, U (resp., V) are the encodings assigned above to the wire bundles that are the inputs (resp., outputs) of $\widehat{g}$.

   Hard-wired into** $R_{\widehat{C}}$**: For each gadget $\widehat{g}$ in $\widehat{C}$ that is not a first gadget or a last gadget, the output of running $R_{\widehat{g}}(U,V)$ is hard-wired into the description of $R_{\widehat{C}}$.

4. On input (X, Y) the reconstructor $R_{\widehat{C}}$ computes for all of the first gadgets and last gadgets. For the first gadgets, the input wire bundles are given in X and the outputs have been hard-wired above. Similarly, for the last gadgets, the inputs have been hard-wired and the outputs are given in Y.

   Hard-wired into $R_{\widehat{C}}$: Hard-wire the description of $R_{\widehat{g}}$ for each gate g in the first gadgets and last gadgets, and the values of the connecting wires that touch it.

   Computed by $R_{\widehat{C}}$ on input (X, Y)***:*** For each gadget g in the first gadgets and last gadgets, compute $R_{\widehat{g}}(\cdot, \cdot)$ and output the result.

We now analyze the size and depth of the reconstructor $\mathsf{REC}_{\widehat{C}}$. For a circuit C with k_{I} inputs and k_{O} outputs, $R_{\widehat{C}} \leftarrow \mathsf{REC}_{\widehat{C}}$ on inputs (X, Y) only needs to compute $k_{\mathrm{I}} + k_{\mathrm{O}}$ reconstructor circuits (for the first gadgets and last gadgets). This requires size at most $(k_{\mathrm{I}} + k_{\mathrm{O}})$ times the maximum size of a single-gadget reconstructor, and same depth as the deepest single-gadget reconstructor. In our case $\widehat{\odot}$ gate is the largest (of size $O(t^2)$) and (of depth 2), which gives the claimed size and depth.

There remains to show that for any plausible (X, Y), and $R_{\widehat{C}} \leftarrow \mathsf{REC}_{\widehat{C}}$, $R_{\widehat{C}}(X, Y)$ is $(\mathcal{L}_{\widehat{C}}, \tau_{\widehat{C}}, \epsilon_{\widehat{C}})$-leakage-indistinguishable from $\mathcal{W}_{\widehat{C}}(X|Y)$. The proof is by a hybrid argument, outlined as follows. First, we replace all gadgets in $\widehat{C}$ by their corresponding reconstructors. Then, we replace all connecting wires with random encodings, keeping the innards of gadgets consistent with these random encodings.

We first prove that we can replace each gadget in $\widehat{C}$ with an appropriate gadget reconstructor keeping the connecting wires consistent. We will use the following notation. Let $\{\widehat{g}_i\}$ for $i \in [1, s]$ denote the gadgets in $\widehat{C}$. Drawing a wire assignment from the distribution $\mathcal{W}_{\widehat{C}}(X|Y)$ of the real circuit, we denote its elements as follows. For the ith gadget $\widehat{g}_i$ in $\widehat{C}$, U_i are its inputs and V_i are its outputs (these are identified with elements of X or Y if $\widehat{g}_i$ is a first gadget or a last gadget). Note that (U_i, V_i) is always plausible for $\widehat{g}_i$, by definition. Let us define the following hybrid wire assignment distributions:

$\mathcal{W}^0_{\widehat{C}}$: $\mathcal{W}_{\widehat{C}}(X|Y)$.

$\mathcal{W}^i_{\widehat{C}}$ $(i \in [1, s])$: Same as $\mathcal{W}^{i-1}_{\widehat{C}}$ except that the assignment to the wires inside $\widehat{g}_i$ is replaced by $R_{\widehat{g}_i}(U_i, V_i)$ with $R_{\widehat{g}_i} \leftarrow \mathsf{REC}_{\widehat{g}_i}$.

The following claim shows that $\mathcal{W}^{i-1}_{\widehat{C}}$ and $\mathcal{W}^i_{\widehat{C}}$ are $(\mathcal{L}_{\widehat{g}_i}, \tau_{\widehat{g}_i}, \epsilon_{\widehat{g}_i})$-leakage-indistinguishable for all $i \in [1, s]$. More precisely,

Claim. Let $\mathcal{L}_{\widehat{g}_i}$ be some class of leakage functions and let $\tau_{\widehat{g}_i} > 0, \epsilon_{\widehat{g}_i} > 0$. For any $i \in [1, s]$, if $\widehat{g}_i$ is $(\mathcal{L}_{\widehat{g}_i}, \tau_{\widehat{g}_i}, \epsilon_{\widehat{g}_i})$-reconstructible, then the distributions $\mathcal{W}^{i-1}_{\widehat{C}}$ and $\mathcal{W}^i_{\widehat{C}}$ are $(\mathcal{L}_{\widehat{g}_i}, \tau_{\widehat{g}_i} - O(st^2), \epsilon_{\widehat{g}_i})$-leakage-indistinguishable.

Proof. For any $i \in [1, s]$ we use Lemma 5 with the following mapping: $\mathcal{W}_1 = \mathcal{W}^{i-1}_{\widehat{C}}, \mathcal{W}'_1 = \mathcal{W}^i_{\widehat{C}}$ and $\mathcal{W}_0 = \mathcal{W}_{\widehat{g}_i}(U_i|V_i), \mathcal{W}'_0 = R_{\widehat{g}_i}(U_i, V_i)$ with $R_{\widehat{g}_i} \leftarrow \mathsf{REC}_{\widehat{g}_i}$. To apply Lemma 5 we need to define the distribution F_S, where $f_S \leftarrow \mathsf{F}_S$:

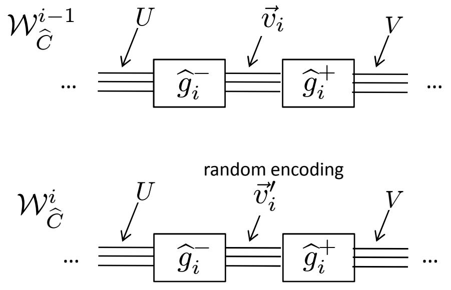

Fig. 5. This shows the notation used in Claim 4.4. In the two consecutive experiments, $\mathcal{W}_{\widehat{C}}^{i-1}$ and $\mathcal{W}_{\widehat{C}}^{i}$, U and V are sampled from the same distribution, whereas in $\mathcal{W}_{\widehat{C}}^{i-1}$ $\vec{v}_i$ is drawn from the honest distribution and in $\mathcal{W}_{\widehat{C}}^{i}$ $\vec{v}_i'$ is a random encoding.

1. For all $j \geq i+1$ sample from $\mathcal{W}_{\widehat{g}_j}(U_j|V_j)$ and hard-wire the result into the description of f_S.
2. For all $j \leq i-1$ sample $R_{\widehat{g}_j} \leftarrow \mathsf{REC}_{\widehat{g}_j}$ and run $R_{\widehat{g}_j}(U_j, V_j)$ to obtain a valid wire assignment for that part of the circuit. Hardwire the result into the description of f_S.
3. For the part of the wire assignment that represents $\widehat{g}_i$, f_S just outputs its input.

Note that f_S takes as long to sample as the time required to either compute or reconstruct the $s-1$ gadgets, which, in our case, is $O(t^2)$ per gadget. It is easy to see that f_S is in $\mathsf{SHALLOW}(0,0)$. Moreover, if f_S takes as input a sample from $\mathcal{W}_{\widehat{g}_i}(U_i|V_i)$ then its output is distributed as $\mathcal{W}_{\widehat{C}}^{i-1}$. On the other hand if the input is $R_{\widehat{g}_i}(U_i, V_i)$, then f_S's output is identically distributed to $\mathcal{W}_{\widehat{C}}^{i}$. These facts, combined with Lemma 5 and the fact that $\mathcal{W}_0$ and $\mathcal{W}_0'$ are $(\mathcal{L}_{\widehat{g}_i}, \tau_{\widehat{g}_i}, \epsilon_{\widehat{g}_i})$-leakage-indistinguishable, show that $\mathcal{W}_1 = \mathcal{W}_{\widehat{C}}^{i-1}$ and $\mathcal{W}_1' = \mathcal{W}_{\widehat{C}}^{i}$ are $(\mathcal{L}_{\widehat{g}_i}, \tau_{\widehat{g}_i} - O(s,t^2), \epsilon_{\widehat{g}_i})$-leakage-indistinguishable. This concludes the claim. □

Next, we show that we can replace the connecting wires in $\widehat{C}$ with random encodings. Let m be the number of connecting wire bundles in $\widehat{C}$ (since every gadget in $\widehat{C}$ has at most two inputs, $m \leq 2s$). Associate each bundle of connecting wires with integer $i \in [1,m]$ and denote the encoding carried by this bundle by $\vec{v}_i$. Denote by $\widehat{g}_-^i$ the gadget that has $\vec{v}_i$ as an output wire bundle, and by $\widehat{g}_+^i$ the gadget that has $\vec{v}_i$ as input (see Figure 5). We define iteratively the following hybrid wire assignment distributions:

$\mathcal{W}_{\widehat{C}}^{i}$ $i \in [s+1, s+m]$: Same as $\mathcal{W}_{\widehat{C}}^{i-1}$ except that $\vec{v}_i$ is replaced with a random encoding $\vec{v}_i'$ (and the internal wires in $\widehat{g}_-^i$ and $\widehat{g}_+^i$ are adjusted accordingly, as the wire bundles are given as inputs to the reconstructors of $\widehat{g}_-^i$ and $\widehat{g}_+^i$).

Intuitively: $\mathcal{W}_{\widehat{C}}^{s}$ is the wire assignment distribution that results from running, for each gadget in $\widehat{C}$, its corresponding reconstructor using honestly-computed connecting wires. Then, in $\mathcal{W}_{\widehat{C}}^{i}$ for $i = s+1, \ldots, s+m$, we replace step-by-step the honest encodings at the connecting wires with random encodings. The final distribution, $\mathcal{W}_{\widehat{C}}^{s+m}$, is identical to $\mathsf{REC}_{\widehat{C}}(X,Y)$.

We next prove a claim stating that for all $i \in [s+1, s+m]$ the distributions $\mathcal{W}_{\widehat{C}}^{i-1}$ and $\mathcal{W}_{\widehat{C}}^{i}$ are $(\mathcal{L}_{\mathcal{W}}, \tau_{\mathcal{W}}, \epsilon_{\mathcal{W}})$-leakage-indistinguishable.

Claim. Let $\mathcal{L}_{\mathcal{W}}$ be some class of leakage functions and let $\tau_{\Pi} > 0, \epsilon_{\Pi} > 0$. If Π is $(\mathcal{L}_{\Pi}, \tau_{\Pi}, \epsilon_{\Pi})$-leakage-indistinguishable, then for all $i \in [s+1, s+m]$ the distributions $\mathcal{W}_{\widehat{C}}^{i-1}$ and $\mathcal{W}_{\widehat{C}}^{i}$ are $(\mathcal{L}_{\mathcal{W}}, \tau_{\mathcal{W}}, \epsilon_{\mathcal{W}})$-leakage-indistinguishable with $\epsilon_{\mathcal{W}} = \epsilon_{\Pi}$, $\tau_{\mathcal{W}} = \tau_{\Pi} - O(st^2)$, and $\mathcal{L}_{\Pi} = \mathcal{L}_{\mathcal{W}} \circ \mathsf{SHALLOW}(2, O(t^2))$.

Proof. To prove this statement for any $i \in [s+1, s+m]$, we apply Lemma 5 with the following assignment for the distributions: $\mathcal{W}_1 = \mathcal{W}_{\widehat{C}}^{i-1}, \mathcal{W}_1' = \mathcal{W}_{\widehat{C}}^{i}$ and $\mathcal{W}_0 = \mathsf{Enc}(v_i), \mathcal{W}_0' = \mathsf{Enc}(v_i')$, with $v_i' \leftarrow \mathcal{K}$. Furthermore, we define the distribution F_S, with $f_S \leftarrow \mathsf{F}_S$ that takes as input a single encoding $\vec{e}$:

1. Sample $R_{\widehat{g}_-^i}$ from $\mathsf{REC}_{\widehat{g}_-^i}$ and $R_{\widehat{g}_+^i}$ from $\mathsf{REC}_{\widehat{g}_+^i}$ and hard-wire their descriptions into f_S.
2. Sample the values for all the connecting wire bundles except $\vec{v}_i$ according to $\mathcal{W}_{\widehat{C}}^{i}$ (which is the same as $\mathcal{W}_{\widehat{C}}^{i-1}$ for those wire bundles).
3. For each gadget $\widehat{g}$ in $\widehat{C}$ except $\widehat{g}_-^i$ and $\widehat{g}_+^i$, pick a reconstructor from the appropriate reconstructor distribution $R_{\widehat{g}} \leftarrow \mathsf{REC}_{\widehat{g}}$, and run $R_{\widehat{g}}(U,V)$, where (U,V) are the sampled values for the input and output wire bundles of $\widehat{g}$. The resulting wire assignments for each gadget are hard-wired into f_S.
4. Pick and hardwire reconstructors $R_{\widehat{g}_-^i} \leftarrow \mathsf{REC}_{\widehat{g}_-^i}$ and $R_{\widehat{g}_+^i} \leftarrow \mathsf{REC}_{\widehat{g}_+^i}$ and wire their descriptions into f_S. On input $\vec{e}$, run on-line the reconstructors $R_{\widehat{g}_-^i}$ and $R_{\widehat{g}_+^i}$, using as their inputs and outputs the wire bundles already sampled and $\vec{v}_i$ set to e. Output their resulting wire assignments together with the hardwired wire assignments for all the other gadget reconstructors.

We claim that

$$\mathcal{W}_{\widehat{C}}^{i-1} \equiv f_S(\vec{e}), \text{ if } \vec{e} \leftarrow \mathsf{Enc}(v_i),$$
$$\mathcal{W}_{\widehat{C}}^{i} \equiv f_S(\vec{e}), \text{ if } \vec{e} \leftarrow \mathsf{Enc}(v_i').$$

Indeed, in either case, all the wires internal to gadgets are computed according to reconstructors, and the connecting wire bundles except $\vec{v}_i$ are sampled identically in the two distributions. If $e \leftarrow \mathsf{Enc} v_i$ then, because all the gadgets are rerandomizing, the joint distribution of e together with all the other wires is indeed $\mathcal{W}_{tC}^{i-1}$ (note that this is the only place where we use the fact that the gadgets are rerandomizing, but the use of this fact here is crucial: if $\mathsf{Enc}(v_i)$ was correlated with some other connecting wire bundle, we could not hardwire that bundle into f_S, because it would not be known until e was given).

Sampling $f_S \leftarrow \mathsf{F}_S$ takes $O(st^2)$ time, because that's how long it takes to sample the reconstructors. Let us now analyze the complexity of f_S. Since most of the wire assignments are hard-wired in advance into f_S, on input $\vec{e}$ f_S only needs to run $\widehat{g}_-^i$ and

$\widehat{g}^i_+$. Thus, we get that functions $f_S \leftarrow \mathsf{F}_S$ can be computed by $\mathsf{SHALLOW}(s_{\mathcal{W}}, d_{\mathcal{W}})$ with $s_{\mathcal{W}} = \mathsf{size}(\mathsf{REC}_{\widehat{g}^i_-}) + \mathsf{size}(\mathsf{REC}_{\widehat{g}^i_+})$ and depth $d_{\mathcal{W}} = \mathsf{max}(\mathsf{depth}(\mathsf{REC}_{\widehat{g}^i_-}), \mathsf{depth}(\mathsf{REC}_{\widehat{g}^i_+}))$. From the analysis of single gadget reconstructors it follows that the size and depth of reconstructors is maximal for the $\widehat{\odot}$ gadget. More precisely, with Lemma 7 we get $d_{\mathcal{W}} = 2$ and size $s_{\mathcal{W}} = O(t^2)$. If we now apply Lemma 5 with the fact that $\mathcal{W}_0$ and $\mathcal{W}'_0$ are $(\mathcal{L}_\Pi, \tau_\Pi, \epsilon_\Pi)$-leakage-indistinguishable, we get that $\mathcal{W}_1 = \mathcal{W}^{i-1}_{\widehat{C}}$ and $\mathcal{W}'_1 = \mathcal{W}^{i}_{\widehat{C}}$ are $(\mathcal{L}_{\mathcal{W}}, \tau_{\mathcal{W}}, \epsilon_{\mathcal{W}})$-leakage-indistinguishable. □

Putting now the results from Claim 4.4 and Claim 4.4 together we get that $\mathcal{W}^0_{\widehat{C}} = \mathcal{W}_{\widehat{C}}(X|Y)$ and $\mathcal{W}^{s+m}_{\widehat{C}} = \mathsf{REC}_{\widehat{C}}(X,Y)$ are $(\mathcal{L}_{\widehat{C}}, \tau_{\widehat{C}}, \epsilon_{\widehat{C}})$-leakage-indistinguishable. Here, $\tau_{\widehat{C}} = \tau_\Pi - O(st^2)$ and

$$\epsilon_{\widehat{C}} = m \cdot \epsilon_\Pi + \sum_{1 \leq i \leq s} \epsilon_{\widehat{g}_i} \leq m \cdot \epsilon_\Pi + s \cdot \max_{1 \leq i \leq s}(\epsilon_{\widehat{g}_i}). \tag{7}$$

Since $\max\limits_{1 \leq i \leq s}(\epsilon_{\widehat{g}_i}) = t\epsilon_\Pi$ we get with 7 and $m \leq 2s$ that

$$\epsilon_{\widehat{C}} = (m + ts)\epsilon_\Pi = \leq \epsilon_\Pi s(t+2).$$

It remains to analyze the complexity of $\mathcal{L}_{\widehat{C}}$. In Lemma 2-4, we can set $\mathcal{L} = \mathcal{L}_{\widehat{C}}$. Furthermore, in Lemma 7 we can set $\mathcal{L}_{\mathsf{REC}_{\widehat{\odot}}}$ to $\mathcal{L}_{\widehat{C}}$, and thus get that if Π is $(\mathcal{L}_\Pi, \tau, \epsilon_\Pi)$-leakage-indistinguishable, then $\widehat{\odot}$ is $(\mathcal{L}_{\mathsf{REC}_{\widehat{\odot}}}, \tau - O(t^2), t\epsilon_\Pi)$-reconstructible with $\mathcal{L}_\Pi = \mathcal{L}_{\mathsf{REC}_{\widehat{\odot}}} \circ \mathsf{SHALLOW}(3, O(t^2))$. Finally, we let $\mathcal{L}_{\mathcal{W}}$ from Claim 4.4 be $\mathcal{L}_{\widehat{C}}$. This gives us

$$\mathcal{L}_\Pi = \max(\mathcal{L}_{\widehat{C}} \circ \mathsf{SHALLOW}(2, O(t^2)), \mathcal{L}_{\widehat{C}} \circ \mathsf{SHALLOW}(3, O(t^2))) = \mathcal{L}_{\widehat{C}} \circ \mathsf{SHALLOW}(3, O(t^2)).$$

Note that depth 3 can be reduced to 2 in the case of $\mathsf{GF}(2)$ (see Lemma 7). This concludes the proof. □

5 Stateful Circuits

We now proceed to prove Theorem 1 (stated in Section 5), thereby establishing bout the security of the transformation.

Proof (Proof of Theorem 1). Let us give an outline of the proof. We have to show that for every q_{TR} adaptive $(\mathcal{L}_{\mathsf{TR}}, \tau_{\mathsf{TR}})$-observer OBS, there exists a simulator $\mathsf{SIM}_{\mathsf{TR}}$ that only has black-box access to the circuit and runs in time at most τ'_{TR} time, such that for every stateful circuit $C[M]$ of size s, with memory of size k, k_{I} inputs and k_{O} outputs, the output distribution of OBS and $\mathsf{SIM}_{\mathsf{TR}}$ are statistically close.

The idea for the proof is simple: $\mathsf{SIM}_{\mathsf{TR}}$ runs OBS as a subroutine and eventually will output whatever OBS outputs. To get the right output $\mathsf{SIM}_{\mathsf{TR}}$ has to simulate $\mathsf{Exp}^{\mathrm{real}}_{\mathsf{TR}}$ without knowledge of the initial secret state M_0. This in particular includes answering the leakage queries of OBS in a way that is consistent with the public inputs and outputs of the circuit.

For the proof we view each clock cycle $1 \leq i \leq q_{\mathsf{TR}}$ of the stateful circuit $C[M]$ as a stateless circuit C^* that runs on public input x_i and outputs a public output y_i. Additionally, during the computation it will make use of the state M_{i-1} that in C^* will be represented as a secret input and returns as additional output M_i (notice that these inputs and outputs are already in encoded form and will *not* require $\widehat{\texttt{encoder}}$ and $\widehat{\texttt{decoder}}$ gadgets). Thus, C^* has $k + k_{\mathrm{I}}$ inputs, $k + k_{\mathrm{O}}$ outputs and s gates (since instead of having $\texttt{mask}$ gates after the memory gates, we put them after the additional inputs). Let now $\widehat{C}[\widehat{M}]$ be the transformation of $C[M]$. $\widehat{C}$ consist of a special encoder sub-circuit $\widehat{E}$ that consists of $\widehat{\texttt{encoder}}$ gadgets taking as input x_i and outputting a valid encoding, and a decoder sub-circuit $\widehat{D}$ made out of $\widehat{\texttt{decoder}}$ gadgets that takes as input Y_i (this is the output of $\widehat{C}[\widehat{M_{i-1}}](x_i)$ in encoded form) and outputs y_i. Furthermore, since $\widehat{C}$ is stateful it has memory gadgets to store $\widehat{M_i}$. As already outlined above, for the proof we will eliminate the memory gadgets and view the secret state $\widehat{M_{i-1}}$ as $\widehat{C}$'s secret input and $\widehat{M_i}$ as its secret output. We will denote this circuit with $\widehat{C}^*$ and write $(\widehat{M_i}, y_i) \leftarrow \widehat{C}^*(\widehat{M_{i-1}}, x_i)$ for the computation in the ith clock cycle.

Notice that since $\widehat{C}^*$ is stateless, it looks promising to apply Lemma 8 which would prove almost instantly the security of the transformation. However, we encounter some problems here: first, Lemma 8 explicitly excludes $\mathsf{encoder}$ and $\mathsf{decoder}$ gates. Second, the secret state $\widehat{M_i}$ for $1 \leq i < q_{\mathsf{TR}}$ can be observed two times (once as the output of the ith cycle, and once as the input to the $(i+1)$th), and, moreover, the observer can pick its leakage functions adaptively. Let us be more precise about the last point: when OBS observes the computation in $\widehat{C}^*(\widehat{M_{i-1}}, x_i)$ with output y_i, he can pick a leakage function f_i and obtains some knowledge about the secret state $\widehat{M_i}$. Adaptively, based on that knowledge (i.e. on the output y_i and the leakage that may depend on $\widehat{M_i}$) the observer may then pick a leakage function f_{i+1} and obtains leakage depending on the computation $\widehat{C}^*(\widehat{M_i}, x_{i+1})$. The crucial observation here is that both the leakage in the ith and $(i+1)$th observation may very well depend on the secret state $\widehat{M_i}$. We will address these issues in the following analysis.

Let us now define how the simulator $\mathsf{SIM}_{\mathsf{TR}}$ works and then show by a hybrid argument that this simulation is indistinguishable for any q_{TR} adaptive $(\mathcal{L}_{\mathsf{TR}}, \tau_{\mathsf{TR}})$-observer OBS. $\mathsf{SIM}_{\mathsf{TR}}$ runs in the experiment $\mathsf{Exp}^{\mathrm{sim}}_{\mathsf{TR}}$ described in Definition 1 and is defined for any circuit C as given in 6. Notice that $\mathsf{SIM}_{\mathsf{TR}}$ does not know the secret state, but instead uses random encodings Z_i to compute a valid wire assignment. Furthermore, it uses Y_i as the encoded public output of $\widehat{C}^*$ on input $(\widehat{M_{i-1}}, X_i)$, which was sampled independently from the distribution $\mathsf{Enc}(y_i)$ (in particular, independent of the inputs $(X_i, \widehat{M_{i-1}})$). If we can show that this simulation is indistinguishable for OBS that expects to run in $\mathsf{Exp}^{\mathrm{real}}_{\mathsf{TR}}$ (cf. Definition 1), then we have proven the theorem. We prove this indistinguishability by a hybrid argument following a similar approach as in Lemma 8, though, due to the adaptivity of the observer, we will not argue about hybrid wire assignment distributions,

```
Simulator $\mathsf{SIM}_{\mathsf{TR}}(\mathsf{OBS}, q_{\mathsf{TR}}, C)$
Sample uniformly at random encodings $(Z_0, \ldots, Z_{q_{\mathsf{TR}}+1})$, where each $Z_i$
        consists of $k$ encodings of random elements of $\mathcal{K}$
For each $i \in [1, q_{\mathsf{TR}}]$ sample $R_{\widehat{C}^*} \leftarrow \mathsf{REC}_{\widehat{C}^*}$
Run $\mathsf{OBS}(q_{\mathsf{TR}}, C)$
For each query $(f_i, x_i)$ of $\mathsf{OBS}$:
    Query $C[M_{i-1}]$ on input $x_i$ to obtain $y_i$
    Sample encoding $Y_i \leftarrow \mathsf{Enc}(y_i)$
    Compute wire assignment $W_E$ for the $\widehat{\texttt{encoder}}$ with input $x_i$ and its output $X_i$
    Compute wire assignment $W_D$ for $\widehat{\texttt{decoder}}$ gadget with input $Y_i$ and output $y_i$
    Sample $W \leftarrow R_{\widehat{C}^*}((Z_{i-1}, X_i), (Z_i, Y_i))$
    Return $(f_i(W_E, W, W_D), y_i)$ to $\mathsf{OBS}$
Return the output of $\mathsf{OBS}$.
```

Fig. 6. Description of the simulator that runs in the experiment $\mathsf{Exp}^{\mathrm{sim}}_{\mathsf{TR}}$.

but rather about hybrid experiments. In each of these hybrid experiments we describe how the simulation of OBS's view can be done.

In the first hybrid experiment $\mathsf{Exp}^{-1}_{\mathsf{TR}}$ the simulator will use the correct honest state M_0 to answer all the queries by computing honestly the wire assignment for all wires in the circuit. Since this is only a syntactic change to $\mathsf{Exp}^{\mathrm{real}}_{\mathsf{TR}}$, trivially for any q_{TR} adaptive $(\mathcal{L}_{\mathsf{TR}}, \tau_{\mathsf{TR}})$-observer, for any circuit C and any initial state M_0, we get that:

$$|\Pr[\mathsf{Exp}^{\mathrm{real}}_{\mathsf{TR}}(\mathsf{OBS}, \mathcal{L}_{\mathsf{TR}}, q_{\mathsf{TR}}, C, M_0, t) = 1] - \Pr[\mathsf{Exp}^{-1}_{\mathsf{TR}} = 1]| = 0.$$

Let us now give an outline of the following hybrid experiments. In the hybrid experiment $\mathsf{Exp}^{0}_{\mathsf{TR}}$, the simulator answers OBS's queries by using as the wire assignment for the sub-circuits $\widehat{C}^*$ the output of an appropriate reconstructor as opposed to the real wire assignment in $\mathsf{Exp}^{-1}_{\mathsf{TR}}$. In the further $(q_{\mathsf{TR}}+2)k$ hybrid experiments, $\mathsf{Exp}^{i,j}_{\mathsf{TR}}$ with $i \in [0, q_{\mathsf{TR}}+1]$ and $j \in [1, k]$, we replace step-by-step the real memory $\widehat{M_i}$ with random encodings Z_i. Notice that in all these experiments we use Y_i independently sampled from the distribution $\mathsf{Enc}(y_i)$ as the encoded public output of $\widehat{C}^*$. As soon as we start to replace the real memory $\widehat{M_i}$ with the random encodings Z_i, $(Y_{i+1}, \widehat{M}_{i+1})$ may no longer be a consistent output of $\widehat{C}^*$. However, we will show that this setting is indistinguishable from the case where all wires are honestly computed.

Let us be more precise and define $\mathsf{Exp}^{0}_{\mathsf{TR}}$ as the hybrid experiment where the simulator knows M_0, which allows him (together with the knowledge of x_i) to sample $\widehat{M_i}, X_i, Y_i$. Then, this simulator answers each of the q_{TR} queries (f_i, x_i) by computing wire assignments for $\mathcal{W}_{\widehat{C}^*}$ by running the appropriate reconstructor $R_{\widehat{C}^*}((X_i, \widehat{M}_{i-1}), (Y_i, \widehat{M_i}))$ and computing W_E, W_D in plain view. We need to show that if each of the wire assignments for $\widehat{C}^*$

is $(\mathcal{L}_{\widehat{C}}, \tau_{\widehat{C}}, \epsilon_{\widehat{C}})$-reconstructible for some $\mathcal{L}_{\widehat{C}}$ and $\tau_{\widehat{C}} \geq 0, \epsilon_{\widehat{C}} \geq 0$, then the simulation in experiment $\mathsf{Exp}^0_{\mathsf{TR}}$ is $\epsilon_{\widehat{C}}$ indistinguishable for any q_{TR}-adaptive $(\mathcal{L}_{\widehat{C}}, \tau_{\widehat{C}} - q_{\mathsf{TR}} O(st^2))$-observer OBS from the simulation in $\mathsf{Exp}^{-1}_{\mathsf{TR}}$.

For each query (f_i, x_i) computing the wire assignment W_E and W_D is easy (and can be done by the simulator) since the inputs to both are known: for the $\widehat{\texttt{encoder}}$ it is x_i and for the $\widehat{\texttt{decoder}}$ Y_i. Moreover, it is not difficult to see that the pair $((X_i, \widehat{M}_{i-1}), (Y_i, \widehat{M}_i))$ is a plausible input for the reconstructor of $\widehat{C}$ using the rerandomizing property. The remaining "inner" parts of $\widehat{C}^*$ are computed by the reconstructor $R_{\widehat{C}^*}$. Since $\widehat{C}^*$ is rerandomizing and $(\mathcal{L}_{\widehat{C}}, \tau_{\widehat{C}}, \epsilon_{\widehat{C}})$-reconstructible, we get by the same argument as in Claim 4.4 that replacing real wire assignments for $\widehat{C}^*$ with reconstructed once is indistinguishable for OBS (given that the inputs to the reconstructor are plausible). Notice also that we need to replace the reconstructors for *all* clock cycles. Proving this can be done by a simple hybrid argument along the lines of Claim 4.4. We omit the details in this sketch and obtain for any q_{TR} adaptive $(\mathcal{L}_{\widehat{C}}, \tau_{\widehat{C}} - q_{\mathsf{TR}} O(st^2))$-observer OBS

$$|\Pr[\mathsf{Exp}^0_{\mathsf{TR}} = 1] - \Pr[\mathsf{Exp}^1_{\mathsf{TR}} = 1]| \leq q_{\mathsf{TR}} \epsilon_{\widehat{C}}. \tag{8}$$

Before moving on and showing that we can replace each encoding of the state by a random encoding, let us analyze how the parameters $\mathcal{L}_{\widehat{C}}, \tau_{\widehat{C}}$ and $\epsilon_{\widehat{C}}$ can be expressed in terms of the parameters for the encoding scheme $\mathit{\Pi}$. If the underlying encoding scheme $\mathit{\Pi}$ is 2-adaptive $(\mathcal{L}_{\mathit{\Pi}}, \tau_{\mathit{\Pi}}, \epsilon_{\mathit{\Pi}})$-leakage indistinguishable, then we get by Lemma 8 $\tau_{\widehat{C}} \leq \tau_{\mathit{\Pi}} - q_{\mathsf{TR}} O(st^2))$,

$$\epsilon_{\widehat{C}} = \epsilon_{\mathit{\Pi}} s(t+2), \text{ and} \tag{9}$$

$$\mathcal{L}_{\mathit{\Pi}} = \mathcal{L}_{\widehat{C}} \circ \mathsf{SHALLOW}(3, O(t^2)), \text{ for some } \mathcal{L}_{\widehat{C}}. \tag{10}$$

Next, we prove that the simulator can replace each single encoding of the state with a random encoding. This proof is similar to Claim 4.4 with some subtleties. Notice that during q_{TR} observations the observer can learn information on $q_{\mathsf{TR}} + 2$ states with each having k elements. Thus, we define $(q_{\mathsf{TR}} + 2)k$ hybrid experiments, with $i \in [0, q_{\mathsf{TR}} + 1], j \in [1, k]$

$\mathsf{Exp}^{i,j}_{\mathsf{TR}}$**:** This is as the previous experiment, but replacing the jth element of the ith state with a random encoding,

and order them as follows

$$\mathsf{Exp}^{0,1}, \ldots, \mathsf{Exp}^{0,k}, \mathsf{Exp}^{1,1}, \ldots, \mathsf{Exp}^{q+1,k-1}, \mathsf{Exp}^{q+1,k}.$$

For ease of notation, we identify $\mathsf{Exp}^{i,0}$ with $\mathsf{Exp}^{i-1,k}$ for $i > 0$ and $\mathsf{Exp}^{0,0}$ with Exp^0.

We next prove that the simulation in these hybrid experiments are indistinguishable for a q_{TR}-adaptive $(\mathcal{L}_{\mathsf{TR}}, \tau_{\mathsf{TR}})$-observers.

Claim. Let $\mathcal{L}_{\mathcal{W}}$ be some class of leakage functions and let $\epsilon_{\mathit{\Pi}} \geq 0, \tau_{\mathit{\Pi}} \geq 0$. If $\mathit{\Pi}$ is 2-adaptive $(\mathcal{L}_{\mathit{\Pi}}, \tau_{\mathit{\Pi}}, \epsilon_{\mathit{\Pi}})$-leakage indistinguishable, then for any q_{TR}-adaptive $(\mathcal{L}_{\mathcal{W}}, \tau_{\mathcal{W}})$-observer OBS

(where $\tau_{\mathcal{W}} = \tau_{\Pi} - q_{\mathsf{TR}} O(st^2)$ and $\mathcal{L}_{\Pi} = \mathcal{L}_{\mathcal{W}} \circ \mathsf{SHALLOW}(2, O(t^2)))$, for any circuit C, any initial state M_0, and for any $i \in [0, q+1], j \in [1, k]$:

$$|\Pr[\mathsf{Exp}_{\mathsf{TR}}^{i,j-1} = 1] - \Pr[\mathsf{Exp}^{i,j} = 1]| < \epsilon_{\Pi}. \qquad (11)$$

Proof. We prove this claim by contradiction. Suppose there exists such an observer OBS, a state M_0, and values $i \in [0, q+1], j \in [1, k]$ such that (11) does not hold, then we will build a 2-adaptive $(\mathcal{L}_{\Pi}, \tau_{\Pi})$-observer OBS_{Π} that will distinguish an encoding of the jth element of the ith state from a random encoding. Such OBS_{Π} will simulate the environment for OBS, placing its target encoding as element number j in the ith state. Notice that OBS_{Π} will use in the simulation of the environment for OBS the fact that it can observe the target encoding two times, as well as the fact that it (unlike $\mathsf{SIM}_{\mathsf{TR}}$) is allowed to depend on the true initial state M_0. Notice again that knowing M_0 and the inputs x_i given by OBS in each round allows to compute all states M_i, $1 \le i \le q_{\mathsf{TR}} + 1$ and, thus, to sample $\widehat{M_i} \leftarrow \mathsf{Enc}(M_i)$.

OBS_{Π} will work as follows: it runs OBS as a subroutine and has to answer its queries. Before we describe how this is done first notice that we will omit details on how to compute the wire assignment of $\widehat{E}$ and $\widehat{D}$ since it is trivial given x_i and Y_i.[13] Depending on the value of i we distinguish three cases for answering the leakage queries (f_ℓ, x_ℓ), $\ell \in [1, q_{\mathsf{TR}}]$. First, the ith state is not part of the wire assignment observed during the ℓth query. Second, the ith state is part of the input of $\widehat{C}^*$ during the ℓth observation, and finally, the ith state represents parts of the output in the wire assignment for the ℓth observation. Let us be more precise and analyze how OBS_{Π} answers the ℓth query:

1. If the ith state is not part of the wire assignment for the ℓth observation, then OBS_{Π} answers the queries in the same way as in the two hybrid experiments (notice that both are identical except for the queries where the ith state is part of the observed wire assignment). We notice in particular that for such queries OBS_{Π} knows the secret input and output state (these can either be the real state or already random encodings) and thus has no problem simulating the answers for these queries (by computing all the wires in the circuits with the reconstructors).
2. If the ith state is part of the input of $\widehat{C}^*$ during the ℓth observation (i.e. $\ell = i+1$), then OBS_{Π} needs to produce a honest wire assignment for $\widehat{C}^*$. This wire assignment depends on the jth element of the ith state. OBS_{Π} is going to put its target encoding at this position. We denote by $\widehat{M_i'}$ the ith state in $\mathsf{Exp}_{\mathsf{TR}}^{i,j-1}$, except that whenever the jth element is used we use the target encoding. If the target encoding is an encoding of the real value at this position then the simulation is identical to $\mathsf{Exp}_{\mathsf{TR}}^{i,j-1}$. On the other hand, if it is an encoding of a random value, then the simulation is identical to $\mathsf{Exp}_{\mathsf{TR}}^{i,j}$. The difficulty is that if OBS_{Π} puts the target encoding at this position, then it has to come up with a wire assignment for $\widehat{C}^*$ that is consistent with the target encoding.

[13] x_i can basically be chosen by OBS himself which together with M_{i-1} allows to compute y_i and to sample $Y_i \leftarrow \mathsf{Enc}(y_i)$. Again notice that Y_i is for *all* experiments the encoding of the same value y_i.

Since the target encoding is only known to the leakage function, this has to be done in a shallow way. For this purpose OBS_{Π} will use an appropriate reconstructor $R_{\widehat{C}^*}$ drawn from $\mathsf{REC}_{\widehat{C}^*}$ (recall that already in experiment $\mathsf{Exp}^0_{\mathsf{TR}}$ the real wire assignments have been replaced by appropriate reconstructors) and run it as part of the leakage function f_{Π} on input $((X_{i+1}, \widehat{M}'_i), (Y_{i+1}, \widehat{M}_{i+1}))$. This would result in a security loss that depends on $k_{\mathrm{I}}, k_{\mathrm{O}}$ and k (since the reconstructor takes as input $((X_{i+1}, \widehat{M}'_i), (Y_{i+1}, \widehat{M}_{i+1}))$). A more thorough analysis will allow us to eliminate this loss.[14] Eventually, OBS_{Π} will do a leakage query f_{Π} to Eval_{Π}. f_{Π} takes as input a single target encoding, computes online the reconstructor $R_{\widehat{C}^*}$ on it, and finally evaluates f_ℓ on the output of $R_{\widehat{C}^*}$. The result of this will be returned to OBS_{Π}.

3. If the ith state represents part of the output in $\widehat{C}^*$'s wire assignment during the ℓth observation (i.e. $\ell = i$), then OBS_{Π} needs to produce a honest wire assignment for $\widehat{C}^*$. This wire assignment depends on the jth element of the ith state. The analysis is similar to step 2.

A crucial point in this simulation is that OBS_{Π} has to query Eval_{Π} twice to obtain a consistent simulation. Once when the ith state is input to $\widehat{C}^*$ and a second time when it is part of the output. This is possible since Π is assumed to be 2-adaptive leakage indistinguishable. For the rest of the reduction refer to Claim 4.4. With Lemma 8 (and the therein defined size and depth of the reconstructor) this gives us the following parameters: If Π is 2-adaptive $(\mathcal{L}_{\Pi}, \tau_{\Pi}, \epsilon_{\Pi})$-leakage indistinguishable, then for all $i \in [0, q_{\mathsf{TR}}+1], j \in [1, k]$ the simulation of two consecutive experiments are $\epsilon_{\mathcal{W}}$-indistinguishable for q_{TR}-adaptive $(\mathcal{L}_{\mathcal{W}}, \tau_{\mathcal{W}})$-observers. □

Putting things together we obtain with Claim 5 and equation 8-10: If Π is 2-adaptive $(\mathcal{L}_{\Pi}, \tau_{\Pi}, \epsilon_{\Pi})$-leakage-indistinguishable, then for any circuit C of size s, any initial state M_0 with size k, any $q_{\mathsf{TR}} \geq 0, t > 0$ and any q_{TR}-adaptive $(\mathcal{L}_{\mathsf{TR}}, \tau_{\mathsf{TR}})$-observer

$$|\Pr[\mathsf{Exp}^{\mathrm{real}}_{\mathsf{TR}}(\mathsf{OBS}, \mathcal{L}_{\mathsf{TR}}, q_{\mathsf{TR}}, C, M_0, t) = 1] - \Pr[\mathsf{Exp}^{\mathrm{sim}}_{\mathsf{TR}}(\mathsf{SIM}_{\mathsf{TR}}, q_{\mathsf{TR}}, C, M_0, t) = 1]| \leq \epsilon_{\mathsf{TR}}.$$

Here, $\tau_{\mathsf{TR}} = \tau_{\Pi} - q_{\mathsf{TR}} O(st^2)$ and

$$\begin{aligned}
\epsilon_{\mathsf{TR}} &= |\Pr[\mathsf{Exp}^{-1}_{\mathsf{TR}} = 1] - \Pr[\mathsf{Exp}^{0}_{\mathsf{TR}} = 1]| + \sum_{i \in [0, q_{\mathsf{TR}}+1], j \in [1,k]} |\Pr[\mathsf{Exp}^{i,j-1}_{\mathsf{TR}} = 1] - \Pr[\mathsf{Exp}^{i,j}_{\mathsf{TR}} = 1]| \\
&= q_{\mathsf{TR}} \epsilon_{\widehat{C}} + (q_{\mathsf{TR}} + 2) k \epsilon_{\Pi} \\
&= q_{\mathsf{TR}} \epsilon_{\Pi} s(t+2) + (q_{\mathsf{TR}} + 2) k \epsilon_{\Pi} \\
&\leq \epsilon_{\Pi} (q_{\mathsf{TR}} + 2)(s(t+2) + k).
\end{aligned}$$

[14] Fortunately, all of the outputs of $\widehat{C}^*$ and all inputs except for the jth element of the ith state are known to OBS_{Π} and, thus, can be hard-coded into the description of $R_{\widehat{C}^*}$. In particular, this guarantees that the size of the reconstructor for $\widehat{C}^*$ is extremely small (i.e. it consists only of a single gadget reconstructor).

It remains to analyze the complexity of $\mathcal{L}_{\mathsf{TR}}$. If we set in the above analysis $\mathcal{L}_{\widehat{C}} = \mathcal{L}_{\mathcal{W}} = \mathcal{L}_{\mathsf{TR}}$, then from Claim 5 and equation 10, we get

$$\mathcal{L}_{\Pi} = \max(\mathcal{L}_{\mathsf{TR}} \circ SHALLOW(2, O(t^2)), \mathcal{L}_{\mathsf{TR}} \circ SHALLOW(3, O(t^2))).$$

This proves the theorem. □

6 Variants

6.1 Unconditional Security against Constant-Depth-Circuits Leakage

The result in Theorem 1 is conditioned on assumptions that decoding is "hard" for functions in $\mathcal{L}$. Lower bounds on computational tasks are notoriously difficult to prove, and therefore, given our current state of knowledge, applying our results will, in most cases, require computational assumptions about hardness of decoding for a given class of leakage functions (or restrictions on how many wires they can observe, as in [18]).

However, we highlight some cases in which the theorem can be applied unconditionally.

6.1.1 AC^0 leakage

Consider circuits over $\mathcal{K} = \mathsf{GF}(2)$ with the decoder Dec being the parity function. It is known that parity is hard to approximate for constant depth (also known as AC^0) circuits. Thus, let $\mathcal{C}(d, s, \lambda)$ denote Boolean circuits made of NOT gates and unbounded fan-in AND and OR gates, with λ bits of output, size s, and depth (not counting NOT gates) d. Let $\mathcal{L}^1_{\mathsf{AC}^0}$ denote $\mathcal{C}(d, 2^{t^{1/d}}, 1)$ for some constant d. Then we can use the result of Håstad [17] (as cited in [21, Corollary 1]), which translated into our definition, says that parity encoding is $(\mathcal{L}^1_{\mathsf{AC}^0}, \tau_{\mathsf{AC}^0}, 2^{-t^{1/d+1}})$-leakage-indistinguishable, for any τ_{AC^0}. More generally, if we set $\mathcal{L}_{\mathsf{AC}^0} = \mathcal{C}(d, \exp(O(t^{(1-\delta)}/d), t^{\delta})$ for some $0 < \delta < 1$, then we can use the result of Dubrov and Ishai [11, Theorem 3.4], which says that parity encoding is $(\mathcal{L}_{\mathsf{AC}^0}, \tau_{\mathsf{AC}^0}, \exp(-\Omega(t^{(1-\delta)}/d)))$-leakage-indistinguishable.[15]

Since Theorem 1 requires that the underlying encoding scheme is leakage-indistinguishable against 2-adaptive observers, we will need the following lemma and prove it specifically for leakage functions modeled by circuits with unlimited fan-in `AND` and `OR` gates, such as AC^0.

Lemma 9 (2-adaptive leakage-indistinguishability). *Let* D, E *be two distributions and* $d, s, \lambda, \tau, \epsilon \geq 0$. *If* D *and* E *are* $(\mathcal{L}, O(\tau 2^{\lambda}), \epsilon)$*-leakage-indistinguishable, then the two distributions are* 2*-adaptive* $(\mathcal{L}', \tau, \epsilon)$*-leakage-indistinguishable, where* $\mathcal{L} = \mathcal{C}(d+2, O(s2^{\lambda}), 2\lambda)$ *and* $\mathcal{L}' = \mathcal{C}(d, s, \lambda)$.

[15] An even better result is obtained [11, Theorem 3.4] if one restricts d to $d = 1$: in that case, the ϵ parameter in leakage-indistinguishability gets reduced to $\exp(-\Omega(t - t^{\delta} \log t))$.

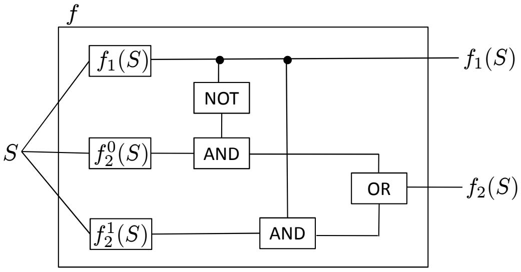

Fig. 7. The structure of f when $\lambda = 1$

Proof. Assume for contradiction that D and E are not 2-adaptive $(\mathcal{L}', \tau, \epsilon)$-leakage-indistinguishable, then there exists a 2-adaptive $(\mathcal{L}', \tau)$-observer OBS' such that 1 does not hold and we are going to show how to build a $(\mathcal{L}, O(\tau 2^{\lambda}))$-observer OBS such that

$$|\Pr[\mathsf{OBS}^{\mathsf{Eval}(\mathsf{D},\cdot)} = 1] - \Pr[\mathsf{OBS}^{\mathsf{Eval}(\mathsf{E},\cdot)} = 1| > \epsilon.$$

OBS runs OBS' as a subroutine and has to answer its 2 leakage queries f_1, f_2. The difficulty will be that OBS is supposed to only query Eval once. We will resolve this by putting all the adaptivity into the leakage function. The resulting leakage function will output the result of both leakage queries. The size of its circuit description will be exponential in λ.

OBS runs in two phases: first, a learning phase, where it is supposed to learn all possible leakage functions from OBS' for the second leakage query. Second, a leakage phase, where it builds a leakage function, obtains valid leakage from Eval with just a single query, and finally returns the reply to OBS'. The learning phase is pretty simple: OBS runs OBS' as a subroutine and gets back f_1. Since OBS is only allowed to query Eval once, it cannot query Eval with f_1 directly. Instead, it needs to find out f_2 that OBS' would use for every possible output $\Lambda \in \{0,1\}^{\lambda}$ of f_1. To do so, it rewinds OBS' 2^{λ} times, and each time gives a different Λ to OBS' to obtain the function f_2^{Λ}. (Observe that some values of Λ may be an invalid return for the leakage function f_1 and by this OBS' might notice that he is run in a simulated environment; in that case, OBS' may take more time than τ', so OBS will stop after τ' steps.)

Let us now describe the leakage phase. OBS will build its leakage function f as follows: on input S, f computes $\Lambda_1 = f_1(S)$, $f_2^{\Lambda_1}(S)$, and outputs both values.

We need to compute the circuit complexity of f. All 2^{λ} possible functions of f_2 need to be hardwired into the circuit, but they can be computed in parallel with each other and together with f_1 (so they increase the size, but not the depth, of the circuit). Then the output of one of these functions needs to be "selected" according to the output of f_1. This selection can be done by adding depth two (not counting `NOT` gates) and $O(2^{\lambda})$ additional gates, as shown in in Figure 7 for the case when $\lambda = 1$. Thus, we get for some $\mathcal{L}' = \mathcal{C}(d, s, \lambda)$ that $\mathcal{L} = \mathcal{C}(d+2, O(s2^{\lambda}), 2\lambda)$ as stated in the lemma.

The rest of the proof is straightforward: OBS can use its return from the oracle Eval to answer the two leakage queries f_1, f_2 of OBS'. Since this is a perfect simulation, we get that if OBS' can distinguish with advantage more than ϵ, then so can OBS. Notice that the running time of OBS is $O(2^{\lambda}\tau)$. □

Generalizing Lemma 9. We can generalize this lemma in two ways: first, by a similar argument we can prove security against p adaptive observers. This, however, increases the function's size exponentially in p *and* λ and moreover requires that the functions in $\mathcal{L}$ can output $p\lambda$ bits. Second, observe that we have proven this Lemma for the special case that the leakage functions are circuits with particular gates. This can be generalized to other function classes as long as they suffice to compute the function f (and in particular, the selection part). Notice that if the function class allows if-branching then the running time of f will only increase by a factor linear in p, whereas the size of the description will still suffer from an exponential blow-up.

If we instantiate in Theorem 1 the encoding scheme with parity, we get from Lemma 9 the following corollary.

Corollary 1. *Let $0 < \delta < 1$, $d \geq 4, t > 0, q_{\mathsf{TR}} \geq 0$ be some constants and let $d < 1/\delta - 1$. There exists a circuit transformation that is $(\mathcal{L}_{\mathsf{TR}}^{\mathsf{AC}^0}, \tau_{\mathsf{TR}}, \tau'_{\mathsf{TR}}, q_{\mathsf{TR}}, \epsilon_{\mathsf{TR}})$-secure for*

- *any $\tau_{\mathsf{TR}} \leq O(\tau_{\mathsf{AC}^0} 2^{-\lambda}) - q_{\mathsf{TR}} O(st^2)$, where s is the number of gates plus the number of input wires in C,*
- *some $\tau'_{\mathsf{TR}} \leq \tau_{\mathsf{TR}} + q_{\mathsf{TR}} O(st^2)$,*
- *some $\epsilon_{\mathsf{TR}} \leq (q_{\mathsf{TR}} + 2)(s_C(t+2) + k)\epsilon_{\mathsf{AC}^0}$, where k is the number of memory gates in C,*
- *$\mathcal{L}_{\mathsf{TR}}^{\mathsf{AC}^0} = \mathcal{C}(d-4, \exp(O(t^{(1-\delta)/d})), \lfloor t^{\delta}/2 \rfloor)$*[16]

Proof. For ease of notation, let $\epsilon_{\mathsf{AC}^0} = \exp(-\Omega(t^{(1-\delta)/d}))$. In [11, Theorem 3.4] the authors showed that the parity encoding is $(\mathcal{L}_{\mathsf{AC}^0}, \tau_{\mathsf{AC}^0}, \epsilon_{\mathsf{AC}^0})$ for any τ_{AC^0}, where $\mathcal{L}_{\mathsf{AC}^0}$ are circuits that output t^{δ} bits and are of depth d and size $\exp(O(t^{(1-\delta)}/d))$. Lemma 9 then shows that the encoding scheme is 2-adaptive $(\mathcal{L}'_{\mathsf{AC}^0}, O(\tau_{\mathsf{AC}^0} 2^{-\lambda}), \epsilon_{\mathsf{AC}^0})$-leakage indistinguishable, where $\mathcal{L}'_{\mathsf{AC}^0}$ are circuits that output $\lfloor t^{\delta}/2 \rfloor$ and have depth $d-2$ and size $\exp(O(t^{(1-\delta)/d}) - \lfloor t^{\delta}/2 \rfloor) = \exp(O(t^{(1-\delta)/d}))$, where the equality follows from $d < 1/\delta - 1$. If we now apply Theorem 1 with $\mathcal{K} = \mathsf{GF}(2)$, and observe that $\mathsf{SHALLOW}(2, O(t^2)$ can be implemented in $\mathcal{C}(2, O(t^2)), \lambda)$ for some λ (by expressing the constant-size depth-2 $\oplus$ gates as a constant-size CNF or DNF), we obtain the desired result. □

Improving the security loss. The bounds from Corollary 1 imply that asymptotically the leakage function classes that parity encoding and our transformed circuits can tolerate are similar as long as $d < 1/\delta - 1$. This restriction can be eliminated by relaxing the security definition. More precisely, if in Definition 1 we restrict the adversary to choose the leakage function f_i, $i \geq 2$, adaptively *only* on the output of the leakage functions $f_1, \ldots f_{i-2}$, then Theorem 1 won't require 2-adaptive leakage-indistinguishability of the encoding scheme. Hence, in Corollary 1 the restriction that $d < 1/\delta - 1$ can be eliminated. Notice though that the choice of f_i and the input x_i may still depend on the circuit's output $y_1, \ldots, y_{i-1}$.

[16] Notice that this is still in the class of AC^0 circuits since $d-4$ is constant.

6.1.2 $\mathsf{ACC}^0[q]$ leakage

A natural way to extend the class of $\mathcal{L}^1_{\mathsf{AC}^0}$ to something more general is to allow parity gates (or more generally, gates that compute modular sums). Clearly, such circuits can compute the parity function, but are there still other functions that cannot be computed by such circuit? This is indeed the case. Let us be a little bit more precise. For any integer n let $\mathtt{MOD}_n$ be the gate that outputs 0 if the sum of its inputs is 0 modulo n, and 1 otherwise. We define the class $\mathcal{L}^1_{\mathsf{ACC}^0[n]}$ as functions computable by Boolean circuits made of unbounded fan-in AND, OR, NOT and $\mathtt{MOD}_n$ gates, of output length 1, depth at most d, and polynomial size. Let Π_p be the simple additive secret sharing scheme modulo p. By a result of Razborov and Smolensky [35,39], for any distinct primes p and q, the encoding Π_p is leakage-indistinguishable for functions in $\mathcal{L}^1_{\mathsf{ACC}^0[q]}$. Since Π_p has a linear decoding function, we can apply Theorem 1 to get security of circuit transformation based on Π_p encoding.

6.2 Replacing opaque gates with reconstructible gadgets

The scheme in Section 3 requires an "opaque" gate $\mathcal{O}$, i.e., a leak-free component that samples string from a certain prescribed distribution. We now show that $\mathcal{O}$ can, in fact, be replaced by a gadget built out of smaller gates operating in plain view — as long as this gadget is reconstructible. Thus, to make *arbitrary* circuits leakage-resilient, it suffices to find a way to build one specific simple circuit in a way that is reconstructible for the given leakage class.

The following composition lemma shows that any opaque gate can be replaced by a reconstructible gadgets; to invoke it for the scheme of Section 3, let $g = \mathcal{O}$.

Lemma 10 (Inside-out composition). *Let C be an arbitrary circuit. Let $\widehat{C}$ be its transformation, and let g be gates in $\widehat{C}$. Let $\widehat{g}$ be a gadget for g_i, i.e., a circuit which implements the same (probabilistic) mapping as g_i but whose internal wires are observable by the leakage function. Let $\widehat{C}'$ be the composite circuit obtained from $\widehat{C}$ by replacing g with $\widehat{g}$. Let τ be the maximal time needed to compute all the wires in $\widehat{g}$ either by using an appropriate reconstructor or the real inputs.*

If $\widehat{C}$ is $(\mathcal{L}_{\widehat{C}}, \tau_{\widehat{C}}, \epsilon_{\widehat{C}})$-reconstructible by $\mathcal{R}_{\widehat{C}}$ for some $\mathcal{L}_{\widehat{C}}, \tau_{\widehat{C}}, \epsilon_{\widehat{C}}$ and $\widehat{g}$ are $(\mathcal{L}_{\widehat{g}}, \tau_{\widehat{g}}, \epsilon_{\widehat{g}})$ reconstructible by $\mathcal{R}_{\widehat{g}}$ for some $\mathcal{L}_{\widehat{g}}, \tau_{\widehat{g}}, \epsilon_{\widehat{g}}$, then $\widehat{C}'$ is $(\mathcal{L}_{\widehat{g}}, \min(\tau_{\widehat{C}}, \tau_{\widehat{g}}) - O(s\tau), \epsilon_{\widehat{g}} + \epsilon_{\widehat{C}})$-reconstructible by $\mathcal{R}'_{\widehat{g}} \circ \mathcal{R}_{\widehat{C}}$, where $\mathcal{R}'_{\widehat{g}} = \{x \mapsto (x, f(x)) | f \in \mathcal{R}_{\widehat{g}}\}$.

Proof (Proof sketch). Since $\widehat{C}$ is rerandomizing, so is $\widehat{C}'$. The reconstructor $\mathsf{REC}_{\widehat{C}'}$ is defined by composing the given reconstructors $\mathsf{REC}_{\widehat{C}}$ and $\mathsf{REC}_{\widehat{g}}$ as follows (see Figure 8 for notation). Sampling $R_{\widehat{C}'} \leftarrow \mathsf{REC}_{\widehat{C}'}$ is done by sampling $R_{\widehat{C}} \leftarrow \mathsf{REC}_{\widehat{C}}$ and $R_{\widehat{g}} \leftarrow \mathsf{REC}_{\widehat{g}}$. Given X and Y, $R_{\widehat{C}'}$ uses $R_{\widehat{C}}(X, Y)$ to assign the wires of $\widehat{C}'$ that come from $\widehat{C}$; in particular this assigns U and V, so $R_{\widehat{g}}(U, V)$ is used to assign the remaining wires inside $\widehat{g}$.

To argue indistinguishability, define the hybrid distribution $\mathcal{W}^1_{\widehat{C}'}(X, Y)$ as assigning the wires that come from $\widehat{C}$ honestly (i.e., drawing them from $\mathcal{W}_{\widehat{C}}(X|Y)$) and then reconstructing the remaining wires that come from $\widehat{g}$ (using $\mathsf{REC}_{\widehat{g}}(U, V)$).

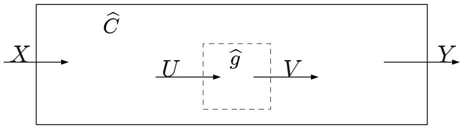

Fig. 8. Notation for $\widehat{C}'$ obtained by composing $\widehat{C}$ and $\widehat{g}$.

A distinguisher between $\mathcal{W}_{\widehat{C}'}(X|Y)$ and $\mathcal{W}^1_{\widehat{C}'}(X, Y)$ yields a distinguisher between $\mathsf{REC}_{\widehat{g}}(U, V)$ and $\mathcal{W}_{\widehat{g}}(U|V)$ (violating the property of $\mathsf{REC}_{\widehat{g}}$): let $W \leftarrow \mathcal{W}_{\widehat{C}}(X|Y)$ and let U and V be the input and output of $\widehat{g}$ in W; then given a challenge wire assignment to $\widehat{g}$, combine it with W to get a full wire assignment to $\widehat{C}'$ and feed it to the given distinguisher.

Similarly, a distinguisher between $\mathcal{W}^1_{\widehat{C}'}(X, Y)$ and $\mathsf{REC}_{\widehat{C}'}(X, Y)$ yields distinguisher between $\mathcal{W}_{\widehat{C}}(X|Y)$ and $\mathsf{REC}_{\widehat{C}}(X, Y)$ (violating the property of $\mathsf{REC}_{\widehat{C}}$): given a challenge wire assignment to $\widehat{C}$, one can complete it to a wire assignment to $\widehat{C}'$ (by filling in the wires of $\widehat{g}$ using $\mathsf{REC}_{\widehat{g}}$) and invoke the given distinguisher.

We omit in this sketch showing how the parameters work out. □

The above holds for replacing a single gate. The proof easily generalizes to replacing multiple gates, with a small loss in parameters.

Alternative security proof. Incidentally, this lemma suggests an alternative proof for the security of the scheme Section 3. One first defines *opaque encoded* gates that perform $\odot$, $\oplus$, etc. on encoded values in an opaque way: they get encoded inputs and output a random encoding of the correct result, without any leakage from within the gate (the wires between gates do, as usual, leak).[17] . One can transform any circuit C into a circuit $\widehat{C}$ using opaque encoded gates in the natural way. This $\widehat{C}$ is readily verified to be reconstructible (by a simplified version of Lemma 8). Then, replace each opaque encoded gate in $\widehat{C}$ with the corresponding reconstructible gadget given in Figure 3; the resulting circuit is, of course, identical to the transformed circuit of Section 3, and its reconstructibility follows from Lemma 10.[18]

Reducing randomness. The fact that opaque gates can be replaced by reconstructible gadgets implies another useful property: we can replace an opaque gate $\mathcal{O}$ by another opaque gate $\mathcal{O}'$ which uses less randomness, as long as the two cannot be distinguished by the observer. This allows a cheaper implementation of the opaque gate. For example, in the case of AC^0, $\mathcal{O}$ (which samples random t-bit strings with parity 0) can be replaced by $\mathcal{O}'$ which uses just $polylog(t)$ random bits, expanded to $t-1$ pseudorandom bits using Nisan's unconditional pseudorandom generator against AC^0 [28]. Similarly, against AC^0 one can use any imperfect source of randomness that is merely $polylog(t)$-independent [8].

[17] For example, $c \leftarrow a \oplus b$ is converted to $\vec{c} \leftarrow \mathsf{Enc}(\mathsf{Dec}(\vec{a}) \oplus \mathsf{Dec}(\vec{b}))$

[18] For tight results, note that, as in the proof of Lemma 8), only the reconstructors for first and last gates need to be computed online (due to the rerandomizing property of our gates).

6.3 Generalization to Arbitrary Reconstructible Gadgets

In Sections 3 through 5 we define and analyze a particular class of constructions, based on linear secret sharing schemes. In Section 6.1 we further specialize this to the case of the parity scheme and AC^0 leakages.

However, the proof techniques introduced along the way are, in fact, more general. Note that Lemma 8 relies essentially only on the fact that the gate gadgets are rerandomizing and reconstructible. One can obtain an analogous result using any encoding method (not necessarily a linear one) and a corresponding set of sound gate gadgets that are rerandomizing and reconstructible. We thus obtain a general composition lemma for reconstructors, informally stated thus:

Lemma 11 (Reconstructor composition for encoding-based circuits (informal)). *Let $\Pi = (\mathsf{Enc}, \mathsf{Dec})$ be any (not necessarily linear) encoding scheme that is $(\mathcal{L}_\Pi, \tau_\Pi, \epsilon_\Pi)$-leakage-indistinguishable for some $\mathcal{L}_\Pi, \tau_\Pi, \epsilon_\Pi$. Let G be a set of gates operating on plain values, and let $\widehat{G}$ be a set of corresponding gadgets, operating on encoded values, which are sound for Π. Suppose each gadget in $\widehat{G}$ is rerandomizing and $(\mathcal{L}_{\widehat{G}}, \tau_{\widehat{G}}, \epsilon_{\widehat{G}})$-reconstructible by $\mathcal{R}_{\widehat{G}}$. Let TR be the circuit transformation defined analogously to Section 3, but changed in the natural way to use Π and $\widehat{G}$. Then for any stateless circuit C of size s (without `encoder` or `decoder` gates) with k_I inputs and k_O outputs, $\mathsf{TR}(C)$ is rerandomizing and $(\mathcal{L}_{\widehat{C}}, \tau_{\widehat{C}}, \epsilon_{\widehat{C}})$ reconstructible by $\mathcal{R}_{\widehat{C}}$, for*

- $\mathcal{L}_\Pi = \max(\mathcal{L}_{\widehat{G}}, \mathcal{L}_{\widehat{C}} \circ (2 \times \mathcal{R}_{\widehat{G}}))$,
- *any* $\tau_{\widehat{C}} \leq \tau_\Pi - st_{\widehat{G}}$*, where* $t_{\widehat{G}}$ *is maximum time require to run or reconstruct a gadget in* $\widehat{G}$*,*
- *some* $\epsilon_{\widehat{C}} \leq s(\epsilon_{\widehat{G}} + i\epsilon_\Pi)$*, where i is the maximal fan-in of the gates in G,*
- $\mathcal{R}_{\widehat{C}} = (k_\mathrm{I} + k_\mathrm{O}) \times \mathcal{R}_{\widehat{G}}$.

Consequentially, the transformation TR is secure for the appropriate values of parameters (similar to Theorem 1).

The proofs are straightforward generalizations of Theorem 1 and Lemma 8, and thus omitted.

Lastly, note that these results further generalize to the case where each wire bundle in $\widehat{C}$ uses a different encoding scheme, since we never rely on the encoding schemes being identical.

6.4 Unconditional Security against Noisy Leakages

Thus far, we considered leakage classes $\mathcal{L}$ that are constrained in computational power per se. As discussed in Section 1.2.3, an alternative assumption, recently introduced by Rabin and Vaikuntanathan [34], is that the leakage is *noisy*, i.e., the observer gets an imperfect copy of the circuit's state subject to some noise. Their work shows a circuit transformation secure against such noisy leakage. Here we show that their model can be recast as a special case of ours, and outline an alternative, concise proof of the security of their scheme using our reconstructor machinery.

Noisy leakage, as defined in [34], assumes that each leaked bit[19] is randomly flipped independently of the rest, with some probability p. In our model, this is captured by the leakage class $\mathcal{L}_p = \{N_p\}$ where N_p is a probabilistic circuit that independently flips each input bit with probability p and outputs the result.

Theorem 2 ([34]). *The circuit transformation of [34], which encodes each bit into t bits, is $(\mathcal{L}_p, \tau, \tau + poly(t, q, s), q, 2^{-p^{\Theta(1)}})$-secure for circuit size s and any q, τ.*

Proof (Alternative proof (sketch)). The key observation (similarly to [34]) is that the parity encoding Π (as used in Section 6.1) is also leakage-indistinguishable against $\mathcal{L}_p$: by Yao's XOR lemma, when the encoding is sufficiently large compared to p, given a noisy string $N_p(x)$, the observer cannot approximate the parity of the original string x. Next, we observe that the gadgets defined in [34] are rerandomizing and, by (by a tight reduction to the leakage-indistinguishability of the encoding Π) also reconstructible against leakages in $\mathcal{L}_p$. By Lemma 11, the claim follows. □

6.5 Circuit Transformation from Opaque Public-key Encryption

We describe a simple circuit transformer, using public-key encryption, that is secure against any polynomial-time measurement (i.e., OBS is polynomial-time and likewise $\mathcal{L} = \mathsf{P}$). Our intention is, chiefly, to demonstrate another application of the general reconstructor-composition lemma of Section 6.3. Since this transformer relies on leak-free components that are large and have to maintain (short-term) secret states, we do not claim it can be implemented realistically or efficiently.

The transformation, $\mathsf{TR}_{\mathsf{PK}}$, is defined thus. Let $(\mathsf{PKGen}, \mathsf{PKEnc}, \mathsf{PKDec})$ be a public-key encryption scheme with IND-CPA security, and let k be a security parameter. Let the original circuit C consist of `AND`, `OR` and `NOT` gates, along with the special gates `encoder`, `decoder`, `mask` and `copy` defined in Section 3. The transformation, like that of Section 3, converts each wire in the circuit into a wire bundle, and each gate g into a gadget $\widehat{g}$. Here, each bundle carries an encryption of the original wire's value.

For the gadgets $\widehat{\texttt{AND}}$, $\widehat{\texttt{OR}}$, $\widehat{\texttt{NOT}}$, $\widehat{\texttt{copy}}$ and $\widehat{\texttt{mask}}$, the gadget consists simply of a single *opaque component* which decrypts all its inputs, applies the suitable operation on plaintexts, and encrypts the outputs; it is assumed that this opaque component is completely leak-free.[20] The encryption keys are separately negotiated along each wire, using the following *wire protocol*: the downstream opaque component generates a key pair $(\mathsf{sk}, \mathsf{pk})$ and sends

[19] We focus on their main model. For simplicity, we considering the binary where $\mathcal{K} = \mathsf{GF}(2)$. These observations are easily generalized to larger fields and suitable noise models.

[20] As in Section 6.2, the scheme remains secure if the opaque components are replaced by arbitrary gadgets which have the same functionality and are reconstructible.

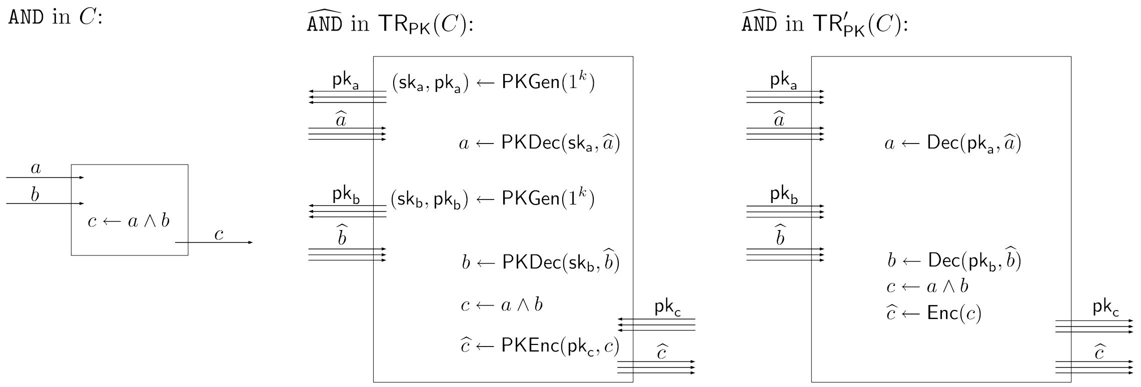
AND in C:
$\widehat{\texttt{AND}}$ in $\mathsf{TR}_{\mathsf{PK}}(C)$:
$\widehat{\texttt{AND}}$ in $\mathsf{TR}'_{\mathsf{PK}}(C)$:
a
b
$c \leftarrow a \wedge b$
c
$\mathsf{pk_a}$
$(\mathsf{sk_a}, \mathsf{pk_a}) \leftarrow \mathsf{PKGen}(1^k)$
$\widehat{a}$
$a \leftarrow \mathsf{PKDec}(\mathsf{sk_a}, \widehat{a})$
$\mathsf{pk_b}$
$(\mathsf{sk_b}, \mathsf{pk_b}) \leftarrow \mathsf{PKGen}(1^k)$
$\widehat{b}$
$b \leftarrow \mathsf{PKDec}(\mathsf{sk_b}, \widehat{b})$
$c \leftarrow a \wedge b$
$\mathsf{pk_c}$
$\widehat{c} \leftarrow \mathsf{PKEnc}(\mathsf{pk_c}, c)$
$\widehat{c}$
$\mathsf{pk_a}$
$\widehat{a}$
$a \leftarrow \mathsf{Dec}(\mathsf{pk_a}, \widehat{a})$
$\mathsf{pk_b}$
$\widehat{b}$
$b \leftarrow \mathsf{Dec}(\mathsf{pk_b}, \widehat{b})$
$c \leftarrow a \wedge b$
$\widehat{c} \leftarrow \mathsf{Enc}(c)$
$\mathsf{pk_c}$
$\widehat{c}$

Fig. 9. An AND gate and its transformation under $\mathsf{TR}_{\mathsf{PK}}$ and $\mathsf{TR}'_{\mathsf{PK}}$. Note how $\widehat{\texttt{AND}}$ conducts 3 instances of the wire protocols.

pk upstream; the upstream opaque components then sends the encryption of the plaintext under pk. This is illustrated in Figure 9 (center).

Lastly: the $\widehat{\texttt{encoder}}$ gadgets consists of a leak-free component which receives a plaintext a as input and a public key pk from the downstream component, and sends $\mathsf{PKEnc}(pk, a)$ to the downstream component. The $\widehat{\texttt{decoder}}$ gadget consists of a leak-free component which generates a key pair $(\mathsf{sk}, \mathsf{pk})$, sends pk upstream, receives a ciphertext $\widehat{a}$ from upstream, and outputs $\mathsf{PKDec}(sk, \widehat{a})$.

Soundness of $\mathsf{TR}_{\mathsf{PK}}$ follows trivially from the soundness of the encryption scheme.

Theorem 3. *The circuit transformation* $\mathsf{TR}_{\mathsf{PK}}$ *is* $(\mathsf{P}, \tau(k), \tau(k) + poly(k), q(k), negl(k))$*-secure for any polynomials* τ,q *and circuits of size* $poly(k)$.

Proof (Proof sketch). We define an artificial circuit transformation $\mathsf{TR}'_{\mathsf{PK}}$ which is secure iff $\mathsf{TR}_{\mathsf{PK}}$ is secure. This $\mathsf{TR}'_{\mathsf{PK}}$ is similar to $\mathsf{TR}_{\mathsf{PK}}$, except that the wire protocol is replaced by a non-interactive one, using the following encoding scheme $\Pi = (\mathsf{Enc}, \mathsf{Dec})$ (see Figure 9). The upstream component transmits each value a using the encoding procedure $\mathsf{Enc}(a)$ which outputs $(\mathsf{pk}, \widehat{a})$, where $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{PKGen}(1^k)$ and $\widehat{a} \leftarrow \mathsf{PKEnc}(pk, a)$; the secret key sk is discarded. The downstream component does not have the decryption key; instead, it recovers a by running $\mathsf{Dec}(\mathsf{pk}, \widehat{a})$, which recovers and outputs a by brute force search.[21] In $\mathsf{TR}'_{\mathsf{PK}}$, $\widehat{\texttt{encoder}}$ and $\widehat{\texttt{decoder}}$ are opaque component that simply run Enc and Dec respectively.

Note that the circuit transformations $\mathsf{TR}_{\mathsf{PK}}$ and $\mathsf{TR}'_{\mathsf{PK}}$ are indeed secure with identical parameters, since their wire assignments have identical distributions (the only difference is in the direction by which the public keys are sent between opaque components). In $\mathsf{TR}'_{\mathsf{PK}}$ all

[21] The inefficiency of the opaque components used by $\mathsf{TR}'_{\mathsf{PK}}$ is irrelevant to the security proof of $\mathsf{TR}_{\mathsf{PK}}$.

gadgets (except $\widehat{\texttt{encoder}}$ and $\widehat{\texttt{decoder}}$) are rerandomizing. Also, by the IND-CPA security of the encryption scheme, these gadgets are $(\mathsf{P}, \mathrm{poly}(k), \mathit{negl}(k))$-reconstructible by the distribution of depth-0 circuits that simply output a precomputed sample from $\mathsf{Enc}(0)$. By Lemma 11, the claim follows. □

7 On the Necessity of Opaque Gates

Our constructions reduce the leakage-resilience of arbitrary circuits to that of simple "opaque gate" components, and Section 6.2 provides significant leeway for the realization of these components. But can large opaque gates be completely eliminated? Alas, as shown next, large opaque gates are necessary for secure and sound transformation of arbitrary circuits, if the transformation's security follows from (or implies) reconstructibility. Moreover, we conjecture that large opaque gates are necessary for some "black-box" constructions of transformers.

7.1 Necessity of opaque gates for reconstruction

We begin by showing that large opaque gates are necessary when using our proof technique, and more generally, in any (sound) circuit transformation in which the transformed circuits contains *some* part which is (shallowly) reconstructible. Otherwise, for any circuit C, there exists a shallow circuit C' that computes the same function — which, for suitable parameters, is obviously false.

Let us first give a general lemma relating the parameters of C' to those of the transformation and the reconstructor.

Notation. In the following, we assume for simplicity that all gates are binary, i.e., $\mathcal{K} = \{0, 1\}$. Let $\mathsf{DS}(d, s)$ denote the class of functions computable by unlimited fan-in Boolean circuits of depth d and size s. For a set of (probabilistic) gates G, define the class of functions that verify gates in G as $\mathcal{V}_G = \{v_g | g \in G\}$, where $v_g(x, y) = 1$ iff the output y is *plausible* for g on input x (i.e., $\Pr[g(x) = y] > 0$).

Lemma 12. *Let* TR *be any circuit transformer, which output circuits* $\widehat{C}$ *using gate set* G*. Suppose* $\widehat{C}$ *is of the form* $\widehat{C} = \widehat{D} \circ\ \widehat{C}' \circ \widehat{E}$*, where* $\widehat{E}$ *is an "encoder" in some function class* $\mathcal{L}_{\widehat{E}}$*,* $\widehat{C}'$ *is a rerandomizing"core" circuit, and* $\widehat{D}$ *is any "decoder" circuit.*

Suppose that for some deterministic single-output circuit C*, function class* $\mathcal{L}_{\mathsf{REC}_{\widehat{C}}}$ *and distinguishing advantage* $\epsilon < 1$*, the core* $\widehat{C}'$ *of* $\widehat{C} \leftarrow \mathsf{TR}(C)$ *is* $(\mathcal{L}_1, O(1), \epsilon)$*-reconstructible by* $\mathcal{L}_{\mathsf{REC}_{\widehat{C}'}}$*. Then the function computed by* C *lies in the class* $\{\texttt{AND}_s\} \circ (s \times \mathcal{V}_G) \circ \mathcal{L}_{\mathsf{REC}_{\widehat{C}}} \circ \mathcal{L}_{\widehat{E}}$*.*

Here, $\mathcal{L}_1 = \{\texttt{AND}_s\} \circ (s \times \mathcal{V}_G)$*, where* s *is the size of* C *and* $\texttt{AND}_s$ *is the* $\texttt{AND}$ *gate of fan-in* s*.*

Proof. We shall show that the function computed by C is also computed by C', defined as follows. Intuitively, C' will ask $\mathcal{L}_{\mathsf{REC}_{\widehat{C}'}}$ to reconstruct the internal wires of C while forcing the output to (an encoding of) 0, and then verify reconstructor's output. If 0 is indeed the correct output then verification will succeed (with high probability), otherwise it must fail.

First, think of C' as drawn from the following distribution over circuits. For a random string r and drawing $R_{\widehat{C'}} \leftarrow \mathsf{REC}_{\widehat{C'}}$, on input x:

1. Encode the inputs: $X \leftarrow \widehat{E}(x)$ using the randomness r.
2. Let Y be an arbitrary (fixed) string that decodes to 0, i.e., $\widehat{D}(Y) = 0$.
3. Compute $W \leftarrow R_{\widehat{C'}}(X, Y)$; then set the input wires in W to X, and the output wires to Y.
4. For each gate g in $\widehat{C}'$, verify (using $v_g \in \mathcal{V}_G$) that the the wires connected to g in the assignment W are plausible for g.
5. If all tests in the previous step succeeded, output 0. Otherwise output 1.

Correctness: if $C(x) = 0$ then (X, Y) is plausible for $\widehat{C}'$ (since $\widehat{C}'$ is rerandomizing) and thus, by the definition of the reconstructor $\mathsf{REC}_{\widehat{C'}}$, the wire assignment distribution W is $(\mathcal{L}_1, O(1), \epsilon)$-leakage-indistinguishable from $\mathcal{W}_{\widehat{C'}}(X|Y)$. Since steps 4+5 are in $\mathcal{L}_1$, the verification will pass with probability at least $1-\epsilon$ (otherwise steps 4+5 form a distinguisher between W and $\mathcal{W}_{\widehat{C'}}(X|Y)$). Thus, C' outputs 0 with probability at least $1 - \epsilon$.

Conversely, if $C'(x)$ outputs 0 then W is plausible for every gate g in $\widehat{C}'$. Thus, W has non-zero probability in the honest wire assignment distribution $\mathcal{W}_{\widehat{C'}}(X)$, since the probabilistic gates in $\widehat{C}'$ use independent randomness.[22] The output wires in W are Y, so by the soundness of TR, this means $C(x_1, \ldots, x_{k_\mathrm{I}}) = \mathsf{Dec}(Y) = 0$.

We have thus shown that the randomly drawn C' computes the same function as C with probability at least $1 - \epsilon$. Fixing the best random choice of r and $R_{\widehat{C'}}$, we get as specific circuit C' that computes the same function as C. Lastly, note that C' indeed lies in the class $\{\mathtt{AND}_s\} \circ (s \times \mathcal{V}_G) \circ \mathcal{L}_{\mathsf{REC}_{\widehat{C}}} \circ \mathcal{L}_{\widehat{E}}$. □

In particular, this means that transformed circuits that have constant-depth reconstructors and constant-depth encoders must use large opaque gates:[23]

Lemma 13. *Let* TR *be any circuit transformer (with any* $\epsilon < 1$*), which output circuits* $\widehat{C}$ *using gate set* G*. Suppose* $\widehat{C}$ *is of the form* $\widehat{C} = \widehat{D} \circ \widehat{C}' \circ \widehat{E}$*, where* $\widehat{E} \in \mathsf{AC}^0$ *is an "encoder" circuit,* $\widehat{C}'$ *is a rerandomizing "core" circuit, and* $\widehat{D}$ *is any "decoder" circuit. Then at least one of the following holds:*

1. Transformed circuits use arbitrarily large opaque gates*, i.e.,*
The gate set G *is infinite.*

[22] The fact that different gates are independent is implicit in the very definition of "gate"; if they were dependent, then functions in $\mathcal{V}_G$ would not be able to meaningfully verify correctness of the circuit.

[23] Indeed, for the unconditionally-secure circuit transformation shown in Section 6.1, to maintain the level of security one has to increase t (the output size of the opaque gate $\mathcal{O}$) logarithmically with the size of the transformed circuit C.

2. Transformed circuits are not reconstructible by AC^0*, i.e.,*
There exist circuits C such that for $\widehat{C} \leftarrow \mathsf{TR}(C)$, the core circuit $\widehat{C}'$ is not $(\mathcal{L}_1, O(1), \epsilon)$-reconstructible by AC^0*.*
Here, $\mathcal{L}_1 = \{\mathtt{AND}_s\} \circ (s \times \mathcal{V}_G)$, where s is the size of C and $\mathtt{AND}_s$ is the `AND` *gate of fan-in s.*

Proof. Suppose condition 2 is false. Then for any circuit C computing a function f_C, we can invoke Lemma 12 with $\mathcal{L}_{\mathsf{REC}_{\widehat{C}}} = \mathsf{AC}^0$ to show $f_C \in \{\mathtt{AND}_s\} \circ (s \times \mathcal{V}_G) \circ \mathsf{AC}^0 \circ \mathsf{AC}^0$.

If condition 1 is violated, the class $\mathcal{V}_G$ is finite and thus has constant-depth circuits; hence so does $(s \times \mathcal{V}_G)$. It follows that $f_C \in \mathsf{DS}(O(1), O(s))$. But letting f_C be the parity function and letting C be a circuit that computes f_C using a XOR-tree, this implies $f_C \in AC^0$ which (for sufficiently large k_{I}) contradicts the circuit lower bound of [14][1][17]. □

Note that the result holds even for very bad transformers that allow a distinguishing advantage ϵ that's arbitrarily close to 1. Also, note that result holds even when the decoding procedure has arbitrary high complexity.

More generally, we can relax the assumptions on the depth of the encoder and reconstructor circuits, and also allow the gadget set G to grow with the number of inputs. We show that if the output of the transformer has reconstructors then at least one of the { encoder, gate set verifier, or reconstructor } classes requires circuits that are deep or large. Note that if the encoder is deep or large then the transformed circuit is inefficient; if the gate set cannot be efficiently verified then it contains complicated opaque gates; and if the reconstructor is deep or large, then the security reduction is inefficient and results in low security.

Lemma 14. *Let $d_G, s_G, d_{\widehat{E}}, s_{\widehat{E}}, d_R, s_R$ be some integer functions of k_{I} (the size of the input of C).*

Let TR *be any circuit transformer (with any $\epsilon < 1$), which output circuits $\widehat{C}$ using gate set G that can be verified in $\mathcal{V}_G \subseteq \mathsf{DS}(d_G, s_G)$. Suppose $\widehat{C}$ is of the form $\widehat{C} = \widehat{D} \circ\ \widehat{C}' \circ \widehat{E}$, where $\widehat{E} \in \mathsf{DS}(d_{\widehat{E}}, s_{\widehat{E}})$ is an "encoder" circuit, $\widehat{C}'$ is a rerandomizing "core" circuit, and $\widehat{D}$ is any "decoder" circuit.*

Suppose that for any deterministic single-output circuit C the core $\widehat{C}'$ of $\widehat{C} \leftarrow \mathsf{TR}(C)$ is $(\mathcal{L}_1, O(1), \epsilon)$-reconstructible by $\mathsf{DS}(d_R, s_R)$ where $\mathcal{L}_1 = \{\mathtt{AND}_s\} \circ (s \times \mathcal{V}_G)$.

Then $k_{\mathrm{I}} s_G + s_R + s_{\widehat{E}} > 2^{{k_{\mathrm{I}}}^{\Omega(1/2d)}}$ where $d = d_G + d_{\widehat{E}} + d_R$.

Proof. By the lemma's hypothesis, for any circuit C computing a function f_C, we can invoke Lemma 12 with $\mathcal{L}_{\mathsf{REC}_{\widehat{C}}} = \mathsf{DS}(d_R, s_R)$ and $\mathcal{L}_{\widehat{E}} = \mathsf{DS}(d_{\widehat{E}}, s_{\widehat{E}})$ to get

$$\begin{aligned} f_C &\in \{\mathtt{AND}_s\} \circ (s \times \mathcal{V}_G) \circ \mathcal{L}_{\mathsf{REC}_{\widehat{C}}} \circ \mathcal{L}_{\widehat{E}} \\ &= \{\mathtt{AND}_s\} \circ \mathsf{DS}(d_G, s_G) \circ \mathsf{DS}(d_R, s_R) \circ \mathsf{DS}(d_{\widehat{E}}, s_{\widehat{E}}) \qquad (12) \\ &= \mathsf{DS}(d_G + d_R + d_{\widehat{E}} + 1, s \cdot s_G + s_R + s_{\widehat{E}} + 1)\ . \end{aligned}$$

As shown by Hastad [17], for any depth d and infinitely many k_{I}, there exists an k_{I}-input Boolean function $f_d^{k_{\mathrm{I}}}$ that has small circuits of depth $d+1$:

$$f_d^{k_{\mathrm{I}}} \in \mathsf{DS}(d+1, O(k_{\mathrm{I}}))$$

but requires exponential size for circuits of depth d:

$$f_d^{k_{\mathrm{I}}} \notin \mathsf{DS}(d, 2^{\Omega({k_{\mathrm{I}}}^{1/2d})}) \ .$$

Let C be a size-$O(k_{\mathrm{I}})$ depth-d circuit that computes $f_{d+1}^{k_{\mathrm{I}}}$. Then by (12),

$$f_C \in \mathsf{DS}(d_G + d_R + d_{\widehat{E}} + 1, O(k_{\mathrm{I}}) s_G + s_R + s_{\widehat{E}})$$

yet setting $d = d_G + d_{\widehat{E}} + d_R + 1$ we get

$$f_C \notin \mathsf{DS}(d_G + d_{\widehat{E}} + d_R + 1, 2^{\Omega({k_{\mathrm{I}}}^{1/2d})}) \ .$$

Hence $O(k_{\mathrm{I}}) s_G + s_R + s_{\widehat{E}} > 2^{\Omega({k_{\mathrm{I}}}^{1/2d})}$. The claim follows.

□

Note that Lemma 13 follows as a special case of Lemma 14 by setting s_G, d_G to $O(1)$ (because a finite gadget set G can be verified by a constant set of circuits), setting $d_R, d_{\widehat{E}}$ to $O(1)$, and setting $s_R, s_{\widehat{E}}$ to ${k_{\mathrm{I}}}^{O(1)}$.

When d approaches $\log(k_{\mathrm{I}})$, Lemma 14 no longer gives a meaningful bound on s. However, Lemma 12 does yield (by the same technique) much stronger conditional lower bounds. The following lemma implies, for example, that if there exists a transformer for which the encoder, gate set verifier and reconstructor all have polylogarithmic-depth polynomial-size circuits, then all of $\mathsf{P/poly}$ has such circuits (and in particular $\mathsf{P/poly} = \mathsf{AC}$ and the AC hierarchy collapses), which would be a very surprising, and non-relativizing [26], complexity-theoretic result.

Lemma 15. *Consider a function $\xi : \mathbb{N} \to \mathbb{N}$. Let TR be any circuit transformer (with any $\epsilon < 1$), which output circuits $\widehat{C}$ using gate set G that can be verified in $\mathcal{V}_G \subseteq \mathsf{DS}(\xi(k_{\mathrm{I}}), {k_{\mathrm{I}}}^{O(1)})$. Suppose $\widehat{C}$ is of the form $\widehat{C} = \widehat{D} \circ \ \widehat{C}' \circ \widehat{E}$, where $\widehat{E} \in \mathsf{DS}(\xi(k_{\mathrm{I}}), {k_{\mathrm{I}}}^{O(1)})$ and $\widehat{C}'$ is rerandomizing.*

Suppose that for any deterministic single-output circuit C the core $\widehat{C}'$ of $\widehat{C} \leftarrow \mathsf{TR}(C)$ is $(\mathcal{L}_1, O(1), \epsilon)$-reconstructible by $\mathsf{DS}(\xi(k_{\mathrm{I}}), {k_{\mathrm{I}}}^{O(1)})$ where $\mathcal{L}_1 = \{\mathtt{AND}_s\} \circ (s \times \mathcal{V}_G)$. Then $\mathsf{P/poly}$ has circuits of depth $3\xi(n) + 1$ and polynomial size.

7.2 Necessity of reconstruction

The above lower bounds leave a major loophole: does secure circuit transformation require circuits whose core is reconstructible and rerandomizing? Indeed the construction of [18] evades our lower bounds: it gives a circuit transformation that is $(\mathcal{L}_{\mathsf{ISW}[t]}, \infty, \mathrm{poly}(ts), \infty, 0)$-secure (where $\mathcal{L}_{\mathsf{ISW}[t]}$ contains functions that directly output t of their inputs, and s is the circuit size), using just AND and NOT gates. The lower bound of Lemma 13 is avoided since for their AND gadget there do not exist shallow reconstructors (indeed, reconstructing their AND gadget requires solving a system of t linear equations). Their proof avoids the need for reconstructors by having a simulator that uses leakage function f (chosen by the observer) in a non-blackbox manner: it inspects f to see what are the wires of $\widehat{C}$ that f reads, and simulates just these few wires. This is possible because $\mathcal{L}_\tau$ has trivial computational power and does not even access most of its input.

However, we conjecture that this reconstructor-free approach cannot be significantly extended to larger leakage classes. For more complicated leakage functions that access the whole wire assignment (i.e., that are not spatially local), one runs into the following difficulty: even simple leakage classes like NC^0 can implement strong cryptographic functionality [4], so it is hard to imagine simulators that handle leakage functions given by nontrivial circuits in any way other than simply invoking them in a black-box fashion. Such invocation requires supplying the leakage function with a full wire assignment to $\widehat{C}$, which is exactly the role of reconstructors. Specifically, we conjecture the following:

Conjecture 1 (Necessity of reconstructibility (informal)). For nontrivial leakage classes $\mathcal{L}$, and for any encoding-based circuit transformation whose security against $\mathcal{L}$ is provable by a tight "black-box reduction" to the leakage-indistinguishability of the encoding scheme, the "core" of the transformed circuit is reconstructible by efficient circuits.

Here, "black-box reduction" means that the reduction (from distinguishing encodings to distinguishing $\mathsf{Exp}^{\mathrm{real}}_{\mathsf{TR}}$ from $\mathsf{Exp}^{\mathrm{sim}}_{\mathsf{TR}}$ in Definition 1) uses black-box access to the latter distinguisher, and that the simulator SIM in Definition 1) uses black-box access to OBS and the leakage function f that OBS generates.

Combining the necessity results of Section 7.1 with the above conjecture, we expect that opaque gates are inherent to "natural" circuit transformer constructions.

Lastly, note that Lemma 11 (on reconstructor composition) also implies lower bounds: if a secure encoding-based circuit transformer TR uses gadgets that happen to be reconstructible and rerandomizing, then the transformed circuits are rerandomizing and reconstructible, and thus (by Section 7.1) TR must use large large leak-free gates.

We note that these results and conjectures apply only to transformers that are perfectly sound, and conjecture that the bounds can be circumvented if the transformation has imperfect soundness.

Acknowledgments: We thank Yuval Ishai for discussion on [18] and pointing out the result in [11]. We thank Tal Rabin and Vinod Vaikuntanathan valuable discussions and, in particular, for sharing the draft of [34]. Thanks also to Ran Raz and Debajyoti Bera for discussions on circuit lower bounds, and Ronen Shaltiel for discussions on alternative constructions.

Sebastian Faust was supported in part by Microsoft Research through its PhD Scholarship Programme, by the IAP Programme P6/26 BCRYPT of the Belgian State (Belgian Science Policy), and FWO grant G.0225.07. Leonid Reyzin was supported in part by NSF grants CNS-0546614 and CNS-0831281. Eran Tromer was supported by NSF CyberTrust grant CNS-0808907 and AFRL grant FA8750-08-1-0088. Views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of AFRL, NSF, the U.S. Government or any of its agencies.

References

1. M. Ajtai, *σ_1^1 formulae on finite structures*, Annals of Pure and Applied Logic **24** (1983), 1–48.
2. Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan, *Simultaneous hardcore bits and cryptography against memory attacks*, TCC, 2009, pp. 474–495.
3. Joel Alwen, Yevgeniy Dodis, and Daniel Wichs, *Leakage resilient public-key cryptography in the bounded retrieval model*, Advances in Cryptology — CRYPTO 2009, 2009, to appear.
4. Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz, *Cryptography in NC^0*, FOCS '04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (Washington, DC, USA), IEEE Computer Society, 2004, pp. 166–175.
5. Daniel J. Bernstein, *Cache-timing attacks on AES*, `http://cr.yp.to/papers.html#cachetiming`, 2005.
6. G.R. Blakley, *Safeguarding cryptographic keys*, **48** (1979), 313–317.
7. Raphael Bousso, *The holographic principle*, Reviews of Modern Physics **74** (2002), 825.
8. Mark Braverman, *Poly-logarithmic independence fools AC0 circuits*, Tech. Report TR09-011, ECCC, 2009.
9. David Brumley and Dan Boneh, *Remote timing attacks are practical*, Comput. Netw. **48** (2005), no. 5, 701–716.
10. Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett, *On cryptography with auxiliary input*, STOC '09: Proceedings of the 41st annual ACM symposium on Theory of computing (New York, NY, USA), ACM, 2009, pp. 621–630.
11. Bella Dubrov and Yuval Ishai, *On the randomness complexity of efficient sampling*, STOC '06: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing (New York, NY, USA), ACM, 2006, pp. 711–720.
12. Stefan Dziembowski and Krzysztof Pietrzak, *Leakage-resilient cryptography*, Foundations of Computer Science, Annual IEEE Symposium on **0** (2008), 293–302.
13. Sebastian Faust, Eike Kiltz, Krzysztof Pietrzak, and Guy Rothblum, *Leakage-resilient signatures*, Cryptology ePrint Archive, Report 2009/282, 2009, `http://eprint.iacr.org/2009/282`.
14. Merrick Furst, James B. Saxe, and Michael Sipser, *Parity, circuits, and the polynomial-time hierarchy*, SFCS '81: Proceedings of the 22nd Annual Symposium on Foundations of Computer Science (Washington, DC, USA), IEEE Computer Society, 1981, pp. 260–270.
15. Oded Goldreich, *Towards a theory of software protection and simulation by oblivious rams*, STOC, 1987, pp. 182–194.
16. Oded Goldreich and Rafail Ostrovsky, *Software protection and simulation on oblivious rams*, J. ACM **43** (1996), no. 3, 431–473.
17. Johan Hastad, *Almost optimal lower bounds for small depth circuits*, Symposium on the Theory Of Computing, 1986.
18. Yuval Ishai, Amit Sahai, and David Wagner, *Private circuits: Securing hardware against probing attacks*, CRYPTO'03, 2003, revised and abbreviated version of [19], pp. 463–481.

19. ———, *Private circuits: Securing hardware against probing attacks*, unpublished manuscript ([18] is a revised and abbreviated version), 2003.
20. Jonathan Katz, *Signature schemes with bounded leakage resilience*, Cryptology ePrint Archive, Report 2009/220, 2009, `http://eprint.iacr.org/2009/220`.
21. Adam Klivans, *On the derandomization of constant depth circuits*, APPROX '01/RANDOM '01, Springer-Verlag, 2001, pp. 249–260.
22. Paul C. Kocher, *Design and validation strategies for obtaining assurance in countermeasures to power analysis and related attacks.*, NIST Physical Security Workshop, 2005.
23. Paul C. Kocher, Joshua Jaffe, and Benjamin Jun, *Differential power analysis*, CRYPTO, 1999, pp. 388–397.
24. Markus G. Kuhn, *Compromising emanations: eavesdropping risks of computer displays*, Ph.D. thesis, University of Cambridge, 2003, Technical Report UCAM-CL-TR-577.
25. Silvio Micali and Leonid Reyzin, *Physically observable cryptography (extended abstract)*, TCC'04, 2004, pp. 278–296.
26. Peter Bro Miltersen, *Circuit depth relative to a random oracle*, Information Processing Letters **42** (1992), no. 6, 295–298.
27. Gil Segev Moni Naor, *Public-key cryptosystems resilient to key leakage*, CRYPTO, 2009, to appear.
28. Noam Nisan, *Pseudorandom bits for constant depth circuits*, Combinatorica **11** (1991), no. 1, 63–70.
29. Dag Arne Osvik, Adi Shamir, and Eran Tromer, *Cache attacks and countermeasures: The case of AES*, CT-RSA, 2006, pp. 1–20.
30. Colin Percival, *Cache missing for fun and profit*, presented at BSDCan 2005, Ottawa, 2005; see `http://www.daemonology.net/hyperthreading-considered-harmful`, 2005.
31. Christophe Petit, François-Xavier Standaert, Olivier Pereira, Tal Malkin, and Moti Yung, *A block cipher based pseudo random number generator secure against side-channel key recovery*, ASIACCS, 2008, pp. 56–65.
32. Krzysztof Pietrzak, *A leakage-resilient mode of operation*, EUROCRYPT, 2009, pp. 462–482.
33. Jean-Jacques Quisquater and David Samyde, *Electromagnetic analysis (EMA): Measures and countermeasures for smart cards*, E-smart, 2001, pp. 200–210.
34. Tal Rabin and Vinod Vaikuntanathan, *Cryptographic defenses against noisy side-channel attacks*, unpublished manuscript, May 2009.
35. Alexander Razborov, *Lower bounds for the size of circuits of bounded depth with basis and, xor*, Math. Notes of the Academy of Science of the USSR 41, 1987, pp. 333–338.
36. Kai Schramm and Christof Paar, *Higher order masking of the AES*, CT-RSA, 2006, pp. 208–225.
37. Adi Shamir, *How to share a secret*, Communications of the ACM **22** (1979), no. 11, 612–613.
38. Adi Shamir and Eran Tromer, *Acoustic cryptanalysis: on nosy people and noisy machines*, presented at the Eurocrypt 2004 rump session; see `http://tromer.org/acoustic`, 2004.
39. Roman Smolensky, *Algebraic methods in the theory of lower bounds for boolean circuit complexity*, STOC, 1987, pp. 77–82.
40. L. Smolin, *The strong and weak holographic principles*, Nuclear Physics B **601** (7 May 2001), 209–247.
41. François-Xavier Standaert, Tal Malkin, and Moti Yung, *A unified framework for the analysis of side-channel key recovery attacks*, EUROCRYPT, 2009, pp. 443–461.
42. Francois-Xavier Standaert, Olivier Pereira, Yu Yu, Jean-Jacques Quisquater, Moti Yung, and Elisabeth Oswald, *Leakage resilient cryptography in practice*, Cryptology ePrint Archive, Report 2009/341, 2009, `http://eprint.iacr.org/2009/341`.
43. L. Susskind, *The world as a hologram*, Journal of Mathematical Physics **36** (1995), 6377–6396.
44. Gerard 't Hooft, *Dimensional reduction in quantum gravity*, Salamfestschrift: A Collection of Talks, Conference on Highlights of Particle and Condensed Matter Physics (SALAMFEST), World Scientific, 1993, p. 284.
45. Jason Waddle and David Wagner, *Towards efficient second-order power analysis*, CHES, 2004, pp. 1–15.