Adaptively Secure MPC with Sublinear Communication Complexity

Ran Cohen[1*], abhi shelat[2†], and Daniel Wichs[3‡]

[1] Boston University and Northeastern University
rancohen@ccs.neu.edu
[2] Northeastern University
abhi@neu.edu
[3] Northeastern University
wichs@ccs.neu.edu

Abstract. A central challenge in the study of MPC is to balance between security guarantees, hardness assumptions, and resources required for the protocol. In this work, we study the cost of tolerating adaptive corruptions in MPC protocols under various corruption thresholds.

In the strongest setting, we consider adaptive corruptions of an arbitrary number of parties (potentially all) and achieve the following results:

- A two-round secure function evaluation (SFE) protocol in the CRS model, assuming LWE and indistinguishability obfuscation (iO). The communication, the CRS size, and the online-computation are sublinear in the *size* of the function. The iO assumption can be replaced by secure erasures. Previous results required either the communication or the CRS size to be polynomial in the function size.
- Under the same assumptions, we construct a "Bob-optimized" 2PC (where Alice talks first, Bob second, and Alice learns the output). That is, the communication complexity and total computation of Bob are sublinear in the function size and in Alice's input size. We prove impossibility of "Alice-optimized" protocols.
- Assuming LWE, we bootstrap adaptively secure NIZK arguments to achieve proof size sublinear in the circuit size of the NP-relation.

On a technical level, our results are based on *laconic function evaluation (LFE)* (Quach, Wee, and Wichs, FOCS'18) and shed light on an interesting duality between LFE and FHE.

Next, we analyze adaptive corruptions of all-but-one of the parties and show a two-round SFE protocol in the threshold PKI model (where keys of a threshold FHE scheme are pre-shared among the parties) with communication complexity sublinear in the circuit size, assuming LWE and NIZK. Finally, we consider the honest-majority setting, and show a two-round SFE protocol with guaranteed output delivery under the same constraints.

*Research supported by the Northeastern University Cybersecurity and Privacy Institute Post-doctoral fellowship, NSF grant TWC-1664445, NSF grant 1422965, and by the NSF MACS project.

†Research supported by NSF grant TWC-1664445 and a Google Faculty fellowship.

‡Research supported by NSF grants CNS-1314722, CNS-1413964, CNS-1750795 and the Alfred P. Sloan Research Fellowship.

1 Introduction

After establishing feasibility in the 1980's [79, 52, 8, 28, 76], the rich literature of multi-party computation (MPC) has focused on several performance aspects of the problem. These aspects include: (a) studying the resources required in terms of communication rounds, total amount of communication, and total amount of computation, (b) minimizing the required complexity assumptions under the various notions, and most importantly, (c) enhancing the notion of security, starting from the simplest notion of static corruptions with semi-honest adversaries in a stand-alone model, to sequential, and concurrent composition, to adaptive corruptions of parties by a malicious adversary.

Recent results have considered a few of these questions simultaneously. Despite several decades of progress, many basic questions about feasibility and asymptotic optimality of MPC protocols remain. The focus of this paper is to study *the price of adaptive security* in light of recent round-optimal and low-communication protocols for the static-security setting.

Recall that *adaptive security* [7, 20] for an MPC protocol models the realistic threat in which the adversary can corrupt a party during the execution of a protocol—in particular, after seeing some of the transcript of a protocol. In contrast, with *static corruptions*, the adversary must choose which parties to corrupt *before* the protocol begins. In this simpler static case, the security argument relies on the fact that the inputs of the corrupted parties are known, and thus the simulator can "work around" these parties to generate a reasonable, and consistent transcript for the remaining parties. Indeed, adaptive security is known to be strictly stronger than static security [20, 22].

While the idea of allowing an adversary to corrupt parties at anytime during protocol executions seems natural, its technical formulation is captured by obliging the simulator in the security definition to support some specific tasks. In particular, the technical difficulty in achieving adaptive security is that the simulator must produce a transcript for the execution *before* knowing which parties are corrupted. In an extreme case, the protocol can already be completed, and the adversary can then *begin* to corrupt all of the parties, one by one.

Two main models are considered for adaptive corruptions. In the first and simpler one, it is assumed that parties can *securely erase* certain parts (and even all) of their random tapes.[4] In this setting, when simulating a party who gets corrupted, the simulator may not be required to provide random coins explaining all the messages previously sent by that party. In the second, *erasures-free* model, there are no assumptions about the ability to erase local information. When a party is corrupted in this adaptive security notion, the adversary can learn all of that party's inputs and internal random coins. In this case, a secure

[4] We note that in certain cases it is reasonable to erase the random coins, e.g., when encrypting a message it is normally fine not to store the encryption randomness; however, is some cases one cannot erase all of its random tape, e.g., when sending a public encryption key it is normally essential to store the decryption key. We refer the reader to [20, 18] for further discussion on secure erasures.

protocol requires a simulator that, after producing the transcript, can "explain" the transcript by generating the coins and inputs for a given party after they are corrupted. In particular, the simulator only learns the input of that party after the corruption (e.g., after the entire execution), and then must "explain" the transcript it produced beforehand in a way that is consistent with the given input.

As a result of these difficulties, most of the literature shows that achieving adaptive security is notoriously harder than achieving static security; in some cases there are outright impossibility results such as the case of fully homomorphic encryption [68], public-key encryption which cannot exist for arbitrary messages [72], constant-round MPC in the plain model (under black-box simulation) [47], MPC protocols with non-expander communication graphs [17], and composable broadcast protocols without an honest majority [62]. All of these lower bounds, with the exception of [47], hold also for the weaker adaptive setting with secure erasures.

1.1 Full Adaptivity: Adaptive Corruptions of All the Parties

We start by considering the strongest adversary that can adaptively corrupt, and arbitrarily control, any subset of the participating parties. We will focus on the resources required for securely evaluating a function, balancing between the number of rounds, the communication complexity, and the online-computational complexity (the work performed between the first and last messages).

The feasibility of *adaptively secure* MPC was established in the seminal CLOS protocol [21] in a resoundingly strong manner in the UC framework [19]. This paper established the notion of fully adaptive security as described above, in the stronger, erasures-free setting, when the adversary can corrupt *all* protocol parties after execution. They then achieved this notion with a brilliant, yet complicated protocol that worked in the *common random string* model.[5] However, that protocol's round complexity depended on the circuit *depth*, and its communication was polynomially larger than the *size* of the circuit being computed. Roughly 15 years later, Canetti et al. [27] constructed a constant-round protocol under standard assumptions, and recently Benhamouda et al. [10] constructed a 2-round protocol assuming 2-round adaptively secure oblivious transfer (OT). But again, both of these recent results require communication that is larger than the circuit size, and thus come at a larger cost than recent protocols for static corruptions that require two rounds and sublinear communication in the circuit size [71].

Another line of recent work overcomes the communication bottleneck, but at the cost of stronger assumptions and a large *common reference string*. Constant-round [38, 24] and 2-round [46, 26] protocols for adaptively secure MPC are known assuming indistinguishability obfuscation (iO) for circuits and one-way

[5] In the *common random string* model, all parties receive a uniformly random string generated in a trusted setup phase. In the *common reference string* model, the common string is sampled according to some pre-defined distribution.

functions (OWF). These protocols have sublinear communication ([38, 24] in the semi-honest model, [46, 26] in the malicious setting[6]), but require a large CRS (at least linear in the circuit size). In particular, the approach of these results is to place an obfuscated universal circuit into the common reference string which can compute any function of a given size. Thus, these results are more aptly described as *bounded-circuit-size* adaptively secure MPC. In contrast, we aim to study a setup model in which the reference string is smaller (preferably independent) of the size of the evaluated function.

Lastly, recent advances in the static setting [75, 1] presented protocols with online-computation that only depends on the function's *depth* but not on its *size*. In the adaptive setting, on the contrary, all known protocols require computing the function during the online part of the computation.

We now present three results in the fully adaptive setting: a resource-efficient MPC protocol; feasibility and infeasibility results regarding one-sided-optimized two-party protocols; and NIZK protocols with a short proof.

Two-Round MPC with Low Communication and Online-Computation. Thus, the first result of this paper is to present a 2-round fully adaptively secure MPC that requires only sublinear communication (i.e., depends only on the inputs, outputs, and depth of the function), sublinear online-computation, and that uses a sublinear common reference string. To achieve our result, we combine the techniques from the recent work on *Laconic Function Evaluation* (LFE) [75] (that can be instantiated under a natural variant of the learning with errors assumption, called *adaptive LWE (ALWE)*.[7]) and *explainability compilers* [38]. In this sense, our answer to the main question regarding the cost of adaptive security versus static security shows a minimal cost to the communication complexity in the secure-erasures model, and the addition of complexity assumptions in the erasures-free setting: namely the need for sub-exponentially secure iO in order to implement the explainability compiler. Table 1 summarizes the performance characteristics of prior work in comparison to our new result.

Theorem 1 (adaptively secure MPC with sublinear communication, informal). *Assuming ALWE and secure erasures (alternatively, sub-exponential iO), every function can be securely computed by a 2-round protocol tolerating a malicious adversary that can adaptively corrupt all of the parties, such that the communication complexity, the online-computation complexity, and the size of the common reference string are sublinear in the function size.*

[6] The protocols in [38, 24] use the CLOS compiler [21] to get malicious security. Since the communication of previously known adaptively secure ZK protocols depends on the NP relation (see [70, 58, 44] and references therein), the communication of the maliciously secure protocols depended on the CRS. Our short NIZK (Theorem 3) can be used to reduce the communication of [38, 24] in the malicious setting as well.

[7] The basic construction in [75] holds under the standard LWE assumption; however, for the purpose of (semi-)malicious MPC, in which the inputs to the protocol can be chosen adaptively, after the CRS is published, we require the stronger variant.

To explain the key bottleneck in achieving our result, note that almost all known methods for succinct MPC in the static setting rely on *fully homomorphic encryption* [49].[8] The general template is for parties to encrypt and broadcast their inputs, independently evaluate the function on said inputs, and then jointly decrypt the output. The problem in the case of adaptive security is that the simulator must produce a transcript for such a protocol, consisting of the input ciphertexts and the output ciphertext, without knowing the inputs of any parties; later after corruption, the simulator would need to provide a decryption key that explains the ciphertexts for any given input and for the final output. Unfortunately, Katz et al. [68] showed that this exact task is not possible for all functions, even assuming secure erasures, since the existence of such a simulator would imply a compact circuit that can be used to compute the function.

To get around the impossibility of adaptively secure FHE, the key insight of our approach is to instead use a recent technique of laconic function evaluation (LFE) [75], itself an extension of the idea of laconic OT [29]. At a high level, LFE allows a party to publish a short *digest* of a function; later any party can encrypt an input to that function such that the resulting ciphertext is still small with respect to the *size* of the function. In particular, both the digest and the ciphertext size are proportional to the *depth* of the function. Because the computational cost of the decryption algorithm is proportional to evaluating the function, LFE avoids the impossibility argument for adaptive security from [68], while preserving the succinct communication pattern. LFE is in some sense a dual notion to FHE. We extend on this duality in the discussion on the two-party case below.

Our starting point follows the statically secure protocol from [75]. The idea is for the parties to each locally compute a digest of the function f (this is done deterministically, using a CRS for LFE parameters), and then use an MPC protocol (possibly not communication efficient) to jointly compute the encryption of the inputs $(x_1, \ldots, x_n)$. The communication and online-computation required are naturally proportional only to the encryption algorithm, which depends on the depth of the original function but not on its size. Finally, each of the parties can then locally decrypt the ciphertext with respect to the digest to recover the output.

Nonetheless, for adaptive security, it is unclear how to simulate the output ciphertext when possibly all n parties can be corrupted. To circumvent this barrier, we first observe that the protocol from [75] achieves adaptive security in the *erasures* model, without any additional assumptions, and then remove the erasures using the explainability compiler technique from [38]. Loosely speaking, an explainability compiler takes a randomized circuit C and compiles it to a circuit $\widetilde{C}$, computing the same function, along with an additional program $\mathsf{Explain}$, such that given any input/output pair (x, y) the program $\mathsf{Explain}$ can produce coins r satisfying $y = \widetilde{C}(x; r)$.

[8] Another approach for compact MPC is using *function secret sharing (FSS)* [15, 16]. This approach does not seem to support adaptive corruptions.

Overall, this framework achieves all of the round, communication, and online-computation complexity goals, but it still requires a common reference string whose size is related to the *depth* of the function being computed, and further in the erasures-free setting, it relies on iO. In contrast, in the static corruption setting, only LWE is required.

Protocol	**Security (erasures)**	**Rounds**	**Communication**	**Online Computation**	**Setup size**	**Setup type**	**Assumption**
MW [71]	static	2	$\text{poly}(\ell_{\text{in}}, \ell_{\text{out}}, d, \kappa, n)$	$\text{poly}(\|C\|, \kappa)$	$\text{poly}(\kappa, d)$	CRS	LWE, NIZK
QWW [75] ABJMS [1]	static	2	$\text{poly}(\ell_{\text{in}}, \ell_{\text{out}}, d, \kappa, n)$	$\text{poly}(\ell_{\text{in}}, \ell_{\text{out}}, d, \kappa, n)$	$\text{poly}(\kappa, d)$	CRS	ALWE LWE
CLOS [21]	adaptive(no)	$O(d)$	$\|C\| \cdot \text{poly}(\kappa, n)$	$\text{poly}(\|C\|, \kappa)$	$\text{poly}(\kappa)$	CRS	TDP, NCE dense-crypto
GS [47]*	adaptive(no)	$O(d)$	$\|C\| \cdot \text{poly}(\kappa, n)$	$\text{poly}(\|C\|, \kappa)$	-	-	CRH TDP, NCE dense-crypto
DKR [38] CGP [24]	adaptive(no)	$O(1)$	$\|C\| \cdot \text{poly}(\kappa, n)$	$\text{poly}(\|C\|, \kappa)$	$\text{poly}(\|C\|, \kappa)$	Ref	OWF, iO
GP [46]	adaptive(no)	2	$\text{poly}(\ell_{\text{in}}, \ell_{\text{out}}, \kappa, n)$	$\text{poly}(\|C\|, \kappa)$	$\text{poly}(\|C\|, \kappa)$	Ref	OWF, iO
CPV [27]	adaptive(no)	$O(1)$	$\|C\| \cdot \text{poly}(\kappa, n)$	$\text{poly}(\|C\|, \kappa)$	$\text{poly}(\kappa)$	CRS	NCE dense-crypto
BLPV [10]	adaptive(no)	2	$\|C\| \cdot \text{poly}(\kappa, n)$	$\text{poly}(\|C\|, \kappa)$	$\text{poly}(\kappa)$	Ref	adaptive 2-round OT
This work	adaptive(yes) adaptive(no)	2	$\text{poly}(\ell_{\text{in}}, \ell_{\text{out}}, d, \kappa, n)$	$\text{poly}(\ell_{\text{in}}, \ell_{\text{out}}, d, \kappa, n)$	$\text{poly}(\kappa, d)$ $\text{poly}(\ell_{\text{in}}, \ell_{\text{out}}, d, \kappa, n)$	CRS Ref	ALWE ALWE, iO

Table 1: Round, communication, and online-computation of MPC tolerating any number of corruptions, for $f : (\{0,1\}^{\ell_{\text{in}}})^n \to \{0,1\}^{\ell_{\text{out}}}$ represented by a circuit C of depth d. *CRS* refers to a common random string, whereas *Ref* refers to a common *reference* string whose sampling coins are secret. $(*)$ The results in [47] only hold in the stand-alone setting.

Alice/Bob-Optimized protocols. Consider a two-message protocol for two parties, where Alice sends the first message, Bob replies with the second, and only Alice learns the output. In this setting, it is possible for one party's total *computation* (and thus also total communication) to be proportional to the size of their input and output, while the other party "does all of the work" of securely evaluating the function. These protocol variants are designated as "optimized for Alice" or "optimized for Bob," depending on which party saves the work.

In the static-corruption setting, Alice-optimized protocols can be constructed assuming FHE, where Alice encrypts her input, Bob homomorphically evaluates the circuit and returns the encrypted result. Quach et al. [75] showed that Bob-optimized protocols can be constructed from LFE, where Alice compresses the function with her input hard-wired, sends the digest to Bob who replies with the encryption of his input. Therefore, in the static setting, FHE and LFE are dual notions with respect to the work-load of the computation. We next show

that in the adaptive setting this duality breaks. On the one hand, we extend the impossibility result of FHE [68] to rule out adaptively secure 2-round Alice-optimized protocols (even assuming secure erasures). On the other hand, we construct an adaptively secure, semi-malicious,[9] Bob-optimized protocol from LFE and explainability compilers (alternatively, just from LFE assuming secure erasures). We note that any 2-round Bob-optimized protocol can be converted into a 3-round Alice-optimized protocol, which is the best one could hope for. Table 2 summarizes our results vis a vis prior work.

Theorem 2 (Alice/Bob-optimized protocols, informal).

1. *Assuming ALWE and secure erasures (alternatively, sub-exponential iO), there exists an adaptively secure semi-malicious 2PC, where the total communication and Bob's computation are sublinear in the function size and in Alice's input size.*
2. *There exists 2-party functions such that in any adaptively secure, semi-honest, 2-round protocol realizing them, Bob's message must grow linearly in his input, even assuming secure erasures.*

Approach	**Security** (erasures)	**CRS**	**Communication** Alice	Bob	**Computation** Alice	Bob	**Assumptions**
GC [79]	static	-	ℓ_A	$\lvert f\rvert$	$\lvert f\rvert$	$\lvert f\rvert$	static OT
LOT [29]	static	$O(1)$	$O(1)$	$\lvert f\rvert$	$\lvert f\rvert$	$\lvert f\rvert$	DDH, etc.
FHE [49]	static	-	ℓ_A	ℓ_{out}	$\ell_A + \ell_{\mathsf{out}}$	$\lvert f\rvert$	LWE
LFE [75]	static	$O(1)$	$O(1)$	$\ell_B + \ell_{\mathsf{out}}$	$\lvert f\rvert$	$\ell_B + \ell_{\mathsf{out}}$	ALWE
equivocal GC [27]	adaptive (no)	-	ℓ_A	$\lvert f\rvert$	$\lvert f\rvert$	$\lvert f\rvert$	adaptive OT
This work	adaptive (yes)	$O(1)$	$O(1)$	$\ell_B + \ell_{\mathsf{out}}$	$\lvert f\rvert$	$\ell_B + \ell_{\mathsf{out}}$	ALWE
	adaptive (no)	$\ell_B + \ell_{\mathsf{out}}$	$O(1)$	$\ell_B + \ell_{\mathsf{out}}$	$\lvert f\rvert$	$\ell_B + \ell_{\mathsf{out}}$	ALWE and iO
	adaptive (yes)	$\lvert f\rvert$	$\lvert f\rvert$	$\ell_{\mathsf{out}} + o(\ell_B)$	$\lvert f\rvert$	$\lvert f\rvert$	impossible

Table 2: Comparison of two-message semi-honest protocols for $f : \{0,1\}^{\ell_A} \times \{0,1\}^{\ell_B} \to \{0,1\}^{\ell_{\mathsf{out}}}$. Alice talks first, Bob the second, and only Alice learns the output. For simplicity, multiplicative factors that are polynomial in the security parameter κ or the circuit depth d are suppressed.

The key idea behind our Bob-optimized protocol is to use the same LFE approach put forth in [75] for static security, and strengthen it to tolerate adaptive corruptions. To support an adaptive corruption of Alice, the simulator will need to produce an *equivocal* first message, i.e., to simulate the digest without knowing the input value of Alice, and upon a later corruption of Alice generate

[9] In the semi-malicious setting, the adversary follows the protocol as in the semi-honest case, but he can choose arbitrary random coins for corrupted parties.

appropriate random coins explaining the message. Our first technical contribution is to create an equivocal version of the LFE scheme of [75]. Similarly, to support an adaptive corruption of Bob, the simulator should be able to generate an equivocal second message, i.e., generate the ciphertext without knowing the input of Bob, and upon a later corruption of Bob provide appropriate random coins. This can be handled either assuming secure erasures, or using explainability compilers.

Succinct Adaptively Secure NIZK. Next, we consider the problem of constructing an adaptively secure non-interactive zero-knowledge protocol (NIZK) that is "succinct," i.e., the size of the proof and of the common reference string should be smaller than the size of the circuit relation. The best we can hope for is for the proof to be the size of the witness (as otherwise, the lower-bound of Gentry and Wichs [50] requires a non-standard complexity assumption). The first adaptively secure NIZK was constructed by Groth et al. [56], however it was not succinct. Gentry [49] and later Gentry et al. [51] combined FHE with a standard NIZK system to construct such schemes that are secure against static corruptions, and as observed in [51] also against adaptive corruptions in the secure-erasures setting. However, these schemes are not secure against adaptive corruptions in the erasure-free setting. In particular, they run into the FHE bottleneck for adaptive security by Katz et al. [68] described above.

Our main technique to overcome this lower bound is to use *homomorphic trapdoor functions (HTDF)* [53]. HTDF schemes are a primitive that conceptually unites homomorphic encryption and homomorphic signatures. In our usage, HTDF can be thought of as fully homomorphic commitment schemes which are equivocal (hence, statistically hiding), where a trapdoor can be used to open any commitment to any desired value. Using HTDF, the prover can commit to the witness (instead of encrypting it), evaluate the circuit over the commitments, and use adaptive but non-succinct NIZK (e.g., from [56]) to prove knowledge of the witness and that the result commits to 1. The verifier evaluates the circuits over the committed witness, and verifies the NIZK to ensure that the result is a commitment to 1. A summary of our results in comparison to prior work appears in Table 3.

Theorem 3 (short NIZK, informal). *Assuming LWE, if there exists adaptively secure NIZK arguments for NP, there exists adaptively secure NIZK arguments for NP with proof size sublinear in the circuit size of the NP relation.*

1.2 Adaptive Corruptions of a Strict Subset of the parties

Recall that the notion of fully adaptive security allows the adversary to corrupt *all* of the parties in the execution—in which case the protocol offers no privacy of inputs. A criticism of this notion is that it may be too strong for certain applications. In fact, the motivation behind this strong notion arises mainly from its application to *composition* of protocols. Namely, in a larger protocol that involves more parties, participants of a sub-protocol may eventually *all*

Protocol	Security (erasures)	CRS size	Proof size	Assumptions
Groth [55]	static	$\|C\| \cdot \text{poly}(\kappa)$	$\|C\| \cdot \text{poly}(\kappa)$	TDP
Groth [55]	static	$\|C\| \cdot \text{polylog}(\kappa) + \text{poly}(\kappa)$	$\|C\| \cdot \text{poly}(\kappa)$	Naccache-Stern
GOS [56]	adaptive (no)	$\text{poly}(\kappa)$	$\|C\| \cdot \text{poly}(\kappa)$	pairing based
Gentry [49]	adaptive (yes)	$\text{poly}(\kappa)$	$\|w\| \cdot \text{poly}(\kappa, d)$	LWE, NIZK
GGIPSS [51]	adaptive (yes)	$\text{poly}(\kappa)$	$\|w\| + \text{poly}(\kappa, d)$	LWE, NIZK
This work	adaptive (no)	$\text{poly}(\kappa)$	$\|w\| \cdot \text{poly}(\kappa, d)$	LWE, NIZK

Table 3: NIZK arguments with security parameter κ, for circuit size $|C|$, depth d, and witness size $|w|$.

become corrupted, and thus security of the larger protocol will depend on the fully adaptive security of the subprotocol.

It is equally justifiable, however, to consider other protocol-design tasks in which the protocol needs only withstand a weaker adversary who can corrupt either all-but-one of the participants, or—weaker still—only a minority of the players. We next consider adaptive security in these two settings.

All-but-one Corruptions. When considering adaptive security for all-but-one corruptions, Ishai et al. [63] constructed a constant-round, information-theoretically secure protocol in the OT-hybrid model. Garg and Sahai [47] showed an elegant way to instantiate the trusted setup required for [63] using non-black-box techniques and thus constructed a constant-round MPC protocol in the plain model, under standard cryptographic assumptions. The communication in both of these protocols is super-linear in the circuit size.

In contrast, for the weaker notion of static security, Asharov et al. [3] presented a 2-round protocol with sublinear communication, albeit in the *threshold-PKI model*. The threshold-PKI model is a setup in which all the participants of the protocol are privately given individualized key shares corresponding to a public key. A single-round protocol for threshold PKI was also given in [3], yielding a 3-round protocol in a standard CRS setup. Mukherjee and Wichs [71] removed the need for this extra round, thereby presented a 2-round MPC with sublinear communication in the common random string model.

We can thus pose our main question regarding the *cost* of adaptive security for communication-optimal protocols. Recently, Damgård et al. [43] constructed an adaptively secure 3-round MPC protocol with sublinear communication complexity in the threshold-PKI model assuming LWE. Their main idea is to use a special threshold FHE scheme that enables equivocating encryptions of 0 to encryptions of 1. Initially, the parties broadcast encryptions of their inputs. Next, each party locally evaluates the circuit, and the parties re-randomize the evaluated ciphertext in the second round by broadcasting (special) encryptions of 0. The third round is a single-round threshold decryption protocol.

To simulate this protocol, the simulator uses the equivocal mode of the public key. This way, all ciphertexts in the first round are simulated as encryptions of 0. After extracting corrupted parties' inputs, and obtaining the output value, the simulator uses the re-randomizing round to carefully add non-zero encryptions, and force the joint ciphertext to be an encryption of the output. Finally, the threshold decryption protocol is simulated. We note that using the approach of [43] (which is based on [41]), the re-randomization round seems to be inherent, and so it is unclear how to obtain optimal two rounds using this technique.

Our result in this setting is to construct an adaptively secure 2-round MPC assuming non-committing encryption (NCE) and threshold equivocal FHE in the threshold-PKI setup model. The setup assumption can be instantiated using the recent 2-round protocol of [10], assuming 2-round adaptively secure OT, resulting in a 4-round variant in the CRS model. All of the necessary primitives can be instantiated from LWE in the semi-malicious setting, and security in the malicious case follows using NIZK. Table 4 summarizes the prior work and our contribution in this model.

Theorem 4 (all-but-one corruptions, informal). *Assuming LWE and adaptively secure NIZK, every function can be securely computed by a 2-round protocol in the threshold-PKI model tolerating a malicious adversary that can adaptively corrupt all-but-one of the parties such that the communication complexity is sublinear in the function size.*

Protocol	**Security**	**Rounds**	**Communication**	**Assumptions**	**Setup**
AJLTVW [3]	static	2 3	$\mathrm{poly}(\ell_{\mathsf{in}}, \ell_{\mathsf{out}}, d, \kappa, n)$	LWE, NIZK	threshold PKI CRS
MW [71]	static	2	$\mathrm{poly}(\ell_{\mathsf{in}}, \ell_{\mathsf{out}}, d, \kappa, n)$	LWE, NIZK	CRS
IPS [63]	adaptive	$O(1)$	$\lvert C\rvert + \mathrm{poly}(d, \log \lvert C\rvert, \kappa, n)$	OT-hybrid	-
GS [47]	adaptive	$O(1)$	$\lvert C\rvert + \mathrm{poly}(d, \log \lvert C\rvert, \kappa, n)$	CRH, TDP, NCE dense crypto	-
DPR [43]	adaptive	3	$\mathrm{poly}(\ell_{\mathsf{in}}, \ell_{\mathsf{out}}, d, \kappa, n)$	LWE, NIZK	threshold PKI
This work	adaptive	2 4	$\mathrm{poly}(\ell_{\mathsf{in}}, \ell_{\mathsf{out}}, d, \kappa, n)$	LWE, NIZK	threshold PKI CRS

Table 4: Comparison of maliciously secure MPC for $f : (\{0,1\}^{\ell_{\mathsf{in}}})^n \to \{0,1\}^{\ell_{\mathsf{out}}}$ represented by a circuit C of depth d, tolerating $n-1$ corruptions. $(*)$ The results in [47] only hold in the stand-alone model.

Our protocol follows the template of [3], where every party encrypts his input in the first round, locally evaluates the circuit over the ciphertexts, uses its key-share to partially decrypt the result, and broadcasts the decrypted share (some additional "smudging" noise is sometimes required to protect the decryption share). The technical challenges are: (1) the ciphertexts in the first round must

be created in an equivocal way, and (2) the simulation strategy used for the threshold decryption in [3] (and similarly in [71]) is inherently static, and does not translate in a straightforward way to the adaptive setting.

We overcome the first challenge by constructing a novel *threshold equivocal FHE* scheme. The scheme is equipped with an equivocal key-generation algorithm. All ciphertexts encrypted in this mode are "meaningless" and carry no information about the plaintext; a trapdoor can be used to equivocate any ciphertext to any message. We instantiate this FHE scheme using the *dual-mode* HTDF scheme of Gorbunov et al. [53] that can generate the homomorphic trapdoor functions in an extractable mode, corresponding to the standard (meaningful) mode of the FHE, and an equivocal mode, corresponding to the meaningless mode.

We proceed to explain the second challenge. As observed in [3, 71], the threshold decryption protocol may leak some information about the shares of the secret key, and the simulator for the decryption protocol can be used to protect *exactly* one party. In the static setting, when the set of corrupted parties is known ahead of time, the simulator can choose one of the honest parties P_h as a special party for simulating the threshold decryption. This approach does not work in the adaptive setting since the party P_h may get corrupted after simulating the decryption protocol. The simulator cannot know in advance which party will be the last to remain honest. For this reason, we use a different simulation strategy which allows the simulator to "correct" his choice of the party that is simulated as honest for the decryption protocol. Technically, this is done by having each party send shares of zero to each other party over a secure channel (that can be instantiated via NCE). These shares are used to hide the partial decryptions without changing their values. Since shares exchanged between pairs of honest parties remain hidden from the eyes of the adversary, the simulator has more freedom to replace the special party P_h upon corruption, by another honest party, even after simulating the decryption protocol.

Thus, as it stands, the cost of adaptive security with respect to the best statically secure protocols is either the threshold-PKI setup assumption, or the requirement of 2 additional rounds. Removing either of these costs remains an interesting open question.

Honest-Majority Setting. In the honest-majority setting, it is possible to guarantee output delivery to all honest parties. Damgård and Ishai [39] demonstrated the feasibility of constructing adaptively secure protocols that use a constant number of rounds and only require one-way functions. However, the communication of their protocol is super-linear in the circuit size.

In the static-corruption setting, Asharov et al. [3] constructed the first protocol with sublinear communication using threshold FHE; their protocol requires 4 rounds in the threshold-PKI model and 5 rounds in the CRS model. Gordon et al. [54] reduced the round complexity to 2 in the threshold-PKI model or 3 in the CRS model. Recently, Ananth et al. [2] showed a 3-round protocol in the plain model with communication polynomial in the circuit size, and Badrinarayanan

et al. [4] showed a similar result with sublinear communication. Moreover, this round complexity is tight because it is known that 2-round fair protocols are impossible in the CRS model [48, 54, 74].[10]

Our result in this setting is to construct an adaptively secure analogue of [3, 4]. In particular, we construct a 2-round adaptively secure MPC with guaranteed output delivery and the same communication complexity as in the static case, assuming NCE and threshold equivocal FHE in the threshold-PKI model in the semi-malicious setting (all assumptions can be based on LWE). Security in the malicious case follows using NIZK. We can compile our 2-round protocol into a constant-round protocol in the plain with the same communication complexity by computing the threshold-PKI setup using the protocol of Damgård and Ishai [39].

Theorem 5 (honest majority, informal). *Assuming LWE and adaptively secure NIZK, every function can be securely computed with guaranteed output delivery by a 2-round protocol in the threshold-PKI model tolerating a malicious adversary that can adaptively corrupt a minority of the parties such that the communication complexity is sublinear in the function size.*

The 2-round protocol is based on the protocol from the all-but-one case, described in Section 1.2. The challenge lies in overcoming aborting parties to guarantee output delivery. We combine techniques from the threshold FHE of [54] that required $n/2$ decryption shares to reconstruct the output into our threshold equivocal FHE. The main idea is to share the decryption key using Shamir's secret sharing instead of additive secret sharing. Both Shamir's reconstruction and the decryption algorithm consist of linear operations, which make them compatible with each other. As observed by Gordon et al. [54] (see also [14]), the problem with a naïve use of this technique is that the "smudging noise" (used to protect partial decryptions from leakage) is multiplied by the Lagrange coefficients, which may cause an incorrect decryption. Following [54], we have each party secret shares his smudging noise using Shamir's scheme, in a way that is compatible with the reconstruction procedure. We show that this technique can support adaptive corruptions.

To conclude, in the threshold-PKI model, the price of adaptive security is *the same* as of static security in terms of assumptions, number of rounds, and communication complexity. In the plain model, the cost is an additional constant number of rounds. Table 5 summarizes prior work and our results.

1.3 Additional Related Work

Adaptive security tolerating an arbitrary number of corruptions has been considered in various models, including protocols in the CRS model [21, 27, 10], the sunspot model [23], the key-registration model [6], the temper-proof hardware

[10] We emphasize that the lower bounds hold given a public-coin setup, where all parties get the same information, and does not hold given a private-coin setup such as threshold PKI.

Protocol	Security	Rounds	Communication	Assumptions	Setup
AJLTVW [3]	static	4 5	$\mathrm{poly}(\ell_{\mathsf{in}}, \ell_{\mathsf{out}}, d, \kappa, n)$	LWE, NIZK	threshold PKI CRS
GLS [54]	static	2 3	$\mathrm{poly}(\ell_{\mathsf{in}}, \ell_{\mathsf{out}}, d, \kappa, n)$	LWE, NIZK	threshold PKI CRS
ACGJ [2]	static	3	$\lvert C\rvert \cdot \mathrm{poly}(\kappa, n)$	PKE and zaps	-
BJMS [4]	static	2 3	$\mathrm{poly}(\ell_{\mathsf{in}}, \ell_{\mathsf{out}}, d, \kappa, n)$	LWE, zaps, dense crypto	threshold PKI -
DI [39]	adaptive	$O(1)$	$\lvert C\rvert \cdot \mathrm{poly}(\kappa, n)$	OWF	-
This work	adaptive	2 $O(1)$	$\mathrm{poly}(\ell_{\mathsf{in}}, \ell_{\mathsf{out}}, d, \kappa, n)$	LWE, NIZK	threshold PKI -

Table 5: Comparison of maliciously secure MPC for $f : (\{0,1\}^{\ell_{\mathsf{in}}})^n \to \{0,1\}^{\ell_{\mathsf{out}}}$ represented by a circuit C of depth d, in the honest-majority setting.

model [61], the super-polynomial simulation model [5, 59], and more generally, based on UC-puzzles [37, 78]. All of these protocols require super-linear communication complexity.

Adaptive security in the secure-erasures model was considered in [7, 69, 64, 9, 73, 42, 60], and in the erasures-free model tolerating all-but-one corruptions in [66, 63, 57, 43] as well as in the honest-majority setting [36, 41, 39]. With the exception of [43], all of these protocols also require super-linear communication complexity.

Garay et al. [45] considered information-theoretic MPC in the client-server setting, where a constant number of clients uses n servers that assist with the computation, and studied sublinear communication in the number of *servers*. They gave a complete characterization for semi-honest security with static corruptions and adaptive corruptions with or without erasures.

In the static setting, MPC with sublinear communication complexity over eventual-delivery asynchronous channels was constructed in [32]. We conjecture that our techniques can also be applied in the asynchronous setting to obtain adaptive security with low communication.

We note that since the protocol of Garg and Polychroniadou [46] has low communication complexity, and its CRS size depends on the circuit size, it is possible to use a more compact representation of the function, e.g., by a Turing machine (TM) (or a RAM program as considered in [26]), and obfuscate it using iO for Turing machines. Nonetheless, the solution provided in this paper is different in several qualitative aspects. First, to make the CRS independent of the computation at hand, it is preferred to obfuscate a *universal* TM, which receives the description of the concrete TM on its input tape; while iO for TM with *bounded* inputs exists under the same assumptions as iO for circuits [11, 25, 12], iO for TM with *unbounded* inputs is only known under the stronger assumptions of public-coin differing-inputs obfuscation [65]. Second, it is not clear how to replace the iO for TM assumption by secure erasures. Third, the

computation may require a large auxiliary information, e.g., access to a large database, whose description is independent of the TM; this may result with a large description of the function. In our solution, the obfuscated circuit is sublinear in the computation size even when a large auxiliary information is used.

1.4 Open Questions

Our main question is to study the price of adaptive security. Dramatic improvements in the answer to this question have emerged over the past 15 years, and this paper is able to establish almost zero cost in terms of round or communication. Our results, however, leave the following questions as future work.

- **Reducing setup assumptions.** Our results for fully adaptive, 2-round, protocols without erasures require a *common reference string*. Are there fully adaptively secure protocols with sublinear communication complexity in the common *random* string (even with super-constant number of rounds)?
- **Reducing hardness assumptions.** Are there fully adaptively secure protocols with sublinear communication without assuming secure erasures or explainability compilers/iO?
- **Improving setup assumptions/round complexity for all-but-one.** Our optimal-round protocol requires a pre-distribution of the FHE keys. We show a 4-round protocol in the CRS model (equivalently, in the plain model for semi-honest). Are there 2 or 3 round protocols with sublinear communication in the CRS model to match the results for static adversaries?

Paper Organization

In §3, we present our results on fully adaptive security, and in §4.1 and §4.2, we present our results on Bob- and Alice-optimized protocols. In §5, we consider the all-but-one corruption case, and in §6, the honest-majority case. We refer the reader to the full version of the paper [35] for formal definitions and complete proofs.

2 Preliminaries

Basic notations. For $n \in \mathbb{N}$ let $[n] = \{1, \cdots, n\}$. We denote by κ the security parameter. Let poly denote the set all positive polynomials and let PPT denote a probabilistic algorithm that runs in *strictly* polynomial time. A function $\nu \colon \mathbb{N} \to \mathbb{R}$ is *negligible* if $\nu(\kappa) < 1/p(\kappa)$ for every $p \in$ poly and sufficiently large κ. Two distribution ensembles $X = \{X(a,\kappa)\}_{a\in\{0,1\}^*,\kappa\in\mathbb{N}}$ and $Y = \{Y(a,\kappa)\}_{a\in\{0,1\}^*,\kappa\in\mathbb{N}}$ are computationally indistinguishable (denoted $X \stackrel{c}{\equiv} Y$) if no PPT algorithm can tell the difference between them except with negligible probability (in κ).

Cryptographic primitives. In this work, we consider secure protocols in various security settings that require different cryptographic primitives. We present formal definitions for all primitives in the full version [35]. An informal description of every primitive is given before it is used in the main body.

Security model. We present our results in the UC framework. We refer the reader to [19] for a detailed description of the framework.

In our secure function evaluation protocols, we will consider two security notions. In the honest-majority setting, we will consider security with *guaranteed output delivery* [33], informally meaning that all honest parties will receive the correct output from the computation. In general, when an honest majority is not assumed this cannot be achieved [31], and the standard requirement is for *security with abort*, informally meaning that the adversary has the capability to first learn the output from the computation and later force all honest parties to output $\perp$.

Guaranteed output delivery and security with abort are not to be confused with *guaranteed termination*, which means that the honest parties actually finish the protocol. We emphasize that UC protocols cannot provide guaranteed termination since the adversary has full control over the communication channels, and he can simply "hang" the computation. Therefore, following the convention of Canetti et al. [21], we exclude trivial protocols, and require that the properties of guaranteed output delivery or security with abort will hold when the environment provides sufficiently many activations to the parties, and the adversary delivers all messages.[11] In particular, unlike the stand-alone model, in the UC model even when a protocol guarantees output delivery, we allow the adversary to learn the output from the computation while the honest parties do not; however, if an honest party terminates it is guaranteed to receive the output. An alternative, is to work in the $\mathcal{F}_{\mathsf{sync}}$-hybrid model [31] or to consider the framework of [67], which ensures guaranteed termination regardless of the adversary's actions (but, still, as long as the environment provides sufficiently many activations to the parties).

3 Sublinear Communication in the Fully Adaptive Setting

In this section, we consider the fully adaptive setting (where the adversary can corrupt all parties) and construct two-round secure protocols with sublinear communication and online-computation complexity (in the circuit size). Our starting point is the protocol of Quach et al. [75] that is based on *laconic function evaluation (LFE)*.

[11] Other properties such as *privacy* and *independence of inputs* are always required to hold.

3.1 Cryptographic Primitives used in the Protocol

Laconic function evaluation. Informally, an LFE scheme consists of 4 algorithms. The CRS generation algorithm generates a common random string given the security parameter and function parameters (e.g., function depth and input length) $\mathsf{crs} \leftarrow \mathsf{LFE.crsGen}(1^\kappa, \mathsf{params})$. The compression algorithm produces a small digest of a circuit $\mathsf{digest}_C = \mathsf{LFE.Compress}(\mathsf{crs}, C; r)$. The encryption algorithm encrypts the input based on the digest $\mathsf{ct} \leftarrow \mathsf{LFE.Enc}(\mathsf{crs}, \mathsf{digest}_C, x)$. The decryption algorithm decrypts the ciphertext using the random coins used in the compression $y = \mathsf{LFE.Dec}(\mathsf{crs}, C, r, \mathsf{ct})$.

We require the LFE to be correct, i.e., using the notation above it holds that $y = C(x)$, and secure, meaning that the ciphertext can be simulated given the output value y without knowing the input x. LFE can be constructed with the function-hiding property, which ensures that the digest can be simulated based on the function parameters without knowing the function itself. If function hiding is not required (as is the case in this section) the compression algorithm can be made deterministic. We consider the "adaptive" version of LFE, where the inputs to the computation can be chosen *after* the CRS has been sampled. Quach et al. [75] constructed LFE schemes satisfying this property assuming *adaptive LWE*.

Explainability compilers. Informally, an explainability compiler takes as input a description of a randomized algorithm Alg, and outputs two algorithms: $\widetilde{\mathsf{Alg}}$ and $\mathsf{Explain}$. The first algorithm $\widetilde{\mathsf{Alg}}$ computes the same functionality as Alg. The second algorithm $\mathsf{Explain}$ takes an input/output pair (x, y) and produces random coins r such that $y = \widetilde{\mathsf{Alg}}(x; r)$.

Assuming iO for circuits and OWF, Dachman-Soled et al. [38] constructed explainability compilers with *selective* security, where the challenge input is selected independently of the compiled circuit. Explainability compilers with *adaptive* security, where the challenge input is selected based on the compiled circuit follows via complexity leveraging [13] assuming iO and OWF with sub-exponential security (see also [24]). Looking ahead, to support adaptive inputs from the environment, our protocol requires the latter variant.

3.2 Adaptive Security with Sublinear Communication: Secure-Erasures Setting

We will show that assuming LFE every function can be securely realized in the common *random* string model with secure erasures, by a 2-round protocol tolerating an arbitrary number of adaptive corruptions with sublinear communication, online-computation, and CRS size. In Section 3.3, we will show how to replace the secure-erasures assumption by assuming explainability compilers, in which case the protocol requires a common *reference* string.

The basis of our protocol is the 2-round protocol of Quach et al. [75, Thm. 6.2] in the common random string model, that is secure against $n - 1$ static

corruptions and achieves sublinear communication and online-computation assuming the existence of LFE. The protocol from [75] is specified in a hybrid model with an ideally secure computation (with abort) of the function $\mathsf{LFE.Enc}$ (i.e., the $\mathcal{F}_{\text{sfe-abort}}^{\mathsf{LFE.Enc}}$-hybrid model). That is, the ideal functionality receives $(\mathsf{crs}, \mathsf{digest}_f, x_i, r_i)$ from each party P_i and computes

$$\mathsf{ct} = \mathsf{LFE.Enc}(\mathsf{crs}, \mathsf{digest}_f, x_1, \ldots, x_n; \oplus_{i\in[n]} r_i).$$

In case of inconsistent inputs, or if the adversary sends abort, the functionality outputs $\perp$.

Given a circuit C_f computing f, the protocol of [75] is defined as follows:

- The common random string is computed as $\mathsf{crs} \leftarrow \mathsf{LFE.crsGen}(1^\kappa, f.\mathsf{params})$.
- Upon receiving $(\mathsf{input}, \mathsf{sid}, x_i)$, every party P_i computes $\mathsf{digest}_f = \mathsf{LFE.Compress}(\mathsf{crs}, C_f)$, samples a uniformly random $r_i \leftarrow \{0,1\}^*$, and invokes the ideal functionality $\mathcal{F}_{\text{sfe-abort}}^{\mathsf{LFE.Enc}}$ with $(\mathsf{input}, \mathsf{sid}, (\mathsf{crs}, \mathsf{digest}_f, x_i, r_i))$.
- Upon receiving $(\mathsf{output}, \mathsf{sid}, \mathsf{ct})$ from the ideal functionality, party P_i checks that $\mathsf{ct} \neq \perp$ (otherwise, P_i outputs $(\mathsf{output}, \mathsf{sid}, \perp)$), computes $y = \mathsf{LFE.Dec}(\mathsf{crs}, C_f, \mathsf{ct})$, and outputs $(\mathsf{output}, \mathsf{sid}, y)$.

Proving security of the protocol against a static adversary corrupting all-but-one of the parties is straightforward. Namely, by definition of LFE schemes, the simulator can simulate the ciphertext ct based on the output y, and without knowing the input values, as $\mathsf{ct} \leftarrow \mathsf{Sim}_{\text{LFE}}(\mathsf{crs}, C_f, \mathsf{digest}_f, y)$. Furthermore, by the properties of LFE, the size of the circuit computing $\mathsf{LFE.Enc}$ is $\text{poly}(\kappa, \ell_{\mathsf{in}}, \ell_{\mathsf{out}}, d, n)$. By instantiating the ideal functionality using a statically secure 2-round protocol (e.g., the one from [71]), Quach et al. [75] achieved a statically secure protocol with sublinear communication and online-computational complexity.

A closer look at the protocol of [75] shows that it remains secure even facing adaptive corruptions of all-but-one of the parties, since a single honest party suffices to keep the randomness used for $\mathsf{LFE.Enc}$ hidden from the adversary. Furthermore, under the additional assumption of secure erasures, each party can erase his random coins r_i immediately after invoking $\mathcal{F}_{\text{sfe-abort}}^{\mathsf{LFE.Enc}}$, and the protocol can satisfy adaptive corruptions of all the parties. By instantiating the functionality $\mathcal{F}_{\text{sfe-abort}}^{\mathsf{LFE.Enc}}$ with the 2-round adaptively secure MPC from [10], we obtain the following theorem.

Theorem 6 (Theorem 1, secure-erasures version, restated). *Assume the existence of LFE schemes for* $\mathsf{P/poly}$*, of* 2*-round adaptively and maliciously secure OT, and of secure erasures, and let* $f : (\{0,1\}^{\ell_{in}})^n \to \{0,1\}^{\ell_{out}}$ *be an* n*-party function of depth* d*.*

Then, $\mathcal{F}_{\text{sfe-abort}}^{f}$ *can be UC-realized tolerating a malicious, adaptive PPT adversary by a* 2*-round protocol in the common random string model. The size of the common random string is* $\text{poly}(\kappa, d)$*, whereas the communication and online-computational complexity of the protocol are* $\text{poly}(\kappa, \ell_{in}, \ell_{out}, d, n)$*.*

Note that following [75, 10], the assumptions in Theorem 6 hold under the adaptive LWE assumption.

3.3 Adaptive Security with Sublinear Communication: Erasures-Free Setting

In the erasures-free setting, it is unclear how to simulate the output ciphertext, and later upon learning all of the inputs values of the parties, explain the random coins that are used to generate it. We get around this barrier by using explainability compilers.

Two-Round Protocol Assuming Adaptive Explainability Compilers. We consider explainability compilers with adaptive security (where the challenge ciphertext is dynamically chosen) that can be realized by sub-exponentially secure iO and OWF. To define the common reference string for the protocol, we define the distribution $D_{\mathsf{lfe}}(\mathsf{params})$ that is parametrized by an LFE scheme and by the parameters of the function to be computed params. The distribution D_{lfe} computes $\mathsf{crs} \leftarrow \mathsf{LFE.crsGen}(1^\kappa, \mathsf{params})$ and $(\widetilde{\mathsf{LFE.Enc}}, \mathsf{Explain}) \leftarrow \mathsf{Comp}(1^\kappa, \mathsf{LFE.Enc})$, and outputs the reference string $(\mathsf{crs}, \widetilde{\mathsf{LFE.Enc}})$.

We would like to define the protocol in the $\widetilde{\mathsf{LFE.Enc}}$-hybrid model; however, the function $\widetilde{\mathsf{LFE.Enc}}$ is only given in the CRS and is not known before the protocol begins. To get around this technicality, we define the function $f_C((C_1, x_1, r_1), \ldots, (C_n, x_n, r_n))$ that receives a circuit C_i, a value x_i, and random coins r_i from each party, and outputs $C_1(x_1, \ldots, x_n; \oplus r_i)$ in case $C_1 = \ldots = C_n$, or $\perp$ otherwise.

Protocol π_{full}

- **Common Input:** An LFE scheme and a circuit C_f computing the function f.
- **Hybrid model:** The parties have access to the CRS functionality $\mathcal{F}_{\mathsf{crs}}^{D_{\mathsf{lfe}}(f.\mathsf{params})}$ that outputs a crs for the LFE scheme and a circuit $\widetilde{\mathsf{LFE.Enc}}$, and to the SFE functionality $\mathcal{F}_{\mathsf{sfe\text{-}abort}}^{f_C}$.
- **The Protocol:**

1. Upon receiving $(\mathsf{input}, \mathsf{sid}, x_i)$, every party P_i invokes $\mathcal{F}_{\mathsf{crs}}^{D_{\mathsf{lfe}}(f.\mathsf{params})}$ to get $(\mathsf{crs}, \widetilde{\mathsf{LFE.Enc}})$, computes $\mathsf{digest}_f = \mathsf{LFE.Compress}(\mathsf{crs}, C_f)$, samples a uniformly random $r_i \leftarrow \{0,1\}^*$, and invokes $\mathcal{F}_{\mathsf{sfe\text{-}abort}}^{f_C}$ with $(\mathsf{input}, \mathsf{sid}, (\widetilde{\mathsf{LFE.Enc}}, (\mathsf{crs}, \mathsf{digest}_f, x_i), r_i))$.
2. Upon receiving ct from the ideal functionality, party P_i checks that $\mathsf{ct} \neq \perp$ (if so P_i outputs $(\mathsf{output}, \mathsf{sid}, \perp)$), computes $y = \mathsf{LFE.Dec}(\mathsf{crs}, C_f, \mathsf{ct})$, and outputs $(\mathsf{output}, \mathsf{sid}, y)$.

Fig. 1: Two-round SFE with adaptive, malicious security

Theorem 7 (Theorem 1, erasures-free version, restated). *Assume the existence of LFE schemes for* $\mathsf{P/poly}$*, of explainability compilers with adaptive security for* $\mathsf{P/poly}$*, and of 2-round adaptively and maliciously secure OT, and let* $f : (\{0,1\}^{\ell_{in}})^n \to \{0,1\}^{\ell_{out}}$ *be a deterministic n-party function of depth d.*

Then, $\mathcal{F}^{f}_{\mathsf{sfe\text{-}abort}}$ can be UC-realized in the $\mathcal{F}^{D_{\mathsf{lfe}}}_{\mathsf{crs}}$-hybrid model tolerating a malicious, adaptive PPT adversary by a 2-round protocol. The size of the common reference string, the communication complexity, and online-computational complexity of the protocol are $\mathrm{poly}(\kappa, \ell_{in}, \ell_{out}, d, n)$.

The proof of the theorem follows from Lemma 1 (proven in the full version [35]) by instantiating the functionality $\mathcal{F}^{f_C}_{\mathsf{sfe\text{-}abort}}$, that is used to compute $\widetilde{\mathsf{LFE.Enc}}$, using the 2-round protocol from [10] that requires 2-round adaptively and maliciously secure OT.

Lemma 1. *Assume the existence of LFE schemes for* $\mathsf{P/poly}$*, and of explainability compilers with adaptive security for* $\mathsf{P/poly}$*, and let f be a deterministic n-party function. Then, the protocol π_{full} UC-realizes $\mathcal{F}^{f}_{\mathsf{sfe\text{-}abort}}$ tolerating a malicious, adaptive PPT adversary in the $(\mathcal{F}^{D_{\mathsf{lfe}}}_{\mathsf{crs}}, \mathcal{F}^{f_C}_{\mathsf{sfe\text{-}abort}})$-hybrid model.*

4 Adaptively Secure Alice/Bob-Optimized Protocols

In this section, we consider 2-message protocols between Alice and Bob, with respective inputs $x_A \in \{0,1\}^{\ell_A}$ and $x_B \in \{0,1\}^{\ell_B}$, where only Alice learns the output $y = f(x_A, x_B)$. We say that a protocol is "Alice-optimized" if Alice's computation and the total communication of the protocol are proportional to $|x_A| + |y|$, while the computation complexity of Bob is proportional to $|f|$. We say that a protocol is "Bob-optimized" if Bob's computation and the total communication are proportional to $|x_B| + |y|$, while the computation complexity of Alice is proportional to $|f|$.

There exist insecure protocols which are Alice-optimized, where Alice sends her input to Bob who computes the function and returns the output to Alice. Similarly, there exist insecure protocols which are Bob-optimized, where Bob sends his input to Alice when she asks for it, and Alice computes the function on her own.

Assuming FHE [49], there exist statically secure Alice-optimized protocols, where Alice sends her encrypted input to Bob who homomorphically evaluates the function and returns the encrypted output to Alice. Alice's computation and the total communication of the protocol are $(|x_A| + |y|) \cdot \mathrm{poly}(\kappa)$. Assuming function-hiding LFE [75], there exist statically secure Bob-optimized protocols, where Alice sends $\mathsf{digest} \leftarrow \mathsf{LFE.Compress}(\mathsf{crs}, f_{x_A}(\cdot))$ to Bob, who replies with his encrypted input $\mathsf{ct} \leftarrow \mathsf{LFE.Enc}(\mathsf{digest}, x_B)$, and finally Alice recovers the output. Bob's computation and the total communication of the protocol are $(|x_B| + |y|) \cdot \mathrm{poly}(\kappa, d)$, where d is the depth of the function f.

The question we consider is whether there exist adaptively secure protocols which are Alice-optimized or Bob-optimized.

4.1 Adaptively Secure Bob-Optimized Protocol

The elegant protocol from [75] is secure in the common random string model tolerating a static corruption of one of the parties by a semi-malicious adver-

sary (that can choose arbitrary random coins for the corrupted party, but acts honestly otherwise).

Adjusting this protocol to the adaptive setting requires overcoming a few obstacles. Namely, the simulator should be able to generate an equivocal first message, i.e., to simulate the digest without knowing the input value of Alice, and upon a later corruption of Alice generate appropriate random coins explaining the message. Similarly, the simulator should be able to generate an equivocal second message, i.e., generate the ciphertext without knowing the input of Bob, and upon a later corruption of Bob provide appropriate random coins.

To support an adaptive corruption of Alice, we enhance the LFE scheme to support an equivocal mode (see Section 4.1). In this mode, the CRS is generated along with a trapdoor information. The trapdoor can be used to explain a simulated digest as a compression of any circuit with the appropriate parameters. Similarly to Section 3, to support an adaptive corruption of Bob, we can use either secure erasures or explainability compilers.

Theorem 8 (Part 1 of Theorem 2, restated). *Assume the existence of equivocal, function-hiding LFE schemes for* P/poly *and of explainability compilers with adaptive security for* P/poly*, and let* $f : \{0,1\}^{\ell_A} \times \{0,1\}^{\ell_B} \to \{0,1\}^{\ell_{out}}$ *be a deterministic two-party function computable by a depth-*d *circuit.*

Then, $\mathcal{F}_{\mathsf{sfe}}^{f}$ *can be UC-realized tolerating a semi-malicious, adaptive PPT adversary by a* 2*-message protocol in the common reference string model with secure channels. The size of the common reference string, the communication complexity (of both parties), and the computational complexity of Bob are* $(\ell_B + \ell_{out}) \cdot \mathrm{poly}(\kappa, d)$.

The proof of Theorem 8 follows from Lemma 3 below. In the secure-erasures setting, we can remove the explainability compilers assumption, and get the following corollary.

Corollary 1. *Assume the existence of equivocal, function-hiding LFE schemes for* P/poly *and let* f *be a two-party function as above. Then,* $\mathcal{F}_{\mathsf{sfe}}^{f}$ *can be UC-realized in the secure-erasures model tolerating a semi-malicious, adaptive PPT adversary by a* 2*-message protocol in the common random string model with secure channels. The size of the common random string is* $\mathrm{poly}(\kappa, d)$*, and the communication complexity and computational complexity of Bob are* $(\ell_B + \ell_{out}) \cdot \mathrm{poly}(\kappa, d)$.

The secure channels can be instantiated over authenticated channels assuming NCE [20, 40, 30, 34]; however, delivering Bob's public key to Alice requires either an additional communication round or a trusted setup.

Equivocal LFE. We start by extending the notion of LFE to support an equivocal mode.

Definition 1 (equivocal LFE). *A function-hiding LFE scheme* Π *is* equivocal *if there exists a PPT simulator* $(\mathsf{Sim}^{1}_{\text{EQUIV-FH}}, \mathsf{Sim}^{2}_{\text{EQUIV-FH}})$ *for the scheme* Π *such*

that for all stateful PPT adversary $\mathcal{A}$, it holds that

$$\left|\Pr\left[\mathsf{Expt}^{\mathsf{EquivFH\text{-}real}}_{\Pi,\mathcal{A}}(\kappa)=1\right]-\Pr\left[\mathsf{Expt}^{\mathsf{EquivFH\text{-}ideal}}_{\Pi,\mathcal{A}}(\kappa)=1\right]\right|\leq \mathsf{negl}(\kappa),$$

for the experiments $\mathsf{Expt}^{\mathsf{EquivFH\text{-}real}}$ *and* $\mathsf{Expt}^{\mathsf{EquivFH\text{-}ideal}}$ *defined below:*

$\mathsf{Expt}^{\mathsf{EquivFH\text{-}real}}_{\Pi,\mathcal{A}}(\kappa)$	$\mathsf{Expt}^{\mathsf{EquivFH\text{-}ideal}}_{\Pi,\mathcal{A}}(\kappa)$
$\mathsf{params}\leftarrow\mathcal{A}(1^\kappa)$ $\mathsf{crs}\leftarrow\mathsf{LFE.crsGen}(1^\kappa,\mathsf{params})$ $C\leftarrow\mathcal{A}(\mathsf{crs})$ *s.t.* $C\in\mathcal{C}$ *and* $C.\mathsf{params}=\mathsf{params}$ $r\leftarrow\{0,1\}^*$ $\mathsf{digest}=\mathsf{LFE.Compress}(\mathsf{crs},C;r)$ *Output* $\mathcal{A}(\mathsf{crs},\mathsf{digest},r)$	$\mathsf{params}\leftarrow\mathcal{A}(1^\kappa)$ $(\mathsf{crs},\mathsf{digest},\mathsf{state})\leftarrow\mathsf{Sim}^1_{\text{EQUIV-FH}}(1^\kappa,\mathsf{params})$ $C\leftarrow\mathcal{A}(\mathsf{crs})$ *s.t.* $C\in\mathcal{C}$ *and* $C.\mathsf{params}=\mathsf{params}$ $r\leftarrow\mathsf{Sim}^2_{\text{EQUIV-FH}}(C,\mathsf{state})$ *Output* $\mathcal{A}(\mathsf{crs},\mathsf{digest},r)$

In the following lemma (proven in the full version [35] We show that the generic construction of function-hiding LFE from standard LFE presented in [75] can be adjusted to provide equivocality.

Lemma 2. *Assuming the existence of standard LFE schemes and semi-malicious, adaptively secure, 2-round OT, there exists a function-hiding, equivocal LFE scheme.*

We note that both LFE [75] and adaptively and maliciously (hence, also semi-maliciously) secure 2-round OT [10] can be instantiated assuming adaptive LWE. Hence, also equivocal FH-LFE can be instantiated assuming adaptive LWE.

Protocol π_{bob}

- **Common Input:** An LFE scheme and a circuit C_f computing the function f.
- **Notation:** Define the algorithm $\mathsf{LFE.Compress}_{\mathsf{crs},C_f}(x)$ by hard-wiring crs and the circuit C_f to the compression algorithm $\mathsf{LFE.Compress}(\mathsf{crs},C_f(x,\cdot))$, and given input x compress the circuit $C_f(x,\cdot)$ with the input x hard-wired.
- **The Protocol:**

1. Upon receiving $(\mathsf{input},\mathsf{sid},x_A)$, Alice samples uniformly at random $r_A\leftarrow\{0,1\}^*$, computes $\mathsf{digest}=\mathsf{LFE.Compress}_{\mathsf{crs},C_f}(x_A;r_A)$, and sends $(\mathsf{sid},\mathsf{digest})$ to Bob.
2. Upon receiving $(\mathsf{sid},\mathsf{digest})$ from Alice, and having received $(\mathsf{input},\mathsf{sid},x_B)$, Bob computes $\mathsf{ct}\leftarrow\widetilde{\mathsf{LFE.Enc}}(\mathsf{crs},\mathsf{digest},x_B)$, and sends $(\mathsf{sid},\mathsf{ct})$ to Alice.
3. Upon receiving a message $(\mathsf{sid},\mathsf{ct})$ from Bob, Alice computes $y=\mathsf{LFE.Dec}(\mathsf{crs},C,r_A,\mathsf{ct})$ and outputs $(\mathsf{output},\mathsf{sid},y)$.

Fig. 2: 2-round, Bob-optimized protocol with adaptive, semi-malicious security

Semi-Malicious Bob-optimized Protocol. We proceed to our Bob-optimized protocol. Recall that the distribution $D_{\mathsf{lfe}}(\mathsf{params})$ samples a crs for the LFE scheme, computes $(\widetilde{\mathsf{LFE.Enc}},\mathsf{Explain})\leftarrow\mathsf{Comp}(1^\kappa,\mathsf{LFE.Enc})$, and outputs $(\mathsf{crs},\widetilde{\mathsf{LFE.Enc}})$.

Lemma 3. *Consider the notations and assumptions in Theorem 8. Then, protocol π_{bob} securely realizes the functionality $\mathcal{F}_{\mathsf{sfe}}^{f}$ tolerating a semi-malicious, adaptive PPT adversary in the $(\mathcal{F}_{\mathsf{smt}}, \mathcal{F}_{\mathsf{crs}}^{D_{\mathsf{lfe}}(f.\mathsf{params})})$-hybrid model.*

The proof of Lemma 3 can be found in full version [35].

4.2 Impossibility of Adaptively Secure Alice-Optimized Protocol

We now turn to show that the impossibility of adaptively secure FHE from [68] can be extended to rule out adaptively secure Alice-optimized protocols. In fact, we prove a stronger impossibility showing that for some functions the size of Bob's message cannot be smaller than his input, even if Alice's message and the CRS are long. Intuitively, if the output of the function is simply Bob's input, then clearly Bob's message cannot be compressing. We show that this is the case even if the output is short.

For $n \in \mathbb{N}$, we define the two-party functionality $f_n(x_A, g_B) = (g_B(x_A), \lambda)$, where Alice has input $x_A \in \{0,1\}^{\log n}$, Bob has input a function $g_B : \{0,1\}^{\log n} \to \{0,1\}$, represented by its truth table as an n-bit string, and Alice learns the output $g_B(x_A)$.

Theorem 9 (Part 2 of Theorem 2, restated). *Let π_n be a 2-message protocol in the common reference string model for computing f_n, where Alice sends first the message m_1 and Bob replies with the message m_2. If the protocol tolerates a semi-honest, adaptive adversary in the secure-erasures model, then $|m_2| \geq n$.*

Intuitively, by adaptively corrupting Alice and equivocating her input, we can essentially recover $g_B(x_A)$ in any choice of x_A from the protocol transcript. This means that the Bob's response must encode the entire truth table of g_B, which is of size n. The formal proof of Theorem 9 can be found in the full version [35].

5 Adaptive Corruptions of All-But-One of the Parties

In this section, we prove an analogue result in the adaptive setting to the result of Asharov et al. [3], who showed how to compute any function tolerating all-but-one corruptions using a two-round protocol in the threshold-PKI model assuming threshold FHE, which in turn can be instantiated using LWE. Our construction relies on *threshold equivocal FHE* (defined in the full version [35]) that allows simulating ciphertexts for honest parties and explaining them properly upon later corruptions.

We note that the simulation technique used in [3] (and similarly in [71]) does not translate to the adaptive setting. As observed in [3, 71], the threshold decryption protocol may leak some information about the shares of the secret key, and the simulator for the decryption protocol can be used to protect *exactly* one party. Since [3, 71] considered static corruptions, the set of corrupted parties was known ahead of time, and the simulator could choose one of the honest

parties P_h as a special party for the simulation. The decryption protocol was simulated with respect to P_h, as if he is the only honest party. For this reason, proving security of *exactly* $n-1$ corruptions in [71] was considerably simpler than proving security of *up to* $n-1$ corruptions.[12]

The simulation strategy that was used in [3, 71] does not translate to the adaptive setting, since the party P_h that is chosen by the simulator may get corrupted after simulating the decryption protocol. The simulator cannot know in advance which party will be the last to remain honest. For this reason, we use a different simulation strategy, which allows the simulator to "correct" his choice of the party that is simulated as honest for the decryption protocol. Technically, this is done by having each party send shares of zero to each other party over a secure channel (that can be instantiated via NCE). These shares are used to hide the partial decryptions without changing their value. Since shares exchanged between pairs of honest parties remain hidden from the eyes of the adversary, the simulator has more freedom to replace the special party P_h upon corruption, by another honest party, even after simulating the decryption protocol.

5.1 Threshold Equivocal FHE

In the full version [35], we define *equivocal FHE* as an FHE scheme that is augmented with the capability to generate a public key in an "equivocal mode," allowing to explain any ciphertext as an encryption of any value. We show how to construct equivocal FHE from an HTDF scheme, which in turn can be based on LWE. This serves as a stepping stone for *threshold equivocal FHE* which is used in the construction below.

In a threshold FHE scheme, the key-generation and the decryption algorithms are in fact n-party protocols. We consider the simplest case of n-out-of-n threshold FHE and require a single round decryption protocol (following [3, 54, 71, 43]). We note that threshold FHE for more general access structures are also known assuming LWE [14].

Definition 2 (TEFHE). *A threshold equivocal fully homomorphic encryption (TEFHE) is a seven-tuple of algorithms* $(\mathsf{TEFHE.Gen}, \mathsf{TEFHE.Enc}, \mathsf{TEFHE.Eval}, \mathsf{TEFHE.PartDec}, \mathsf{TEFHE.FinDec}, \mathsf{TEFHE.GenEquiv}, \mathsf{TEFHE.Equiv})$ *satisfying the following properties:*

- $\mathsf{TEFHE.Gen}(1^\kappa, 1^d) \to (\mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n)$*: on input the security parameter* κ *and a depth bound* d*, the key-generation algorithm outputs a public key* pk *and* n *secret key shares* $\mathsf{sk}_1, \ldots, \mathsf{sk}_n$*.*
- $\mathsf{TEFHE.Enc}(\mathsf{pk}, \mu) \to \mathsf{ct}$*: on input a public key* pk *and a plaintext* $\mu \in \{0,1\}$*, the encryption algorithm outputs a ciphertext* ct*.*
- $\mathsf{TEFHE.Eval}(\mathsf{pk}, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_\ell) \to \mathsf{ct}$*: on input a public key* pk*, a circuit* $C : \{0,1\}^\ell \to \{0,1\}$*, and a tuple of ciphertexts* $(\mathsf{ct}_1, \ldots, \mathsf{ct}_\ell)$*, the homomorphic evaluation algorithm outputs a ciphertext* ct*.*

[12] We note that the same problem arises also in the threshold FHE scheme for more general access structures [14, Def. 5.5], where the simulation is defined only for *maximal* invalid party sets.

– $\mathsf{TEFHE.PartDec}(i, \mathsf{sk}_i, \mathsf{ct}) \to \mathsf{p}_i$*: on input a secret key share* sk_i *and a ciphertext* ct*, the partial decryption algorithm outputs a partial decryption* p_i*.*
– $\mathsf{TEFHE.FinDec}(\mathsf{pk}, \mathsf{p}_1, \ldots, \mathsf{p}_n) \to \tilde{\mu}$*: on input a public key* pk *and a set* $\{\mathsf{p}_i\}_{i\in[n]}$*, the final decryption algorithm outputs* $\tilde{\mu} \in \{0, 1, \perp\}$*.*
– $\mathsf{TEFHE.GenEquiv}(1^\kappa, 1^d) \to (\mathsf{pk}, \mathsf{td})$*: on input the security parameter* κ *and a depth bound* d*, the equivocal key-generation algorithm outputs a public-key* pk *and a trapdoor* td*.*
– $\mathsf{TEFHE.Equiv}(\mathsf{td}, \mathsf{ct}, m) \to r$*: on input a trapdoor* td*, a ciphertext* ct*, and a plaintext* m*, the equivocation algorithm outputs random coins* r*.*

We require the following properties:

1. The FHE scheme that is defined by setting the decryption key $\mathsf{sk} = (\mathsf{sk}_1, \ldots, \mathsf{sk}_n)$ and the decryption algorithm is composed of executing $\mathsf{TEFHE.PartDec}(i, \mathsf{sk}_i, \mathsf{ct})$ for every $i \in [n]$ followed by $\mathsf{TEFHE.FinDec}(\mathsf{pk}, \mathsf{p}_1, \ldots, \mathsf{p}_n)$ is a correct, compact, and secure equivocal FHE scheme for circuits of depth d.
2. Simulatability of partial decryption: there exists a PPT simulator $\mathsf{Sim}_{\text{TEFHE}}$ such that on input $i \in [n]$, and all decryption keys except of the i'th one $\{\mathsf{sk}_j\}_{j\neq i}$ The following distributions are statistically close:

$$\{\mathsf{p}_i \mid \mathsf{p}_i \leftarrow \mathsf{TEFHE.PartDec}(i, \mathsf{sk}_i, \mathsf{ct})\} \stackrel{c}{\equiv} \{\mathsf{p}'_i \mid \mathsf{p}'_i \leftarrow \mathsf{Sim}_{\text{TEFHE}}(i, \mathsf{ct}, \mu, \{\mathsf{sk}_j\}_{j\neq i})\}\,,$$

   where the keys are set as $(\mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n) \leftarrow \mathsf{TEFHE.Gen}(1^\kappa, 1^d)$, the ciphertext is set as $\mathsf{ct} \leftarrow \mathsf{TEFHE.Eval}(\mathsf{pk}, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_\ell)$ for a circuit $C : \{0,1\}^\ell \to \{0,1\}$ and for $i \in [\ell]$ ciphertext $\mathsf{ct}_i \leftarrow \mathsf{TEFHE.Enc}(\mathsf{pk}, \mu_i)$ with $\mu_i \in \{0,1\}$, and $\mu = C(\mu_1, \ldots, \mu_\ell)$.

In the protocol, we will require some additional properties regarding the key-generation and threshold-decryption protocols.

Definition 3 (special TEFHE). *A special TEFHE is a TEFHE scheme satisfying the following properties:*

1. *On input* 1^κ *and* 1^d*, the key-generation algorithm* $\mathsf{TEFHE.Gen}$ *outputs* $(\mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n)$ *where the public key* pk *defines a prime number* q*, and each secret key* sk_i *is uniformly distributed in* $\mathbb{Z}_q^{n'}$ *for some* $n' = \mathrm{poly}(\kappa, d)$*.*
2. *The partial decryption algorithm* $\mathsf{p}_i \leftarrow \mathsf{TEFHE.PartDec}(i, \mathsf{sk}_i, \mathsf{ct})$ *operates by computing* $\mathsf{p}_i = \langle \mathsf{ct}, \mathsf{sk}_i \rangle + e \mod q$*.*
3. *For every* $v_1, \ldots, v_n \in \mathbb{Z}_q$*, the final decryption algorithm* $\mathsf{TEFHE.FinDec}(\mathsf{pk}, \mathsf{p}_1, \ldots, \mathsf{p}_n)$ *satisfies the following* linearity *property*

$$\mathsf{TEFHE.FinDec}(\mathsf{pk}, \mathsf{p}_1 + v_1, \ldots, \mathsf{p}_n + v_n) = \mathsf{TEFHE.FinDec}(\mathsf{pk}, \mathsf{p}_1, \ldots, \mathsf{p}_n) + \sum_{i\in[n]} v_i.$$

Lemma 4. *Assuming LWE there exist special TEFHE schemes.*

The lemma is proved in the full version [35].

5.2 The Protocol

We define the protocol in the threshold-PKI hybrid model, where a trusted party generates the keys of the TEFHE scheme $(\mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n) \leftarrow \mathsf{TEFHE.Gen}(1^\kappa, 1^d)$ and $(\mathsf{pk}, \mathsf{sk}_i)$ to every P_i. In the full version [35], we probe the following theorem.

Theorem 10. *Assume that special TEFHE exists, let $t < n$, and let $f : (\{0,1\}^{\ell_{in}})^n \to \{0,1\}^{\ell_{out}}$ be an efficiently computable function of depth d. Then, $\mathcal{F}^f_{\mathsf{sfe\text{-}abort}}$ can be UC-realized in the $(\mathcal{F}_{\mathsf{thresh\text{-}pki}}, \mathcal{F}_{\mathsf{smt}})$-hybrid model, tolerating an adaptive, semi-malicious, PPT t-adversary, by a two-round protocol with communication complexity* $\mathrm{poly}(\ell_{in}, \ell_{out}, d, \kappa, n)$.

Protocol $\pi_{\mathsf{allbutone}}$

- **Private Input:** Every party P_i, for $i \in [n]$, has private input $x_i \in \{0,1\}^{\ell_{\mathsf{in}}}$.
- **Common Input:** A special TEFHE scheme Π and a circuit C_f of depth d.
- **The Protocol:**

1. Upon receiving $(\mathsf{input}, \mathsf{sid}, x_i)$, party P_i proceeds as follows:
   (a) Invoke $\mathcal{F}_{\mathsf{thresh\text{-}pki}}(\Pi, d)$ with $(\mathsf{init}, \mathsf{sid})$ to receive $(\mathsf{sid}, \mathsf{pk}, \mathsf{sk}_i)$. Let q be the prime associated with the public key pk (as per Definition 3).
   (b) Encrypt the input as $\mathsf{ct}_i \leftarrow \mathsf{TEFHE.Enc}(\mathsf{pk}, x_i)$.
   (c) Sample random $\mathsf{s}_i^1, \ldots, \mathsf{s}_i^n \leftarrow \mathbb{Z}_q$, conditioned on $\sum_{j=1}^n \mathsf{s}_i^j = 0 \mod q$.
   (d) Send $(\mathsf{sid}, \mathsf{ct}_i, \mathsf{s}_i^j)$ to P_j over a secure channel (via $\mathcal{F}_{\mathsf{smt}}$).
2. In case some party aborts, output $(\mathsf{output}, \mathsf{sid}, \perp)$ and halt. Otherwise, upon receiving $(\mathsf{sid}, \cdot)$ messages from all the parties, party P_i proceeds as follows:
   (a) Compute $\mathsf{ct} = \mathsf{TEFHE.Eval}(\mathsf{pk}, C_f, \mathsf{ct}_1, \ldots, \mathsf{ct}_n)$.
   (b) Partially decrypt the result as $\mathsf{p}_i = \mathsf{TEFHE.PartDec}(i, \mathsf{sk}_i, \mathsf{ct})$.
   (c) Set $\mathsf{m}_i = \mathsf{p}_i + \sum_{j=1}^n \mathsf{s}_j^i \mod q$ and send $(\mathsf{sid}, \mathsf{m}_i)$ to every party.
3. In case some party aborts, output $(\mathsf{output}, \mathsf{sid}, \perp)$ and halt. Otherwise, upon receiving $(\mathsf{sid}, \cdot)$ from all the parties, party P_i runs the final decrypt as $y = \mathsf{TEFHE.FinDec}(\mathsf{pk}, \{\mathsf{m}_1, \ldots, \mathsf{m}_n\})$ and outputs $(\mathsf{output}, \mathsf{sid}, y)$.

Fig. 3: 2-round MPC with semi-malicious security

Malicious Security with Sublinear Communication. Asharov et al. [3] provided a round-preserving compiler from semi-maliciously security to maliciously security in the static setting assuming NIZK. In the full version [35], we prove security of this compiler in the adaptive setting. We note that following the GMW paradigm, it is important that the semi-malicious protocol can be defined purely over a broadcast channel, however, the protocol in Section 5.2 uses secure channels. To resolve this issue, the secret shares of zero that were sent over secure point-to-point channels should be encrypted and transmitted over the broadcast channel. As we consider adaptive corruptions, we need to use non-committing encryption and each non-committing public key should be used to encrypt n elements in $\mathbb{Z}_q$. We consider the distribution of the NCE public keys as part of the threshold-PKI functionality. Alternatively, the public keys can be exchanged at the cost of an additional communication round.

Theorem 11 (Theorem 4, restated). *Assume the existence of special TEFHE schemes and NCE schemes, let $t < n$, and let $f : (\{0,1\}^{\ell_{in}})^n \to \{0,1\}^{\ell_{out}}$ be an efficiently computable function of depth d. Then, $\mathcal{F}^f_{\mathsf{sfe\text{-}abort}}$ can be UC-realized in the $(\mathcal{F}_{\mathsf{thresh\text{-}pki}}, \mathcal{F}_{\mathsf{bc}}, \mathcal{F}_{\mathsf{nizk}})$-hybrid model, tolerating an adaptive, malicious, PPT t-adversary, by a two-round protocol with communication complexity* $\mathrm{poly}(\ell_{in}, \ell_{out}, d, \kappa, n)$.

6 The Honest-Majority Setting

In this section, we show how to adjust the protocol from Section 5 that provides security with abort, into guaranteeing output delivery in the honest-majority setting. We apply some of the techniques from [54] on our adaptively secure protocol designed for the all-but-one setting, and achieve a matching result tolerating adaptive corruptions.

In the all-but-one case (Section 5) the decryption key was shared using additive secret sharing. As observed in [54], since the decryption of the GSW-based threshold FHE consists of linear operations, it is possible to use Shamir's secret sharing [77] instead. The problem with a naïve use of this idea is that when the partial decryptions are reconstructed, each decryption share is multiplied by the Lagrange coefficient, and thus also the smudging noise. This will result in blowing up the noise and may end up with an incorrect decryption. Gordon et al. [54] overcame this problem by having each party secret share (using Shamir's scheme) its smudging noise in the first round of the protocol, and parties added shares of the smudging noise of non-aborting parties in a way that is compatible with the decryption algorithm.[13]

In the full version [35], we adjust the definition of TEFHE to support $n/2$-out-of-n secret sharing, prove existence under LWE, and use it for proving the following theorem.

Theorem 12. *Assume the existence of special $n/2$-out-of-n TEFHE schemes, let $t < n/2$, and let $f : (\{0,1\}^{\ell_{in}})^n \to \{0,1\}^{\ell_{out}}$ be an efficiently computable function of depth d. Then, $\mathcal{F}^f_{\mathsf{sfe\text{-}god}}$ can be UC-realized in the $(\mathcal{F}_{\mathsf{thresh\text{-}pki}}, \mathcal{F}_{\mathsf{smt}})$-hybrid model, tolerating an adaptive, semi-malicious, PPT t-adversary, by a two-round protocol with communication complexity* $\mathrm{poly}(\ell_{in}, \ell_{out}, d, \kappa, n)$.

Similarly to the previous section, using the semi-malicious to malicious compiler, we obtain the following corollary.

Theorem 13. *Consider the same assumptions as in Theorem 12. Then, $\mathcal{F}^f_{\mathsf{sfe\text{-}god}}$ can be UC-realized in the $(\mathcal{F}_{\mathsf{thresh\text{-}pki}}, \mathcal{F}_{\mathsf{bc}}, \mathcal{F}_{\mathsf{nizk}})$-hybrid model, tolerating an adaptive, malicious PPT t-adversary, by a two-round protocol with communication complexity* $\mathrm{poly}(\ell_{in}, \ell_{out}, d, \kappa, n)$.

[13] Recently, Boneh et al. [14] showed that this problem can be overcome in a different way, by using a special secret sharing scheme that ensures the Lagrange coefficients are binary values.

Bibliography

[1] P. Ananth, S. Badrinarayanan, A. Jain, N. Manohar, and A. Sahai. From FE combiners to secure MPC and back. *IACR Cryptology ePrint Archive*, 2018:457, 2018.

[2] P. Ananth, A. R. Choudhuri, A. Goel, and A. Jain. Round-optimal secure multiparty computation with honest majority. In *CRYPTO '18, part II*, pages 395–424, 2018.

[3] G. Asharov, A. Jain, A. López-Alt, E. Tromer, V. Vaikuntanathan, and D. Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. In *EUROCRYPT '12*, pages 483–501, 2012.

[4] S. Badrinarayanan, A. Jain, N. Manohar, and A. Sahai. Secure MPC: laziness leads to GOD. *IACR Cryptology ePrint Archive*, 2018:580, 2018.

[5] B. Barak and A. Sahai. How to play almost any mental game over the net - concurrent composition via super-polynomial simulation. In *FOCS*, pages 543–552, 2005.

[6] B. Barak, R. Canetti, J. B. Nielsen, and R. Pass. Universally composable protocols with relaxed set-up assumptions. In *FOCS*, pages 186–195, 2004.

[7] D. Beaver and S. Haber. Cryptographic protocols provably secure against dynamic adversaries. In *EUROCRYPT '92*, pages 307–323, 1992.

[8] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In *STOC*, pages 1–10, 1988.

[9] R. Bendlin, I. Damgård, C. Orlandi, and S. Zakarias. Semi-homomorphic encryption and multiparty computation. In *EUROCRYPT '11*, pages 169–188, 2011.

[10] F. Benhamouda, H. Lin, A. Polychroniadou, and M. Venkitasubramaniam. Two-round adaptively secure multiparty computation from standard assumptions. In *TCC '18, part I*, pages 175–205, 2018.

[11] N. Bitansky, S. Garg, H. Lin, R. Pass, and S. Telang. Succinct randomized encodings and their applications. In *STOC*, pages 439–448, 2015.

[12] N. Bitansky, R. Canetti, S. Garg, J. Holmgren, A. Jain, H. Lin, R. Pass, S. Telang, and V. Vaikuntanathan. Indistinguishability obfuscation for RAM programs and succinct randomized encodings. *SICOMP*, 47(3):1123–1210, 2018.

[13] D. Boneh and X. Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *EUROCRYPT '04*, pages 223–238, 2004.

[14] D. Boneh, R. Gennaro, S. Goldfeder, A. Jain, S. Kim, P. M. R. Rasmussen, and A. Sahai. Threshold cryptosystems from threshold fully homomorphic encryption. In *CRYPTO '18, part I*, pages 565–596, 2018.

[15] E. Boyle, N. Gilboa, and Y. Ishai. Breaking the circuit size barrier for secure computation under DDH. In *CRYPTO '16, part I*, pages 509–539, 2016.

[16] E. Boyle, N. Gilboa, and Y. Ishai. Group-based secure computation: Optimizing rounds, communication, and computation. In *EUROCRYPT '17, part II*, pages 163–193, 2017.

[17] E. Boyle, R. Cohen, D. Data, and P. Hubáček. Must the communication graph of MPC protocols be an expander? In *CRYPTO '18, part III*, pages 243–272, 2018.

[18] R. Canetti. Security and composition of multiparty cryptographic protocols. *JCRYPTOL*, 13(1):143–202, 2000.

[19] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *FOCS*, pages 136–145, 2001.
[20] R. Canetti, U. Feige, O. Goldreich, and M. Naor. Adaptively secure multi-party computation. In *STOC*, pages 639–648, 1996.
[21] R. Canetti, Y. Lindell, R. Ostrovsky, and A. Sahai. Universally composable two-party and multi-party secure computation. In *STOC*, pages 494–503, 2002.
[22] R. Canetti, I. Damgård, S. Dziembowski, Y. Ishai, and T. Malkin. Adaptive versus non-adaptive security of multi-party protocols. *JCRYPTOL*, 17(3):153–207, 2004.
[23] R. Canetti, R. Pass, and A. Shelat. Cryptography from sunspots: How to use an imperfect reference string. In *FOCS*, pages 249–259, 2007.
[24] R. Canetti, S. Goldwasser, and O. Poburinnaya. Adaptively secure two-party computation from indistinguishability obfuscation. In *TCC '15, part II*, pages 557–585, 2015.
[25] R. Canetti, J. Holmgren, A. Jain, and V. Vaikuntanathan. Succinct garbling and indistinguishability obfuscation for RAM programs. In *STOC*, pages 429–437, 2015.
[26] R. Canetti, O. Poburinnaya, and M. Venkitasubramaniam. Better two-round adaptive multi-party computation. In *PKC*, pages 396–427, 2017.
[27] R. Canetti, O. Poburinnaya, and M. Venkitasubramaniam. Equivocating yao: constant-round adaptively secure multiparty computation in the plain model. In *STOC*, pages 497–509, 2017.
[28] D. Chaum, C. Crépeau, and I. Damgård. Multiparty unconditionally secure protocols (extended abstract). In *STOC*, pages 11–19, 1988.
[29] C. Cho, N. Döttling, S. Garg, D. Gupta, P. Miao, and A. Polychroniadou. Laconic oblivious transfer and its applications. In *CRYPTO '17, part II*, pages 33–65, 2017.
[30] S. G. Choi, D. Dachman-Soled, T. Malkin, and H. Wee. Improved non-committing encryption with applications to adaptively secure protocols. In *ASIACRYPT '09*, pages 287–302, 2009.
[31] R. Cleve. Limits on the security of coin flips when half the processors are faulty (extended abstract). In *STOC*, pages 364–369, 1986.
[32] R. Cohen. Asynchronous secure multiparty computation in constant time. In *PKC*, pages 183–207, 2016.
[33] R. Cohen and Y. Lindell. Fairness versus guaranteed output delivery in secure multiparty computation. *JCRYPTOL*, 30(4):1157–1186, 2017.
[34] R. Cohen and C. Peikert. On adaptively secure multiparty computation with a short CRS. In *SCN*, pages 129–146, 2016.
[35] R. Cohen, A. Shelat, and D. Wichs. Adaptively secure MPC with sublinear communication complexity, 2019. https://eprint.iacr.org/2018/1161.
[36] R. Cramer, I. Damgård, S. Dziembowski, M. Hirt, and T. Rabin. Efficient multiparty computations secure against an adaptive adversary. In *EUROCRYPT '99*, pages 311–326, 1999.
[37] D. Dachman-Soled, T. Malkin, M. Raykova, and M. Venkitasubramaniam. Adaptive and concurrent secure computation from new adaptive, non-malleable commitments. In *ASIACRYPT '13, part I*, pages 316–336, 2013.
[38] D. Dachman-Soled, J. Katz, and V. Rao. Adaptively secure, universally composable, multiparty computation in constant rounds. In *TCC '15, part II*, pages 586–613, 2015.
[39] I. Damgård and Y. Ishai. Constant-round multiparty computation using a black-box pseudorandom generator. In *CRYPTO '05*, pages 378–394, 2005.
[40] I. Damgård and J. B. Nielsen. Improved non-committing encryption schemes based on a general complexity assumption. In *CRYPTO '00*, pages 432–450, 2000.

[41] I. Damgård and J. B. Nielsen. Universally composable efficient multiparty computation from threshold homomorphic encryption. In *CRYPTO '03*, pages 247–264, 2003.
[42] I. Damgård, V. Pastro, N. P. Smart, and S. Zakarias. Multiparty computation from somewhat homomorphic encryption. In *CRYPTO '12*, pages 643–662, 2012.
[43] I. Damgård, A. Polychroniadou, and V. Rao. Adaptively secure multi-party computation from LWE (via equivocal FHE). In *PKC*, pages 208–233, 2016.
[44] C. Ganesh, Y. Kondi, A. Patra, and P. Sarkar. Efficient adaptively secure zero-knowledge from garbled circuits. In *PKC*, pages 499–529, 2018.
[45] J. A. Garay, Y. Ishai, R. Ostrovsky, and V. Zikas. The price of low communication in secure multi-party computation. In *CRYPTO '17, part I*, pages 420–446, 2017.
[46] S. Garg and A. Polychroniadou. Two-round adaptively secure MPC from indistinguishability obfuscation. In *TCC '15, part II*, pages 614–637, 2015.
[47] S. Garg and A. Sahai. Adaptively secure multi-party computation with dishonest majority. In *CRYPTO '12*, pages 105–123, 2012.
[48] R. Gennaro, Y. Ishai, E. Kushilevitz, and T. Rabin. On 2-round secure multiparty computation. In *CRYPTO '02*, pages 178–193, 2002.
[49] C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178, 2009.
[50] C. Gentry and D. Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In *STOC*, pages 99–108, 2011.
[51] C. Gentry, J. Groth, Y. Ishai, C. Peikert, A. Sahai, and A. D. Smith. Using fully homomorphic hybrid encryption to minimize non-interative zero-knowledge proofs. *JCRYPTOL*, 28(4):820–843, 2015.
[52] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *STOC*, pages 218–229, 1987.
[53] S. Gorbunov, V. Vaikuntanathan, and D. Wichs. Leveled fully homomorphic signatures from standard lattices. In *STOC*, pages 469–477, 2015.
[54] S. D. Gordon, F. Liu, and E. Shi. Constant-round MPC with fairness and guarantee of output delivery. In *CRYPTO '15, part II*, pages 63–82, 2015.
[55] J. Groth. Short pairing-based non-interactive zero-knowledge arguments. In *ASIACRYPT '10*.
[56] J. Groth, R. Ostrovsky, and A. Sahai. New techniques for noninteractive zero-knowledge. *J. ACM*, 59(3):11:1–11:35, 2012.
[57] C. Hazay and A. Patra. Efficient one-sided adaptively secure computation. *JCRYPTOL*, 30(1):321–371, 2017.
[58] C. Hazay and M. Venkitasubramaniam. On the power of secure two-party computation. In *CRYPTO '16, part II*, pages 397–429, 2016.
[59] C. Hazay and M. Venkitasubramaniam. Composable adaptive secure protocols without setup under polytime assumptions. In *TCC '16-B, part I*, pages 400–432, 2016.
[60] C. Hazay, Y. Lindell, and A. Patra. Adaptively secure computation with partial erasures. In *PODC*, pages 291–300, 2015.
[61] C. Hazay, A. Polychroniadou, and M. Venkitasubramaniam. Constant round adaptively secure protocols in the tamper-proof hardware model. In *PKC*, pages 428–460, 2017.
[62] M. Hirt and V. Zikas. Adaptively secure broadcast. In *EUROCRYPT '10*, pages 466–485, 2010.
[63] Y. Ishai, M. Prabhakaran, and A. Sahai. Founding cryptography on oblivious transfer - efficiently. In *CRYPTO '08*, pages 572–591, 2008.

[64] Y. Ishai, M. Prabhakaran, and A. Sahai. Secure arithmetic computation with no honest majority. In *TCC '09*, pages 294–314, 2009.
[65] Y. Ishai, O. Pandey, and A. Sahai. Public-coin differing-inputs obfuscation and its applications. In *TCC '15, part II*, pages 668–697, 2015.
[66] J. Katz and R. Ostrovsky. Round-optimal secure two-party computation. In *CRYPTO '04*, pages 335–354, 2004.
[67] J. Katz, U. Maurer, B. Tackmann, and V. Zikas. Universally composable synchronous computation. In *TCC '13*, pages 477–498, 2013.
[68] J. Katz, A. Thiruvengadam, and H. Zhou. Feasibility and infeasibility of adaptively secure fully homomorphic encryption. In *PKC*, pages 14–31, 2013.
[69] Y. Lindell. Adaptively secure two-party computation with erasures. In *CT-RSA*, pages 117–132, 2009.
[70] Y. Lindell and H. Zarosim. Adaptive zero-knowledge proofs and adaptively secure oblivious transfer. *JCRYPTOL*, 24(4):761–799, 2011.
[71] P. Mukherjee and D. Wichs. Two round multiparty computation via multi-key FHE. In *EUROCRYPT '16, part II*, pages 735–763, 2016.
[72] J. B. Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In *CRYPTO '02*, pages 111–126, 2002.
[73] J. B. Nielsen, P. S. Nordholt, C. Orlandi, and S. S. Burra. A new approach to practical active-secure two-party computation. In *CRYPTO '12*, pages 681–700, 2012.
[74] A. Patra and D. Ravi. On the exact round complexity of secure three-party computation. In *CRYPTO '18, part II*, pages 425–458, 2018.
[75] W. Quach, H. Wee, and D. Wichs. Laconic function evaluation and applications. In *FOCS*, pages 859–870, 2018.
[76] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In *FOCS*, pages 73–85, 1989.
[77] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.
[78] M. Venkitasubramaniam. On adaptively secure protocols. pages 455–475, 2014.
[79] A. C. Yao. How to generate and exchange secrets (extended abstract). In *FOCS*, pages 162–167, 1986.