Bounded-Communication Leakage Resilience via Parity-Resilient Circuits*

Vipul Goyal† Yuval Ishai‡ Hemanta K. Maji§ Amit Sahai¶
Alexander A. Sherstov‖

Abstract

We consider the problem of distributing a computation between two parties, such that any bounded-communication leakage function applied to the local views of the two parties reveals essentially nothing about the input. This problem can be motivated by the goal of outsourcing computations on sensitive data to two servers in the cloud, where both servers can be simultaneously corrupted by viruses that have a limited communication bandwidth.

We present a simple and efficient reduction of the above problem to that of constructing *parity-resilient circuits*, namely circuits that map an encoded input to an encoded output so that the parity of any subset of the wires is essentially independent of the input. We then construct parity-resilient circuits from circuits that are resilient to *local* leakage, which can in turn be obtained from protocols for secure multiparty computation. Our main reduction builds on a novel generalization of the "ε-biased masking lemma" that applies to interactive protocols.

Applying the above, we obtain two-party protocols with resilience to bounded-communication leakage either in the information-theoretic setting, relying on random oblivious transfer correlations, or in the computational setting, relying on non-committing encryption which can be based on a variety of standard cryptographic assumptions.

Keywords: Leakage-resilient cryptography, communication complexity, ε-biased masking.

*A preliminary version of this paper appears in the proceedings of FOCS 2016.

†Microsoft Research, India. vipul.goyal@gmail.com.

‡Department of Computer Science, Technion and University of California, Los Angeles. yuval.ishai@gmail.com.

§Department of Computer Science, Purdue University. hmaji@purdue.edu. Work done in part at the University of California, Los Angeles.

¶Department of Computer Science, University of California, Los Angeles. amitsahai@gmail.com.

‖Department of Computer Science, University of California, Los Angeles. sherstov@cs.ucla.edu.

Contents

1 Introduction

The goal of *leakage-resilient cryptography* is to maintain the traditional guarantees of cryptography even when partial information about internal secrets can be leaked. A central theme of research in this area is that of securing *general computations* against leakage. Originating from [ISW03, MR04, FRR+10], this goal has been pursued in a variety of computational models and with different types of natural leakage functions.

In this work we focus on securing general computations against leakage in the following simple scenario. Suppose that a client wishes to outsource the computation of a function f on a sensitive input x to the cloud. The client trusts the cloud servers to perform the computation correctly, but would like to ensure that no information about x is revealed. A direct solution is to have the client encrypt x using fully homomorphic encryption (FHE) [Gen09], and have a cloud server compute f on the encrypted input. However, FHE is currently still quite far from being practical, its existence relies on a relatively narrow class of cryptographic assumptions related to the intractability of lattice problems, and its fully compact form requires an ad-hoc circular security assumption.

A potentially more efficient alternative approach is to distribute the computation of f between two non-colluding servers. That is, the client starts by secret-sharing x between the two servers. This step should be done very efficiently, e.g., in quasi-linear time, and should be independent of the function f. Given a description of f, the servers then engage in an interactive secure two-party computation protocol [Yao82, GMW87] for evaluating the shares of $y = f(x)$ from the shares of x. Finally, the servers send the shares of y back to the client, who reconstructs the output. Again, the final reconstruction step should be efficient and independent of f.

In addition to not requiring the use of FHE, such two-server solutions offer several other advantages over the single-server solution: They can minimize the amount of work performed by the client (eliminating expensive "cryptographic" computations), they can provide information-theoretic security given correlated randomness between the servers that can be set up before the input x is known, and they do not require the client to maintain a secret state for reconstructing the output. The latter feature is useful for accommodating more general scenarios in which inputs originate from and/or outputs are delivered to multiple clients.

Bounded-communication leakage. If such a two-server solution is implemented using standard protocols for secure two-party computation, the client is protected against any *single* corrupted server. The question we consider is that of providing a meaningful protection even when the two servers are *simultaneously* corrupted. We envision a scenario where both servers are infected by cooperating viruses, but the viruses are subject to a bound on the total number of bits they can communicate with each other. We refer to this type of leakage as *bounded-communication leakage* (BCL). Our goal is to design BCL-resilient two-party protocols, namely ones that allow the servers to interactively compute shares of $f(x)$ from shares of x while completely hiding x from the viruses. See Figure 1 for an illustration of this motivating scenario.

Note that the standard notion of security for two-party protocols coincides with BCL-resilience when the communication bound is 0 (i.e., the viruses cannot communicate at all). However, the security guarantees of standard protocols for secure two-party computation break down if even a small amount of leakage is allowed [BCH12]. In particular, both "GMW-style" protocols and "Yao-style" protocols effectively generate a simple secret-sharing of all intermediate values of the computation,

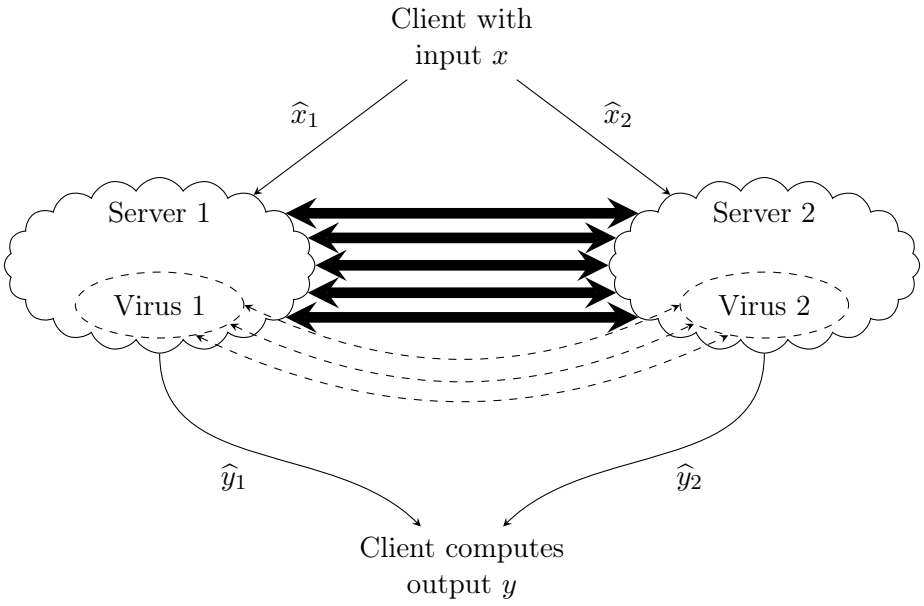

Figure 1: Motivating Example.

and any such intermediate value can be recovered using a single bit of leakage regardless of the amount of communication required to compute this value given the inputs alone.

Other than the restriction on the total amount of communication between the viruses, we do not impose any other restriction on the way they may interact. For instance, the viruses may first store their local views of the entire protocol execution, and only then run a multi-round (bounded-communication) protocol for recovering information about the secret input x from their local views. We do assume, however, that the viruses are *passive* in the sense that they do not tamper with the messages sent by the infected servers during the protocol execution.

The BCL assumption can be justified by the existence of virus detection mechanisms that make it difficult for the viruses to directly communicate with each other or to alter the messages sent by the servers without being detected. Still, the viruses can communicate at a slow rate, e.g., by carefully controlling the timing of the messages. Earlier works that impose bounds on the communication of adversarial parties include [Dzi06, DLW06, DP07, DNW08].

A very different motivating scenario for the BCL assumption is that of protecting the computation performed by the servers against unintentional leakage of partial information resulting from the computation process itself. This type of leakage is captured by the "only computation leaks" (OCL) assumption put forward in the influential work of Micali and Reyzin [MR04]. The OCL assumption is typically motivated by side-channel leakage in hardware implementations, where the servers correspond to different hardware components. The BCL assumption we consider is less restrictive in that it can apply globally to the entire views of the servers throughout the protocol execution, regardless of the way in which computations are carried out. Making the stronger OCL assumption does not seem to make the problem considerably easier. On the other hand, unlike previous works on the OCL model, in this work we focus on a simple *single-execution* setting, where only one (stateless) computation is being performed, as opposed to the more challenging *continuous leakage*

setting in which a sequence of computations with a common secret state are subject to leakage. The following comparison captures the best adaptations of previous results to our simpler model, ignoring FHE-based solutions [GR10, JV10] that are trivialized in our single-execution setting.

The first goal of the present work is to study the feasibility of BCL-resilient computation.

> *Under which cryptographic assumptions or setup assumptions can general computations be protected against bounded-communication leakage?*

The prior state of the art can be summarized as follows. In the information-theoretic setting, a construction of Dziembowski and Faust [DF12] (the "DF-construction" for short) provides unconditional security by employing leak-free hardware components whose size must inherently grow with both the leakage bound and the statistical security parameter. Concretely, each hardware component samples a random pair of orthogonal vectors that are distributed to the two servers. The security of the construction breaks down if the entire output of any component is leaked. While some form of setup seems necessary in the information-theoretic setting even without leakage,[1] the DF-construction leaves open the possibility of using *constant-size* (or "finite") hardware components, namely ones whose size does not depend on the leakage bound or the statistical security parameter.

The breakthrough work of Goldwasser and Rothblum [GR12] and subsequent variants of Bitansky et al. [BDL14] show that information-theoretic OCL and BCL security is possible even without any setup. However, these protocols require a large number of servers, whereas in this work we focus on 2-server solutions.

In the computational security model, Dachman-Soled et al. [DLZ15] showed how to instantiate the hardware components of the DF-construction in the plain model by using a strong form of deniable encryption [CDNO97], whose only known instantiations rely on indistinguishability obfuscation (iO) [GGH+13, SW14]. The possibility of computational solutions in the plain model that do not rely on FHE or iO remained open.

In addition to the feasibility questions, we will be interested in the achievable *leakage rate*, measured as the ratio between the leakage communication bound c and the size S of the circuits required for implementing the protocol for f. (This captures the fractional information leakage about internal computation steps.) We will also be interested in the *computational overhead* of protocols, measured as the ratio between S and the circuit size of f, denoted by s.

> *What are the best achievable leakage rate and computational overhead of BCL-resilient protocols?*

In previous two-party solutions that do not rely on FHE the leakage rate is worse than $1/cs$, which is inherited from the parameters of the DF-construction (see Table 1 in [DLZ15]). Moreover, the computational overhead of these solutions is at least quadratic in c. The latter holds also for protocols from [GR12, BDL14] that involve more than two servers.

[1] Indeed, all known approaches for information-theoretic secure two-party computation using correlated randomness require the entropy of the correlated randomness to be bigger than the circuit size, except for the extreme case of exponential-size circuits [BIKK14]. Thus, the small amount of correlated randomness provided by the client's messages is unlikely to be sufficient.

1.1 Our Results

We introduce a simple and general technique for constructing and analyzing BCL-resilient protocols. Our technique yields efficient protocols that achieve information-theoretic security using a minimal setup, or alternatively provide computational security under a variety of standard assumptions.

The BCL-resilient protocols we construct can be obtained from any standard protocol for secure multiparty computation (MPC), where the security threshold of the MPC protocol serves as a leakage communication bound in the BCL-resilient protocol. Alternatively, they can be obtained from any *multi-server* OCL-resilient protocol (such as the one from [GR12]), where the statistical error of the multi-server protocol determines the leakage bound of the two-server protocol.

More concretely, we obtain the following new results on BCL-resilient protocols.

- Feasibility: information-theoretic setting. We construct (2-server) BCL-resilient protocols with information-theoretic security using only an oblivious transfer (OT) setup. That is, the servers either have access to an OT oracle or to constant-size leak-free hardware components that produce OT correlations.[2] As discussed above, this is the best one can hope for barring a major breakthrough in information-theoretic cryptography. This should be contrasted with the hardware components required by the DF-construction, whose size should grow with both the leakage bound and the security parameter.

- Feasibility: computational setting. We obtain the first 2-server computational BCL-resilient protocols (in the plain model) that do not rely on FHE or iO. Concretely, our construction only requires the use of *non-committing encryption* (NCE) [CFGN96], which can be based on a variety of standard cryptographic assumptions that include the intractability of factoring Blum integers [CDMW09] or Decisional Diffie Hellman [DN00]. These instantiations make a crucial use of the simple setup of our information-theoretic protocols.

- Leakage rate and computational overhead. Our BCL-resilient protocols, in both settings, offer qualitative improvements in leakage rate and efficiency over previous protocols. Recall that in the information-theoretic setting, the 2-server DF-construction [DF12] the leakage bound is inherently smaller than the size of the trusted hardware compontents. Moreover, previous solutions in both settings [DF12, GR12, BDL14, DLZ15] involve multiplicative computational overhead on the server side which is bigger than the leakage bound. Our protocols get around these limitations. In particular, whenever f can be computed by a circuit whose size s and depth h satisfy the mild conditions $s \geq c^2$ and $s \geq h^2$, where c is the leakage communication bound, we can tolerate $\tilde{\Omega}(\sqrt{s})$ bits of leakage with $\log^{O(1)}(c)$ computational overhead using either constant-size hardware (alternatively, OT-correlations) or NCE.

  More generally, if $f\colon \{0,1\}^n \to \{0,1\}^n$ is computable by a circuit of size s and depth h, then we get BCL-resilient protocols with security against c bits of leakage where the client can be implemented by circuits of size $\widetilde{O}(n+c)$, and where the servers can be implemented by circuits of size $\widetilde{O}(s+ch+c^2)$ in the information-theoretic protocols, or alternatively invoke an NCE protocol a similar number of times in the computational protocols. These parameters are inherited from known honest-majority MPC protocols [DIK10]. The information-theoretic

[2] An OT correlation delivers a pair of random bits (s_0, s_1) to one server and the pair (b, s_b) to the other, where b is a random bit.

protocols can be efficiently implemented by having the client distribute the instances of the OT correlation to the servers in an offline preprocessing phase, before the input x is known.[3]

The above results are based on two technical ingredients that may be considered as independently interesting results: (1) the construction of so-called *parity-resilient circuits*, a natural computational analogue of small-bias generators [NN90]; and (2) a generalization of the so-called *ε-biased masking* lemma [DS05a] that applies to interactive protocols. These are described in the next section.

1.2 Overview of Techniques

We start by observing what goes wrong when trying to evaluate f using a very simple protocol for secure two-party computation, namely the "GMW protocol" [GMW87, Gol04] for passive adversaries when implemented over ideal OT. In this protocol, the local views of the two parties essentially form an *additive secret sharing* of the gate values in a circuit computing f. That is, letting z denote the vector of gate values on the input x, the joint views are distributed as $(z \oplus r, r)$ where r is a uniformly random bit-string.

This protocol miserably fails in achieving our goal: any intermediate value z_i in the computation of f can be revealed by leaking only a single bit from a local view. Moreover, by the $\mathbb{F}_2$-linearity of the secret sharing of z, revealing an arbitrary *parity* of bits from a local view reveals the corresponding parity of the bits of z.

The first idea is that in order to protect the protocol against such simple parity attacks, it suffices to protect the computation of f via a *parity-resilient circuit*: a randomized circuit that receives a randomized encoding $\widehat{x}$ of an input x and produces an encoded output $\widehat{y}$, where the parity of any subset of the gate values (when evaluating the circuit on $\widehat{x}$) reveals essentially nothing about x. A bit more precisely, the circuit is ε-parity-resilient if for any inputs x, x', the distributions of the gate values $z(\widehat{x})$ and $z(\widehat{x}')$ are ε-indistinguishable by parity functions.

Using such a parity-resilient circuit, we can easily obtain a protocol with resilience to a *single bit of parity leakage*. The client locally encodes the input x, and additively shares the encoding $\widehat{x}$ between the two servers. The servers now run the GMW protocol to obtain additive shares of an output encoding $\widehat{y}$. The shares of $\widehat{y}$ are sent back to the client, who decodes y.

Constructing parity-resilient circuits. We turn to the question of constructing parity-resilient circuits. Despite this being a very natural object, we are not aware of any previous related study. Our first observation is that any multi-party protocol which is ε-resilient to a single bit of OCL leakage per computation step, such as the protocol of [GR12], directly implies an ε-parity-resilient circuit. Indeed, arbitrarily representing each computation step by a boolean circuit, the parity of an arbitrary subset of gates can be recovered using a single bit of leakage from each computation step. However, given the complexity and the relatively poor parameters of the construction from [GR12] and its variants, we seek an alternative and more direct construction.

Our main construction of parity-resilient circuits can be based on any general MPC protocol that

[3] Note that the DF-construction can also be implemented in this setting by having the client distribute (large) instances of an inner product correlation instead of OT correlation. However, our protocol has the efficiency advantages discussed above.

offers security against k passively corrupted parties. The construction is broken into several modular steps, where the first two steps mimic previous constructions of small-bias PRGs from bounded independence [MST03, IKOS09]. The first step is a transformation of a k-secure MPC protocol into a k-private circuit, namely a circuit with the same syntax as parity-resilient circuits, except that the joint values of any k gates should reveal nothing about x. The transformation, pointed out in [ISW03], is straightforward: let f' denote a randomized function which maps a secret-shared input for f to a secret-shared output for f. Then, a k-secure MPC protocol for f' can be directly implemented as a k-private circuit. The second step replaces each atomic gate in the k-private circuit by a constant-size ("finite") randomized gadget that maps a non-linear encoding of the inputs to a non-linear encoding of the output. The existence of a gadget with the required properties follows by a probabilistic argument (cf. [DDV10]), however we also give a simple explicit construction of a gadget $g : \{0,1\}^9 \to \{0,1\}^3$.

The combination of the above two steps gives a construction of $2^{-\Omega(k)}$-parity-resilient circuits over the finite gadget, where the size of a parity resilient-circuit for f is linear in the circuit size of the k-secure MPC protocol for f'. Somewhat surprisingly, turning a circuit over g to a circuit over a standard binary gates is not as straightforward as it may seem. In particular, a naive implementation of g will compromise parity resilience. Instead, we will consider for now a natural generalization of the GMW protocol that works over g-gates instead of binary gates by using a finite two-party oracle H instead of the standard OT oracle. When applied to a parity-resilient circuit over g, this protocol generates a pair of views distributed as $(z \oplus r, r)$, where z is a parity-resilient encoding of the input. The protocol still offers resilience to a single bit of parity-leakage.

From parities to bounded communication via generalized ε-biased masking. Our next, and most technical, step is arguing that parity-resilience *automatically* implies general BCL-resilience with related parameters. Concretely, we prove the following theorem. Suppose that μ_0 and μ_1 are two distributions that are ε-indistinguishable by parities. Then any interactive two-party protocol with c bits of communication can have at most an $\varepsilon \cdot 2^{c/2}$ advantage in distinguishing between a secret sharing of μ_0 and μ_1, namely between the distributions $(\mu_0 \oplus r, r)$ and $(\mu_1 \oplus r, r)$, where r is a random bit-string chosen independently of μ_0, μ_1. This means that as long as $\varepsilon \ll 2^{-c/2}$, an ε-parity-resilient circuit (combined with the GMW protocol) offers good BCL-resilience against c bits of leakage. Together with the above, we get a transformation from k-secure MPC protocols to BCL-resilient protocols over a finite two-party oracle H that tolerate $\Omega(k)$ bits of leakage.

Specializing the above theorem to 1-message protocols and for the case where one of the two distributions is uniform, the theorem can be shown to be essentially equivalent to the so-called ε-*biased masking* lemma from [DS05a]. The ε-biased masking lemma says that $M \oplus X$, where M is a high entropy source and S is an independent small-bias distribution, is statistically close to uniform. See Appendix C for a proof of the equivalence. Our theorem is more general in two orthogonal ways: it applies to multi-round protocols, and it extends pseudorandomness to indistinguishability. In light of the usefulness of the original ε-biased masking lemma in cryptography (see, e.g., [DS05b, AIK08, FS08, IKOS09]) we expect our generalized version to find additional applications.

Wrapping up. The above is almost enough to get our results for the information-theoretic model. The only remaining step is to replace the finite oracle H by a standard OT oracle, or alternatively an OT correlation. (The latter protocols directly imply parity-resilient circuits over the binary basis.) This step follows by observing that a simple deterministic reduction from H to OT re-

spects BCL-resilience. More generally, to respect BCL-resilience we need such reductions to satisfy a strong notion of security we refer to as *joint simulation security*. This notion, previously considered in [DLZ15], strengthens the standard simulation-based definition of secure computation by considering the outputs of the two simulators *jointly*. To make this possible, the two simulators share a common source of randomness. Finally, we obtain our results for the computational model by observing that the standard construction of OT from NCE is also secure under the strong joint simulation requirement.

These final steps crucially rely on the constant-size setup feature of our information-theoretic protocols. Indeed, the setup required previous DF-construction could only be instantiated in the plain model using iO [DLZ15]. This qualitative difference between our constant-size setup and the computationally simple inner-product setup required by DF may seem surprising. However, natural attempts to realize the DF setup via standard protocols for secure two-party computation (even ones that offer adaptive security [CFGN96]) fail. This can be attributed to the fact, already discussed above, that applying a low-communication leakage attack to standard secure computation protocols [Yao82, GMW87] reveals intermediate values of the computation that cannot be computed via a low-communication protocol.

More broadly, obtaining "leakage tolerant" forms of information-theoretic protocols for functions with a super-polynomial input domain appears to be a difficult task even for simple functions [BGI+14]. We bypass this problem by using constant-size oracles, whose brute-force secure evaluation using a truth-table representation is trivially leakage tolerant.

See Figure 2 for a roadmap of the different steps of our construction and the relations between different types of leakage-resilient objects and Section 2 for the relevant definitions and formal theorem statements.

1.3 Open Problems

Our work gives rise to two natural open questions: extending the results from the single-execution setting to the more challenging continuous leakage setting, and obtaining similar information-theoretic results in a multi-party setting without a setup (reproducing the result of [GR12] with better parameters). We believe that the ideas introduced in this work will serve as a useful basis for such extensions.

2 Outline of Definitions and Theorems

2.1 Definitions

We consider boolean circuits over a basis $\mathbb{B}$ of gates. Each gate in $\mathbb{B}$ computes a function of the form $g : \{0,1\}^\alpha \to \{0,1\}^\beta$. When the basis is not specified, it is understood to be the full binary basis $\mathbb{B} = \{\mathsf{AND}, \mathsf{OR}, \mathsf{NOT}, \mathsf{XOR}\}$ where $\mathsf{AND}, \mathsf{OR}, \mathsf{XOR}$ gates have fan-in 2. We will consider both deterministic and randomized circuits (a circuit is deterministic by default). We let $|C|$ denote the number of gates in C, also including inputs and randomness gates.

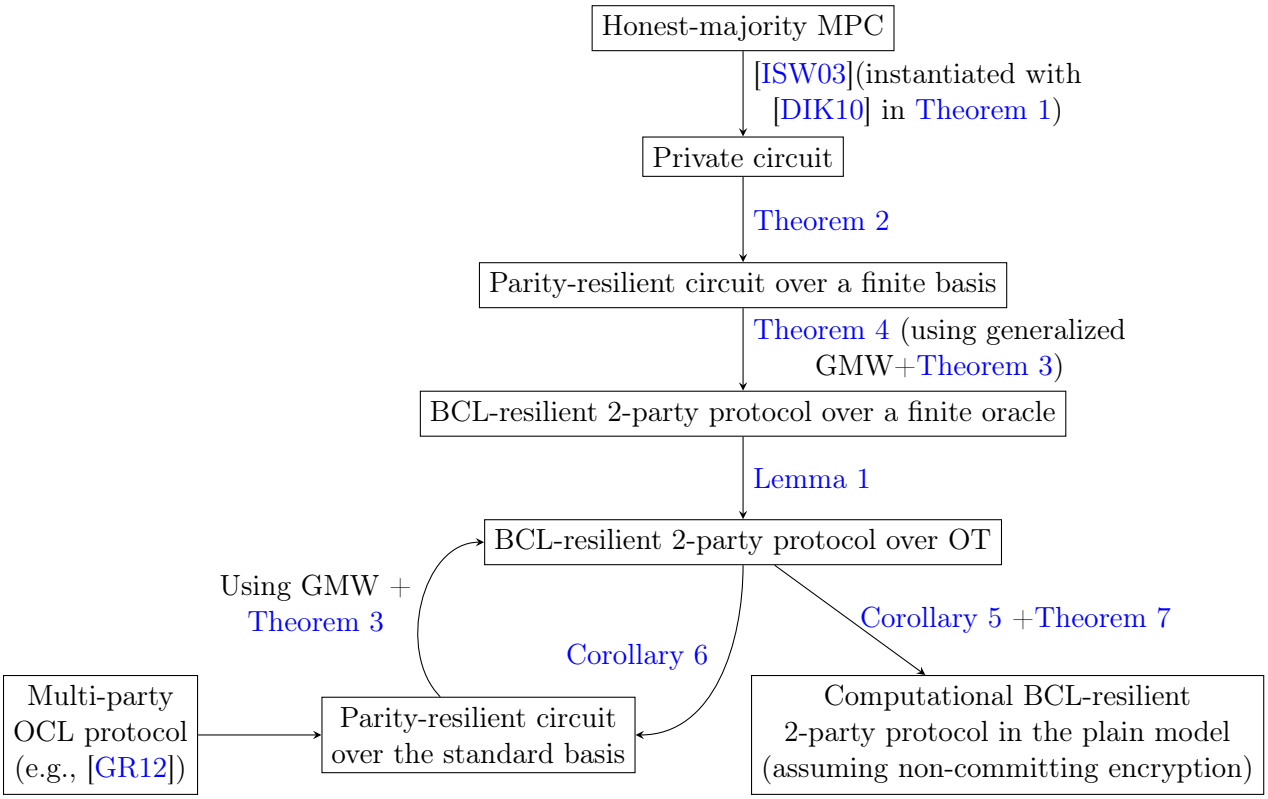

Figure 2: Relations between different notions of leakage-resilient circuits and protocols.

We consider a simple model for leakage-resilient circuits that generalizes the stateless variant of private circuits from [ISW03] (see also [FRR+10, BIVW15]). Such circuits map an encoded input for a function f into an encoded output, where the internal gate values of C should hide the input in the presence of partial leakage.

Definition 1 (Leakage-resilient circuits and k-private circuits)**.** *For $f\colon \{0,1\}^{n_i} \to \{0,1\}^{n_o}$, a leakage-resilient circuit for f is defined by (I, C, O), where:*

- $I\colon \{0,1\}^{n_i} \to \{0,1\}^{\widehat{n}_i}$ *is a randomized* input encoder *circuit, which maps an input x to an encoded input $\widehat{x}$;*
- *C is a randomized circuit, mapping an encoded input $\widehat{x} \in \{0,1\}^{\widehat{n}_i}$ to an encoded output $\widehat{y} \in \{0,1\}^{\widehat{n}_o}$;*
- $O\colon \{0,1\}^{\widehat{n}_o} \to \{0,1\}^{n_o}$ *is a deterministic* output decoder *circuit, which maps an encoded output $\widehat{y}$ to an output y.*

We say that (I, C, O) is an (L, ε)-leakage-resilient implementation of f*, for a leakage function $L\colon \{0,1\}^{|C|} \to \{0,1\}^*$ and $\varepsilon \geqslant 0$, if the following requirements hold:*

- Correctness*: For any input $x \in \{0,1\}^{n_i}$, we have $\Pr[O(C(I(x))) = f(x)] = 1$, where the probability is over the randomness of I and C;*
- Leakage-resilience*: For any $x, x' \in \{0,1\}^{n_i}$, the statistical distance between the distributions $L(C[I(x)])$ and $L(C[I(x')])$ is at most ε, where $C[\widehat{x}]$ denotes the joint distribution of the $|C|$ gate values in the computation of C on input $\widehat{x}$.*

For a class $\mathcal{L}$ of leakage functions, we say that (I, C, O) is an $(\mathcal{L}, \varepsilon)$-leakage-resilient implementation of f*, if it is an (L, ε)-leakage-resilient implementation of f for all $L \in \mathcal{L}$. We say that (I, C, O) is a* k-private implementation of f *if it is an $(\mathcal{L}, 0)$-leakage-resilient implementation of f for the class $\mathcal{L}$ of all k-bit projection functions (which output k fixed entries of the input).*

Without any requirements on I and O, the above definition can be met by having I compute a leakage-resilient secret sharing of the input that is passed by C directly to the output decoder. The decoder decodes the circuit output and computes f. To rule out such a solution, we require the encoder and the decoder to be *universal* (i.e., depend only on n_i, n_o and the circuit size of f but not on f itself). Furthermore, we would like the encoder and decoder size to be considerably smaller than the circuit size f. These requirements effectively force C to perform the bulk of the computation in a leakage-resilient manner.

The following bound on the efficiency of k-private circuits follows by implementing a secure multiparty computation protocol from [DIK10] as a circuit, via the general transformation suggested in [ISW03]. The protocol achieves security against k semi-honest parties by using $n = O(k)$ parties. The private circuit is obtained by applying the protocol to the function which maps a secret sharing of the input for f to a secret sharing of the output of f. In the private circuits model, the secret sharing of the input is implemented by the input encoder and the reconstruction of the output from its shares is implemented by the output decoder.

Theorem 1 (Implicit in [DIK10])**.** *There is an efficient algorithm Q such that for every positive integer k and every circuit C_f of size s and depth h computing a function $f\colon \{0,1\}^{n_i} \to \{0,1\}^{n_o}$, the output of $Q(1^k, C_f)$ is a k-private implementation (I, C, O) of f with $|I| = \widetilde{O}(n_i + k)$, $|C| = \widetilde{O}(s + kh + k^2)$ and $|O| = \widetilde{O}(n_o + k)$.*

We will be particularly interested in the following special case of leakage-resilient circuits.

Definition 2 (Parity-resilient circuits)**.** *We say that (I, C, O) is an ε-*parity-resilient implementation *of f if it is an $(\mathcal{L}, \varepsilon)$-leakage-resilient implementation of f for the class $\mathcal{L}$ of all parity functions, namely the class of functions that output the parity of a subset of the wires.*

The main goal of this work is to construct two-party protocols that are resilient to bounded-communication leakage. We consider two-party protocols that start with encoded inputs and end with encoded outputs. Furthermore, we also consider protocols that receive correlated random inputs, or alternatively invoke a two-argument function as an oracle.

Definition 3 (Two-party protocol with encoded input and output)**.** *A two-party protocol for $f\colon \{0,1\}^{n_i} \to \{0,1\}^{n_o}$ is defined by $\Pi = (I, (R_1, R_2), (M_1, M_2), O)$, where:*

- *$I\colon \{0,1\}^{n_i} \to \{0,1\}^{\widehat{n}_i} \times \{0,1\}^{\widehat{n}_i}$ is a randomized* input encoder *circuit, which maps an input x for f to a pair of protocol inputs $(\widehat{x}_1, \widehat{x}_2)$.*
- *R_1 and R_2 are distributions over $\{0,1\}^{n_r}$ that capture the random inputs of the two parties. They are assumed to be uniform and independent by default. Otherwise, we say that Π uses* correlated randomness *(R_1, R_2).*
- *M_1 and M_2 are deterministic* next message functions*, where M_j determines the next message to be sent by party j as a function of its input $\widehat{x}_j$, random input r_j, and the sequence of previous messages received from the other party. Messages are sent in rounds, where party 1 sends messages in odd rounds and party 2 in even rounds. After a predetermined number of rounds, the function M_j returns a local output $\widehat{y}_j \in \{0,1\}^{\widehat{n}_o}$ for party j.*
- *$O\colon \{0,1\}^{\widehat{n}_o} \times \{0,1\}^{\widehat{n}_o} \to \{0,1\}^{n_o}$ is a deterministic* output decoder *circuit, which maps a pair of protocol outputs $(\widehat{y}_1, \widehat{y}_2)$ to an output y of f.*

For $x \in \{0,1\}^{n_i}$, we denote by $\Pi(x)$ the output of π on input x, namely the result of applying the input encoder I to x, interacting as specified by $(R_1, R_2), (M_1, M_2)$, and applying the output decoder O to the pair of protocol outputs. We say that Π correctly *computes $f\colon \{0,1\}^{n_i} \to \{0,1\}^{n_o}$ if for every input $x \in \{0,1\}^{n_i}$ we have $\Pr[\Pi(x) = f(x)] = 1$.*

We denote by $\mathsf{view}(x)$ the joint distribution $(\mathsf{view}_1, \mathsf{view}_2)$ obtained by running Π on input x, where view_j includes the encoded input $\widehat{x}_j$, the random input r_j (sampled from R_j), and the sequence of messages received *by party j. (The messages sent by party j as well as its output $\widehat{y}_j$ are uniquely determined by view_j.) We denote by $|\Pi|$ the total circuit size required for implementing all invocations of the next message functions, including the input and randomness gates.*

Finally, we also consider oracle-aided protocols*, where the parties can invoke a two-party oracle $H : \{0,1\}^{\alpha_1} \times \{0,1\}^{\alpha_2} \to \{0,1\}^{\beta}$. After receiving an input from each party, the oracle delivers the*

output to party 2. In an oracle-aided protocol, the outputs of the next message function M_j also include messages sent by party j as inputs to H and the inputs of M_2 include messages received by party 2 as outputs of H. The oracle may be invoked several times in parallel, where each party can send inputs to several invocations of H in a single round.

We now define our main notion of bounded-communication leakage resilience.

Definition 4 (BCL-resilient protocol)**.** *We say that Π is a (c, ε)-*bounded-communication leakage resilient protocol for f *(or (c, ε)-BCL-resilient for short) if Π correctly computes f, and the following security requirement holds. For any communication protocol $\pi : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}$ with communication complexity at most c, and any pair of inputs $x, x' \in \{0,1\}^{n_i}$ we have $|\Pr[\pi(\mathsf{view}(x)) = 1] - \Pr[\pi(\mathsf{view}(x')) = 1]| \leq \varepsilon$, where $n_j = |\mathsf{view}_j|$.*

For a polynomial-time computable $f : \{0,1\}^* \to \{0,1\}^*$, the above definition can be naturally extended to capture *computational* BCL-resilient protocols for f. Such a protocol is specified by PPT algorithms $\Pi = (I, (M_1, M_2), O)$. We say that Π is computationally $c(n)$-BCL-resilient if for every input length n it is $(c(n), \varepsilon(n))$-BCL-resilient for some negligible function ε, with respect to every leakage protocol π that is implemented by circuits of size $\mathsf{poly}(n)$.

2.2 Main Theorems

Following are the main theorems and corollaries that correspond to the steps of the construction described in Section 1.2. See Figure 2 for a roadmap. Transformations between different objects are always polynomial-time computable, even when we do not say so explicitly.

Theorem 2 (Private circuits ⇒ parity-resilient circuits over a finite basis)**.** *There is a function $g\colon \{0,1\}^9 \to \{0,1\}^3$ such that every k-private implementation (I, C, O) of a function f can be efficiently transformed into a $2^{-\Omega(k)}$-parity-resilient implementation (I', C', O') of f over the basis $\mathbb{B} = \{g\}$, with $|I'| = O(|I|)$, $|C'| = O(|C|)$, and $|O'| = O(|O|)$.*

Theorem 3 (Generalized ε-biased masking)**.** *Let μ_0, μ_1 be probability distributions over $\{0,1\}^n$ that are ε-indistinguishable by parities. Then any communication protocol $\pi : \{0,1\}^n \times \{0,1\}^n \mapsto \{0,1\}$ with communication complexity at most c obeys:*

$$\left| \mathop{\mathbb{E}}_{x \sim \mu_0} \mathop{\mathbb{E}}_{r \xleftarrow{\$} \{0,1\}^n} [\pi(x \oplus r, r)] - \mathop{\mathbb{E}}_{x \sim \mu_1} \mathop{\mathbb{E}}_{r \xleftarrow{\$} \{0,1\}^n} [\pi(x \oplus r, r)] \right| \leqslant 2^{c/2}\varepsilon$$

Theorem 4 (Parity-resilient circuits ⇒ BCL-resilient oracle-aided protocols)**.** *Suppose (I', C', O') is a 2^{-k}-parity-resilient implementation of f over a basis $\mathbb{B} = \{g\}$, where $g : \{0,1\}^\alpha \to \{0,1\}^\beta$, and where C' has depth h. Then there is a (c, ε)-BCL-resilient $O(h)$-round two-party protocol $\Pi = (I'', (R_1, R_2), (M_1, M_2), O'')$ for f using an oracle $H : \{0,1\}^{\alpha+\beta} \times \{0,1\}^\alpha \to \{0,1\}^\beta$, with $c = \Omega(k)$, $\varepsilon = 2^{-\Omega(k)}$, $|I''| = O(|I'|)$, $|R_1| = \beta \cdot |C'|$, $|R_2| = 0$, $|O''| = O(|O'|)$, and $|\Pi| = O(|C'|)$.*

Lemma 1 (Finite oracle ⇒ OT oracle)**.** *For any positive integers $\alpha_1, \alpha_2, \beta$, a (c, ε)-BCL-resilient H-aided protocol Π for f, where $H : \{0,1\}^{\alpha_1} \times \{0,1\}^{\alpha_2} \to \{0,1\}^\beta$, can be efficiently transformed to a similar OT-aided protocol Π' for f, where Π' has the same input encoder and output decoder as Π, and where $|\Pi|' \leq 2^{\alpha_2} \cdot \beta \cdot |\Pi|$.*

Corollary 5 (BCL-resilient protocols over OT)**.** *For every positive integer k and circuit C_f of size s and depth h computing a function $f\colon \{0,1\}^{n_i} \to \{0,1\}^{n_o}$, there is a $(k, 2^{-k})$-BCL-resilient OT-aided protocol $\Pi = (I, (R_1, R_2), (M_1, M_2), O)$ for f, where $|\Pi| = \widetilde{O}(s + kh + k^2)$, $|I| = \widetilde{O}(n_i + k)$, and $|O| = \widetilde{O}(n_o + k)$, and where the OT oracle is called $\widetilde{O}(s + kh + k^2)$ times. Alternatively, there is a similar protocol that uses independent instances of OT correlation instead of an OT oracle.*

Corollary 6 (Parity-resilient circuits over a binary basis)**.** *For every positive integer k and circuit C_f of size s and depth h computing a function $f\colon \{0,1\}^{n_i} \to \{0,1\}^{n_o}$, there is a 2^{-k}-parity-resilient implementation (I, C, O) of f over the full binary basis, with $|I| = \widetilde{O}(n_i + k)$, $|C| = \widetilde{O}(s + kh + k^2)$, and $|O| = \widetilde{O}(n_o + k)$.*

Theorem 7 (Computational BCL-resilient protocols in the plain model)**.** *Suppose the DDH assumption holds. Then, for every polynomial-time computable $f : \{0,1\}^n \to \{0,1\}^{m(n)}$ and polynomial $c(n)$, there is a computational $c(n)$-BCL-resilient implementation $\Pi = (I, (M_1, M_2), O)$ of f, where the running time of I is $\widetilde{O}(n + c(n))$ and the running time of O is $\widetilde{O}(m(n) + c(n))$.*

3 Private Circuits imply Parity Resilience

In this section, we shall prove the following theorem:

Theorem 2 Restated (Private circuits ⇒ parity-resilient circuits over a finite basis)**.** *There is a function $g\colon \{0,1\}^9 \to \{0,1\}^3$ such that every k-private implementation (I, C, O) of a function f can be efficiently transformed into a $2^{-\Omega(k)}$-parity-resilient implementation (I', C', O') of f over the basis $\mathbb{B} = \{g\}$, with $|I'| = O(|I|)$, $|C'| = O(|C|)$, and $|O'| = O(|O|)$.*

Recall that a k-private implementation (I, C, O) of f has the property that (the joint distribution of) the values of any k wires of C is independent of the input x. We transform the k-private implementation into a parity-resilient implementation (I', C', O') by encoding *each* wire of C using a small-bias encoding of constant block-length. In more detail, the circuit C' mimics the computation of C while maintaining the small-bias encoding of each wire of C. To emulate the computation of a gate g in C, the circuit C' takes as input the encodings of the respective input wires and additional fresh randomness to produce the encoding of the output wire of g. We start by assuming that this entire procedure is *atomically* performed by a finite-size gadget. In this case, parity resilience is ensured because the parity of a small number of wires in C' is independent of the input x, due to k-privacy guarantee of C. On the other hand, a parity of a large number of wires in C' is close to uniform because it XORs a large number of independent small bias distributions (this crucially relies on the fact that the encoding of each wire of C was created using independent random bits). We note, however, that because the finite-size gadget is deterministic and receives its randomness as an external input, we need the small-bias property to hold even with respect to parities that involve the random inputs.

To formally present our transformation, we need to define small-bias distributions and encodings with small bias.

3.1 Small-bias Encoding

For a subset $S \subseteq [n]$, the character $\chi_S \colon \{0,1\}^n \to \{0,1\}$ is defined as $\chi_S(x) := \oplus_{i \in S} x_i$. Note that $\chi_\emptyset(x) = 0$ for all $x \in \{0,1\}^n$, and there are a total of 2^n different possible characters.

Definition 5 (Small-bias Distribution)**.** *A distribution D over the sample space $\{0,1\}^n$ is* ε-biased *if for all non-empty subsets $S \subseteq [n]$, the following condition is satisfied.*

$$\left| \Pr_{x \sim D}[\chi_S(x) = 0] - \Pr_{x \sim D}[\chi_S(x) = 1] \right| \leqslant \varepsilon$$

For $n = 1$, if a distribution has small bias then it is close to the uniform random bit. And, if its bias is less than 1, then the outcome of the distribution is unpredictable. More generally, any linear test cannot distinguish an ε-biased distribution from the uniform distribution over n-bits.

Definition 6 ((Strong) Small-bias Generator)**.** *A function $G \colon \{0,1\}^m \to \{0,1\}^n$ is an* ε-bias generator *if the distribution $G(U_m)$ is ε-biased, where U_m is the uniform distribution over m-bit strings. The generator G is a* strong ε-bias generator *if the distribution $(U_m, G(U_m))$ over $\{0,1\}^{m+n}$ is ε-biased.*

Next, we define small-biased encodings. Let $\mathsf{Enc} \colon \{0,1\} \times \{0,1\}^{c'} \to \{0,1\}^c$ be a function. For $x \in \{0,1\}$, the *encoding of* x is defined by $(r, \mathsf{Enc}(x; r))$, where $r \sim U_{c'}$. Let $\mathsf{Enc}[x]$ represent the corresponding joint distribution, i.e. the distribution $(U_{c'}, \mathsf{Enc}(x; U_{c'}))$ over $\{0,1\}^{c'+c}$. The function Enc is a *valid encoding function*, if $\mathsf{Supp}(\mathsf{Enc}(0; U_{c'}))$ is disjoint from $\mathsf{Supp}(\mathsf{Enc}(1; U_{c'}))$.

The corresponding (canonical) decoding function $\mathsf{Dec} \colon \cup_{x \in \{0,1\}} \mathsf{Supp}(\mathsf{Enc}(x; U_{c'})) \to \{0,1\}$ is defined as follows: $\mathsf{Dec}(\widehat{x}) = x \in \{0,1\}$, where $\widehat{x} \in \mathsf{Supp}(\mathsf{Enc}(x; U_{c'}))$.

Definition 7 (Small-bias Encoding)**.** *A valid encoding function $\mathsf{Enc} \colon \{0,1\} \times \{0,1\}^{c'} \to \{0,1\}^c$ is an ε-biased encoding if, for any $x \in \{0,1\}$, the function $\mathsf{Enc}(x; \cdot)$ is a strong ε-bias generator.*

For $t \in \mathbb{N}$, let $\mathsf{Enc} \colon \{0,1\} \times \{0,1\}^{(2t+1)} \to \{0,1\}^{(2t+1)}$ be defined as follows.

$$\mathsf{Enc}(x; r_0 \dots r_{2t}) = \begin{cases} r_0\ r_1 \dots r_{2t}, & \text{if maj } \{r_0, \dots, r_{2t}\} = x \\ \overline{r_{2t}}\ \overline{r_0} \dots \overline{r_{(2t-1)}}, & \text{if maj } \{r_0, \dots, r_{2t}\} = \overline{x} \end{cases}$$

The corresponding canonical decoding function is the following.

$$\mathsf{Dec}(\widehat{x}_0, \dots, \widehat{x}_{2t}) = \text{maj } \{\widehat{x}_0, \dots, \widehat{x}_{2t}\}$$

Figure 3: Strong Small-bias Encoding Scheme.

Lemma 2 (Small-bias Encodings)**.** *There exists a valid encoding function $\mathsf{Enc} \colon \{0,1\} \times \{0,1\}^{c'} \to \{0,1\}^c$ and (constant) $\varepsilon \in (0,1)$ such that the encoding Enc is ε-biased.*

Proof. The construction provided in Figure 3 is a small-bias distribution for all $t \in \mathbb{N}$. Claim 5 proves that the bias of the distribution $\mathsf{Enc}[x]$ is $\varepsilon = 1/2 + \binom{2t}{t} 2^{-(2t+1)}$, for all $x \in \{0,1\}$. □

3.2 Small-bias Gadget

Let $\mathsf{Enc}\colon \{0,1\} \times \{0,1\}^{c'} \to \{0,1\}^{c}$ be a function that defines a valid encoding and let $\Sigma = \cup_{x\in\{0,1\}}\mathsf{Supp}(\mathsf{Enc}(x; U_{c'}))$.

Definition 8 (Gadget of a Function)**.** *Given a function $f\colon \{0,1\}^n \to \{0,1\}$, the* gadget of f using encoding function Enc *is a function $\mathcal{G}_{\mathsf{Enc},f}\colon \Sigma^n \times \{0,1\}^{c'} \to \{0,1\}^{c}$ defined as follows:*

1. *Let the input to the gadget be $(\widehat{x}_1, \ldots, \widehat{x}_n, r) \in \Sigma^n \times \{0,1\}^{c'}$*
2. *For $i \in [n]$, let $x_i = \mathsf{Dec}(\widehat{x}_i)$*
3. *Define $y = f(x_1, \ldots, x_n)$ and output $\widehat{y} = \mathsf{Enc}(y; r)$.*

Consider any $(x_1, \ldots, x_n) \in \{0,1\}^n$. Let r_i be uniformly random and $\widehat{x}_i = \mathsf{Enc}(x_i; r_i)$, for all $i \in [n]$, and r is also uniformly random. Recall that the encoding distribution $\mathsf{Enc}[x_i]$ is the joint distribution $(r_i, \widehat{x}_i)$ for uniformly random r_i, for all $i \in [n]$, and $\mathsf{Enc}[y]$ represents the joint distribution (r, y) for uniformly random r. Then $\mathcal{G}_{\mathsf{Enc},f}[x_1, \ldots, x_n]$ represents the joint distribution $(\mathsf{Enc}[x_1], \ldots, \mathsf{Enc}[x_n], \mathsf{Enc}[y])$.

Definition 9 (Small-bias Gadget)**.** *The gadget $\mathcal{G}_{\mathsf{Enc},f}$ is ε-*biased *if, for all $(x_1, \ldots, x_n) \in \{0,1\}^n$ the distribution $\mathcal{G}_{\mathsf{Enc},f}[x_1, \ldots, x_n]$ is ε-biased.*

Lemma 3. *If $\mathsf{Enc}\colon \{0,1\} \times \{0,1\}^{c'} \to \{0,1\}^{c}$ is ε-biased, then for any function $f\colon \{0,1\}^n \to \{0,1\}$, the gadget $\mathcal{G}_{\mathsf{Enc},f}$ is ε-biased.*

Proof. Fix $x_1, \ldots, x_n$ (thus, fixing y) and a non-empty subset S.

Suppose S contains some wires of $\mathsf{Enc}[x_i]$. Let T be the restriction of S to the wires in $\mathsf{Enc}[x_i]$. Then, for every fixing of $(r_1, \widehat{x}_1), \ldots, (r_{i-1}, \widehat{x}_{i-1}), (r_{i+1}, \widehat{x}_{i+1}), \ldots, (r_n, \widehat{x}_n), (r, \widehat{y})$, the bias of $\chi_S(\mathcal{G}_{\mathsf{Enc},f}[x_1, \ldots, x_n])$ is identical to the bias of $\chi_T(\mathsf{Enc}[x_i])$, which is at most ε. Averaging over all fixings, the overall bias of $\chi_S(\mathcal{G}_{\mathsf{Enc},f}[x_1, \ldots, x_n])$ is at most ε.

If S does not contain any wire of $\mathsf{Enc}[x_i]$ for any $i \in [n]$, then S must contain wires only from $\mathsf{Enc}[y]$. Then, for every fixing of $(r_1, \widehat{x}_1), \ldots, (r_n, \widehat{x}_n)$, the bias of $\chi_S(\mathcal{G}_{\mathsf{Enc},f}[x_1, \ldots, x_n])$ is identical to the bias of $\chi_S(\mathsf{Enc}[y])$, which is at most ε. Again, averaging over all fixings, the overall bias of $\chi_S(\mathcal{G}_{\mathsf{Enc},f}[x_1, \ldots, x_n])$ is at most ε. □

3.3 Proof

Suppose (I, C, O) is a k-private implementation of a function $f\colon \{0,1\}^{n_i} \to \{0,1\}^{n_o}$ given by Theorem 1 over a basis $\mathbb{B}$. In particular, we get $\mathbb{B} = \{\mathsf{NAND}\}$. Let Enc be an $(\varepsilon^* = 3/4)$-biased encoding as defined in Figure 3 when $t = 1$.

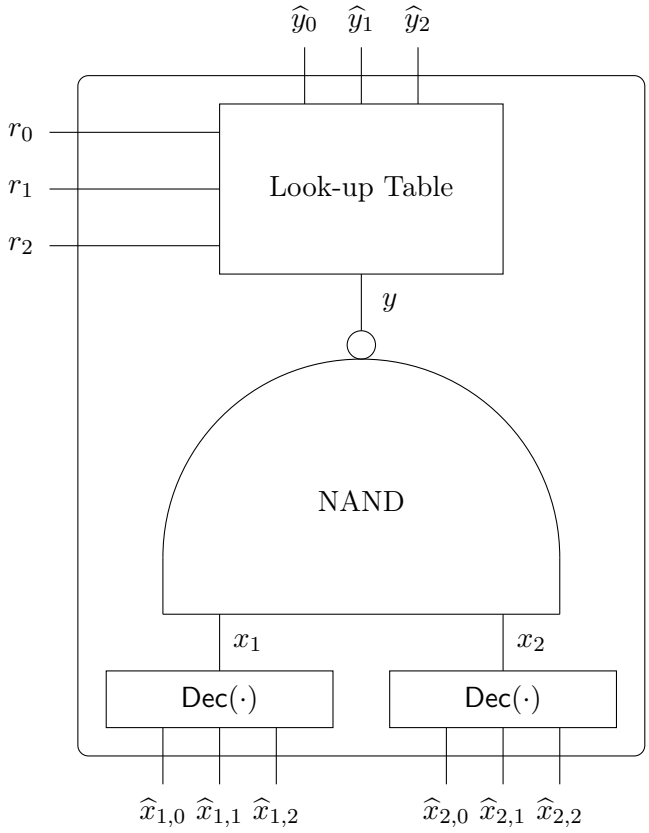

Figure 4: The NAND Gadget: $\mathcal{G}_{\mathsf{Enc},\mathsf{NAND}}$. Enc is the encoding function in Figure 3 with $t = 1$. The inputs are $\widehat{x}_1 = (\widehat{x}_{1,0}, \widehat{x}_{1,1}, \widehat{x}_{1,2})$ and $\widehat{x}_2 = (\widehat{x}_{2,0}, \widehat{x}_{2,1}, \widehat{x}_{2,2})$ and randomness $r = (r_0, r_1, r_2)$. The decoding function Dec is the majority function. Output is $\widehat{y} = (\widehat{y}_0, \widehat{y}_1, \widehat{y}_2)$. The lookup table implements the deterministic function $\widehat{y} = \mathsf{Enc}(\mathsf{NAND}(x_1, x_2)\ ;\ r)$.

Ensure:

1. (I, C, O) is a k-private implementation of f over the basis $\mathbb{B}$.
2. Let $\mathsf{Enc}\colon \{0,1\} \times \{0,1\}^{\alpha} \to \{0,1\}^{\beta}$ be an ε^*-bias encoding scheme.

Construction of (I', C', O'):

1. Let $\mathbb{B}' = \mathcal{G}_{\mathsf{Enc},\mathbb{B}} = \{\mathcal{G}_{\mathsf{Enc},g}\colon g \in \mathbb{B}\}$.
2. Definition of I': On input x, compute $I(x) \in \{0,1\}^{\widehat{n}_i}$. For each bit $\widehat{x}_k$ in $\widehat{x}$, pick α bits of random bits r_k and compute $\widehat{x}'_k = \mathsf{Enc}(\widehat{x}_k; r_k)$. Output the concatenation of these bits $(\widehat{x}'_1, \ldots, \widehat{x}'_{\widehat{n}_i}) \in \{0,1\}^{\beta\widehat{n}_i}$.
3. Definition of C': The circuit C' mimics the computation of C gate-wise as follows. Suppose C computes a function $g \in \mathbb{B}$ (say, $g = \mathsf{NAND}$) of two wires w_1 and w_2 and the output wire is w_3. In C' we will have an encoding of wire values of w_1 and w_2 using Enc and it will compute an encoding of the output wire value w_3 using $\mathcal{G}_{\mathsf{Enc},g} \in \mathbb{B}'$. Formally, let $\widehat{v}_{w_1}$ and $\widehat{v}_{w_2}$ be the encodings of values of wire w_1 and w_2 using the encoding function Enc. C' picks fresh randomness $r \xleftarrow{\$} \{0,1\}^{\alpha}$ and uses these as input to $\mathcal{G}_{\mathsf{Enc},g}$. The output is an encoding $\widehat{v}_{w_3} = \mathcal{G}_{\mathsf{Enc},g}(\widehat{v}_1, \widehat{v}_2, r)$ that corresponds to an encoding of the wire value of w_3 using Enc. The output wires of C' correspond to the encoding of each output wire in C using the encoding scheme Enc.
4. Definition of O': Given inputs $(\widehat{y}'_1, \ldots, \widehat{y}'_{\widehat{n}_o}) \in \{0,1\}^{\beta\widehat{n}_o}$, decode $\widehat{y}_k = \mathsf{Dec}(\widehat{y}'_k)$, for $k \in [\widehat{n}_o]$, and output $O(\widehat{y}'_1, \ldots, \widehat{y}'_{\widehat{n}_o})$.
5. Privacy Guarantee: This is an $(\varepsilon^*)^k$-parity-resilient implementation of f over the basis $\mathbb{B}'$.

Particular Instantiation:

1. Let (I, C, O) be the construction provided by Theorem 1.
2. Let Enc be the construction in Figure 3 with $t = 1$, $\alpha = \beta = 3$ and $\varepsilon^* = 3/4$.

Figure 5: Using a Private Implementation of f to construct a Parity-resilient Implementation of f.

Transformation. The transformation of a private implementation of f into a parity-resilient implementation of f is provided in Figure 5. Intuitively, the input encoder I' computes $I(x)$ and then encodes each of its bits with Enc. The circuit C' obtains the input of C with each bit encoded by Enc. Thereafter, C' mimics the computation of each wire of C but keeps it encoded using Enc. If C performs a g-gate computation, then C' performs a corresponding $\mathcal{G}_{\mathsf{Enc},g}$ computation. Finally, O' decodes each bit using Dec and then applies O to output y.

We will show that (I', C', O') is an ε-parity resilient implementation, where $\varepsilon = (\varepsilon^*)^k$. Let $\chi_S \in \mathcal{L}$ be a function that, on input $C'[x]$, outputs the parity of wires indexed by S. We say that a wire w in C is *touched by* S if any wire of $\mathsf{Enc}[w]$ lies in S. Let $\mathcal{T}$ be the set of all wires in C that are touched by S and $k_S = |\mathcal{T}|$ be the number of wires in C that are touched by S.

If $k_S \leqslant k$, then the output distribution $\chi_S(C'[x])$ is independent of the input x.

If $k_S > k$, then consider a fixing v of values of all wires in C when the input is x, i.e. $C[x] = v$. Consider the following equivalent technique of sampling $C'[x]$ consistent with $C[x] = v$: Encode the value corresponding to each wire $w \in C$ independently according to the distribution $\mathsf{Enc}[v(w)]$, where $v(w)$ represents the value of the wire w according to v.

Let $S(w)$ be the restriction of S to the wires in C' that are encoding of the wire $w \in C$. So, we have the following:

$$\begin{aligned}\chi_S(C'[x]) &= \sum_v \Pr[C[x] = v] \cdot \left(\chi_S(C'[x]) \mid C[x] = v \right) \\ &= \sum_v \Pr[C[x] = v] \cdot \bigoplus_{w \in \mathcal{T}} \chi_{S(w)}(\mathsf{Enc}[v(w)])\end{aligned}$$

Each random variable $\chi_{S(w)}(\mathsf{Enc}[v(w)])$, for $w \in \mathcal{T}$, is ε^*-biased for every fixing v. So, the distribution $\oplus_{w \in \mathcal{T}} \chi_{S(w)}(\mathsf{Enc}[v(w)])$ is $(\varepsilon^*)^{k_S}$-biased for every fixing v (by Claim 2). Thus, $\chi_S(C'[x])$ is $(\varepsilon^*)^{k_S}/2$ close to uniform (by Claim 3). Consequently, the statistical distance between $\chi_S(C'[x])$ and $\chi_S(C'[x'])$ is at most $(\varepsilon^*)^k$.

Note that the circuit C' uses $\mathcal{G}_{\mathsf{Enc},\mathbb{B}}$ and, in particular, $\mathcal{G}_{\mathsf{Enc},\mathsf{NAND}}$ suffices. Further, I', C' and O' can be implemented using circuits that are a constant times the size of a circuit implementing I, C and O, respectively.

4 Communication Complexity Bound

In this section, we present a generalization of the popular "Small-bias Masking Lemma" [NN90, AR94, GW97, DS05a] (refer to Appendix C) in the two-party setting.

Theorem 3 Restated (Generalized ε-biased masking)**.** *Let μ_0, μ_1 be probability distributions over $\{0,1\}^n$ that are ε-indistinguishable by parities. Then any communication protocol $\pi : \{0,1\}^n \times \{0,1\}^n \mapsto \{0,1\}$ with communication complexity at most c obeys:*

$$\left| \mathop{\mathbb{E}}_{x \sim \mu_0} \mathop{\mathbb{E}}_{r \xleftarrow{\$} \{0,1\}^n} [\pi(x \oplus r, r)] - \mathop{\mathbb{E}}_{x \sim \mu_1} \mathop{\mathbb{E}}_{r \xleftarrow{\$} \{0,1\}^n} [\pi(x \oplus r, r)] \right| \leqslant 2^{c/2}\varepsilon$$

Intuitively, the theorem states the following. Suppose we have a distribution μ_0 and μ_1 that are ε-indistinguishable from each other w.r.t. any linear test. That is, given a sample that is drawn according to the distribution μ_b (where $b \xleftarrow{\$} \{0,1\}$), any linear test can predict b with at most ε advantage.

Now, we additively secret share a sample according to the distribution μ_b, for $b \xleftarrow{\$} \{0,1\}$, among two parties, i.e. the joint views of parties is $(U, \mu_b \oplus U)$. Next, the parties run a low communication protocol (communication complexity bounded by c) among themselves. Now, given this communication, the advantage to predict b is at most $2^{c/2}\varepsilon$, i.e. at most $2^{c/2}$ times the advantage to predict b using linear tests on a sample drawn according to μ_b.

To prove this result, we will need some elementary Fourier analysis. The characters $\chi_S \colon \{0,1\}^n \to \mathbb{R}$ of the Fourier transform are given by $\chi_S(x) = (-1)^{S \cdot x}$. The Fourier coefficients of a function $f \colon \{0,1\}^n \to \mathbb{R}$ are denoted by: $\widehat{f}(S) = \frac{1}{N} \sum_{x \in \{0,1\}^n} f(x)\chi_S(x)$, where $N = 2^n$.

Definition 10 (ε-Indistinguishability by Parities)**.** *Two probability distributions μ_0, μ_1 over $\{0,1\}^n$ are called* ε-indistinguishable by parities *if, for every $S \subseteq [n]$,*

$$\left| \sum_{x \in \{0,1\}^n} \mu_0(x)\chi_S(x) - \sum_{x \in \{0,1\}^n} \mu_1(x)\chi_S(x) \right| \leqslant \varepsilon$$

Equivalently, for every $S \subseteq [n]$,

$$\left| \Pr_{x \sim \mu_0}[\chi_S(x) = 1] - \Pr_{x \sim \mu_1}[\chi_S(x) = 1] \right| \leqslant \varepsilon$$

We need to show:

$$\left| \mathop{\mathbb{E}}_{x \sim \mu_0} \mathop{\mathbb{E}}_{r \xleftarrow{\$} \{0,1\}^n} [\pi(x \oplus r, r)] - \mathop{\mathbb{E}}_{x \sim \mu_1} \mathop{\mathbb{E}}_{r \xleftarrow{\$} \{0,1\}^n} [\pi(x \oplus r, r)] \right| \leqslant 2^{c/2}\varepsilon \tag{1}$$

Let Δ stand for the left-hand side of Equation 1. Consider the matrix:

$$P = 2^{-n} \left[\mu_0(x \oplus r) - \mu_1(x \oplus r) \right]_{x, r \in \{0,1\}^n}$$

We have

$$\begin{aligned}
\Delta &= \left| \left(2^{-n} \sum_{x,r \in \{0,1\}^n} \mu_0(x)\pi(x \oplus r, r) \right) - \left(2^{-n} \sum_{x,r \in \{0,1\}^n} \mu_1(x)\pi(x \oplus r, r) \right) \right| \\
&= \left| 2^{-n} \sum_{x,r \in \{0,1\}^n} (\mu_0((x \oplus r) \oplus r) - \mu_1((x \oplus r) \oplus r)) \cdot \pi(x \oplus r, r) \right| \\
&= \left| 2^{-n} \sum_{x,r \in \{0,1\}^n} (\mu_0(x \oplus r) - \mu_1(x \oplus r)) \cdot \pi(x, r) \right| \\
&= |\langle P, \pi \rangle| ,
\end{aligned}$$

where we view π as a matrix $\pi = [\pi(x,r)]_{x,r}$. Decompose the protocol as a sum of combinatorial rectangles:

$$\pi = \sum_{i \in [2^c]} \mathbf{1}_{A_i} \mathbf{1}_{B_i}^{\mathsf{T}},$$

where $A_i, B_i \subseteq \{0,1\}^n$ are some sets and $\mathbf{1}_{A_i}$, $\mathbf{1}_{B_i}$ are their, respective, characteristic vectors. Then,

$$\begin{aligned}
\Delta &= |\langle P, \pi \rangle| \\
&\leqslant \sum_{i \in [2^c]} \left| \mathbf{1}_{A_i}^{\mathsf{T}} P \mathbf{1}_{B_i} \right| \\
&\leqslant \sum_{i \in [2^c]} \sqrt{|A_i| \cdot |B_i|}\, \|P\| \\
&\leqslant 2^{c/2} \sqrt{\sum_{i \in [2^c]} |A_i| \cdot |B_i|}\, \|P\| \\
&\leqslant 2^{c/2} 2^n \|P\| \\
&= 2^{n+c/2} \|P\|,
\end{aligned}$$

where $\|\cdot\|$ denotes the spectral norm (equivalently, the largest singular value).

Claim 1. $\|P\| \leqslant \varepsilon 2^{-n}$

Proof. The singular value decomposition of P is found as follows:

$$\begin{aligned}
P &= \left[\sum_{S \subseteq [n]} 2^{-n} \left(\widehat{\mu_0}(S) - \widehat{\mu_1}(S) \right) \cdot \chi_x(S) \chi_r(S) \right]_{x,r} \\
&= [\chi_S(x)]_{x,S} \cdot \begin{bmatrix} \ddots & & \\ & 2^{-n} \left(\widehat{\mu_0}(S) - \widehat{\mu_1}(S) \right) & \\ & & \ddots \end{bmatrix} \cdot [\chi_S(r)]_{S,r} \\
&= H \cdot \begin{bmatrix} \ddots & & \\ & 2^{-n} \left(\widehat{\mu_0}(S) - \widehat{\mu_1}(S) \right) & \\ & & \ddots \end{bmatrix} \cdot H^{\mathsf{T}},
\end{aligned}$$

where $H = 2^{-n/2} [\chi_S(x)]_{x,S}$ is a unitary matrix. In particular,

$$\|P\| = \max_{S \subseteq [n]} |\widehat{\mu_0}(S) - \widehat{\mu_1}(S)| \leqslant \varepsilon 2^{-n},$$

where the final step uses the assumption that μ_0, μ_1 are ε-indistinguishable. □

Thus, we get that $\Delta \leqslant 2^{n+c/2} \|P\| \leqslant 2^{c/2} \varepsilon$.

5 Parity Resilience implies Bounded-Communication Leakage Resilience

In this section we prove the following theorem.

Theorem 4 Restated (Parity-resilient circuits ⇒ BCL-resilient oracle-aided protocols)**.** *Suppose (I', C', O') is a 2^{-k}-parity-resilient implementation of f over a basis $\mathbb{B} = \{g\}$, where $g : \{0,1\}^{\alpha} \to \{0,1\}^{\beta}$, and where C' has depth h. Then there is a (c, ε)-BCL-resilient $O(h)$-round two-party protocol $\Pi = (I'', (R_1, R_2), (M_1, M_2), O'')$ for f using an oracle $H : \{0,1\}^{\alpha+\beta} \times \{0,1\}^{\alpha} \to \{0,1\}^{\beta}$, with $c = \Omega(k)$, $\varepsilon = 2^{-\Omega(k)}$, $|I''| = O(|I'|)$, $|R_1| = \beta \cdot |C'|$, $|R_2| = 0$, $|O''| = O(|O'|)$, and $|\Pi| = O(|C'|)$.*

Recall that a parity-resilient implementation (I', C', O') of f fools all linear tests on its wire values and it is constructed over a basis $\mathbb{B}'$. To construct a bounded-communication resilient implementation Π of f, we leverage Theorem 3. Parties begin with an additive secret sharing of the input encoding. Thereafter, parties implement a GMW-style [GMW87] protocol where the invariant that parties have additive secret shares of the wires in C' is maintained. To compute the output wires of a (possibly, randomized) gate h in C', our construction Π uses a trusted hardware that takes as input the additive shares of the inputs to h from both parties, additive secret shares of the randomness needed to compute the gate h and the additive secret share of the output of h from the first party. Then, it performs the computation of h based on these inputs and provides an additive share to the second party that is consistent with the inputs provided by both parties and the additive share of the output provided by the first party. The size of the trusted hardware that implements this deterministic computation is finite, if the oracle h is finite.

The construction is formally provided in Figure 6.

Intuitively, the functionality $\mathcal{F}(h)$, described in Figure 6, takes as input the additive secret shares (x_1, x_2) of the input x and re-computes an additive secret share of the output $y = h(x)$, where the share of party 1 is y_1. Let $\mathcal{T}_h$ represent an oracle that implements the functionality $\mathcal{F}(h)$.

Suppose (I', C', O') is an ε'-parity-resilient implementation of f over the basis $\mathbb{B}'$. Then we will construct a (c, ε)-BCL-resilient two-party protocol Π using oracles in $\mathcal{T}_{\mathbb{B}'} = \{\mathcal{T}_g : g \in \mathbb{B}'\}$, where $\varepsilon = 2^{c/2} \cdot \varepsilon'$. For example, $\mathbb{B}' = \{\mathcal{G}_{\mathsf{Enc},\mathsf{NAND}}\}$ suffices. Our construction, intuitively, additively secret shares the values of wires in C' among the parties. And to perform a g-gate evaluation in C', our protocol will invoke the finite oracle $\mathcal{T}_g$.

So, the input encoder I'', on input x, additively secret shares the output of $I'(x)$. Then, similar to the GMW [GMW87] protocol, parties recursively maintain the additive secret-shares of wire-values of C' among themselves. Corresponding to a g-gate evaluation in C', our protocol invokes $\mathcal{T}_g$ to obtain the additive secret-shares of the output wires of g. Finally, the output decoder O'' adds the shares of the two parties and uses the decoder O' to compute the output y.

Note that the joint distribution $(\mathsf{view}_1, \mathsf{view}_2)$ is identically distributed as $(U, U + C'[x])$ and, applying Theorem 3, we get that $\pi(\mathsf{view}[x])$ and $\pi(\mathsf{view}[x'])$ have statistical distance at most $2^{c/2} \cdot \varepsilon'$, for any two-party protocol π with communication complexity c.

Trusted Oracle. Given an oracle $h\colon \{0,1\}^{\alpha} \to \{0,1\}^{\beta}$, consider the two-party functionality $\mathcal{F}(h)$ defined below:

1. It takes input $(x_1, y_1) \in \{0,1\}^{\alpha} \times \{0,1\}^{\beta}$ from party 1 and $x_2 \in \{0,1\}^{\alpha}$ from party 2.
2. It computes $y = h(x_1 \oplus x_2)$.
3. It outputs $y_2 = y \oplus y_1$ to party 2.

Let $\mathcal{T}_h$ represent an oracle that implements the functionality $\mathcal{F}(h)$.

Ensure:

1. Suppose (I', C', O') is an ε'-parity-resilient implementation of f over the basis $\mathbb{B}'$.

Transformation:

1. $\mathcal{T}_{\mathbb{B}'} = \{\mathcal{T}_g\colon g \in \mathbb{B}'\}$
2. Input Encoder Description I'': Given input $x \in \{0,1\}^{n_i}$, compute $\widehat{x} = I'(x) \in \{0,1\}^{\widehat{n}'_i}$, $\widehat{x}_1 \xleftarrow{\$} \{0,1\}^{\widehat{n}'_i}$ and $\widehat{x}_2 = \widehat{x} \oplus \widehat{x}_1$.
3. Randomness Distributions (R_1, R_2): Let $R_1 = U_{|C'|}$, where $|C'|$ represents the number of wires in C', and $R_2 = \emptyset$.
4. Next Message Functions (M_1, M_2): Iteratively, the protocol performs the gate-wise computation of C' such the value of each wire in C' is additively secret-shared between the two parties. Suppose a gate $g\colon \{0,1\}^{\alpha} \to \{0,1\}^{\beta}$ takes as input wires $(w_1, \ldots, w_{\alpha})$ and the output wires are $(w_{\alpha+1}, \ldots, w_{\alpha+\beta})$. Let $\widehat{w}_{k,1}$ and $\widehat{w}_{k,2}$ be the additive secret-shares of the value corresponding to the wire w_k, and party 1 has $\widehat{w}_{k,1}$ and party 2 has $w_{k,2}$, where $k \in [\alpha]$. Party 1, using fresh bits from its local randomness r_1, defines $\widehat{w}_{\alpha+\ell,1} \xleftarrow{\$} \{0,1\}$, where $\ell \in [\beta]$. Now, parties invoke $H = \mathcal{T}_g$ with, respective, inputs $(\widehat{w}_{1,1}, \ldots, \widehat{w}_{\alpha+\beta,1})$ and $(\widehat{w}_{1,2}, \ldots, \widehat{w}_{\alpha,2})$. Party 2 defines the output of the instance of $\mathcal{T}_g$ as $(\widehat{w}_{\alpha+1,2}, \ldots, \widehat{w}_{\alpha+\beta,2})$. Note that the round complexity of this protocol is the depth of the circuit C', because all gate computations at a certain level of C' can be performed in parallel in one round of the protocol.
5. Output Decoder Description O'': Given $\widehat{y}_1 \in \{0,1\}^{\widehat{n}'_o}$ and $\widehat{y}_2 \in \{0,1\}^{\widehat{n}'_o}$, output $O'(\widehat{y}_1 \oplus \widehat{y}_2)$.
6. Privacy Guarantee: For a leakage protocol with communication complexity c, this is a $2^{c/2} \cdot \varepsilon'$-BCL-resilient protocol.

Particular Instantiations:

1. Let (I', C', O') be the $(3/4)^k$-parity-resilient implementation of f over the basis $\mathcal{G}_{\mathsf{Enc},\mathbb{B}}$ as produced by Theorem 2.

Figure 6: Using a Parity-resilient Implementation of f to construct a BCL-resilient Implementation of f.

6 Replacing Finite Oracles with 2-choose-1 OT

In this section, we shall prove the following lemma:

Lemma 1 Restated (Finite oracle ⇒ OT oracle)**.** *For any positive integers $\alpha_1, \alpha_2, \beta$, a (c, ε)-BCL-resilient H-aided protocol Π for f, where $H : \{0,1\}^{\alpha_1} \times \{0,1\}^{\alpha_2} \to \{0,1\}^{\beta}$, can be efficiently transformed to a similar OT-aided protocol Π' for f, where Π' has the same input encoder and output decoder as Π, and where $|\Pi|' \leq 2^{\alpha_2} \cdot \beta \cdot |\Pi|$.*

First, we shall show that $(t,1)$-OT, for constant t, suffices and then we shall show that $(2,1)$-OT suffices.

Let Π be an (c, ε)-BCL implementation of f using trusted oracles in $\mathbb{B}''$. While executing Π, every invocation of an oracle $H \in \mathbb{B}''$ is replaced as follows.

1. Let $H\colon \{0,1\}^{\alpha_1} \times \{0,1\}^{\alpha_2} \to \{0,1\}^{\beta}$. Define $H_i\colon \{0,1\}^{\alpha_1} \times \{0,1\}^{\alpha_2} \to \{0,1\}^{\beta}$ as the i-th output bit of H, where $i \in [\beta]$.
2. Suppose parties invoke H with respective inputs $x_1 \in \{0,1\}^{\alpha_1}$ and $x_2 \in \{0,1\}^{\alpha_2}$.
3. For each $i \in [\beta]$, do the following. Perform $\binom{2^{\beta}}{1}$-OT where the sender bits are: $H_i(x_1, 0)$, $\ldots$, $H_i(x_1, 2^{\beta} - 1)$, and the receiver choice index is $x_2 \in \{0, \ldots, 2^{\beta} - 1\}$. Party 2 receives $H_i(x_1, x_2)$.

The protocol incurs an additional 2^{β} multiplicative overhead and the view of the transformed protocol is identical to the view of the protocol using oracles in $\mathbb{B}''$. This new protocol uses oracles in $\{\binom{2^{\beta}}{1}\text{-OT}\colon H \in \mathbb{B}''$ has output domain $\{0,1\}^{\beta}\}$.

Next, we shall show that $(2,1)$-OT suffices. For this, we need the notion of "joint simulation of views," (Section 6.1) and the final proof is presented in Section 6.2.

The intuitive reasoning underlying our approach is the following. The number of $(m,1)$-OT instances is typically significantly larger than the additional communication that our BCL-protocol can tolerate. So, simply implementing $(m,1)$-OT using a secure protocol and basing the security of the composed protocol on a small loss in the security parameter of the original BCL protocol is not possible (a technique akin to "complexity leveraging"). To circumvent this issue, we implement $(m,1)$-OT using a *stronger* notion of security, namely *joint simulation security*. This stronger security notion ensures that the *joint views* of the composed protocol can be simulated by the parties using coordinated-simulators that they run on their respective *local views*. Under joint simulation security, we provide protocols that: (1) Reduce $(m,1)$-OT to $(2,1)$-OT, (2) Reduce $(2,1)$-OT to precomputed ROT-correlations, and (3) Reduce $(2,1)$-OT to the plain model using computational hardness assumptions. Towards this direction we present a composition theorem in Theorem 8.

6.1 Joint Simulation Security

Party $i \in \{1,2\}$ has private input x_i and local private randomness r_i. The randomness (r_1, r_2) can either be sampled independently in the plain model or according to some fixed joint distribution in

the correlated private randomness model. The protocol $\pi((x_1, r_1), (x_2, r_2))$ is a two-party protocol starting with respective inputs (x_b, r_x) with party i. We also consider protocols that are oracle-aided, i.e. parties can interact via a deterministic 2-party functionality oracle. Without loss of generality, we assume that party i uses no randomness other than r_i.

For fixed inputs (x_1, x_2), the *view of the protocol*, represented by $\mathsf{view}(x_1, x_2)$, is the random variable obtained by executing $\pi((x_1, R_1), (x_2, R_2))$ and outputting $((x_1, R_1, T_1), (x_2, R_2, T_2))$, where T_i represents all messages received by party i either from the other party or from the oracle functionalities.

Definition 11 (Joint Simulation Security)**.** *Let $f\colon X_1 \times X_2 \to Y_1 \times Y_2$ be a deterministic function such that $f(x_1, x_2) = (y_1, y_2)$. We say that a protocol $\pi(\cdot, \cdot)$ realizes f with* ε-joint simulation security *if the following conditions hold:*

1. *(Perfect) Correctness: For every inputs (x_1, x_2), we have $\Pr[\pi(x_1, x_2)$ outputs $f(x_1, x_2)] = 1$.*
2. *Joint Simulation Security: There exists a pair of simulators S_1 and S_2, and a randomness distribution T shared between the simulators such that for all inputs we following holds:*

$$\mathrm{SD}\left(\left(S_1(x_1, y_1, T), S_2(x_2, y_2, T)\right) , \mathsf{view}(x_1, x_2) \right) \leqslant \varepsilon$$

In particular, if $\varepsilon = 0$ then we say that π perfectly realizes *f* with joint simulation security*. And, if the distributions $(S_1(x_1, y_1, T), S_2(x_2, y_2, T))$ and $\mathsf{view}(x_1, x_2)$ can only be ε-distinguished by computationally bounded distinguishers then we say that π realizes f with* ε-computational joint simulation security*.*

We highlight that this definition is similar to the notion of "strong oblivious simulation" as introduced by [DLZ15].

6.1.1 Composition Theorem

In this section we present a composition theorem that is useful to modularly construct BCL-resilient protocols.

Theorem 8 (Composition Theorem)**.** *Let $\Pi_{f|g}$ is a (c, ε)-BCL-resilient implementation of f that uses oracle calls to g. Let Π_g perfectly realizes g with perfect joint simulation security. Consider the protocol Π_f obtained by replacing every invocation of g in $\Pi_{f|g}$ by an independent execution of the protocol Π_g. Then Π_f is a (c, ε)-BCL-resilient implementation of f.*

Further, if Π_g realizes g with ε'-computational joint simulation security then Π_f is a computationally c-BCL-resilient implementation of f.

Proof. Let Π' be $\Pi_{f|g}$ with one of its g-invocations replaced by Π_g. Let π' be a protocol running on views generated by Π' and has communication complexity c. Let $\mathsf{view}_{\Pi'}[x]$ be the joint distribution of view of parties generated by Π' on input x.

We will study the random variable $\pi'(\mathsf{view}_{\Pi'[x]})$ using a hybrid argument.

Suppose the inputs to the substituted instance of g are (u_1, u_2) and the outputs are (v_1, v_2). Consider a new protocol $\widetilde{\pi}$ that runs identical to π, except that it erases the execution of Π_g and substitutes it with view_{Π_g} using $(S_1(u_1, v_1, T), S_2(u_2, v_2, T))$. The views of π' and $\tilde{\pi}$ are identical (by perfect joint simulation security of Π_g). So, $\pi'(\mathsf{view}_{\Pi'[x]})$ is identical to $\widetilde{\pi}(\mathsf{view}_{\Pi'[x]})$.

Next, consider the following. Run $\Pi_{g|f}$ on input x. Consider the protocol π that, for $i \in \{1, 2\}$, instructs party i to execute $S_i(u_1, v_1, T)$ and then runs the protocol $\widetilde{\pi}$. It is clear that $\widetilde{\pi}(\mathsf{view}_{\Pi'[x]})$ is identical to $\pi(\mathsf{view}_{\Pi_{f|g}[x]})$.

Note that π', $\widetilde{\pi}$ and π have identical communication complexity. So, Π' is also (c, ε)-BCL-resilient implementation.

Analogously, all invocations to g-oracle can be replaced by the protocol Π_g and the result follows.

In the computational case, since, S_1 and S_2 are efficient, the overall distinguishing advantage can be at most $\varepsilon + \left|\Pi_{f|g}\right| \cdot \varepsilon'$. And, hence, the result. □

6.1.2 Protocols with Joint Simulation Security

In this section we present some protocols with joint simulation security.

Theorem 9 (Joint Simulation: Instantiations)**.** *There exists a $\pi_{f|g}$ that perfectly realizes f with joint simulation strategy while invoking oracle calls to g, when:*

1. *$f = (n, 1)$-OT, $g = (2, 1)$-OT and $R_1 = R_2 = \emptyset$.*
2. *$f = (2, 1)$-OT, $g = \emptyset$ and (R_1, R_2) is the ROT-correlated private randomness.*
3. *$f = (m, 1)$-OT, $g = \emptyset$ and (R_1, R_2) are sufficiently long independent random strings in the computational setting.*

Proof. 1. **Construction 1.** Consider the protocol presented below:

Ensure:

(a) Input to party 1: $(x_1, \ldots, x_n)$ and input to party 2: $i \in [n]$.

(b) Output of party 1: $\emptyset$ and Output of party 2: x_i.

Protocol:

(a) Party 2 computes $(c_1, \ldots, c_n)$ such that $c_j = 1$ if $j = i$, otherwise $c_j = 0$.

(b) Parties invoke n independent instances of $(2, 1)$-OT with party 1 inputs $(0, x_i)$ and party 2 input c_i. Party 2 receives $z_j = x_{c_j}$, for $j \in [n]$.

(c) Party 2 outputs z_i.

Consider the following simulation strategies. Let $S_1((x_1, \ldots, x_n))$ be the simulator that outputs the view that it invoked n instances of $(2, 1)$-OT with respective inputs $(0, x_i)$. Let

$S_2(i, z = x_i)$ be the simulator that outputs the view that it invoked n instances of $(2,1)$-OT with respective input c_j, where $c_j = 1$ if $j = i$, otherwise $c_j = 0$. And it receives z_j as output from the j-th instantiation of $(2,1)$-OT, where $z_j = z$ if $j = i$, otherwise $z_j = 0$.

It clear that this is a perfect joint simulation.

2. **Construction 2.** Consider the protocol presented below.

Ensure:
(a) Input to party 1 is (x_0, x_1) and input to party 2 is c.
(b) Output of party 1 is $\emptyset$ and output of party 2 is x_c.
Protocol:
(a) Party 1 obtains (y_0, y_1) and party 2 obtains $(b, w = y_b)$, where $y_0, y_1, b \xleftarrow{\$} \{0,1\}$ (this is the ROT private correlated randomness).
(b) Party 2 sends $m = b + c$.
(c) Party 1 sends $\alpha_0 = x_0 + y_m$ and $\alpha_1 = x_1 + y_{\overline{m}}$.
(d) Party 2 computes $z = \alpha_c + w$.

Consider the following simulation strategies. Let the randomness sample shared among the simulators be: $(\widetilde{\alpha}_0, \widetilde{\alpha}_1, \widetilde{m})$, where each bit is chosen uniformly at random. $S_1((x_0, x_1), (\widetilde{\alpha}_0, \widetilde{\alpha}_1, \widetilde{m}))$ generates a view where:

(a) It obtains $y_0 = \widetilde{\alpha}_m + x_m$ and $y_1 = \widetilde{\alpha}_{\overline{m}} + x_{\overline{m}}$ from the ROT correlation.
(b) It receives $m = \widetilde{m}$ from party 2, and
(c) It sends $\alpha_0 = \widetilde{\alpha}_0$ and $\alpha_1 = \widetilde{\alpha}_1$ to party 2.

$S_2(c, z = x_c, (\widetilde{\alpha}_0, \widetilde{\alpha}_1, \widetilde{m}))$ generates a view where:

(a) It obtains $b = \widetilde{m} + c$ and $w = \widetilde{\alpha}_c + x_c$ from the ROT correlation.
(b) It sends $m = \widetilde{m}$ to party 1,
(c) It receives $\alpha_0 = \widetilde{\alpha}_0$ and $\alpha_1 = \widetilde{\alpha}_1$ from party 1, and
(d) It computes $z = \alpha_c + w = x_c$.

This is a perfect simulation because, given a fixed (x_0, x_1, c), the distribution (y_0, y_1, b) is uniformly random. And the random variables corresponding to messages $(w, m, \alpha_0, \alpha_1)$ in the protocol are deterministically defined in terms of (x_0, x_1, c) and (y_0, y_1, b). One can verify that the random variables as output by the simulators satisfy those relations.

3. **Construction 3.** We present a protocol for $(2,1)$-OT inspired by the construction in [BCH12]. The protocol for $(m,1)$-OT can be constructed analogously. Consider the protocol presented below:

Ensure:

(a) Input to party 1 is (x_0, x_1) and input to party 2 is i.

(b) Output of party 1 if $\emptyset$ and output of party 2 is x_i.

(c) (NC-Gen, NC-Enc, NC-Dec, NC-Sim, NC-oGen, NC-oSim) is a special non-committing bit encryption scheme that has oblivious key sampling (see Appendix B for definition).

Protocol:

(a) Party 2 computes $(e_i, d_i) = \text{NC-Gen}(1^k; r_G)$ and $e_{\bar{i}} = \text{NC-oGen}(1^k; r_{\widehat{G}})$. Party 2 sends encryption keys (e_0, e_1) to party 1.

(b) Party 1 computes $c_b = \text{NC-Enc}_{e_b}(x_b; r_{E,b})$, for $b \in \{0,1\}$. Party 1 sends (c_0, c_1) to party 2.

(c) Party 2 outputs $x_i = \text{NC-Dec}_{d_i}(c_i)$.

The joint simulation strategy samples two random strings (r_0, r_1) as the common random string. The simulation strategy $S_1(x_0, x_1, (r_0, r_1))$ does the following.

(a) Obtain $(e_b, c_b, r^0_{G,b}, r^0_{E,b}, r^1_{G,b}, r^1_{E,b}) \leftarrow \text{NC-Sim}(1^k, r_b)$, for $b \in \{0,1\}$.

(b) Create the view of party 1 as follows. Message (e_0, e_1) is received from party 2.

(c) And message (c_0, c_1) is sent to party 2, where c_b is encryption of x_b using the key e_b and randomness $r^{x_b}_{E,b}$, for $b \in \{0,1\}$.

The simulation strategy $S_2(i, x_i, (r_0, r_1))$ does the following.

(a) Obtain $(e_b, c_b, r^0_{G,b}, r^0_{E,b}, r^1_{G,b}, r^1_{E,b}) \leftarrow \text{NC-Sim}(1^k, r_b)$, for $b \in \{0,1\}$.

(b) Create the view of party 2 as follows. Party 2 sent (e_0, e_1) such that $(e_i, d_i) \leftarrow \text{NC-Gen}(r^{x_i}_{G,i})$ and claim that $e_{\bar{i}}$ was generated by NC-oGen using randomness $r' \leftarrow \text{NC-oSim}(e_{\bar{i}})$.

(c) Party 2 receives (c_0, c_1) and outputs x_i as a decoding of c_i with decryption-key d_i.

We prove the security of the joint simulation strategy using a hybrid argument. Consider the following sequence of hybrids.

(a) Hybrid 0 (Real World). Parties follow the protocol honestly.

(b) Hybrid 1. Party 2 generates both $(e_b, d_b) \leftarrow \text{NC-Gen}(1^k; r_{G,b})$, for $b \in \{0,1\}$. And explains $e_{\bar{i}}$ using the randomness $r' \xleftarrow{\$} \text{NC-oSim}(e_{\bar{i}})$. The "special" property of our scheme ensures that this hybrid is indistinguishable from the previous hybrid.

(c) Hybrid 2. Substitute $(e_0, c_0, r_{G,0}, r_{E,0})$ by the simulated string $(e_0, c_0, r^{x_0}_{G,0}, r^{x_0}_{E,0})$ from the outputs of $\text{NC-Sim}(1^k, r_0)$ and let $(e_0, d_0) \leftarrow \text{NC-Gen}(1^k; r^{x_0}_{G,0})$. This hybrid is indistinguishable from the previous hybrid due to the security of non-committing encryption.

(d) Hybrid 3. Substitute $(e_1, c_1, r_{G,1}, r_{E,1})$ by the simulated string $(e_1, c_1, r^{x_1}_{G,1}, r^{x_1}_{E,1})$ from the outputs of $\text{NC-Sim}(1^k, r_1)$ and let $(e_1, d_1) \leftarrow \text{NC-Gen}(1^k; r^{x_1}_{G,1})$. This hybrid is indistinguishable from the previous hybrid due to the security of non-committing encryption.

(e) Hybrid 4 (Joint Simulation World). Note that this is identical to the previous hybrid.

This shows that the joint simulation of the views is computationally indistinguishable from the joint views of the parties in the real world. Similarly, we can also prove the joint simulation security for the analogous protocol for $(m, 1)$-OT.

□

6.2 Final Part of Proof

Consider a (c, ε)-BCL-resilient protocol using $\{(2^\beta, 1)\text{-}\mathsf{OT} \colon H \in \mathbb{B}''$ has output domain $\{0,1\}^\beta\}$. Substitute each invocation of $(2^\beta, 1)$-OT with the protocol presented in Theorem 9 (Part 1.) that uses an oracle implementing $(2, 1)$-OT. Using the composition theorem Theorem 8, we get the result.

7 Protocol Instantiations

Here we prove the main corollaries of our theorems by instantiating them with appropriate constructions. Our aim is to reduce the construct bounded-communication resilient implementation of a functionality f (that is implemented by a circuit C_f) using random OT pre-computations or using computational assumptions in the plain model.

7.1 Proof of Corollary 5

For any function $f \colon \{0,1\}^{n_i} \to \{0,1\}^{n_o}$, there exists an $(\kappa, 2^{-\kappa})$-BCL-resilient protocol of f using $(2, 1)$-OT-oracle. The construction proceeds as follows:

1. Let (I, C, O) be a k-private implementation of the function f using Theorem 1 over the basis $\mathbb{B}$.
2. Let (I', C', O') be an ε_1-parity-resilient implementation of f using Theorem 2 over the basis $\mathcal{G}_{\mathsf{Enc},\mathbb{B}}$, where Enc is the encoding function in Figure 3 with $t = 1$ and $\varepsilon_1 = (3/4)^k$.
3. Let Π be a (c, ε_2)-BCL-resilient implementation of f using Theorem 4 over the oracles in $\mathcal{T}_{\mathcal{G}_{\mathsf{Enc},\mathbb{B}}}$, where $\varepsilon_2 = 2^{c/2} \cdot (3/4)^k$.
4. Let Π' be a (c, ε_2)-BCL-resilient implementation of f using Lemma 1 over the oracle $\{(2, 1)\text{-}\mathsf{OT}\}$.

Using $c = \kappa$ and $k = 2\kappa \log_{4/3} 2$ in the construction, (I'', C'', O'') is a $(\kappa, 2^{-\kappa})$-parity resilient implementation of f over the standard basis $\mathbb{B}$, and there exists circuit implementations such that $|I''| = \widetilde{O}(n_i + \kappa)$, $|C''| = \widetilde{O}(s + \kappa h + \kappa^2)$ and $|O''| = \widetilde{O}(n_o + \kappa)$.

Using Theorem 8 and Theorem 9 (Part 2), we get that the same construction using precomputed ROT-correlated private randomness is also $(\kappa, 2^{-\kappa})$-bounded-communication resilient implementation of f.

7.2 Proof of Corollary 6

Fix $c = 1$ and consider the Π' as constructed in Section 7.1. Now construct a circuit C'' over the standard basis that generates the joint view of parties (the input encoder I'' and the output decoder O'' remain unchanged). In this circuit, any invocation of the $(2,1)$-OT with inputs (x_0, x_1) and b is computed by: $t = x \wedge b$ and $z = x_0 \oplus t$. Note that we can express the intermediate wire t as $x_0 \oplus z$, and the values x_0 and z already occur in the views of party 1 and 2, respectively, generated by Π'. Any parity of wires in C'' can, thus, be expressed as parities of values that exists in the views of parties as generated in Π'. So, any parity of wires of C'' can be computed by performing 1 bit of communication in Π'. Therefore, (I'', C'', O'') is an ε_2-parity resilient implementation of f over the standard basis $\mathbb{B}$. Using $k = (1+\kappa) \cdot \log_{4/3} 2$ in the construction, (I'', C'', O'') is a $2^{-\kappa}$-parity resilient implementation of f over the standard basis $\mathbb{B}$, and there exists circuit implementations such that $|I''| = \widetilde{O}(n_i + \kappa)$, $|C''| = \widetilde{O}(s + \kappa h + \kappa^2)$ and $|O''| = \widetilde{O}(n_o + \kappa)$.

8 Computational Bounded-communication Leakage Resilience

In this section we prove the following theorem.

Theorem 7 Restated (Computational BCL-resilient protocols in the plain model)**.** *Suppose the DDH assumption holds. Then, for every polynomial-time computable $f : \{0,1\}^n \to \{0,1\}^{m(n)}$ and polynomial $c(n)$, there is a computational $c(n)$-BCL-resilient implementation $\Pi = (I, (M_1, M_2), O)$ of f, where the running time of I is $\widetilde{O}(n + c(n))$ and the running time of O is $\widetilde{O}(m(n) + c(n))$.*

This result follows by using the composition theorem (Theorem 8) on the following two results: (1) Construction of BCL-resilient protocol using $(2,1)$-OT by Corollary 5, and (2) Computational joint simulation secure protocol for $(m,1)$-OT presented in Theorem 9 (Part 3) instantiated using the special NCE construction with oblivious key sampling based on the computational hardness of factoring Blum integers or Decisional Diffie-Hellman by [CDMW09].

Acknowledgements. This research was done in part while the first four authors were visiting the Simons Institute for the Theory of Computing, supported by the Simons Foundation and by the DIMACS/Simons Collaboration in Cryptography through NSF grant #CNS-15-23467. The individual authors were supported by the following grants: BSF grant 2012378, ISF grant 1709/14, ERC starting grant 259426, NSF grant CNS-1566499, a DARPA/ARL SAFEWARE award, NSF Frontier Award 1413955, NSF grants 12-28984, 11-36174, 11-18096, and 10-65276, NSF CAREER award CCF-1149018, a Xerox Faculty Research Award, a Google Faculty Research Award, an equipment grant from Intel, an Okawa Foundation Research Grant, and an Alfred P. Sloan Foundation Research Fellowship. This material is in part based upon work supported by the Defense Advanced Research Projects Agency through the ARL under Contract W911NF-15-C-0205. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense, the National Science Foundation, or the U.S. Government.

References

[AIK08] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. On pseudorandom generators with linear stretch in NC^0. *Computational Complexity*, 17(1):38–69, 2008. 6

[AR94] Noga Alon and Yuval Roichman. Random cayley graphs and expanders. *Random Struct. Algorithms*, 5(2):271–285, 1994. 17, 35

[BCH12] Nir Bitansky, Ran Canetti, and Shai Halevi. Leakage-tolerant interactive protocols. In Ronald Cramer, editor, *TCC 2012: 9th Theory of Cryptography Conference*, volume 7194 of *Lecture Notes in Computer Science*, pages 266–284, Taormina, Sicily, Italy, March 19–21, 2012. Springer, Heidelberg, Germany. 1, 25

[BDL14] Nir Bitansky, Dana Dachman-Soled, and Huijia Lin. Leakage-tolerant computation with input-independent preprocessing. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014, Part II*, volume 8617 of *Lecture Notes in Computer Science*, pages 146–163, Santa Barbara, CA, USA, August 17–21, 2014. Springer, Heidelberg, Germany. 3, 4

[BGI+14] Amos Beimel, Ariel Gabizon, Yuval Ishai, Eyal Kushilevitz, Sigurd Meldgaard, and Anat Paskin-Cherniavsky. Non-interactive secure multiparty computation. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014, Part II*, volume 8617 of *Lecture Notes in Computer Science*, pages 387–404, Santa Barbara, CA, USA, August 17–21, 2014. Springer, Heidelberg, Germany. 7

[BIKK14] Amos Beimel, Yuval Ishai, Ranjit Kumaresan, and Eyal Kushilevitz. On the cryptographic complexity of the worst functions. In Yehuda Lindell, editor, *TCC 2014: 11th Theory of Cryptography Conference*, volume 8349 of *Lecture Notes in Computer Science*, pages 317–342, San Diego, CA, USA, February 24–26, 2014. Springer, Heidelberg, Germany. 3

[BIVW15] Andrej Bogdanov, Yuval Ishai, Emanuele Viola, and Christopher Williamson. Bounded indistinguishability and the complexity of recovering secrets. *Electronic Colloquium on Computational Complexity (ECCC)*, 22:182, 2015. 9

[CDMW09] Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. Improved non-committing encryption with applications to adaptively secure protocols. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 287–302, Tokyo, Japan, December 6–10, 2009. Springer, Heidelberg, Germany. 4, 28, 35

[CDNO97] Ran Canetti, Cynthia Dwork, Moni Naor, and Rafail Ostrovsky. Deniable encryption. In Burton S. Kaliski Jr., editor, *Advances in Cryptology – CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 90–104, Santa Barbara, CA, USA, August 17–21, 1997. Springer, Heidelberg, Germany. 3

[CFGN96] Ran Canetti, Uriel Feige, Oded Goldreich, and Moni Naor. Adaptively secure multiparty computation. In *28th Annual ACM Symposium on Theory of Computing*, pages 639–648, Philadephia, Pennsylvania, USA, May 22–24, 1996. ACM Press. 4, 7, 34

[DDV10] Francesco Davì, Stefan Dziembowski, and Daniele Venturi. Leakage-resilient storage. In Juan A. Garay and Roberto De Prisco, editors, *SCN 10: 7th International Conference on Security in Communication Networks*, volume 6280 of *Lecture Notes in Computer Science*, pages 121–137, Amalfi, Italy, September 13–15, 2010. Springer, Heidelberg, Germany. 6

[DF12] Stefan Dziembowski and Sebastian Faust. Leakage-resilient circuits without computational assumptions. In Ronald Cramer, editor, *TCC 2012: 9th Theory of Cryptography Conference*, volume 7194 of *Lecture Notes in Computer Science*, pages 230–247, Taormina, Sicily, Italy, March 19–21, 2012. Springer, Heidelberg, Germany. 3, 4

[DIK10] Ivan Damgård, Yuval Ishai, and Mikkel Krøigaard. Perfectly secure multiparty computation and the computational overhead of cryptography. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 445–465, French Riviera, May 30 – June 3, 2010. Springer, Heidelberg, Germany. 4, 8, 9, 10

[DLW06] Giovanni Di Crescenzo, Richard J. Lipton, and Shabsi Walfish. Perfectly secure password protocols in the bounded retrieval model. In Shai Halevi and Tal Rabin, editors, *TCC 2006: 3rd Theory of Cryptography Conference*, volume 3876 of *Lecture Notes in Computer Science*, pages 225–244, New York, NY, USA, March 4–7, 2006. Springer, Heidelberg, Germany. 2

[DLZ15] Dana Dachman-Soled, Feng-Hao Liu, and Hong-Sheng Zhou. Leakage-resilient circuits revisited - optimal number of computing components without leak-free hardware. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part II*, volume 9057 of *Lecture Notes in Computer Science*, pages 131–158, Sofia, Bulgaria, April 26–30, 2015. Springer, Heidelberg, Germany. 3, 4, 7, 23

[DN00] Ivan Damgård and Jesper Buus Nielsen. Improved non-committing encryption schemes based on a general complexity assumption. In Mihir Bellare, editor, *Advances in Cryptology – CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 432–450, Santa Barbara, CA, USA, August 20–24, 2000. Springer, Heidelberg, Germany. 4

[DNW08] Ivan Damgård, Jesper Buus Nielsen, and Daniel Wichs. Isolated proofs of knowledge and isolated zero knowledge. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 509–526, Istanbul, Turkey, April 13–17, 2008. Springer, Heidelberg, Germany. 2

[DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008. 35

[DP07] Stefan Dziembowski and Krzysztof Pietrzak. Intrusion-resilient secret sharing. In *48th Annual Symposium on Foundations of Computer Science*, pages 227–237, Providence, USA, October 20–23, 2007. IEEE Computer Society Press. 2

[DS05a] Yevgeniy Dodis and Adam Smith. Correcting errors without leaking partial information. In Harold N. Gabow and Ronald Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 654–663, Baltimore, Maryland, USA, May 22–24, 2005. ACM Press. 5, 6, 17, 35

[DS05b] Yevgeniy Dodis and Adam Smith. Entropic security and the encryption of high entropy messages. In Joe Kilian, editor, *TCC 2005: 2nd Theory of Cryptography Conference*, volume 3378 of *Lecture Notes in Computer Science*, pages 556–577, Cambridge, MA, USA, February 10–12, 2005. Springer, Heidelberg, Germany. 6

[Dzi06] Stefan Dziembowski. Intrusion-resilience via the bounded-storage model. In Shai Halevi and Tal Rabin, editors, *TCC 2006: 3rd Theory of Cryptography Conference*, volume 3876 of *Lecture Notes in Computer Science*, pages 207–224, New York, NY, USA, March 4–7, 2006. Springer, Heidelberg, Germany. 2

[FRR+10] Sebastian Faust, Tal Rabin, Leonid Reyzin, Eran Tromer, and Vinod Vaikuntanathan. Protecting circuits from leakage: the computationally-bounded and noisy cases. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 135–156, French Riviera, May 30 – June 3, 2010. Springer, Heidelberg, Germany. 1, 9

[FS08] Serge Fehr and Christian Schaffner. Randomness extraction via δ-biased masking in the presence of a quantum attacker. In Ran Canetti, editor, *TCC 2008: 5th Theory of Cryptography Conference*, volume 4948 of *Lecture Notes in Computer Science*, pages 465–481, San Francisco, CA, USA, March 19–21, 2008. Springer, Heidelberg, Germany. 6

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st Annual ACM Symposium on Theory of Computing*, pages 169–178, Bethesda, Maryland, USA, May 31 – June 2, 2009. ACM Press. 1

[GGH+13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th Annual Symposium on Foundations of Computer Science*, pages 40–49, Berkeley, CA, USA, October 26–29, 2013. IEEE Computer Society Press. 3

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In Alfred Aho, editor, *19th Annual ACM Symposium on Theory of Computing*, pages 218–229, New York City,, New York, USA, May 25–27, 1987. ACM Press. 1, 5, 7, 20

[Gol04] Oded Goldreich. *Foundations of Cryptography: Basic Applications*, volume 2. Cambridge University Press, Cambridge, UK, 2004. 5

[GR10] Shafi Goldwasser and Guy N. Rothblum. Securing computation against continuous leakage. In Tal Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 59–79, Santa Barbara, CA, USA, August 15–19, 2010. Springer, Heidelberg, Germany. 3

[GR12] Shafi Goldwasser and Guy N. Rothblum. How to compute in the presence of leakage. In *53rd Annual Symposium on Foundations of Computer Science*, pages 31–40, New Brunswick, NJ, USA, October 20–23, 2012. IEEE Computer Society Press. 3, 4, 5, 7, 8

[GW97] Oded Goldreich and Avi Wigderson. Tiny families of functions with random properties: A quality-size trade-off for hashing. *Random Struct. Algorithms*, 11(4):315–343, 1997. 17, 35

[IKOS09] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Extracting correlations. In *50th Annual Symposium on Foundations of Computer Science*, pages 261–270, Atlanta, Georgia, USA, October 25–27, 2009. IEEE Computer Society Press. 6

[ISW03] Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481, Santa Barbara, CA, USA, August 17–21, 2003. Springer, Heidelberg, Germany. 1, 6, 8, 9

[JV10] Ali Juma and Yevgeniy Vahlis. Protecting cryptographic keys against continual leakage. In Tal Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 41–58, Santa Barbara, CA, USA, August 15–19, 2010. Springer, Heidelberg, Germany. 3

[MR04] Silvio Micali and Leonid Reyzin. Physically observable cryptography (extended abstract). In Moni Naor, editor, *TCC 2004: 1st Theory of Cryptography Conference*, volume 2951 of *Lecture Notes in Computer Science*, pages 278–296, Cambridge, MA, USA, February 19–21, 2004. Springer, Heidelberg, Germany. 1, 2

[MST03] Elchanan Mossel, Amir Shpilka, and Luca Trevisan. On e-biased generators in NC0. In *44th Annual Symposium on Foundations of Computer Science*, pages 136–145, Cambridge, Massachusetts, USA, October 11–14, 2003. IEEE Computer Society Press. 6

[NN90] Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. In *22nd Annual ACM Symposium on Theory of Computing*, pages 213–223, Baltimore, Maryland, USA, May 14–16, 1990. ACM Press. 5, 17, 35

[O'D14] Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014. 33

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In David B. Shmoys, editor, *46th Annual ACM Symposium on Theory of Computing*, pages 475–484, New York, NY, USA, May 31 – June 3, 2014. ACM Press. 3

[Yao82] Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science*, pages 160–164, Chicago, Illinois, November 3–5, 1982. IEEE Computer Society Press. 1, 7

A Technical Results

This section contains technical results needed to prove our main results.

A.1 Bias of Distributions

Claim 2. *Let $X_1, \ldots, X_t$ be independent random variables such that, for all $1 \leqslant j \leqslant t$, the distribution X_j is ε_j-biased. Then, the random variable $\left(\oplus_{i=1}^{t} X_i\right)$ is $\left(\prod_{i=1}^{t} \varepsilon_i\right)$-biased.*

Proof. Proof for $t = 2$ suffices. Let $\Pr[X_1 = 0] - \Pr[X_1 = 1] = \alpha_1$ and $\Pr[X_2 = 0] - \Pr[X_2 = 1] = \alpha_2$. Let $X = X_1 \oplus X_2$. Then, we have:

$$\begin{aligned}\Pr[X = 0] - \Pr[X = 1] &= \Pr[X_1 = 0]\left(\Pr[X_2 = 0] - \Pr[X_2 = 1]\right) + \Pr[X_1 = 1]\left(\Pr[X_2 = 1] - \Pr[X_2 = 0]\right) \\ &= \left(\Pr[X_1 = 0] - \Pr[X_1 = 1]\right) \cdot \left(\Pr[X_2 = 0] - \Pr[X_2 = 1]\right) = \alpha_1\alpha_2\end{aligned}$$

Taking absolute value both sides, we get the result. □

Claim 3. *If the distribution X is ε-biased then the statistical distance of X from uniform distribution is at most $\varepsilon/2$.*

A.2 Small-bias Encodings

Claim 4. *For $t \geqslant 1$, let $\mathsf{Maj}_{(2t+1)}[x]$ be the uniform distribution over $(2t+1)$-length bit strings whose majority of bits are x. For any non-empty $S \subseteq [2t+1]$, the distribution $\chi_S(\mathsf{Maj}_{(2t+1)}[x])$ is ε-biased, where $\varepsilon = \binom{2t}{t} 2^{-2t}$.*

Proof. For a non-empty subset $S \subseteq [n]$, note that the bias of $\chi_S(\mathsf{Maj}_{(2t+1)}[x])$ is equal to $\left|2^{2t+1} \cdot \widehat{\mathsf{Maj}_{(2t+1)}}[x](S)\right|$. Let $r \in \{0,1\}^{2t+1}$. Interpreting the probability distribution $\mathsf{Maj}_{(2t+1)}[x]$ as a function over $(2t+1)$-length bit strings, observe that $\mathsf{Maj}_{(2t+1)}[x](r)$ is 0 if the majority of the bits in r is not x, and is 2^{-2t} if the majority of the bits in r is x.

Let $f\colon \{0,1\}^{2t+1} \to \{-1,+1\}$ be a function such that $f(r) = 1$ if the majority of the bits in r is 0, and $f(r) = -1$ if the majority of the bits in r is 1. Note that $f \equiv 2^{2t+1} \cdot \mathsf{Maj}_{(2t+1)}[0] - 1$.

Suppose $x = 0$. Using the linearity of Fourier transform, the bias of $\chi_S(\mathsf{Maj}_{(2t+1)}[x])$ is equal to $\left|\widehat{f}(S)\right|$. We know the following about the Fourier coefficients of the function f [O'D14]:

$$\widehat{f}(S) = \begin{cases} 0, & \text{if } |S| \text{ is even} \\ (-1)^v \dfrac{\binom{t}{v}}{\binom{2t}{2v}} \cdot \binom{2t}{t} 2^{-2t}, & \text{if } |S| = 2v+1 \end{cases}$$

Thus, we have the following upper bound: $\left|\widehat{f}(S)\right| \leqslant \binom{2t}{t} 2^{-2t}$, because $\binom{t}{v} \leqslant \binom{2t}{2v}$.

For $x = 1$, use the fact that $f \equiv 1 - 2^{2t+1} \cdot \mathsf{Maj}_{(2t+1)}[1]$ to obtain an identical upper bound on the bias of $\chi_S(\mathsf{Maj}_{(2t+1)}[x])$ □

Claim 5. *Consider the encoding function* Enc *defined in Figure 3. For any $x \in \{0,1\}$ and non-empty subset S, the distribution $\chi_S(\mathsf{Enc}[x])$ is ε-biased, where $\varepsilon = 1/2 + \binom{2t}{t} 2^{-(2t+1)}$.*

Proof. Let S be partitioned into two subsets S_R and S_Y by restricting S to the indices in the randomness r and the indices in $\mathsf{Enc}(x; r)$, respectively. We make the following observations:

1. Conditioned on majority of the bits in r being x, $\chi_S(\mathsf{Enc}[x])$ is identical to $\chi_{(S_R \Delta S_Y)}(\mathsf{Maj}_{(2t+1)}[x])$.
2. Conditioned on majority of the bits in r being $\overline{x}$, $\chi_S(\mathsf{Enc}[x])$ is identical to $z \oplus \chi_{(S_R \Delta S'_Y)}(\mathsf{Maj}_{(2t+1)}[x])$, where S'_Y is the left-rotation of S_Y by one position and z is the parity of $|S'_Y|$.
3. And each of the above two events occur with probability $1/2$ when r is drawn uniformly at random.

For a non-empty S, note that $S_R \Delta S_Y \neq \emptyset$ or $S_R \Delta S'_Y \neq \emptyset$. So, the bias of $\chi_S(\mathsf{Enc}[x])$ is at most $\frac{1}{2} \cdot 1 + \frac{1}{2} \cdot \binom{2t}{t} 2^{-2t}$ (using Claim 4). □

B Special Non-Committing Encryption

In this section we introduce the definition of a special non-committing encryption. To begin, we recall the definition of non-committing encryption:

Definition 12 (Non-Committing Encryption [CFGN96])**.** *A* non-committing (bit) encryption *(NCE) scheme consists of a tuple (*NC-Gen,NC-Enc,NC-Dec,NC-Sim*), where (*NC-Gen,NC-Enc,NC-Dec*) is a semantically secure (public-key) encryption scheme (with negligible decryption error), and* NC-Sim *is a PPT simulation algorithm that on input 1^k outputs a tuple $(e, c, r_G^0, r_E^0, r_G^1, r_E^1)$ such that for every $b \in \{0,1\}$, the following distributions are computationally indistinguishable:*

1. *The joint view of an honest sender and an honest receiver in a normal encryption of b:*

$$\{(e, c, r_G, r_E) \colon (e, d) = \text{NC-Gen}(1^k; r_G), c = \text{NC-Enc}_e(b; r_E)\}$$

2. *A simulated view of an encryption of b:*

$$\{(e, c, r_G^b, r_E^b) \colon (e, c, r_G^0, r_E^0, r_G^1, r_E^1) \leftarrow \text{NC-Sim}(1^k)\}$$

The scheme is said to have oblivious key sampling *if in addition to the above:*

- *There is an oblivious public key sampling algorithm* NC-oGen*, which on input* $(1^k, r_{\widehat{G}})$ *samples a public key* $\widehat{e}$ *which is indistinguishable from the public keys generated by* NC-Gen(1^k).

Next, we define a *special* form of non-committing encryption that is crucial for our joint simulation security.

Definition 13 (Special Non-committing Encryption)**.** *Let* (NC-Gen, NC-Enc, NC-Dec, NC-Sim, NC-oGen, NC-oSim) *be a scheme such that:*

1. (NC-Gen, NC-Enc, NC-Dec, NC-Sim, NC-oGen) *is a non-committing encryption scheme that has oblivious key sampling, and*
2. *The following two distributions are computationally indistinguishable:*

$$\left\{(\widehat{e}, r_{\widehat{G}}) \colon \widehat{e} \leftarrow \text{NC-oGen}(1^k, r_{\widehat{G}})\right\},$$

and

$$\left\{(e, r') \colon (e, d) \leftarrow \text{NC-Gen}(1^k; r_G), r' \leftarrow \text{NC-oSim}(1^k, e)\right\}$$

Then, (NC-Gen, NC-Enc, NC-Dec, NC-Sim, NC-oGen, NC-oSim) *is a* special non-committing encryption scheme with oblivious key sampling.

Based on the computational hardness of factoring Blum integers or the computational hardness of the Decisional Diffie-Hellman (DDH), constructions of special non-committing encryption schemes with oblivious key sampling exist [CDMW09].

C Small-bias Masking

Theorem 10 (Small-bias Masking Lemma [NN90, AR94, GW97, DS05a])**.** *Let* M *be a source over the sample space* $\{0,1\}^n$ *with min-entropy at least* $(n-\ell)$*. Let* X *be an independent source over the same sample space* $\{0,1\}^n$ *such that it has bias at most* ε*. Then the following holds:*

$$2\text{SD}\left(\ M \oplus X\ ,\ U_n\ \right) \leqslant 2^{\ell/2} \cdot \varepsilon$$

To understand the relation between Theorem 10 and the "one-round version" of Theorem 3, consider the following pairs of reductions.

First Reduction. Suppose X is an ε-bias distribution over $\{0,1\}^n$. We additively share X between two parties, party 1 and party 2. That is, party 1 receives U_n and party 2 receives $X \oplus U_n$. Now, party 1 obtains an ℓ-bit leakage on the view of party 1. Conditioned on this leakage, the view of party 1 has min-entropy $(n-\ell)$.[4] The view of party 2, despite ℓ-bits of leakage, remains $2^{\ell/2} \cdot \varepsilon$-close to uniform. So, the "one-round version" of our Theorem 3 is implied by Theorem 10.

[4] If L is the leakage random variable, then we have $\widetilde{H}_\infty(X|L) \geqslant (n-\ell)$, i.e., the *average min-entropy* of X is high [DORS08]. Theorem 10 also holds if X has $(n-\ell)$ average min-entropy. At an intuitive level, a source with average min-entropy is close to convex linear combination of min-entropy sources.

Second Reduction. Suppose X is an ε-bias distribution over $\{0,1\}^n$ and party 1 has view U_n and party 2 has view $(X \oplus U_n)$. Without loss of generality, assume that M is a flat with support $2^{n-\ell}$. The leakage communication protocol does the following: Party 1 sends 1, if its view is in the support of M; otherwise it sends 0. The statistical distance of party two's view (at the end of the 1-bit protocol) from uniform is at most $2^{1/2}\varepsilon$.

With probability $2^{-\ell}$, party 1 sends 1. Conditioned on the message being 1, the distance of party 2's view from uniform be: Δ. Then we get that: $2^{-\ell}\Delta \leqslant 2^{1/2}\varepsilon$, that is $\Delta \leqslant 2^{\ell+1/2}\varepsilon$. So, instead of a $2^{\ell/2}\varepsilon$ bound we get the slightly weaker bound of $2^{\ell}\varepsilon$.

The tighter version of this direction of the reduction can be proven for the following small-bias masking result.

$$\mathrm{SD}\left((X \oplus U_n, \mathcal{L}(U_n)) \ , \ (U'_n, \mathcal{L}(U_n)) \right) \leqslant \sqrt{2^{\ell}} \cdot \varepsilon,$$

where X is an ε-bias source and $\mathcal{L}\colon \{0,1\}^n \to \{0,1\}^{\ell}$ be any leakage function. We know that $\widetilde{H}_{\infty}(U_n|\mathcal{L}(U_n)) \geqslant (n-\ell)$.

Suppose the respective views of party 1 and party 2 are U_n and $X \oplus U_n$. Now, party 1 sends $\mathcal{L}(U_n)$ to party 2. Given this leakage, using Theorem 3 the view of party 2 is $2^{\ell/2}\varepsilon$ close to uniform. Thus, our theorem for one round leakage protocols yields the small-bias masking result for average min-entropy source obtained by performing ℓ-bits of leakage.