



# Adaptive Security, Erasures, and Network Assumptions in Communication-Local MPC

Nishanth Chandran<sup>1</sup>, Juan Garay<sup>2</sup>, Ankit Kumar Misra<sup>3</sup>, Rafail Ostrovsky<sup>3</sup>, and Vassilis Zikas<sup>4</sup>

<sup>1</sup> Microsoft Research, Bengaluru, India

[nichandr@microsoft.com](mailto:nichandr@microsoft.com)

<sup>2</sup> Texas A&M University, Texas, USA

[garay@tamu.edu](mailto:garay@tamu.edu)

<sup>3</sup> University of California, Los Angeles, USA

[{ankitkmisra,rafaail}@cs.ucla.edu](mailto:{ankitkmisra,rafaail}@cs.ucla.edu)

<sup>4</sup> Purdue University, West Lafayette, USA

[vzikas@cs.purdue.edu](mailto:vzikas@cs.purdue.edu)

**Abstract.** The problem of reliable/secure all-to-all communication over low-degree networks has been essential for *communication-local* (CL)  $n$ -party MPC (i.e., MPC protocols where every party directly communicates only with a few, typically polylogarithmic in  $n$ , parties) and more recently for communication over *ad hoc* networks, which are used in blockchain protocols. However, a limited number of adaptively secure solutions exist, and they all make relatively strong assumptions on the ability of parties to act in some specific manner before the adversary can corrupt them. Two such assumptions were made in the work of Chandran *et al.* [ITCS '15]—parties can (a) *multisend* messages to several receivers simultaneously; and (b) *securely erase* the message and the identities of the receivers, before the adversary gets a chance to corrupt the sender (even if a receiver is corrupted).

A natural question to ask is: Are these assumptions necessary for adaptively secure CL MPC? In this paper, we characterize the feasibility landscape for all-to-all reliable message transmission (RMT) under these two assumptions, and use this characterization to obtain (asymptotically) tight feasibility results for CL MPC.

- First, we prove a strong impossibility result for a broad class of RMT protocols, termed here *store-and-forward* protocols, which includes all known communication protocols for CL MPC from standard cryptographic assumptions. Concretely, we show that no such protocol with a certain expansion rate can tolerate a constant fraction of parties being corrupted.
- Next, under the assumption of only a PKI, we show that assuming secure erasures, we can obtain an RMT protocol between all pairs of

---

R. Ostrovsky—This research was supported in part by NSF grants CNS-2246355, CCF-2220450, US-Israel BSF grant 2022370, and by Sunday Group.

parties with polylogarithmic locality (even without assuming multi-send) for the honest majority setting. We complement this result by showing a negative result for the setting of dishonest majority.

- Finally, and somewhat surprisingly, under stronger assumptions (i.e., trapdoor permutations with a reverse domain sampler, and compact and malicious circuit-private FHE), we construct a polylogarithmic-locality all-to-one RMT protocol, which is adaptively secure and tolerates any constant fraction of corruptions, without assuming either secure erasures or multisend. This last result uses a novel combination of adaptively secure (e.g., non-committing) encryption and (static) FHE to bypass the impossibility of compact adaptively secure FHE by Katz *et al.* [PKC’13], which we believe may be of independent interest. Intriguingly, even such assumptions do not allow reducing all-to-all RMT to all-to-one RMT (a reduction which is trivial in the non-CL setting). Still, we can implement what we call *sublinear output-set* RMT (SOS-RMT for short). We show how SOS-RMT can be used for SOS-MPC under the known bounds for feasibility of MPC in the standard (i.e., non-CL) setting assuming, in addition to SOS-RMT, an anonymous PKI.

**Keywords:** Secure Multiparty Computation · Communication Locality

## 1 Introduction

### 1.1 Communication Locality and Adaptive Security

Secure multi-party computation (MPC) [4, 19, 31, 47] allows a set of  $n$  parties to securely compute a function on their joint private data. Initial work on MPC focused on feasibility, and it was followed by a series of works on improving round and communication complexity. Envisioning the potential need to deploy MPC on massive networks, novel works on *scalable* MPC (e.g., [20, 21, 25, 34]) have investigated settings and techniques that allowed for protocols with communication complexity that grows (asymptotically) slower than the size of the player set. Boyle, Goldwasser, and Tessaro [8] put forth a different metric that is very relevant for the design of massive-scale MPC, namely, *communication locality* (CL). The CL of a party in a protocol is the number of parties that this party sends/receives messages to/from, via a direct point-to-point channel, through the execution of the protocol; as such, the CL of a protocol is the maximum CL of any party. Motivated by the potential application of MPC to privately executing sublinear algorithms in a distributed manner, [8] proposed a solution which achieves MPC with a sublinear (i.e., polylogarithmic in  $n$ ) CL tolerating a (sub-optimal) number of  $t < (1/3 - \epsilon)n$  actively corrupted parties, for  $\epsilon > 0$ .

The original solution in [8] only considered static corruptions and relied on the existence of a public-key infrastructure (PKI), a common reference string (CRS), semantically secure public-key encryption and existentially unforgeable signatures. Chandran *et al.* [15] improved on the above result to tolerate an asymptotically *optimal* number of  $t < (1/2 - \epsilon)n$  *adaptive* active corruptions,

for an arbitrary small constant  $\epsilon$ . Their construction relied on the same assumptions except for the CRS, which was replaced by a *hidden (random) graph*: A suitable random graph on  $n$  vertices (with sublinear degree) that is sampled by a trusted entity and where each party is given its neighborhood in this graph. However, parties do not know the other parties' neighborhoods, and, most importantly, the *adversary* does not know the (honest) neighbors of honest parties. As shown in [15], this random graph can be realized via a standard symmetric-key infrastructure (SKI)—wherein every two parties share a (secret) symmetric-key encryption key. The emulation is simple: Every two nodes locally use their symmetric key as a seed to a PRF to derive sufficiently long pseudorandomness that can be utilized to decide (locally and independently) whether or not the parties should have an edge between them in any given round. In fact, we mention that, as noted in [14], the above hidden graph (or, equivalently, SKI) assumption can be replaced by standard number-theoretic cryptographic assumptions (such as DDH and its resulting PKI), allowing non-interactive key exchange (NIKE, for short—cf. [24]). The simple idea is that the PKI can be used to non-interactively establish the SKI required in [15], which can be used to derive the hidden graph.

The core challenge associated with such sublinear CL protocols is propagating information and connecting any two parties using a sparse (i.e., sublinear degree) communication graph. In such a context, one needs to route messages through the induced (incomplete) communication graph so that the adversary cannot block (even indirect) communication between any two honest nodes, thus disconnecting the graph. Indeed, since any party can only directly communicate with a sublinear number of neighbors, the only way for it to reach all parties in the network is by means of a *gossiping* protocol. In [8], gossiping was done via a routing protocol based on hierarchical routing and sorting networks that cleverly knit the paths to ensure each message travels over sufficiently many paths, making it impossible for the adversary to block it.

The above gossiping protocol works for a static adversary corrupting  $t < (1/3 - \epsilon)n$  parties. However, when one considers stronger adversaries, with an asymptotically optimal (for MPC) corruption threshold—i.e.,  $t < (1/2 - \epsilon)n$ —and, most importantly, adaptive adversaries, the problem becomes even more challenging, as message routing through the incomplete graph turns into a “cat-and-mouse” game—more formally, a *graph discovery* game—with the adversary using an initial set of corrupted parties to try to discover possible message routes and block them. In [15], properties of a hidden Erdős-Rényi graph were used along with a clever use of edges in a disposable manner (where every edge was used only once) in order to win the above graph discovery game, and devise a sublinear-locality communication protocol for the problem of *reliable message transmission* (RMT) between any two honest nodes, which allowed every honest party to reach every other honest party. The protocol from [15] tolerates an arbitrary constant fraction of the parties being adaptively (and actively) corrupted. RMT protocols can then be used to trivially construct secure message transmission (SMT) protocols—informally, these are protocols which emulate a secure, i.e., private and authenticated, channel between a sender and a receiver—in the

model assuming a PKI, which can then be used to build communication local MPC protocols<sup>1</sup>.

## 1.2 Erasures and Network Assumptions in CL Protocols

The protocol in [15] relied on the aforementioned setup and hardness assumptions, namely, a PKI, an SKI, and the existence of enhanced one-way permutations—the latter being a common minimal assumption for MPC. In addition, an assumption was made in [15] which is not essential for MPC, but, as we show in this paper, turns out to be *necessary* for sublinear communication when natural gossiping protocols are used. In more detail, the assumption of *secure erasures* [12]—namely, that honest parties can erase whichever part of their state they wish, in a way that if they are corrupted later on, the adversary cannot recover the erased information—is only needed for a subset of adaptively secure MPC protocols without the sublinear CL restriction [28]. The construction from [15], however, assumes not just erasures, but actually two levels of strengthening of the assumption: First, it also assumes an *atomic multisend* capability [33], which in a nutshell ensures that if a player  $p$  attempts to send a message to a subset  $Q$  of the player set in some given round, then either all honest parties will receive the message or none of them will<sup>2</sup>. However, even assuming secure erasures in addition to such a rushing adversary proves not to be sufficient for the protocol in [15]. The reason is that when a message is sent to a polylogarithmic (in  $n$ ) number of parties, then one of these parties may be corrupted, in which case the adversary can corrupt the sender and learn who the other receivers were, *before* the sender has had a chance to erase their identities, and corrupt them too, thereby completely neutralizing the sender. In fact, the inability of the adversary to mount such an attack is essential in [15]’s security proof. In order to exclude this attack, [15] also assumes that multisend and erase can jointly be done as an atomic operation—i.e.,  $p$  can send his message so that it is received by all the parties in  $Q$  and erase their identities before the adversary is able to corrupt him.

## 1.3 Our Results

The above state of affairs leaves open several questions regarding the minimal assumptions required for sublinear locality in all-to-all communication (and therefore also in MPC) in the adaptive security setting. In particular, it leaves open the question of the necessity and sufficiency of secure erasures, atomic multisend, and their atomic combination as mentioned above. In this paper we

---

<sup>1</sup> As a side note, an interesting side effect of the recent popularity of blockchain protocols is that, as they also rely on gossiping for communication, results on the feasibility of sublinear-communication protocols provide insights on basic feasibility questions in the blockchain context as well (see related work for further details).

<sup>2</sup> As shown in [33, 36], this property is impossible to obtain from simple point-to-point communication in the standard adaptive and rushing adversary setting [10].

provide a characterization of this landscape, as depicted in Table 1. The impossibility results in Table 1 are for adaptively secure all-to-all reliable message transmission, i.e., the task of allowing every party  $p_i$  to send a (potentially different) message to every party  $p_j$  in a reliable, i.e., authenticated manner—where  $p_j$  becomes aware that the message was sent by  $p_i$ —so that the adversary cannot block or alter the message exchanges between any pair of honest parties. As noted earlier, this serves as a building block for SMT and CL MPC protocols. These results apply to a broad class of protocols which we call *store-and-forward* (SF) protocols, which, intuitively, allow intermediate parties to only store and forward previously received messages, and (for the non-erasure case) under an *expansion-rate* assumption, which mandates that messages originating from any neighbors of a sender will reach a large (polylogarithmic size) set relatively fast (i.e., before they reach their respective receiver) (see Definitions 1 and 2). This includes several natural message propagation and gossiping protocols, and in particular all those used in the CL literature (see Sect. 2 for a discussion).

The positive results in Table 1 are for all-to-one RMT, i.e., there is one receiver who everyone wishes to send messages to. As we shall show, assuming erasures, the feasibility results can be extended to all-to-all RMT. However, intriguingly, in the non-erasure setting, the protocol can only be extended to allow for a sublinear (polylogarithmic) set of receivers. Even so, the assumption of an additional setup, namely, an *anonymous* PKI, allows us to rescue the situation: We show that we can implement a new notion of MPC, which we term *sublinear output-set* MPC (SOS-MPC for short). Likewise, we use the term SOS-RMT to refer to the formerly obtained notion of RMT with a sublinear set of receivers. Intuitively, in an anonymous PKI setting, parties have access to a PKI but do not know which public-key corresponds to which party. Such a setup is common in YOSO-style MPC protocols which have become highly relevant in the blockchain literature [30]. Our newly defined SOS-MPC is similar to standard MPC (i.e., the inputs of all parties are accounted for in the computation) but only a (random) subset of the parties of sublinear size receives the output from the computation. We note that SOS-MPC is sufficient for the motivating applications of CL MPC, namely, secure computation of sublinear algorithms, where the output is by definition far smaller than the input (cf. [9]). This leaves open the question of feasibility or impossibility of (all-to-all RMT and) standard MPC in this setting.

Our positive results are of two flavors. Assuming secure erasures under standard assumptions (i.e., one-way functions), we provide SF protocols for RMT tolerating the (asymptotically) optimal number of corruptions, as implied by the impossibility result of Theorem 3. This leaves open the following important question:

*Is it possible to construct an RMT protocol in the no-erasures setting that is adaptively secure against a constant fraction of corruptions?*

We answer this question in the affirmative, albeit using strong cryptographic assumptions. Concretely, assuming trapdoor permutations with a reversed domain sampler and the existence of a malicious, compact, and circuit-private

**Table 1.** A characterization of feasibility of reliable communication (RMT) under the different assumptions: Atomic multisend (A-MS), secure erasures (Erasures), and multi-send and secure erasures as an atomic operation (A-MSE). All negative results are for all-to-all RMT. The positive results are for all-to-one RMT; but all except Theorem 4 are extended to all-to-all RMT, whereas Theorem 4 is extended to SOS-RMT. Note that SF stands for store-and-forward as defined in Definition 1, and expansion rate is as defined in Definition 2.

A-MS	Erasures	A-MSE	
✓	✓	✓	[15]: Assuming <i>PKI and one-way functions</i> , there exists an (SF) sublinear locality RMT protocol tolerating $t < (1 - \epsilon)n$ corruptions, for any $\epsilon > 0$ .
✓	✗	✗	<b>Theorem 1:</b> There exists no SF polylogarithmic locality RMT protocol with expansion rate $(\text{polylog}(n), \frac{k \log n}{c \log \log n})$ tolerating a constant fraction of corruptions, for any $k < 1$ , where the degree of the communication graph is $O(\log^c n)$ .
✗	✓	✗	<b>Theorem 2:</b> Assuming <i>PKI and one-way functions</i> , there exists an (SF) polylogarithmic locality RMT protocol tolerating $t < (\frac{1}{2} - \epsilon)n$ corruptions, for any $0 < \epsilon < \frac{1}{2}$ . <b>Theorem 3:</b> There exists no SF polylogarithmic locality RMT protocol tolerating $t > (\frac{1}{2} + \epsilon)n$ corruptions, for any $0 < \epsilon < \frac{1}{2}$ .
✗	✗	✗	<b>Theorem 4:</b> Assuming <i>PKI, trapdoor permutations with a reversed domain sampler and compact and malicious circuit-private FHE</i> , there exists a polylogarithmic locality RMT protocol tolerating $t < (1 - \epsilon)n$ corruptions, for any $\epsilon > 0$ .

fully-homomorphic encryption (FHE) scheme [42], we can construct a protocol, which is not SF, and thus circumvents our impossibility result, allowing for RMT tolerating any constant fraction of corruptions. Its construction relies on a novel combination of adaptively secure (e.g., non-committing) encryption [12] and (statically) secure FHE to obtain a homomorphic encryption scheme that, although not fully homomorphic, allows to compute a class of circuits sufficient for our RMT, while providing adaptive security (including deniability), a property which is known to be impossible for general FHE [37].

Regarding our MPC feasibility results, using techniques from [15], we can “lift” all the above all-to-all RMT feasibility results on adaptive all-to-all communication to adaptively secure MPC, under the assumption of enhanced trapdoor permutations, or any other assumption that would allow for corruption-optimal adaptively secure MPC over a complete point-to-point network. Similarly, we can lift the SOS-RMT results to SOS-MPC.

## 1.4 Related Work

Although introduced as a notion explicitly for standard MPC by Boyle, Goldwasser, and Tessaro [9], the idea of low communication locality was already implicit in a number of works on *almost-everywhere* secure (communication, Byzantine agreement, and computation) protocols [17, 18, 23, 27, 39, 40, 44]. Such protocols can operate over incomplete networks (or under-utilize a complete

network to achieve low CL as we do here) but “give up” security for a number of parties; such so-called *doomed* parties might lose their input privacy, and/or contribute a false input to or receive a false output from the computation. The goal of such almost-everywhere secure protocols is then to achieve optimal tradeoffs between number of corruptions and number of doomed parties. We refer to [16] for a recent formal treatment of and detailed literature review of almost-everywhere secure protocols.

As already mentioned, Chandran *et al.* [15] improved on the resiliency of the protocol from [9] and brought adaptive security to the model, at the cost, however, of the strong atomic erase-and-multisend assumption, which restricts the ability of an adaptive adversary to attack the protocol, as we discuss in detail in Sect. 2. The results from [15] relied on a hidden-graph setup which, by construction, was an expander graph. The follow-up work by Boyle *et al.* [6] provided the first solutions to the problem for PRAM-based MPC, which in addition achieved some load balancing properties. Following that, Boyle *et al.* [7] investigated the question of whether an expander is in fact needed for sublinear locality MPC, answering it in the negative.

Also related to our goals are works that explicitly target sublinear per-party communication complexity. In this context, Dani *et al.* [22] presented a statically secure information-theoretic MPC protocol with a per-party communication complexity of  $O(\sqrt{n})$  tolerating  $t < n/3$  corruptions. King and Saia [38] showed how to construct a Byzantine agreement (BA) protocol that is secure against adaptive corruptions, where the communication complexity of every party is  $\tilde{O}(\sqrt{n})$ , which leads to a BA protocol with  $\tilde{O}(n)$  communication locality tolerating  $t < (\frac{1}{3} - \epsilon)n$  corruptions.

Also related to our work is the work of Matt *et al.* [41], who consider a weakening of the adversary’s adaptivity, which they term *delayed adaptive corruption*. Here the adversary who wants to corrupt a party needs to first indicate its intention, say, in round  $r$ , but the actual corruption does not take effect until a few rounds later. Despite being useful for making statements about the load balance and delivery guarantees of blockchain-inspired message propagation protocols, in the context of sublinear locality we are considering, this assumption trivializes feasibility questions. Indeed, the latter protocols tend to use parties (network nodes) as “disposable” relays, i.e., once a party successfully relays its message to its neighbors, corrupting it does not buy the adversary anything. This fact, in combination with the delayed adaptive corruption assumption (which would imply that the adversary’s ability to corrupt is **slower** than the message propagation), would prevent the adversary from adaptively blocking discovered paths.

## 2 Model

**Notation.** Here we present some basic notation used throughout the paper. We denote by  $[n]$  the set  $[n] = \{1, \dots, n\}$ .  $\mathcal{P} = \{p_1, \dots, p_n\}$  denotes the set of parties participating in the MPC protocol. We will often refer to parties as nodes in a network. We will assume that the adversary is able to corrupt a number  $t < n$

of the parties; it will be convenient for our exposition to express corruption in terms of a fraction  $\tau$  of the total number of parties; hence  $t = \tau n$ , for  $0 < \tau < 1$ .

In any directed graph over  $\mathcal{P}$ , we will use  $\rho_{i,j}$  to denote the length of the shortest path from node  $i \in \mathcal{P}$  to node  $j \in \mathcal{P}$ . Further, we will denote by  $\Gamma_q(u)$  the set of all nodes (not including  $u$ ) that are at forward distance  $\leq q$  from node  $u \in \mathcal{P}$ , and we will define  $\gamma_q(u) = |\Gamma_q(u)|$ . Hence,  $\Gamma_1(u)$  denotes the set of all outgoing neighbors of  $u$ . Analogously, we will denote by  $\Gamma_q^{\text{in}}(u)$  the set of all nodes  $v \in \mathcal{P}$  such that  $v \in \Gamma_q(v)$ . Hence,  $\Gamma_1^{\text{in}}(u)$  denotes the set of all incoming neighbors of  $u$ .

## 2.1 Adversarial Model

Next, we turn to defining the communication and adversary model for adaptively secure computation with communication locality. We note that most works in this area leave several of the model assumptions implicit or unspecified. Instead, since our goal is to provide a complete feasibility landscape given such assumptions, we need to take a more rigorous and detailed approach to the specification of the model.

Consistently with the classical MPC literature, we assume that parties are connected with each other via a complete network of secure (i.e., authenticated and private) point-to-point channels [5, 32]. However, since each party can “talk” to only a sublinear (i.e., polylogarithmic in  $n$ ) number of other parties, no party will be using all its point-to-point channels. The communication is synchronous, which means all parties advance in a round-based manner, where whenever the round switches everyone is informed, and messages sent in any round  $r$  are guaranteed to be delivered by the beginning of the following round  $r + 1$ —unless the sender gets corrupted during  $r$  and no *multi-send* capability is assumed (see below). We will consider an adaptive and rushing adversary who might actively corrupt parties during the protocol execution.

The combination of sublinear communication locality and synchrony with such an adversary brings up a number of modeling challenges, described below.

**Localized Notification.** In classical synchronous point-to-point networks, the adversary is always informed when a party  $p_i$  sends a message to another party  $p_j$  via their direct point-to-point channel. Notifying the adversary about such a transmission implicitly captures the assumption that the adversary has a **global** view of the entire network, including runtime-observable events, such as messages being transmitted. This gives him the ability to induce a worst-case (arbitrary/adversarial) scheduling of messages that makes for stronger security.

However, when we shift to settings with vastly large sets of parties—these are the settings where sublinear-locality protocols become relevant—the assumption that the adversary has a complete view of all the events that occur in the network might be too strong. In fact, it is impossible to achieve adaptive security with sublinear locality in this worst-case setting. Indeed, in low-locality settings<sup>3</sup>, a

<sup>3</sup> Here “low” specifically means asymptotically smaller than the adversary’s corruption budget.

party might only communicate with a small number of its neighbors. Hence, if the adversary is able to detect that an honest party  $p$  attempts to send a message to another honest party, then he can simply corrupt the receiver and block this transmission path; by performing this attack on every transmission of  $p$  the adversary will be able to isolate the party from the rest of the network, making *all-to-all* communication (and therefore “full” MPC) impossible<sup>4</sup>. Hence, for such settings, it is natural and relevant to limit the visibility to the adversary in message transmission events to only events happening near his neighborhood, i.e., assume that the adversary only observes message transmissions on channels that are incident to a neighborhood where he is present—e.g., channels in the immediate neighborhoods of parties currently under his control. In our model, we make this assumption—which is necessary for sublinear-locality communication and MPC, and therefore implicit in all relevant works—explicit by assuming that the adversary is only able to observe a transmission when the sender or the receiver of that transmission is corrupted.

**Adaptive Adversarial Scheduling.** The above localized notification assumption is natural in large networks (and necessary for adaptive corruption), but it does create a challenge with allowing the adversary to perform worst-case scheduling: Since the adversary does not have a full view of which honest-to-honest channels are used in each round (recall that only a small, sublinear per party, number is utilized), how can he induce a worst-case scheduling? In fact, although the above localized-notification assumption was implicit in [15], this delicate issue was not addressed.

To address it, we look back at the classical way of defining scheduling for an adaptive and rushing adversary in [10]: Every round is split in “mini-rounds”, where in each mini-round the adversary can have one party  $p_i$  send his message to another party  $p_j$ . If in the protocol every party communicates with all other parties in each round (a common protocol structure in feasibility results) then this means that each round has  $n^2$  mini-rounds. The adaptive adversary is able to corrupt parties between any two mini-rounds. Note that, as explicitly discussed in [10], this means that a worst-case adversary would first deliver all messages sent to corrupted receivers and then schedule the remaining messages in an adaptive manner. To allow for worst-case scheduling in our model, we rely on the exact same idea: In each round  $r$ , the adversary operates in  $n^2$  mini-rounds, where each such mini-round corresponds to a unique (ordered) pair of parties  $(p_i, p_j)$ , allowing  $p_i$  to send its  $r$ th round message to  $p_j$ . The only difference here is that if and while both  $p_i$  and  $p_j$  are honest, the adversary does not learn whether or not a message was sent on the  $(p_i, p_j)$  channel in that round. (Of course, if either  $p_i$  or  $p_j$  gets corrupted down the line, then the adversary will find out at the point of corruption whether a message was exchanged in round  $r$ , unless the corrupted party has had a chance to erase before becoming corrupted—see discussion about erasures below.)

---

<sup>4</sup> Here, we use “full” MPC to refer to the standard MPC formulation where no party is left out; this is in contrast to “almost-everywhere” MPC [27], where some of the parties are not given any correctness and privacy guarantees.

**Trusted Setup Assumptions.** We assume classical correlated randomness setups: The parties have access to a public-key infrastructure (PKI), which they can use for digital signatures and public-key encryption. In addition, the parties are given an appropriately sampled hidden random graph setup with polylogarithmic degree [15]. As discussed in [15], and already mentioned earlier, under standard hardness assumptions (i.e., existence of pseudo-random generators) this graph setup can be replaced by a different correlated randomness setup, namely, a secret key infrastructure (SKI); alternatively, if the PKI allows for non-interactive key exchange (NIKE), then any other setup assumption is not needed since an SKI can be created by pairwise invocation of the NIKE protocol—since this is non-interactive, it does not result in increased communication locality.

**Secure Erasures.** Secure erasures are common (and, in fact, necessary in SF protocols) for adaptively secure MPC and, as we prove here, for point-to-point communication over a sublinear-degree network graph. The assumption is that (honest) parties are able to erase any part of their internal state (including parts of their setup and/or randomness) so that if they get corrupted later on, the adversary does not have access to the erased information. We note that in a model where such erasures are possible, such actions take place responding to protocol instructions, and therefore the adversary corrupting a party is allowed to learn that an erasure was performed by the party in the past.

## 2.2 Atomicity Assumptions

**Atomicity of Actions.** One of the most important parameters of any adaptive adversary setting, which is often left as implicit, is the question of atomicity of operations for the honest party. A block of operations is considered atomic if, once an honest party starts performing them, it is allowed to complete them before the adversary gets a chance to act (e.g., perform additional corruptions). Clearly, the higher the number of operations that are bundled in an atomic block, the harder the job of the adversary becomes. One of the standard uses of atomicity in the distributed computing and cryptographic protocols literature is the so-called *atomic multi-send*, where if in a given round a party is supposed to send a message to multiple parties, it is allowed to do so without any in-between adversarial interference. One can view this assumption as a way to restrict the rushing ability of the adaptive adversary. As such, in the recent literature the (setting of an) adaptive adversary over a network of standard (point-to-point) channels (i.e., with non-atomic multisend) is at times referred to as *strongly rushing* [1, 46] and the term “rushing” is used to refer to the setting where the adaptive adversary operates over a network with atomic-multisend channels.

In this work we use the term “rushing” consistently with [10, 11, 26, 33] to characterize the adversary (i.e., its ability to schedule the delivery of honest parties’ messages) rather than the setting of network and adversary. Hence, atomic multisend is a network/protocol-related atomicity assumption: When the protocol instructs a party  $p$  to deliver a message to several parties in some set  $\mathcal{Q}$  in the same round (or even a vector  $(m_{p_1}, \dots, m_{p_Q})$  of messages, where each  $m_{p_i}$  is

to be sent to  $p_i \in \mathcal{Q}$ ), then the party can send all those messages as an atomic operation, meaning that these messages will be delivered to their intended recipients at the beginning of the following round; in particular, the adversary cannot corrupt  $p$  in the middle of the transmission and enforce that some  $p_i \in \mathcal{Q}$  receives the message  $m_{p_i}$  from  $p$  while some other  $p_j \in \mathcal{Q}$  does not.

**Atomicity Assumptions Used in This Work.** The way different feasibility assumptions are bound together in an atomic operation has an impact on the feasibility of reliable communication (and therefore MPC) that one is able to prove. As our goal is to investigate the different relevant assumptions for sublinear- (polylogarithmic-) locality MPC, we next include a detailed discussion on the ways these assumptions can be bound (and have been bound in prior literature):

- *No erasures/no (atomic) multisend (NE-NAMS):* This is the worst-case network and erasures model (fourth row of Table 1): (1) No honest party is allowed to erase its internal state and the adversary, upon corrupting a party, learns that party’s entire prior and current state, and (2) the parties send their messages one by one as discussed in the adaptive scheduling above.
- *No erasures/(atomic) multisend (NE-AMS):* This corresponds to the second row of Table 1: (1) (Lack of) erasures are as above, and (2) the parties can atomically multisend their messages.
- *Erasures/no (atomic) multisend (E-NAMS):* This corresponds to the third row of Table 1. We assume erasures but no atomic multisend. Hence: (1) In each mini-round where a party  $p_i$  is allowed to “speak” (we say the party is *activated*); i.e., in each miniround corresponding to a pair  $(p_i, p_j)$  for some  $p_j$ , the party  $p_i$  can first erase and then send, but it cannot erase the message it is about to send (including the identity of the receiver) until the next miniround when the party is activated, and (2) messages are sent in a one-by-one manner (in mini-rounds as above), where between any two mini-rounds the adversary can act.
- *Erasures/(atomic) multisend (E-AMS):* (1) Erasures are as in the previous case, and (2) when a party is activated for sending, it is allowed to erase and then send to a set  $\mathcal{Q}$  of parties (but as above, it cannot erase this set or the messages it sends until the next time it is activated). Note that this case allows an adversary who has corrupted a party in  $\mathcal{Q}$  to learn the message, corrupt the sender, learn the identities of all other parties in  $\mathcal{Q}$  (since the sender is not given a chance to erase before) and corrupt all of these parties thereby blocking this message.
- *Atomic erasures and multisend (AE-AMS):* This is the strongest of the atomicity assumptions (first row of Table 1) and corresponds to the model considered in past works on sublinear locality MPC with adaptive corruptions (e.g., [7, 15]). In this case, whenever a party is activated for sending a message, it is allowed to send to a set  $\mathcal{Q}$  of parties (where all are guaranteed to receive their messages at the beginning of the following round) *and* perform erasures after sending is complete and before the adversary has a chance to corrupt this party. This, in particular, means that even if the adversary controls one

of the parties in  $\mathcal{Q}$ , he is still unable to learn who the other parties in  $\mathcal{Q}$  are even by corrupting the sender (as before corruption the sender is able to erase their identities).

### 3 Technical Overview

**Impossibility of Store-and-Forward Without Erasures.** Our first technical contribution (see Sect. 4) is an impossibility result for all-to-all (in fact, even one-to-all) store-and-forward RMT with a high expansion rate, if we do not assume erasures. In particular, we show this impossibility for expansion rate  $(\log^z(n), \frac{k \log n}{c \log \log n})$ , for all  $z > 1$  and  $k < 1$ , where the degree of the communication graph is  $O(\log^c n)$ . Intuitively, our definition of  $(L, \ell)$  expansion rate (see Definition 2) captures the constraint that, in an honest execution of the protocol, when the sender’s message reaches parties up to a distance  $\ell$  from himself, his message also reaches at least  $L$  parties through each of his neighbors. To our knowledge, this property can be shown to hold for all CL MPC protocols in the current literature. The result holds independently of whether or not we assume atomic multisend. The proof utilizes a combination of graph-theoretic and protocol results and can be summarized as follows: The first step in the proof shows that, due to the polylogarithmic locality assumption, there must exist a pair  $(p_s, p_r)$  of sender and receiver that are far enough from one another, concretely, in distance greater than  $\frac{k \log n}{c \log \log n}$  (Lemma 2). Looking ahead, this will be the pair the adversary will try to disconnect. The expansion rate assumption will then ensure that for each forward neighbor  $p$  of  $p_s$  in the RMT to  $p_r$ , the graph rooted at  $p$  that is created by coloring the communication graph as the message of  $p_s$  (intended for  $p_r$ ) passes through the intermediate nodes, will have sufficiently many colored nodes before it reaches  $p_r$ , so that with high probability one of them can be adversarial. Once this happens, the adversary will be able to follow the thread backwards all the way to this neighbor  $p$  and then corrupt  $p$  and all parties that  $p$  has sent the message to, thereby eliminating  $p$  as a possible relay. Since each such  $p$  can be eliminated with high probability, and  $p_s$  had at most polylogarithmically many neighbors, the above adversary will be able to capture all these neighbors (and the colored graphs rooted at them) before the message reaches  $p_r$  (and without corrupting  $p_s$ ). This adversary can drop all the message passing from captured nodes, thereby disconnecting  $p_s$  and  $p_r$  and violating the security of the RMT. The detailed proof is given in Sect. 4.

We remark that, although the above SF subclass might appear somewhat simple, it captures the structure of several natural message-propagation and gossiping protocols—in particular those used in the context of blockchains, as well as in so-called *store-and-forward* (switching) networks (cf. [3]). In fact, as demonstrated in Lemma 1, the class of SF RMT protocols with expansion-rate parameter that fall within our impossibility range includes all known message-propagation protocols in the sublinear locality MPC realm. We stress that our impossibility is not intended as a way to tightly characterize the feasibility landscape of CL RMT in the non-erasure setting, but rather to abstract the core of

the above protocols that makes them inadequate against adaptive adversaries in this setting. Notwithstanding, we are not aware of any technique that yields a protocol in the non-erasure setting, as even common approaches for anonymous communication, e.g., onion-routing-based protocols, seem insufficient in this highly adversarial setting (as discussed in Sect. 1.4).

**The Power of Secure Erasures for CL Protocols.** As discussed above, the above impossibility (for SF RMT) protocols renders existing communication protocols in the CL MPC (and blockchain-via-gossip) literature insecure when erasures are not assumed. Continuing our exploration of the landscape, we turn to the question of how far can the secure-erasures assumption take us in terms of feasibility of RMT. Recall that [15] proved that if erasures and multisend can be performed/bundled in an atomic operation, then any adversary corrupting any constant fraction of the parties can be tolerated in RMT. Here we ask what happens if we unbounded these two assumptions, i.e., assume either only erasures, or erasures and multi-send as separate operations. And we shoot for the strongest possible results: (1) Feasibility even without atomic multisend and (2) Impossibility even with atomic multisend.

For the first (feasibility) result, our starting point is the RMT protocol from [15], which at a high level operates as follows: The sender sends his signed inputs to his hidden graph Round-1 neighbors<sup>5</sup> and whenever a party receives a message in some round  $rnd$ , he relays it (in the next round,  $rnd + 1$ ) to his  $(rnd + 1)$ -round hidden-graph (forward) neighbors<sup>6</sup>.

The above protocol does not work here, since the communicating parties can be cut off by the attack described in Sect. 1.2; namely, a corrupted node  $p$  who receives a message from an honest neighbor  $q$  can corrupt  $q$  and all of  $q$ 's neighbors before any chance of erasure. To make the above protocol secure when (just) erasures are assumed, we make the following modification: Every round is assigned to exactly one party in  $\mathcal{P}$ , who, if he has a message to send, sends this message to exactly one of his hidden graph (forward) neighbors at a time, and then (in the next activation/round) erases both the fact that he sent it and the relevant (used) edge from his hidden graph setup (i.e., the edge pointing to the neighbor he just contacted). Importantly, unlike [15], where as soon as a party relays a message he does not need to do any more relaying, we require every party that has received a message to keep relaying it to its hidden graph(s) neighbors. By forwarding messages to neighbors one at a time and erasing in between, the adversary is prevented from corrupting the neighbors of honest relayers who get corrupted, enabling a message to propagate into the network even after a past relayer is compromised. Note that to allow for worst-case attacks, one needs to devise a careful structure that gives the adversary sufficient attack-opportunities. We ensure this by forcing on our protocol a structure that makes erasures slow enough, so that the adversary is given enough time to attack (see Sect. 5.1.1).

<sup>5</sup> Since we will be running at most polylogarithmic-round protocols, we can assume wlog that the hidden graph is a multi-graph consisting of polylogarithmic, independent copies of polylogarithmic-degree hidden graph setup.

<sup>6</sup> Recall that the hidden graph is directed.

The above protocol is clearly SF, and it even has an expansion rate that matches the parameter of our non-erasure impossibility theorem. Hence, one might be tempted to believe that if the adversary corrupts a constant fraction of the parties, then with good probability he will be able to block the message in a similar way as in our no-erasures impossibility proof. However, contrary to the above intuition, we show that any adversary corrupting at most  $t < (1/2 - \epsilon)n$  of the parties can be tolerated (with overwhelming probability). The proof follows a careful probabilistic argument: Since, from a Chernoff bound, more than half of the hidden graph neighbors of any party are honest, we can show that the above protocol will create an avalanche effect: With good probability, for several rounds from the start of the above protocol the set of honest parties that has received the message will keep growing and it will become large enough to be guaranteed to include a party who is one hop (in the hidden graph) from the RMT receiver. Once this happens, it is game over for the adversary since this party will relay the message to the RMT receiver in the following round.

The above result establishes (one-to-one) RMT assuming an honest majority and erasures only (Theorem 2) and as a corollary, (one-to-one) RMT assuming honest majority, erasures, and multisend, not necessarily bulked as a single atomic operation (Corollary 1). Furthermore, it can be trivially extended to the all-to-all RMT case (where every party wants to send a message reliably to every other party) by using batching techniques from [15] (Corollary 2). This settles the feasibility question.

We next turn to impossibility. Here we use an argument that can be seen as mirroring the proof of Theorem 2: We prove that if the majority of the parties can be corrupted (in particular,  $t > (1/2 + \epsilon)n$  for any constant  $\epsilon$ ) then in *any* RMT-protocol candidate, the adversary can with noticeable probability make the expansion of the set of parties that learn the sender's input shrink, resulting in the message dying in the network before it reaches the receiver. This yields a strong impossibility of RMT in the erasures model, which as we show holds even if we assume multisend (Theorem 3).

**Beyond Store-and-Forward.** Next, we turn our attention to the question of whether one can design RMT protocols in the non-erasure model, by devising non-SF protocols, thereby circumventing our impossibility. The answer to this question is far from simple, which further underpins the challenges that CL protocol design poses.

Our key idea is to hide the store-and-forward procedure under the hood of fully homomorphic encryption (FHE). This will hide the message and signature (and in particular, origin and path) for transmitted messages. But as as we shall see, this seemingly simple intuition needs several modification to work.

The protocol structure is similar to the previous protocol (which assumed secure erasures), with the main difference being that the sender encrypts his original message and signature with the receiver's HE public key, and this ciphertext is what is diffused through the low locality network. In particular, instead of checking signatures in the clear, every party uses HE to check if any of the received ciphertexts is an encoding of a message signed by the sender. If so,

it (the circuit evaluated by HE) outputs a new, rerandomized HE ciphertext that encrypts the sender's message and signature (if several such messages are received then use any of them); otherwise encrypt the all-zero message along with a default signature on it. We denote the above protocol by  $\Pi_{\text{FHE}}^{\text{RMT}}$ .

In order for the above approach to work we need the HE scheme to satisfy several properties that are common in the fully homomorphic encryption (FHE) literature, namely *compactness* and *malicious circuit-privacy* [29, 43, 45]. Informally, compactness ensures that the ciphertext size depends only on the plaintext and the security parameter (in particular it does not grow when the  $\text{Eval}_{\text{FHE}}$  operation is applied). On the other hand, malicious circuit privacy ensures that the ciphertext that is computed by  $\text{Eval}_{\text{FHE}}$  leaks no information about the circuit that was homomorphically evaluated, even when the ciphertext on which  $\text{Eval}_{\text{FHE}}$  is computed is maliciously formed. This last property will ensure that applying  $\text{Eval}_{\text{FHE}}$  automatically rerandomizes the ciphertext. We note that both of these properties are common in standard FHE schemes (cf. [29, 42, 43, 45]).

A caveat in the above idea is that it is known that adaptively secure FHE which satisfies compactness is impossible [37]. Nonetheless, we construct a compact adaptively secure FHE scheme which allows for the homomorphic evaluation of the specific function needed by our RMT protocol. In particular the specific function we require does not actually modify the contents of the underlying message that was encrypted (but only checks validity of a signature “under-the-hood” and then retains or disregards the underlying message). We stress that existence of such a scheme does not contradict the impossibility result of Katz *et al.* [37], as the circuits we consider are not in the class considered there.

Next, we describe the idea to circumvent the above impossibility. We consider only the single-pair RMT setting, where one sender  $u$  wishes to send a message to one receiver  $v$ . In order to ensure that ciphertexts can be simulated without knowledge of the plaintext and, if, later on, the sender and/or receiver is corrupted, the simulator can present randomness matching this ciphertext, we use the following idea: The sender first encrypts the message  $m$  it wishes to send with the receiver's public key, using an adaptively secure encryption scheme (e.g., the non-committing encryption of Canetti *et al.* [13]). (Looking ahead, this will allow a simulator to equivocate the message if needed.) Next, the sender signs the resulting ciphertext  $c$ , and encrypts the resulting pair  $(c, \sigma)$  using the receiver's public key for a (*statically secure*) FHE scheme. For brevity, we will refer to the above operation of encrypting with FHE an authenticated version of the adaptively encrypted plaintext as *adaptively authenticated homomorphic encryption* (aaHE for short). Then the sender propagates the aaHE ciphertext through the hidden communication graph, identically to the original protocol described in Sect. 5.1.

However, unlike the original protocol, parties that are at distance  $> 1$  from the sender cannot tell if the aaHE ciphertext they receive is encrypting a valid message (i.e., one actually originating from the sender). Looking ahead, this property is the key part where our protocol deviates from the SF protocol structure. One solution would be for relayers to propagate all messages they receive.

This, however, would result in an exponentially growing message and could compromise the security of the protocol, as the number of relayed messages would leak information about the position of the relayer in the hidden graph.

This is where FHE comes to the rescue. Upon receiving several such ciphertexts, a relayer homomorphically evaluates the circuit which on input all the (plaintexts of the) received ciphertexts and the sender's verification key, checks if any of these plaintexts is of the form  $(c, \sigma)$ , where  $\sigma$  is a valid sender's signature on  $c$ , and if it finds one it outputs (an FHE ciphertext  $\tilde{c}$  encrypting) it. (If more than one such message is found then output the first one encountered in the above search.) This will make sure that every relayer sends out ciphertexts of the same length which encrypt either  $(c, \sigma)$ , in case the sender was honest and the relayer is on an honest path between the sender and the receiver, or some arbitrary pair  $(*, *)$  (chosen by the adversary) otherwise. In other words, the above scheme ensures that the information transmitted by the above scheme is exactly an  $\text{aaHE}$  encryption of the message  $m$  which our original SF protocol from Sect. 5.1 would propagate, except that since this information is encrypted, the adversary cannot use it to trace the message/ $\text{aaHE}$  ciphertext back to the sender. We stress that the properties of compactness as well as malicious circuit privacy, of FHE play a crucial role here, as we elaborate in the full version.

However, there is still a way the adversary can obtain information in the above scheme that allows him to potentially link the sender to the transmitted  $\text{aaHE}$  ciphertext, by observing which parties “speak” in which round. This can easily be mitigated by decoy traffic: every party sends some message in every round, where in the first round, parties other than the actual sender create and send to their neighbors an  $\text{aaHE}$  ciphertext using dummy strings  $d$  and  $s$  in place of  $m$  and  $\sigma$  respectively, with  $|d| = |m|$  and  $|s| = |\sigma|$ .<sup>7</sup>

The final missing piece in the protocol is to ensure that we can use the above protocol for transmitting multiple messages (as is needed in MPC). It is not hard to see that for this to be possible, we need to exclude replay attacks. Indeed, although the adversary cannot create an  $\text{aaHE}$  of a new message corresponding to honest senders, he can replay past  $\text{aaHE}$  of messages created by the sender. The way to mitigate this is, as is common in the security literature, to use unique publicly agreed identifiers—e.g., round-number and/or message ID—and make sure that the circuit which  $\text{Enc}_{\text{FHE}}$  is run on, also takes as input (and checks) the corresponding identifier. Yet, one needs to be careful about where the identifier is placed inside the  $\text{aaHE}$ . If one encrypts  $(m, \text{msg\_ID})$  with  $\text{aaHE}$  then  $\text{Eval}_{\text{FHE}}$  will be unable to check the message ID  $\text{msg\_ID}$  under the (adaptively secure) encryption. Therefore, the actual message which is encrypted with FHE will be of the form  $((c, \text{msg\_ID}), \sigma)$ , where  $\sigma$  is a signature on the pair  $(c, \text{msg\_ID})$ .

This completes the high-level protocol description. In Theorem 4 we prove that the above protocol is an adaptively secure single-pair RMT protocol with polylogarithmic locality. Intuitively, the fact that the message is transmitted successfully between two honest parties follows from the fact that the protocol view of the adversary in this case is fully simulatable, hence any attack by an

---

<sup>7</sup> WLOG, we assume that valid RMT messages are from a fixed-size domain.

adaptive adversary who does not corrupt the sender or receiver can be reduced to the static case proven in [15]. If, on the other hand, the adversary corrupts the sender or the receiver, then the only challenge for the simulator is to be able to come up with coins that are consistent with the actual input  $m$  of the sender. But this is straightforward in  $\text{aaHE}$  as the FHE ciphertext (i.e., encrypting  $((c, \text{msg\_ID}), \sigma)$ ) is generated by the simulator and hence he can simply reveal the keys used for this encryption. Having these coins, the simulator needs to simply show how to open  $c$  to the message  $m$ . But for that, he can use the adaptive security of the underlying encryption scheme.

**From One-to-One CL RMT to Many-to-Many CL RMT.** The FHE-based protocol described above is for one-to-one RMT. One might be tempted to assume that having such a protocol gives us also all-to-all RMT in this model. Indeed, at first thought, it appears that one can achieve this by simply running  $\frac{n(n-1)}{2}$  instances of  $\Pi_{\text{FHE}}^{\text{RMT}}$  in parallel over the same hidden graph (i.e., a joint state) on separate slots, one for each sender-receiver pair, following the idea of Sect. 5.1.2. However, the following major flaw breaks such a protocol.

Let  $u$  and  $v$  be honest parties in the network, and  $z$  be a party corrupted by adaptive adversary  $\mathcal{A}$ , with  $u$  being a sender while  $v$  and  $z$  act as receivers. Further, suppose the shortest honest path from  $u$  to  $z$  is shorter than that from  $u$  to  $v$ ; i.e.,  $z$  receives her message from  $u$  before (in an earlier round than)  $v$ . Now, adversary  $\mathcal{A}$  can decrypt  $z$ 's message in that same round and learn that it is a valid message originating from  $u$ . Moreover,  $\mathcal{A}$  can corrupt the neighbor who sent her this valid message and decrypt that neighbor's previously received set of messages for  $z$ . In this manner,  $\mathcal{A}$  can corrupt everyone who this message has passed through by following the inverse path the message travelled. Clearly, this is identical to the adversarial strategy employed in Sect. 4 to prove the impossibility of SF RMT in the NE-NAMS model, and it is easy to see that the same reasoning breaks the protocol here.

Intuitively, the root of the problem is that using FHE under the receiver's key renders messages "unlinkable" for everyone *but* the receiver. In single-pair RMT, there is a single receiver and correctness is trivial if he is corrupted, so  $\Pi_{\text{FHE}}^{\text{RMT}}$  suffices. But with multiple receivers, corrupted receivers can evidently cause trouble for honest ones.

The above discussion, points to a restricted class of all-to-all RMT, which we call *sublinear-output-set RMT* (in short, SOS-RMT) to come to the rescue. SOS RMT allows every party (a sender) to send a (potentially different) message to every receiver *in a subset of  $\mathcal{P}$  of size  $o(n)$* —in our case this will be of polylogarithmic size. Here is how we proceed towards the design of SOS RMT:

1. First we observe that the above one-to-one RMT can be trivially turned into an all-to-one RMT, by having honest parties replace their decoy messages with the actual message they want to send to the (single) receiver and adjusting the homomorphic operation to keep (one copy of) all these messages as the ciphertext is diffused through the network. We provide the specification of this operation and the corresponding statement of security in Sect. 6.2. Note

that this protocol uses the same edges of the underlying communication graph as the one-to-one RMT protocol it “piggybacks” on.

- Having such an all-to-one RMT it is straightforward to turn it to an SOS RMT by using an independent (part of the) hidden-graph setup for each of the receivers. Since there are polylogarithmic receivers and each of these hidden (sub-)graphs has polylogarithmic degree, the resulting protocol will also have polylogarithmic locality (see Corollary 4).

**From CL RMT to CL MPC.** Last but not least, we show how to turn the above feasibility results on RMT into feasibility for CL multi-party computation (CL MPC). For the erasures case, we can use the same approach as the one used in [15] for this reduction: Use a constant-round MPC, e.g., [2], where calls to a broadcast channels are replaced by a polylogarithmic-round (in the worst-case) byzantine broadcast protocol. Because the expected constant-round protocol of [35] is guaranteed to terminate with overwhelming probability after polylogarithmically many round, we can simply employ this protocol. This will result in polylogarithmically many invocations of the all-to-all RM from Sect. 5.1, which consumes a polylogarithmic hidden graph setup, thereby yielding a CL MPC protocol in the secure-erasures model (with or without atomic multisend), which is secure under an honest-majority (adaptive) adversary. We refer to Sect. 7.1 for details.

The more challenging case is the non-erasures setting. Here, we do not have an all-to-all CL RMT, so we cannot hope for standard CL MPC—as the latter would imply the former. Instead, we go for (CL) SOS MPC, which as with SOS RMT, computes a function with inputs from all parties, but only distributes the output to a sublinear (polylogarithmic) set of parties. We believe that the notion of SOS MPC is interesting in its own accord, as it appears to be a best-possible security notion in the CP non-erasures setting. Furthermore this notion is already reasonable for the core application of CL MPC, namely computing sublinear algorithms. Indeed, such algorithms typically have output asymptotically smaller than  $n$ . In this case, having the output-set of SOS MPC distribute the outputs to the whole player set (using the complete graph) does not incur a big overhead in communication complexity.

To implement CL SOS MPC, we assume an additional setup, namely, anonymous PKIs for the FHE, NCE, and signature schemes used in our non-erasures RMT protocol: Parties are given public keys but they do not know who has the corresponding secret key. Given this setup and SOS RMT, SOS MPC can be designed as follows: Let  $C$  denote a polylogarithmic size subset of the (owners of the secret keys for) the anonymous public keys. (Any subset will do, but for simplicity we can assume that this is the first polylog(n) public key in a lexicographic order). The parties use SOS RMT (where the FHE and NCE encryptions, and the underlying signatures are with the anonymous PKIs) to share their input to  $C$ . Then the parties in  $C$  run an MPC over SOS RMT (again with these anonymous keys). However, to avoid leaking their identities through communication pattern, *all*  $n$  parties participate in these RMTs, where parties not in  $C$  simply send decoy traffic as in our one-to-one version of the

non-erasure RMT protocol. The details on this construction and security proof are given in the end of Sect. 7.1.

## 4 Impossibility in the NE-NAMS and NE-AMS Models

In this section, we show an impossibility result for a natural class of reliable message transmission (RMT) protocols (and therefore also MPC) which we term store-and-forward protocols, with low communication locality in NE-NAMS and NE-AMS models—i.e., a model where secure erasures are not allowed—tolerating an adaptive adversary corrupting a linear number of parties. As discussed above, this class includes most if not all, gossip-style communication protocols which have been used in the CL MPC and CL communication literature.

Let us first define the class of store-and-forward RMT protocols:

**Definition 1 (Store-and-Forward).** *A PKI-hybrid RMT protocol with sender  $p_s$  and receiver  $p_r$ , using parties in  $\mathcal{P}$  who communicate over point-to-point channels, is a store-and-forward (in short, SF) protocol if it has the following structure:*

- *In the first round of the protocol,  $p_s$  sends the message  $m$  (that he wishes to transmit to  $p_r$ ) to any neighbors of his choosing, along with his signature  $\sigma_s$  on  $m$ .  $p_s$  does not participate in any other transmission.*
- *In every round  $j > 1$ , any party who has received a pair  $(m, \sigma_s)$ , where  $\sigma_s$  is a valid signature (with respect to  $p_s$ 's verification key) on  $m$ , may forward the pair  $(m, \sigma_s)$  to any neighbors of his choosing.*

We remark that the above definition allows parties to selectively decide (using their current state) when and to whom they forward the pair  $(m, \sigma_s)$ . As such it also captures protocols that use delays to hide communication patterns. Furthermore, the assumption that in the first round,  $p_s$  sends all his protocol messages does not pose a restriction with respect to such scheduling—i.e., on when  $p_s$  sends his message to each neighbor—as such delays can be trivially simulated by  $p_s$  telling his neighbors (in the first round) to apply the intended delay. We also point out that the above single-pair RMT can be trivially extended to all-to-all RMT, by allowing each relayer  $p$  to forward vectors of pairs  $((m_{i_1}, \sigma_{i_1}), \dots, (m_{i_\ell}, \sigma_{i_\ell}))$ , where each  $(m_{i_j}, \sigma_{i_j})$  is a (message,  $p_{i_j}$ -signature)-pair that  $p$  heard in a previous round.

Our impossibility result assumes a restriction on the above class of SF protocols, which relates the maximum length of a path traversed by the sender's message to the size of the set of parties that have seen the message. To define this, we introduce the following graph theoretic notation.

**Notation.** We will denote by  $G_{s,r}$  the *labeled* graph with vertices  $V = \mathcal{P}$  which corresponds to a protocol's execution in the following manner: an edge  $(w_1, w_2)$  is added to  $G_{s,r}$  when  $w_1 \in V$  sends the message  $(m, \sigma_s)$  to  $w_2 \in V$  through their point-to-point channel, in the RMT protocol between sender  $p_s$  and receiver  $p_r$ . The label  $l_{w_1, w_2}$  of each such edge  $(w_1, w_2)$  is defined as the round of the RMT

protocol in which this edge was added to  $G_{s,r}$ . Further, we denote by  $G_{s,r}^{\text{rnd}}$  the subgraph of  $G_{s,r}$  that only contains edges  $(w_1, w_2)$  having labels  $l_{w_1, w_2} \leq \text{rnd}$ .

**Definition 2 (Expansion Rate of SF RMT Protocols).** *We say that an SF RMT protocol has expansion rate  $(L, \ell)$ , where  $L \in [n]$  and  $\ell \in \mathbb{N}$ , if the following property holds at every round  $\text{rnd}$  in the protocol execution: If the maximum size of a path in  $G_{s,r}^{\text{rnd}}$  from  $p_s$  to a sink (i.e., a node which has out-degree 0 in  $G_{s,r}^{\text{rnd}}$ ) is  $\ell$ , then for any (forward) neighbor  $p$  of  $p_s$  in  $G_{s,r}^{\text{rnd}}$ , the number of nodes in the subgraph of  $G_{s,r}^{\text{rnd}}$  (with in-degree at least 1) rooted at  $p$  is at least  $L$ .*

The above definition can easily be extended to all-to-all SF RMT protocols (resp. one-to-all), by requiring that for every pair  $(p_s, p_r)$  (resp. for sender  $p_s$  and all receivers  $p_r$ ) the expansion rate of the transmission is as above.

Looking ahead, we will prove our impossibility result for SF RMT protocols with expansion rate  $(\log^z(n), \frac{k \log n}{c \log \log n})$  for all  $k < 1$  and  $z > 1$ , where the degree of the communication graph is  $O(\log^c n)$ . To get a better intuition of how the expansion rate affects the security of RMT protocols against adaptive adversaries, it is worth looking at the simpler case with expansion rate  $(d^{\xi-1}, \xi)$ , where  $d = O(\log^c n)$  is the degree of the underlying communication graph and  $\xi$  is a constant. We note that this seemingly simple case corresponds to (the first of  $\xi$  rounds of) the “vanilla” SF strategy which, to our knowledge, is employed by all CL MPC protocols in the literature.

**Lemma 1.** *Assuming no erasures, there exists no polylogarithmic-locality SF one-to-all RMT protocol with sender  $p_s$  that has expansion rate  $((\log^c n)^2, 3)$  for some constant  $c > 1$ , and tolerates an adaptive adversary corrupting a constant fraction of the parties.*

Note that this is the case with  $\xi = 3$ . We give a proof sketch of the above lemma in the full version. Since this is just a special case of the general theorem (Theorem 1) proved later in this section, we keep the proof at an informal level, to allow the reader to grasp the main ideas, and refer to the remainder of this section for formal claims (that even cover the more general case).

**The Case of Expansion Rate**  $(\log^z n, \frac{k \log n}{c \log \log n})$ . We assume that the adversary is able to corrupt a constant fraction  $\tau$  of the nodes; hence  $t = \tau n$ . Our impossibility result holds for any constant  $\tau > 0$ . Our proof relies on a series of lemmas (see the full version for proofs) on the (polylogarithmic-degree) communication graph and how the adversary attacks any RMT protocol  $\pi$  over such a graph. Concretely, towards our impossibility result, we prove the following:

1. First, we will show (Lemma 2) that there exists a sender  $p_s$  and a receiver  $p_r$ , such that the length of the shortest path between  $p_s$  and  $p_r$  (in the communication graph  $G_{s,r}$  of  $\pi$ ) is strictly greater than  $q$ , where  $q = \frac{k \log n}{c \log \log n}$ , for any  $k < 1$  and where the communication locality of the RMT protocol is  $O(\log^c n)$ . In other words, let  $\text{rnd}$  be the round of  $\pi$  in which the  $(p_s, p_r)$ -RMT message first reaches a distance  $q$  from  $p_s$ ; then,  $p_r$  is not connected to any node in  $G_{s,r}^{\text{rnd}}$ .

2. The remainder of our proof strategy is as follows: Consider an execution of RMT with sender  $p_s$  and receiver  $p_r$  as above; the communication graph after  $\text{rnd}$  rounds is  $G_{s,r}^{\text{rnd}}$ . The goal of the adversary (and what we will prove he can achieve) is to corrupt each of the neighbors of  $p_s$  in this graph, and also corrupt everyone to whom they have (directly or indirectly) conveyed information on  $p_s$ 's message, *before* this information reaches the receiver  $p_r$ . Thus, our proof focuses on each of the neighbors of  $s$  individually, and shows that the above is achieved with overwhelming probability. Using the fact that the total number of neighbors of  $p_s$  is  $O(\log^c n)$ , and by the choice of  $q$ , we can then prove that the probability of an adversary successfully attacking all of  $p_s$ 's neighbors and cutting  $p_s$  off before (information on) his intended message reaches  $p_r$  is noticeable (Lemma 3). This forms a successful attack on the RMT protocol  $\pi$ , as it disconnects  $p_s$  and  $p_r$ , completing the proof.

The formal statement of Lemma 2 now follows.

**Lemma 2.** *For any given SF RMT protocol  $\pi$  with communication locality  $O(\log^c n)$ , let  $q = \frac{k \log n}{c \log \log n}$  for any  $k < 1$ . Consider a sender  $p_s$ . There exists a receiver  $p_r$  such that  $p_r$  is not connected to any node in  $G_{s,r}^{\text{rnd}}$ , where  $\text{rnd}$  denotes the round of  $\pi$  in which the  $(p_s, p_r)$ -RMT message first reaches distance  $q$  from  $p_s$ .*

The most interesting step which captures the essence of our proof is Lemma 3 below. The intuition of the proof is that because the distance between  $p_s$  and  $p_r$  is larger than  $q$ , each  $p_s$ -neighbor in  $G_{s,r}$  will have distance of at least  $q$  to  $p_r$  in  $G_{s,r}$ . Thus, any message originating from such a  $p_s$ -neighbor needs at least  $q$  hops to reach the receiver. Now consider the subgraph of  $G_{s,r}$  which grows from a forward neighbor  $p$  of  $p_s$  only, i.e., the graph consisting of  $p$ , his neighbours, his neighbors' neighbors and so on. To prevent  $p_s$  from communicating with  $p_r$  via  $p$ , the adversary will first corrupt nodes at random with the hope of corrupting at least one node in this subgraph. More specifically, let us consider an adversary that initially corrupts a  $\beta < \frac{\tau}{4}$  fraction of random nodes in the whole graph  $G_{s,r}$ , and show that such an adversary leaves both  $p_s$  and  $p_r$  initially uncorrupted—and looking ahead, will avoid corrupting  $p_s$  and  $p_r$  after this initial step. The first observation is that the message sent by  $p_s$  and relayed by  $p$  will, with overwhelming probability, hit some party in this initially corrupted set before it hits the receiver  $p_r$ . Once this happens, the adversary corrupts everyone who this message has passed through by following the inverse path the message travelled. This is feasible because nodes cannot erase any information, and in particular where messages came from and where they were relayed. If the adversary is able to do this for every  $p_s$ -neighbor, then he successfully cuts  $p_s$  off from  $p_r$ . To complete the proof we need to argue that this strategy can be launched within the adversary's corruption budget. Intuitively, this is the case, because in each step the total set of parties who have received information about the sender's message grows by a polylogarithmic factor. Hence, by the above choice of  $q$  we are guaranteed that the size of the set remains sublinear. Since the adversary

has only spent a fraction of his linear budget in his initial corruption choice, he still has sufficiently many corruptions to perform the above attack. The formal statement follows.

**Lemma 3.** *For any given SF RMT protocol  $\pi$  with communication locality  $O(\log^c n)$ , let  $q = \frac{k \log n}{c \log \log n}$  for any  $k < 1$ . Further, let  $\pi$  have expansion rate  $(\log^z n, q)$  for any  $z > 1$ . Consider a sender  $p_s$  and a receiver  $p_r$ , and an adversary  $\mathcal{A}$  who corrupts each node in  $\mathcal{P}$  at random with constant probability  $\beta < \frac{\tau}{4}$ . Then, with noticeable probability  $\frac{1}{\text{poly}(n)}$ ,  $\mathcal{A}$  (i) does not corrupt more than  $\frac{\tau n}{2}$  nodes in total; (ii) does not corrupt nodes  $p_s$  and  $p_r$ ; but (iii) corrupts at least one node in  $\{p\} \cup \Gamma_{(q-1)}(p)$  for every  $p \in \Gamma_1(p_s)$ .*

We now combine Lemmas 2 and 3 to obtain our impossibility result, in the following theorem, the proof of which is given in the full version.

**Theorem 1.** *In NE-NAMS and NE-AMS models (i.e. models not assuming erasures), there exists no SF protocol, with  $(\log^z n, q)$  expansion-rate, for all-to-all RMT with polylogarithmic (i.e.,  $O(\log^c n)$ ) communication locality tolerating an adaptive adversary corrupting a linear number  $t = \tau n$  (for any constant  $\tau$ ) of parties, where  $q = \frac{k \log n}{c \log \log n}$ , for any  $k < 1$  and  $z > 1$ . The statement holds even assuming an arbitrary correlated randomness setup, atomic multisend, and any cryptographic hardness assumptions.*

## 5 Positive Results in the E-NAMS/E-AMS Models

In this section, we assume that parties can erase their state. In our positive result, we do not assume that parties have an atomic multisend operation available to them, and the operations of sending a message and erasing state are not atomically bound either. This corresponds to the E-NAMS model. We will first show an all-to-all RMT protocol in this model with polylogarithmic locality tolerating  $t < (\frac{1}{2} - \epsilon)n$  corruptions (Sect. 5.1); this automatically also implies a protocol in the stronger E-AMS model. To complement this result, we also show that tolerating  $t > (\frac{1}{2} + \epsilon)n$  corruptions is impossible in the E-NAMS model (Sect. 5.2).

### 5.1 Polylogarithmic Locality RMT in the E-NAMS Model

Our protocol is a standard RMT protocol that allows a sender  $p_s$  to reliably transmit a message to a remote recipient  $p_r$  over a polylogarithmic degree graph. The all-to-all RMT is then obtained by having each pair use this RMT simultaneously. In addition to a PKI (for digital signatures) our protocol uses a hidden graph setup as in [15] as follows: the setup picks a directed random Erdős-Rényi graph  $G(n, p) = (V, E)$ , where  $V = \mathcal{P}$  is the vertex set and  $E$  is the set of edges in  $G$ , and for every  $i, j \in V$ ,  $\Pr[(i, j) \in E] = p$ . This graph is given to the parties such that every party learns its incoming and outgoing edges in  $G$  (and nothing else). From Sect. 2, recall that for any node  $u \in V$ , we denote its set of outgoing neighbors by  $\Gamma_1(u)$ . The set of nodes at distance  $\leq i$  on a forward (directed) path starting from a node  $u$  are denoted by  $\Gamma_i(u)$ .

### 5.1.1 Single-Pair RMT Protocol in the E-NAMS Model

We now describe our RMT protocol from honest sender  $p_s$  to honest receiver  $p_r$ , where  $p_s, p_r \in V$ , denoted by  $\Pi_{RMT}$ . Our protocol assumes a PKI, a hidden graph setup, existentially unforgeable digital signatures (equivalently, one-way functions), worst-case secure erasures (as discussed in Sect. 2), and no atomic multisend. As a corollary of our statement, at the end of the section we prove that the protocol with a minor tweak works even if we assume atomic multisend instead of just point-to-point communication. (The latter corollary is not that surprising, as atomic multisend restricts the adversary's power, but needs to be nonetheless done with care to make sure the adversary cannot abuse it to discover the hidden graph.)

**Protocol Structure.** To make the protocol and proof simplest, we will (implicitly) induce the following structure: The protocol advances in blocks of two sequential (mini-)rounds, where in the first of these two minirounds a specific sender gets a chance to send a message to a specific receiver, and in the second one that sender gets a chance to erase his state (e.g., the information of the previous receiver). These miniround-blocks are advanced in a round-robin fashion: the first  $n$  blocks of such minirounds are with sender  $p_1$  and receiver each party  $p_j$  in the party set; the next  $n$  blocks of minirounds are for sender  $p_2$  and receiver each party  $p_j$  in the party set, and so on; after  $n$  such sets of  $n$  blocks, the  $(n + 1)$ st set of  $n$  blocks is again with sender  $p_1$ , etc. Thus, a sequence of  $2n^2$  minirounds (where all parties have had a chance to send all the messages they have for any other parties) constitutes a round in the protocol. We induce the above structure as it makes the influence of an adaptive adversary clean, no matter what model one is used to. Recall that we consider an adversary  $\mathcal{A}$  who can adaptively corrupt up to a  $\tau$  fraction of nodes in the network. Hence,  $\mathcal{A}$  is allowed to order the blocks of minirounds within a single round (worst-case adaptive scheduling), and he can corrupt any party in between any two minirounds. Observe that this protocol structure does not increase the CL of the protocol, as a party will utilize its associated minirounds if and only if it has something to send to the corresponding receiver.

More concretely, the protocol starts at round 0, and we call the entire block below a round in the protocol.

- For  $\text{spkr} = 1$  to  $n$ , do the following:
  - If Party  $p_{\text{spkr}}$  has a message to send to  $p_0$  it will do so.
  - Party  $p_{\text{spkr}}$  is given a chance to erase.
  - If Party  $p_{\text{spkr}}$  has a message to send to  $p_1$  it will do so.
  - Party  $p_{\text{spkr}}$  is given a chance to erase.
  - ⋮
  - If Party  $p_{\text{spkr}}$  has a message to send to  $p_{n-1}$  it will do so.
  - Party  $p_{\text{spkr}}$  is given a chance to erase.

The protocol then proceeds to the next round, completing a total of  $R$  rounds. The protocol  $\Pi_{RMT}$  itself is defined as follows.

**RMT Protocol Between  $p_s$  and  $p_r$ .** Our protocol proceeds for a total of  $R = \log^{\tilde{c}} n$  rounds, for some constant  $\tilde{c} > 1$  (where rounds are defined as above). All the verification keys of all nodes (denoted  $\text{vk}_w$  for each party  $w \in \mathcal{P}$ , with corresponding signing key  $\text{sk}_w$ ) are known to all parties.

1. First,  $p_s$  signs  $(m, p_r)$  with  $\text{sk}_{p_s}$ . Denote the signed message (which comprises of the  $(m, p_r)$  as well as the signature on it) by  $\mu_m$ . Party  $p_s$  also initializes  $\text{ctr}_{p_s}$  to the index of a random neighbor in  $\Gamma_1(p_s)$ .
2. Now, at every round  $0 \leq j \leq R$ , every node  $w$  does the following:
  - $w$  checks if he possesses a single valid message  $\mu_m$  - i.e., a message of the form  $(m, p_r)$  that has been signed by  $p_s$ . If so, then  $w$  sends  $\mu_m$  to  $\Gamma_1(w)[\text{ctr}_w]$  and sets  $\text{ctr}_w = \text{ctr}_w + 1$ . (This constitutes a mini-round, and is immediately followed by another mini-round in which  $w$  is given a chance to erase the information of the node in  $\Gamma_1(w)[\text{ctr}_w]$  he sent to in the previous miniround.) Node  $w$  repeats the above over  $d$  many neighbors in  $\Gamma_1(w)$  by iteratively incrementing  $\text{ctr}_w$ , where  $d = O(\log^c n)$  (for some  $c > 1$ ) is the communication locality. Otherwise, if he possesses no valid message  $\mu_m$ , then he does nothing.
  - $w$  disregards all messages from  $w^* \notin \Gamma_1^{\text{in}}(w)$ .

We now prove that the above protocol is an RMT between  $p_s$  and  $p_r$ . Define set,  $\text{GOOD}_j$ ,  $0 \leq j \leq R$  to be the set of nodes, who at the beginning of round  $j$  of the protocol are a) honest and b) are in possession of the message  $\mu_m$ . Let  $g_j = |\text{GOOD}_j|$  for all  $j$ . The idea of our proof is that no matter what the adversary does, the set of parties that know the message (and will therefore forward it in the next round) grows multiplicatively in each round of the protocol. Hence, in  $\log^{\tilde{c}} n$  rounds, the message will reach a large enough honest set, so that one of them will be a neighbor of  $p_r$  and will therefore forward the message to  $p_r$ . Once this happens, RMT will have succeeded.

To this end, we prove a series of lemmas, which can be found in the full version. We first use a probabilistic argument to show that the good set  $\text{GOOD}_1$  of honest parties that have seen message  $\mu_m$  at the beginning of round 1 has size  $\text{polylog}(n)$ , with overwhelming probability. Next, we show that, as long as the adversary corrupts a minority of parties, the size of the good set increases multiplicatively in every round by a constant greater than 1. Finally, we prove that the above constant expansion will ensure that, in  $\log^{\tilde{c}} n$  rounds,  $\mu_m$  will arrive at its intended receiver  $p_r$ . This implies the following theorem.

**Theorem 2.** *Assuming a PKI, a hidden graph setup as above, secure erasures, one-way functions (for existentially unforgeable signatures) and an adaptive adversary corrupting at most  $t < (\frac{1}{2} - \epsilon)n$  parties for any  $0 < \epsilon < \frac{1}{2}$ , the protocol  $\Pi_{RMT}$  realizes reliable message transmission from  $p_s$  to  $p_r$ . The statement holds in the E-NAMS as well as E-AMS models.*

Since atomic multisend is a stronger model than the non-atomic multisend setting considered above, the possibility result applies also to this case. Indeed, given atomic multisend one can trivially simulate point-to-point communication

between a sender  $p_s$  and receiver  $p_r$ , by having  $p_s$  multisend the vector that includes the intended message to the location corresponding to  $p_r$  and 0 to all other parties in its outgoing neighborhood of the hidden graph setup. This proves the following statement about the protocol  $\Pi_{RMT}^{(MS)}$  which results from instantiating  $\Pi_{RMT}$  via the above invocation of the atomic multisend primitive.

**Corollary 1.** *Assuming a PKI, a hidden graph setup as above, secure erasures, one-way functions (for existentially unforgeable signatures), atomic multisend, and an adaptive adversary corrupting at most  $t < (\frac{1}{2} - \epsilon)n$  parties for any  $0 < \epsilon < \frac{1}{2}$ , the protocol  $\Pi_{RMT}^{(MS)}$  described above realizes reliable message transmission from  $p_s$  to  $p_r$  in the E-AMS model.*

### 5.1.2 All-Pairs (aka All-to-All) RMT in the E-NAMS Model

We now describe our protocol for RMT between all pairs of parties, denoted by  $\Pi_{a2aRMT}$ . This will allow every party  $u$  to send a message to every other party  $v$  in a total of  $R$  rounds (where a round is defined as earlier). At a high level,  $\Pi_{a2aRMT}$  works as follows. We will execute a total of  $\frac{n(n-1)}{2}$  instances of protocol  $\Pi_{RMT}$  from Sect. 5.1.1 in parallel. For every receiver  $v$ , every sender  $u$  signs  $(m, v)$  with  $\text{sk}_u$ . Denote the signed message (which comprises of the  $(m, v)$  as well as its signature) by  $\mu_{u,v}$ . Every party  $w$  will maintain  $\frac{n(n-1)}{2}$  slots, each corresponding to one  $(u, v)$  pair. Now, at every round  $0 \leq j \leq R$ ,  $w$  checks if it possesses any valid message that has been sent by sender  $u$  to receiver  $v$  (i.e. a message  $\mu_{u,v}$  of the form  $(m, v)$  that has been signed by  $u$ ). It places this message in the slot corresponding to the pair  $(u, v)$ .  $w$  then sends (potentially) all  $\frac{n(n-1)}{2}$  messages to  $\Gamma_1(w)[\text{ctr}_w]$  and sets  $\text{ctr}_w = \text{ctr}_w + 1$ . It is easy to see that the communication locality of any party does not increase through this process – only the number of messages sent by a party at a time increases from a single message to a collection of  $\frac{n(n-1)}{2}$  messages. Applying a union bound over all pairs of senders and receivers, one can obtain the following corollary to Theorem 2.

**Corollary 2.** *Assuming a PKI, a hidden graph setup as above, secure erasures, one-way functions (for existentially unforgeable signatures) and an adaptive adversary corrupting at most  $t < (\frac{1}{2} - \epsilon)n$  parties for any  $0 < \epsilon < \frac{1}{2}$ , the protocol  $\Pi_{a2aRMT}$  realizes reliable message transmission between all pairs of parties ( $u \in \mathcal{P}, v \in \mathcal{P}$ ). The statement holds in the E-NAMS as well as E-AMS models.*

## 5.2 Impossibility of Dishonest Majority in the E-NAMS Model

In this section, we shall show that it is impossible to construct SF reliable message transmission (RMT) protocols (and therefore also MPC) with low communication locality, in a model even with erasures, if the corruption threshold is  $\frac{1}{2} + \epsilon$  for any constant  $\epsilon > 0$ . To do so, we shall prove that an adversary can break correctness of any RMT protocol between some pair of honest nodes in any such protocol. The proof idea can be seen as symmetric to the proof of Theorem 2. In

particular, in Theorem 2 we showed that if the adversary corrupts  $t < (\frac{1}{2} - \epsilon)n$  parties, for any constant  $0 < \epsilon < 1/2$ , then the set of honest nodes that learns (and forwards) the sender's message grows exponentially fast, and therefore in  $\log^{\tilde{c}} n$  rounds it will be large enough to hit a neighbor of the receiver. Here we prove that if  $t \geq (\frac{1}{2} + \epsilon)n$  for any constant  $0 < \epsilon < 1/2$ , then there is a strategy making the above set shrink exponentially fast, which can make the message disappear before reaching the receiver  $p_r$ .

Consider an RMT protocol between an honest sender  $p_s$  and an honest receiver  $p_r$ . We shall show that for any SF RMT protocol from  $p_s$  to  $p_r$ , an adversary that corrupts  $\frac{1}{2} + \epsilon$  parties can prevent the message  $m$  from reaching  $p_r$  in any polynomial number of rounds. As in Sect. 5.1, let the RMT protocol from  $p_s$  to  $p_r$  begin at round 0, and define  $\text{GOOD}_j$ ,  $0 \leq j \leq R$  to be the set of nodes, who at round  $j$  of the protocol are a) honest and b) are in possession of the message  $\mu_m$ . Let  $g_j = |\text{GOOD}_j|$  for all  $j$ .

**Adversarial Strategy.** Our adversarial strategy is as follows: First, corrupt nodes in the graph uniformly at random (i.e., every node is corrupted with probability  $\frac{1}{2} + \frac{\epsilon}{4}$ ). Next, if an adversarial node receives a message (that was a part of the RMT protocol between  $p_s$  and  $p_r$ ) from some node  $w$  (other than  $p_s$ ), then corrupt  $w$ . Do not forward any messages.

We now prove a series of lemmas (see full version) to show that our adversary violates the assumed security of RMT. We first show that our adversary corrupts at most  $\frac{1}{2} + \epsilon$  fraction of nodes, with overwhelming probability. We next show that after the initial round, only a small set of honest nodes are in possession of the message. Finally, we show that after every round in the protocol, the number of honest nodes having the message reduces. This implies the following theorem.

**Theorem 3.** *In the E-NAMS model (i.e., without atomic multisend), there exists no all-to-all store-and-forward RMT protocol with polylogarithmic CL tolerating an adaptive adversary corrupting  $t \geq (\frac{1}{2} + \epsilon)n$  (for any constant  $0 < \epsilon < \frac{1}{2}$ ) parties. The statement holds even assuming an arbitrary correlated randomness setup, secure erasures, and any cryptographic hardness assumption.*

*Remark 1.* We note that the above argument holds if we do not assume atomic multisend (i.e., in E-NAMS). Indeed, in the stronger E-AMS model, the nodes might be able to do some smart multisend-based relay that prevents the set of parties that know the message from shrinking, or slows down the rate. We leave this interesting question as a future research direction.

## 6 Polylogarithmic Locality RMT in the NE-NAMS Model

In this section, we propose RMT protocols which are not store-and-forward, and can therefore circumvent the impossibility result from Sect. 4. Our key idea is to remove the ability of the adversary to identify the sender by looking at intermediate messages, with the use of fully homomorphic encryption (FHE) to hide the contents (and in particular, origin and path) of transmitted messages.

The resulting protocol for single-pair RMT is described in Sect. 6.1. However, we subsequently note that the same protocol loses its security when composed in parallel for the purpose of all-pairs RMT. In the following Sect. 6.2, we show how the protocol can be extended to obtain RMT from all senders to  $\text{polylog}(n)$  receivers, which we term *sublinear output-set* RMT (or SOS-RMT) and later use directly to achieve communication-local SOS-MPC in Sect. 7.2.

## 6.1 Single-Pair RMT Using Fully Homomorphic Encryption

We next provide a description of our (one-to-one) RMT protocol in the non-erasure case under strong cryptographic assumptions. Here we denote the sender by  $u$  and the receiver by  $v$ . See full version for the proof of Theorem 4.

**Single-pair RMT Protocol  $\Pi_{\text{FHE}}^{\text{RMT}}$  Between  $u$  and  $v$  from FHE and Adaptively Secure (Non-committing) Encryption.** Similarly to the protocol from Sect. 5.1, our protocol proceeds for a total of  $R = \log^{\tilde{c}} n$  rounds for any constant  $\tilde{c} > 1$  (where rounds are as defined in Sect. 5.1). The protocol assumes setup for the following schemes:

- An existentially unforgeable digital signature scheme  $(\text{KeyGen}, \text{Sign}, \text{Verify})$ . Denote by  $\text{vk}_u$  the verification key of the sender  $u$  and by  $\text{sk}_u$  the corresponding signing key.
- A non-committing encryption scheme  $(\text{KeyGen}_{\text{NCE}}, \text{Enc}_{\text{NCE}}, \text{Dec}_{\text{NCE}})$ . Denote by  $\text{pk}_v^{\text{NCE}}$  and  $\text{sk}_v^{\text{NCE}}$  the encryption and decryption keys of the receiver.
- A compact and malicious circuit-private FHE scheme  $(\text{KeyGen}_{\text{FHE}}, \text{Enc}_{\text{FHE}}, \text{Eval}_{\text{FHE}}, \text{Dec}_{\text{FHE}})$ . Denote by  $\text{pk}_v^{\text{FHE}}$  and  $\text{sk}_v^{\text{FHE}}$  the encryption and decryption keys of the receiver, respectively.

The protocol also assumes that the parties have agreed on unique public message IDs  $\text{msg\_ID}$  for the transmitted messages (this will include the protocol ID, the party ID, and the current round). The protocol proceeds as follows:

1. Computation of each party when the protocol starts (to compute the first message they will send):
  - *Code for the sender  $u$ :* First,  $u$  encrypts  $m$  with  $v$ 's (non-committing) encryption key  $\text{pk}_v^{\text{NCE}}$ ; denote the resulting ciphertext by  $c$ . Then,  $u$  signs  $(c, v, \text{msg\_ID})$  with  $\text{sk}_u$ ; denote the corresponding signature by  $\sigma$ . Finally,  $u$  encrypts the pair  $((c, v, \text{msg\_ID}), \sigma)$  with  $v$ 's FHE encryption key  $\text{pk}_v^{\text{FHE}}$ ; denote the resulting (aaHE) ciphertext by  $\tilde{c}_u$ .
  - *Code for each party  $w \neq u$ :* Party  $w$  computes  $c$  as an encryption of the all-zero message of size  $|m|$  with  $v$ 's (non-committing) encryption key  $\text{pk}_v^{\text{NCE}}$  and sets  $\sigma$  to the all-zero string of same size as the actual signature of  $u$  above. Then,  $w$  encrypts  $((c, v, \text{msg\_ID}), \sigma)$  with  $v$ 's FHE encryption key  $\text{pk}_v^{\text{FHE}}$ ; denote the ciphertext by  $\tilde{c}_w$ .
2. Next, at any round  $0 \leq j \leq R$ , every node  $w$  does the following: Let  $C_{w,j} = \{\tilde{y}_1, \dots, \tilde{y}_q\}$  be the ciphertexts that party  $w$  has received in the previous rounds ( $C_{w,j} = \{\tilde{c}_w\}$  if no messages have been received yet.)

- $w$  applies the homomorphic evaluation function  $\text{Eval}_{\text{FHE}}$  on input the ciphertexts in  $C_{w,j}$ , the verification key  $\text{vk}_u$  of the sender  $u$ , and the pre-agreed message ID  $\text{msg\_ID}$  to compute the following function: If any  $\tilde{y} \in C_{w,j}$  can be parsed as  $((c, v, \text{msg\_ID}), \sigma)$ , where  $\sigma$  is a valid signature on  $(c, v, \text{msg\_ID})$  according to the sender’s verification key  $\text{vk}_u$ , then output  $((c, v, \text{msg\_ID}), \sigma)$ . (If there are multiple such  $\tilde{y}$ , output the one with the smallest  $c$ .) Party  $w$  denotes the resulting FHE ciphertext by  $\tilde{c}_{w,j}$ , sends it to  $\Gamma_1(w)[\text{ctr}_w]$  and sets  $\text{ctr}_w = \text{ctr}_w + 1$ .
- $w$  disregards all messages from  $w^* \notin \Gamma_1^{\text{in}}(w)$ .

3. At round  $R$ ,  $v$  uses his FHE decryption key to decrypt each FHE ciphertext in  $C_{v,R}$  (i.e., all ciphertexts received in the protocol). If any  $\tilde{y} \in C_{v,R}$  can be parsed as  $((c, v, \text{msg\_ID}), \sigma)$ , where  $\sigma$  is a valid signature on  $(c, v, \text{msg\_ID})$  according to the sender’s verification key  $\text{vk}_u$ , then  $v$  uses his NCE decryption key  $\text{sk}_v^{\text{NCE}}$  to decrypt  $c$  and outputs the corresponding message as the one sent by  $u$  (if more than one such  $c$  exists, then  $v$  takes the one corresponding to the smallest message  $m$ ). Otherwise,  $v$  outputs 0 as the message received from  $u$ .

**Theorem 4.** *Assuming a PKI, hidden graph setup, trapdoor permutations with a reversed domain sampler, and compact and malicious circuit-private FHE [42], protocol  $\Pi_{\text{FHE}}^{\text{RMT}}$  securely realizes single-pair RMT, tolerating an adaptive adversary who corrupts  $t < \epsilon n$  parties for any  $0 \leq \epsilon < 1$ , in the NE-NAMS model.*

## 6.2 Multi-sender RMT

While  $\Pi_{\text{FHE}}^{\text{RMT}}$  cannot be composed in parallel to achieve all-pairs RMT as discussed in Sect. 3, we show in this section that simple joint state parallel composition of single-pair RMT is sufficient to construct an all-to-one RMT protocol  $\Pi_{\text{FHE}}^{\text{a21RMT}}$ , and the usage of multiple all-to-one RMT instances over independent hidden graphs can further extend this to an SOS-RMT protocol  $\Pi_{\text{FHE}}^{\text{SOS-RMT}}$ . See the full version for detailed protocols and proofs.

**Corollary 3.** *Assuming a PKI, a hidden graph setup, trapdoor permutations with a reversed domain sampler, and compact and malicious circuit-private FHE,  $\Pi_{\text{FHE}}^{\text{a21RMT}}$  securely realizes all-to-one RMT, tolerating an adaptive adversary who corrupts  $t < \epsilon n$  parties for any constant  $0 \leq \epsilon < 1$ , in the NE-NAMS model.*

**Corollary 4.** *Assuming a PKI, a hidden graph setup, trapdoor permutations with a reversed domain sampler, and compact and malicious circuit-private FHE,  $\Pi_{\text{FHE}}^{\text{SOS-RMT}}$  securely realizes SOS-RMT, tolerating an adaptive adversary who corrupts  $t < \epsilon n$  parties for any constant  $0 \leq \epsilon < 1$ , in the NE-NAMS model.*

## 7 Communication-Local MPC

We finally turn to the question of adaptively secure MPC with polylogarithmic communication locality. In Sect. 7.1, we show that all-pairs RMT can be used to realize CL MPC, and we outline our impossibility and feasibility results for CL MPC in the NE-AMS, E-NAMS, and E-AMS models. In Sect. 7.2, we state our final feasibility result for CL SOS-MPC in the NE-NAMS model.

## 7.1 CL MPC in the NE-AMS, E-NAMS, and E-AMS Models

All our negative results on all-to-all RMT trivially apply to MPC, since the former is a special case. Next, we show the feasibility of CL MPC, under the combination of the feasibility bounds of all-pairs RMT proven here and the classical  $t < n/2$  bound, necessary and sufficient for standard (non-CL) MPC.

The idea for the above is as follows: Execute the MPC protocol where the point-to-point communication is replaced by encrypting the message with the public key of the receiver and sending it using a fresh all-pairs RMT execution (as constructed here). As noticed in [15], to achieve adaptive security, each round of the MPC protocol will require RMT on a new hidden graph setup which, in the worst case, induces an additive polylogarithmic increase in the CL in every round. To keep the overall CL of the MPC polylogarithmic, one needs to be careful that the total number of point-to-point rounds in the MPC protocol is at most polylogarithmic. To this direction, we adopt the following solution from [15]: Invocations to the (typically round-intensive) broadcast channel are replaced by a polylogarithmic-round broadcast protocol provided in [15]. This protocol can be used within an adaptively secure constant-round MPC protocol (e.g., [2]) to get an overall polylogarithmic-round MPC protocol.

**Theorem 5.** *Assuming a PKI, a polylogarithmic-degree hidden graph setup<sup>8</sup>, and trapdoor permutations with a reversed domain sampler, the following feasibility and impossibility statements hold for the existence of a store-and-forward protocol for securely evaluating any given  $n$ -party function against an adaptive  $t$ -adversary satisfying the following two conditions with overwhelming probability:*

- **Locality.** *Every party communicates with at most  $\mathcal{O}(\log^{1+\epsilon} n)$  other parties, for some constant  $\epsilon > 0$ .*
- **Rounds.** *The protocol terminates after  $\mathcal{O}(\log^{\epsilon'} n)$  rounds, for some constant  $\epsilon' > 0$ .*

1. *In the NE-AMS and NE-NAMS models, i.e. if we do not assume erasures, then no such MPC exists if  $t = O(n)$  and the protocol has an expansion rate of  $(\log^z n, \frac{k \log n}{(1+\epsilon) \log \log n})$ , for some  $k < 1$  and  $z > 1$ .*
2. *In the E-NAMS and E-AMS models, i.e., if we assume erasures (with or without atomic multisend), then there exists such an MPC protocol if  $t < (\frac{1}{2} - \epsilon'')n$  for some constant  $\epsilon''$ .*
3. *In the E-NAMS model, no such MPC exists if  $t > (\frac{1}{2} + \epsilon'')n$  for some constant  $\epsilon''$ .<sup>9</sup>*

<sup>8</sup> Recall that this can be replaced by an SKI or a NIKE scheme assuming the PKI supports it.

<sup>9</sup> Note that Theorem 3 implies that if we assume erasures as an atomic operation and no atomic multisend, then no MPC as in the above theorem exists if  $t > (\frac{1}{2} + \epsilon'')n$  for some constant  $\epsilon''$ . However, this is anyway implied by the tightness of the condition  $t < n/2$  for adaptive security even in the complete (i.e., non-CL) point-to-point channels setting, and is therefore omitted.

## 7.2 CL Sublinear-Output-Set (SOS) MPC in the NE-NAMS Model

Since we do not have all-pairs RMT in the NE-NAMS model, we propose a protocol for sublinear output-set MPC (SOS-MPC), where a sublinear (here polylogarithmic) set of parties learns the output. A high-level outline is as follows.

We first select a committee of some  $\text{polylog}(n)$  parties; these parties perform the actual MPC and learn the output. The first task here is to select the committee members and keep their identities hidden from the adversary, while still allowing parties to send messages to committee members. We resolve this by introducing an *anonymous* PKI setup, for all three schemes involved, namely the signature scheme, the non-committing encryption scheme, and the FHE scheme. Each party receives its secret keys, while the public keys are made known to everyone without disclosing identities. The setup also selects some  $\text{polylog}(n)$  parties at random to form the committee, and publishes their public keys.

Next, each party creates  $\text{polylog}(n)$  secret shares of his MPC input, and distributes these shares to the committee with a single instance of SOS-RMT (using  $\Pi_{\text{FHE}}^{\text{SOS-RMT}}$ ). Finally, the committee members simulate an arbitrary MPC protocol to obtain the output. This can be realized by simulating each round of communication in the MPC protocol via a new instance of SOS-RMT, wherein each committee member sends the appropriate message for the MPC protocol to other committee members, while other parties just send a dummy all-zeros message to each committee member. After every RMT instance, each committee member decrypts messages received from other committee members.

A detailed description of our final protocol can be found in the full version, and the following theorem is immediate.

**Theorem 6.** *Assuming an anonymous PKI, a polylogarithmic-degree hidden graph setup, trapdoor permutations with a reversed domain sampler, and compact and malicious circuit-private FHE, there exists a protocol, satisfying the following two constraints with overwhelming probability:*

- **Locality.** *Every party communicates with at most  $\mathcal{O}(\log^{1+\epsilon} n)$  other parties, for some constant  $\epsilon > 0$ , and*
- **Rounds.** *The protocol terminates after  $\mathcal{O}(\log^{\epsilon'} n)$  rounds, for some constant  $\epsilon' > 0$ ,*

*which securely evaluates any given  $n$ -party function against an adaptive  $t$ -adversary corrupting up to  $t < n/2$  parties in the NE-NAMS model, and delivers the output to any  $\mathcal{O}(\log^{\epsilon''} n)$  parties, for constant  $\epsilon'' > 0$ .*

## References

1. Abraham, I., et al.: Communication complexity of byzantine agreement, revisited. In: Robinson, P., Ellen, F. (eds.) Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, 29 July–2 August 2019, pp. 317–326. ACM (2019). <https://doi.org/10.1145/3293611.3331629>
2. Beaver, D., Micali, S., Rogaway, P.: The round complexity of secure protocols (extended abstract), pp. 503–513 (1990). <https://doi.org/10.1145/100216.100287>
3. Beck, M., et al.: Logistical computing and internetworking: middleware for the use of storage in communication. In: 3rd Annual International Workshop on Active Middleware Services (AMS 2001), 6 August 2001, San Francisco, CA, USA, pp. 12–21. IEEE Computer Society (2001). <https://doi.org/10.1109/AMS.2001.993716>
4. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In: Simon, J. (ed.) Proceedings of the 20th Annual ACM Symposium on Theory of Computing, 2–4 May, 1988, Chicago, Illinois, USA, pp. 1–10. ACM (1988). <https://doi.org/10.1145/62212.62213>
5. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract), pp. 1–10 (1988). <https://doi.org/10.1145/62212.62213>
6. Boyle, E., Chung, K.-M., Pass, R.: Large-scale secure computation: multi-party computation for (parallel) RAM programs. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9216, pp. 742–762. Springer, Heidelberg (2015). [https://doi.org/10.1007/978-3-662-48000-7\\_36](https://doi.org/10.1007/978-3-662-48000-7_36)
7. Boyle, E., Cohen, R., Data, D., Hubáček, P.: Must the communication graph of MPC protocols be an expander? In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018. LNCS, vol. 10993, pp. 243–272. Springer, Cham (2018). [https://doi.org/10.1007/978-3-319-96878-0\\_9](https://doi.org/10.1007/978-3-319-96878-0_9)
8. Boyle, E., Goldwasser, S., Tessaro, S.: Communication locality in secure multi-party computation. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 356–376. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-36594-2\\_21](https://doi.org/10.1007/978-3-642-36594-2_21)
9. Boyle, E., Goldwasser, S., Tessaro, S.: Communication locality in secure multi-party computation. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 356–376. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-36594-2\\_21](https://doi.org/10.1007/978-3-642-36594-2_21)
10. Canetti, R.: Security and composition of multiparty cryptographic protocols **13**(1), 143–202 (2000). <https://doi.org/10.1007/s001459910006>
11. Canetti, R.: Universally composable security: a new paradigm for cryptographic protocols, pp. 136–145 (2001). <https://doi.org/10.1109/SFCS.2001.959888>
12. Canetti, R., Feige, U., Goldreich, O., Naor, M.: Adaptively secure multi-party computation. In: Miller, G.L. (ed.) Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, 22–24 May, 1996, pp. 639–648. ACM (1996). <https://doi.org/10.1145/237814.238015>
13. Canetti, R., Feige, U., Goldreich, O., Naor, M.: Adaptively secure multi-party computation, pp. 639–648 (1996). <https://doi.org/10.1145/237814.238015>
14. Chandran, N., Chongchitmate, W., Garay, J.A., Goldwasser, S., Ostrovsky, R., Zikas, V.: Optimally resilient and adaptively secure multi-party computation with low communication locality. IACR Cryptol. ePrint Arch, vol. 2014, p. 615 (2014). <http://eprint.iacr.org/2014/615>
15. Chandran, N., Chongchitmate, W., Garay, J.A., Goldwasser, S., Ostrovsky, R., Zikas, V.: The hidden graph model: communication locality and optimal resiliency with adaptive faults, pp. 153–162 (2015). <https://doi.org/10.1145/2688073.2688102>

16. Chandran, N., Forghani, P., Garay, J.A., Ostrovsky, R., Patel, R., Zikas, V.: Universally composable almost-everywhere secure computation. In: Dachman-Soled, D. (eds.) 3rd Conference on Information-Theoretic Cryptography, ITC 2022. LIPIcs, 5–7 July, 2022, Cambridge, MA, USA, vol. 230, pp. 14:1–14:25. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2022). <https://doi.org/10.4230/LIPIcs.HTC.2022.14>
17. Chandran, N., Garay, J., Ostrovsky, R.: Improved fault tolerance and secure computation on sparse networks. In: Abramsky, S., Gavoille, C., Kirchner, C., Meyer auf der Heide, F., Spirakis, P.G. (eds.) ICALP 2010. LNCS, vol. 6199, pp. 249–260. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-14162-1\\_21](https://doi.org/10.1007/978-3-642-14162-1_21)
18. Chandran, N., Garay, J., Ostrovsky, R.: Edge fault tolerance on sparse networks. In: Czumaj, A., Mehlhorn, K., Pitts, A., Wattenhofer, R. (eds.) ICALP 2012. LNCS, vol. 7392, pp. 452–463. Springer, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-31585-5\\_41](https://doi.org/10.1007/978-3-642-31585-5_41)
19. Chaum, D., Crépeau, C., Damgård, I.: Multiparty unconditionally secure protocols (extended abstract). In: Simon, J. (eds.) Proceedings of the 20th Annual ACM Symposium on Theory of Computing, 2–4 May, 1988, Chicago, Illinois, USA, pp. 11–19. ACM (1988). <https://doi.org/10.1145/62212.62214>
20. Damgård, I., Ishai, Y.: Scalable secure multiparty computation. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 501–520. Springer, Heidelberg (2006). [https://doi.org/10.1007/11818175\\_30](https://doi.org/10.1007/11818175_30)
21. Damgård, I., Ishai, Y., Krøigaard, M.: Perfectly secure multiparty computation and the computational overhead of cryptography. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 445–465. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-13190-5\\_23](https://doi.org/10.1007/978-3-642-13190-5_23)
22. Dani, V., King, V., Movahedi, M., Saia, J.: Brief announcement: breaking the  $O(nm)$  bit barrier, secure multiparty computation with a static adversary. In: ACM Symposium on Principles of Distributed Computing, PODC 2012, Funchal, Madeira, Portugal, 16–18 July 2012, pp. 227–228 (2012)
23. Dwork, C., Peleg, D., Pippenger, N., Upfal, E.: Fault tolerance in networks of bounded degree (preliminary version), pp. 370–379 (1986). <https://doi.org/10.1145/12130.12169>
24. Freire, E.S.V., Hofheinz, D., Kiltz, E., Paterson, K.G.: Non-interactive key exchange. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 254–271. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-36362-7\\_17](https://doi.org/10.1007/978-3-642-36362-7_17)
25. Garay, J., Ishai, Y., Ostrovsky, R., Zikas, V.: The price of low communication in secure multi-party computation. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10401, pp. 420–446. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-63688-7\\_14](https://doi.org/10.1007/978-3-319-63688-7_14)
26. Garay, J.A., Katz, J., Kumaresan, R., Zhou, H.: Adaptively secure broadcast, revisited. In: Gavoille, C., Fraigniaud, P. (eds.) Proceedings of the 30th Annual ACM Symposium on Principles of Distributed Computing, PODC 2011, San Jose, CA, USA, 6–8 June, 2011, pp. 179–186. ACM (2011). <https://doi.org/10.1145/1993806.1993832>
27. Garay, J.A., Ostrovsky, R.: Almost-everywhere secure computation. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 307–323. Springer, Heidelberg (2008). [https://doi.org/10.1007/978-3-540-78967-3\\_18](https://doi.org/10.1007/978-3-540-78967-3_18)

28. Garg, S., Sahai, A.: Adaptively secure multi-party computation with dishonest majority. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 105–123. Springer, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-32009-5\\_8](https://doi.org/10.1007/978-3-642-32009-5_8)
29. Gentry, C.: Fully homomorphic encryption using ideal lattices, pp. 169–178 (2009). <https://doi.org/10.1145/1536414.1536440>
30. Gentry, C., et al.: YOSO: you only speak once. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021. LNCS, vol. 12826, pp. 64–93. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-84245-1\\_3](https://doi.org/10.1007/978-3-030-84245-1_3)
31. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority. In: Aho, A.V. (eds.) Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA, pp. 218–229. ACM (1987). <https://doi.org/10.1145/28395.28420>
32. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority, pp. 218–229 (1987). <https://doi.org/10.1145/28395.28420>
33. Hirt, M., Zikas, V.: Adaptively secure broadcast. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 466–485. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-13190-5\\_24](https://doi.org/10.1007/978-3-642-13190-5_24)
34. Ishai, Y., Prabhakaran, M., Sahai, A.: Founding cryptography on oblivious transfer – efficiently. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 572–591. Springer, Heidelberg (2008). [https://doi.org/10.1007/978-3-540-85174-5\\_32](https://doi.org/10.1007/978-3-540-85174-5_32)
35. Katz, J., Koo, C.-Y.: On expected constant-round protocols for byzantine agreement. In: Dwork, C. (eds.) CRYPTO 2006. LNCS, vol. 4117, pp. 445–462. Springer, Heidelberg (2006). [https://doi.org/10.1007/11818175\\_27](https://doi.org/10.1007/11818175_27)
36. Katz, J., Maurer, U., Tackmann, B., Zikas, V.: Universally composable synchronous computation. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 477–498. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-36594-2\\_27](https://doi.org/10.1007/978-3-642-36594-2_27)
37. Katz, J., Thiruvengadam, A., Zhou, H.-S.: Feasibility and infeasibility of adaptively secure fully homomorphic encryption. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 14–31. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-36362-7\\_2](https://doi.org/10.1007/978-3-642-36362-7_2)
38. King, V., Saia, J.: Breaking the  $o(n^2)$  bit barrier: scalable byzantine agreement with an adaptive adversary. In: Proceedings of the 29th Annual ACM Symposium on Principles of Distributed Computing, PODC 2010, Zurich, Switzerland, 25–28 July 2010, pp. 420–429 (2010)
39. King, V., Saia, J., Sanwalani, V., Vee, E.: Scalable leader election. In: SODA, pp. 990–999 (2006)
40. King, V., Saia, J., Sanwalani, V., Vee, E.: Towards secure and scalable computation in peer-to-peer networks. In: FOCS, pp. 87–98 (2006)
41. Matt, C., Nielsen, J.B., Thomsen, S.E.: Formalizing delayed adaptive corruptions and the security of flooding networks. In: Dodis, Y., Shrimpton, T. (eds.) Advances in Cryptology, CRYPTO 2022. LNCS, vol. 13508, pp. 400–430. Springer, Cham (2022). [https://doi.org/10.1007/978-3-031-15979-4\\_14](https://doi.org/10.1007/978-3-031-15979-4_14)
42. Ostrovsky, R., Paskin-Cherniavsky, A., Paskin-Cherniavsky, B.: Maliciously circuit-private FHE. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8616, pp. 536–553. Springer, Heidelberg (2014). [https://doi.org/10.1007/978-3-662-44371-2\\_30](https://doi.org/10.1007/978-3-662-44371-2_30)

43. Smart, N.P., Vercauteren, F.: Fully homomorphic encryption with relatively small key and ciphertext sizes. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 420–443. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-13013-7\\_25](https://doi.org/10.1007/978-3-642-13013-7_25)
44. Upfal, E.: Tolerating linear number of faults in networks of bounded degree. In: PODC, pp. 83–89 (1992)
45. van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully homomorphic encryption over the integers. In: Gilbert, H. (eds.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 24–43. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-13190-5\\_2](https://doi.org/10.1007/978-3-642-13190-5_2)
46. Wan, J., Xiao, H., Devadas, S., Shi, E.: Round-efficient byzantine broadcast under strongly adaptive and majority corruptions. In: Pass, R., Pietrzak, K. (eds.) TCC 2020. LNCS, vol. 12550, pp. 412–456. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-64375-1\\_15](https://doi.org/10.1007/978-3-030-64375-1_15)
47. Yao, A.C.C.: Protocols for secure computations (extended abstract), pp. 160–164 (1982). <https://doi.org/10.1109/SFCS.1982.38>