Edge Fault Tolerance on Sparse Networks*

Nishanth Chandran[1,**], Juan Garay[2], and Rafail Ostrovsky[3,***]

[1] Microsoft Research, Redmond
nish@microsoft.com
[2] AT&T Labs – Research
garay@research.att.com
[3] Departments of Computer Science and Mathematics, UCLA
rafail@cs.ucla.edu

Abstract. Byzantine agreement, which requires n processors (nodes) in a completely connected network to agree on a value dependent on their initial values and despite the arbitrary, possible malicious behavior of some of them, is perhaps the most popular paradigm in fault-tolerant distributed systems. However, partially connected networks are far more realistic than fully connected networks, which led Dwork, Peleg, Pippenger and Upfal [STOC'86] to formulate the notion of *almost-everywhere (a.e.) agreement* which shares the same aim with the original problem, except that now not all pairs of nodes are connected by reliable and authenticated channels. In such a setting, agreement amongst all correct nodes cannot be guaranteed due to possible poor connectivity with other correct nodes, and some of them must be given up. The number of such nodes is a function of the underlying communication graph and the adversarial set of nodes.

In this work we introduce the notion of *almost-everywhere agreement with edge corruptions* which is exactly the same problem as described above, except that we additionally allow the adversary to completely control some of the communication channels between two correct nodes—i.e., to "corrupt" edges in the network. While it is easy to see that an a.e. agreement protocol for the original node-corruption model is also an a.e. agreement protocol tolerating edge corruptions (albeit for a reduced fraction of edge corruptions with respect to the bound for node corruptions), no polynomial-time protocol is known in the case where a constant fraction of the edges can be corrupted and the degree of the network is sub-linear.

* A full version of this paper, entitled "Almost-Everywhere Secure Computation with Edge Corruptions," is available at http://eprint.iacr.org/2012/221.

** Part of this work was done at UCLA.

*** Supported in part by NSF grants 0830803, 09165174, 1065276, 1118126 and 1136174, US-Israel BSF grant 2008411, OKAWA Foundation Research Award, IBM Faculty Research Award, Xerox Faculty Research Award, B. John Garrick Foundation Award, Teradata Research Award, and Lockheed-Martin Corporation Research Award. This material is based upon work supported by the Defense Advanced Research Projects Agency through the U.S. Office of Naval Research under Contract N00014-11-1-0392. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

A. Czumaj et al. (Eds.): ICALP 2012, Part II, LNCS 7392, pp. 452–463, 2012.
© Springer-Verlag Berlin Heidelberg 2012

We make progress on this front, by constructing graphs of degree $O(n^{\epsilon})$ (for arbitrary constant $0 < \epsilon < 1$) on which we can run a.e. agreement protocols tolerating a constant fraction of adversarial edges. The number of given-up nodes in our construction is μn (for some constant $0 < \mu < 1$ that depends on the fraction of corrupted edges), which is asymptotically optimal. We remark that allowing an adversary to corrupt edges in the network can be seen as taking a step closer towards guaranteeing a.e. agreement amongst honest nodes even on adversarially chosen communication networks, as opposed to earlier results where the communication graph is specially constructed.

In addition, building upon the work of Garay and Ostrovsky [Eurocrypt'08], we obtain a protocol for *a.e. secure computation* tolerating edge corruptions on the above graphs.

Keywords: Fault tolerance, almost-everywhere agreement, bounded-degree network, secure multiparty computation.

1 Introduction

Byzantine agreement [12,10] is perhaps the most popular paradigm in fault-tolerant distributed systems. It requires n parties (processors) to agree upon a common value that is dependent on their inputs, even when some of them may behave arbitrarily. More specifically, n parties $P_1, \cdots, P_n$, each holding input v_i, must run a protocol such that at the end of the protocol, all "honest" (i.e., not misbehaving) parties output the same value and if $v_i = v$ for all the honest parties, then the honest parties output v. Traditionally, protocols for Byzantine agreement assume that any two of the n parties share a reliable and authenticated channel which they use for communication. However, for protocols executed over large networks such as the Internet, in which nodes are typically connected by a communication graph of small degree, this assumption is unreasonable.

In light of this, the seminal work of Dwork, Peleg, Pippenger, and Upfal [5] considered the problem of reaching agreement over networks that are *not* fully connected, and even of low degree, where every party shares a reliable and authenticated channel only with a few of the other $n - 1$ parties. More formally, in the Dwork *et al.* formulation, the n parties (or nodes) are connected by a communication network $\mathcal{G}$. Nodes that are connected by an edge in $\mathcal{G}$ share a reliable and authentic channel, but other nodes must communicate via paths in the graphs that may not be available to them (due to the adversarial "corruption" of some of the nodes). Naturally, in such a setting, one may not be able to guarantee agreement amongst all honest parties; for example, one cannot hope to be able to communicate at all with an honest party whose neighbors are all adversarial. Given this fact—and ubiquitously—Dwork *et al.* termed the new problem *almost-everywhere (a.e.) agreement*, wherein the number of such abandoned nodes (which henceforth will be called "doomed") introduces another

parameter of interest, in addition to the degree of the communication graph (which we wish to minimize), and the number of adversarial nodes that can be tolerated (which we wish to maximize) in reaching agreement.

Indeed, in [5], Dwork *et al.* provide a.e. agreement protocols for various classes of low-degree graphs and bounds on the number of adversarial nodes as well as doomed nodes. For example, they construct a graph of constant degree and show an agreement protocol on this graph tolerating a $\frac{\alpha}{\log n}$ fraction of corrupted nodes (for constant $0 < \alpha < 1$), guaranteeing agreement amongst $(1 - \alpha - \mu)n$ of the honest nodes (for constant $0 < \mu < 1$). In another construction, they give a graph of degree $\mathcal{O}(n^{\epsilon})$ (for constant $0 < \epsilon < 1$) and show an agreement protocol on this graph tolerating a constant α ($0 < \alpha < 1$) fraction of corrupted nodes, and again guaranteeing agreement amongst $(1 - \alpha - \mu)n$ nodes. In a subsequent and remarkable result, Upfal [13] constructed a constant-degree graph and showed the existence of an a.e. agreement protocol on this graph tolerating a constant fraction of corrupted nodes, while giving up a constant fraction of the honest nodes. Unfortunately, the protocol of [13] runs in exponential time (in n). More recently, Chandran, Garay, and Ostrovsky [3] constructed a graph of degree $\mathcal{O}(\log^k n)$ (for constant $k > 1$) and show an agreement protocol on this graph tolerating a constant fraction of corrupted nodes, while giving up only $\mathcal{O}(\frac{n}{\log n})$ honest nodes.

Edge corruptions. All existing work on a.e. agreement considers the case where only nodes may be corrupted and misbehave, while communication on all edges is assumed to be perfectly reliable and authentic. In many settings, however, such a guarantee might again be unrealistic and a bit optimistic. Think for example, of the communication subsystem of a networked computer (e.g., network interface controller card) being infected by malicious software designed to disrupt or alter operation. This would affect the communication between honest parties. Or worse, of a scenario where the secret keys shared by two parties in the system is compromised, yet the parties themselves are honest.

In this work, we address this situation and endow the adversary with additional powers which allow him, in addition to corrupting nodes, to corrupt some of the *edges* in the network—i.e., we consider *a.e. agreement with edge corruptions.* When he does (corrupt an edge), he is able to completely control the communication channel between the two honest nodes, from simply preventing them to communicate, to injecting arbitrary messages that the receiving end will accept as valid. As in the node-only corruption case, in this case also some of the honest nodes in the network must be abandoned. As further motivation towards considering edge corruptions, we remark that allowing an adversary to corrupt edges in the network moves us a step closer towards guaranteeing a.e. agreement on adversarially chosen communication networks. In an ideal scenario, we would like to construct a.e. computation protocols on arbitrary adversarially chosen communication networks. Unfortunately, this is impossible in general[1].

[1] The adversary could simply design networks where several nodes have extremely poor connectivity and hence corrupting a few edges could create several disconnected components in the network of small size.

Table 1. *Agreement against edge corruptions from agreement against node corruptions. (Expl., Det., and Rand. denote Explicit, Deterministic and Randomized, resp.)*

Reference	Graph degree	Frac. of corrupt edges	Graph/Protocol	Running time
[5]	$\mathcal{O}(n^{\epsilon})$	$\frac{\alpha}{n^{\epsilon}}$; adaptive	Expl./Det.	Polynomial
[5]	$\mathcal{O}(1)$	$\frac{\alpha}{\log n}$; adaptive	Expl./Det.	Polynomial
[13]	$\mathcal{O}(1)$	α; adaptive	Expl./Det.	Exponential
[9]	$\mathcal{O}(\log^k n)$	$\frac{\alpha}{\log^k n}$; static	Expl./Rand.	Polynomial
[3]	$\mathcal{O}(\log^k n)$	$\frac{\alpha}{\log^k n}$; adaptive	Expl./Det.	Polynomial
[This work]	$\mathcal{O}(n^{\epsilon})$	α; adaptive	Rand./Det.	Polynomial

However, we can take a step in this direction by allowing the adversary to corrupt edges in the network that we design, thereby "modifying" the network.

Observe that an a.e. agreement protocol for node corruptions can be readily transformed into an a.e. agreement protocol for edge corruptions, albeit for a reduced fraction of edge corruptions. More specifically, let d be the maximum degree of any node in a graph $\mathcal{G}$ on n nodes that admits an a.e. agreement protocol Π amongst $p < n$ nodes, in the presence of x corrupt nodes. Then, it is easy to see that $\mathcal{G}$ admits an a.e. agreement protocol Π' amongst p nodes in the presence of x corrupt edges[2]. However, this means that the graph will only admit an agreement protocol for an $\frac{x}{nd}$ fraction of corrupted edges, as opposed to an $\frac{x}{n}$ fraction of corrupted nodes in the former case. Therefore, the result that we get for the case of edge corruptions using this method is asymptotically weaker than in the case of node corruptions (except when d is a constant). Unfortunately, by applying this method, none of the existing protocols for a.e. agreement against node corruptions give us an a.e. agreement protocol tolerating a constant fraction of edge corruptions. This is depicted in Table 1, where we outline the results obtained via this approach using the results on a.e. agreement for node corruptions from the works of [5,13,9,3], and compare them with the results we obtain in this work. In all the results listed in the table, $0 < \alpha < 1$ is a constant, $0 < \epsilon < 1$ can be any arbitrary constant, and $k > 1$ is a constant.

Note that all the previous results (except for the result obtained as a corollary to [13], in which the protocol's running time is exponential) cannot handle the case where we have a *constant* fraction of corrupted edges. In this work, we are precisely interested in this case. Specifically, we construct the first a.e. agreement protocol on graphs with sub-linear degree that can tolerate a constant fraction of edge corruptions. We remark here that while the above graph constructions are deterministic, we construct our graph (upon which we obtain an a.e. agreement protocol tolerating constant fraction of corrupted edges) probabilistically, and our result holds with high probability. However, a graph satisfying the conditions required for our a.e. agreement protocol to be successful can be sampled with probability $1 - \mathsf{neg}(n)$, where $\mathsf{neg}(n)$ denotes a function that is negligible in n, and furthermore, one can also efficiently check if the graph thus sampled satisfies the necessary conditions for our protocol.

[2] To simulate an adversary corrupting edge (u, v), simply corrupt either node u or v.

Our results and techniques. In this work, we show that for every constant ϵ, $0 < \epsilon < 1$, there exists a graph, call it $\mathcal{G}_{\text{main}}$, on n nodes, with maximum degree $d_m = \mathcal{O}(n^{\epsilon})$, and such that it admits an a.e. agreement protocol that guarantees agreement amongst a $\gamma_m n$ fraction of honest nodes (for some constant $0 < \gamma_m < 1$), even in the presence of an α_m fraction of corrupted edges (i.e., at most $\frac{\alpha_m n d_m}{2}$ corrupted edges), for some constant $0 < \alpha_m < 1$. Our protocol works against an adversary that is *adaptive* (i.e., the adversary can decide which edges to corrupt on the fly during the protocol after observing messages of honest parties) and *rushing* (i.e., in every round, the adversary can decide on its messages after seeing the messages from the honest parties). We now outline the high-level ideas behind our construction:

1. The first step in our construction is to build a graph with higher degree, $\mathcal{O}(\sqrt{n}\log n)$, on which we can have an a.e. agreement protocol tolerating a constant fraction of corrupted edges.
   - To do this, we first observe a property of a graph that is sufficient for such a construction (besides, obviously, every node having degree $\mathcal{O}(\sqrt{n}\log n)$), namely, that any two nodes in the graph have $\mathcal{O}(\log^2 n)$ number of paths of length 2 between them.
   - Second, we observe that the Erdős-Renyi random graph $\mathcal{G}(n, \frac{\log n}{\sqrt{n}})$ satisfies the above two properties with high probability. That is, graph $\mathcal{G}$ on n nodes satisfying the above two properties can be easily sampled by putting an edge (u, v) in $\mathcal{G}$, independently, with probability $p = \frac{\log n}{\sqrt{n}}$.
   - Once we have a graph satisfying these properties, the construction is fairly straightforward: to obtain reliable communication between any two nodes, say, u and v, u simply sends the message to all nodes in the network via all the paths of length 2, and all the nodes then send the message to v, again via all their paths of length 2. One can then show that if v takes a simple majority of the received values, then a constant fraction of the nodes can communicate reliably even in the presence of a constant fraction of corrupted edges. (As shown in [5], reliable pairwise communication is sufficient to obtain an a.e. agreement protocol amongst those nodes.)
2. Next, we show how to construct a graph, $\mathcal{G}'$, recursively from $\mathcal{G} \leftarrow G(n, \frac{\log n}{\sqrt{n}})$ above such that the new graph is of size n^2 and its degree at most twice that of $\mathcal{G}$, and yet we can have an agreement protocol on $\mathcal{G}'$ tolerating a constant fraction of corrupted edges.
   - We construct $\mathcal{G}'$ by taking n "copies" of $\mathcal{G}$ to form n "clouds," and then connecting the clouds using another copy of $\mathcal{G}$. We connect two clouds by connecting the i^{th} node in one with the i^{th} node in the other.
   - Now our hope is to be able to simulate the communication between two nodes u and v in the following way: u will send the message to all nodes in its cloud (call this cloud C_u). Cloud C_u will then send the message to cloud C_v (the cloud which v is a part of). Finally, v will somehow receive the message from cloud C_v.

- The problem with this approach is that we need to have a protocol that will allow two clouds to communicate reliably. But clouds themselves are comprised of nodes, some of which might be corrupted or doomed; hence, the transmission from cloud C_u to cloud C_v might end up being unreliable. To get over this problem, we make use of a specific type of agreement protocol known as *differential agreement* [6], which, informally and whenever possible, allows parties to agree on the majority value of the honest parties' inputs. Careful application of this protocol allows us to perform a type of "error-correction" of the message when it is being transferred from one cloud to another.
- Combining the above techniques leads us to our main result, an a.e. agreement protocol on graphs of degree $\mathcal{O}(n^\epsilon)$ (for all constants $0 < \epsilon < 1$), tolerating a constant fraction of corrupted edges, while giving up μn honest nodes (for a constant $0 < \mu < 1$).

In [8], Garay and Ostrovsky considered the problem of (unconditional, or information-theoretic) secure multi-party computation (MPC) [1,4] in the context of partially connected networks with adversarial nodes. By applying our new a.e. agreement protocol to the construction in [8], we obtain an a.e. MPC protocol tolerating both node and edge corruptions for graphs of degree $\mathcal{O}(n^\epsilon)$ and same parameters as above.

Due to lack of space, a more detailed overview of related work, as well as supplementary material and proofs, are presented in the full version [2].

2 Model, Definitions and Building Blocks

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a graph with n nodes (i.e., $|\mathcal{V}| = n$). The nodes of the graph represent the processors (parties, "players") participating in the protocol, while the edges represent the communication links connecting them. We assume a synchronous network and that the protocol communication is divided into rounds. In every round, all parties can send a message on all of their communication links (i.e., on all edges incident on the node representing the party); these messages are delivered before the next round.

An adversary $\mathcal{A}$ can "corrupt" a set of nodes (as in taking over them and completely control their behavior), $\mathcal{T}_{\text{nodes}} \subset \mathcal{V}$, as well as a set of edges, $\mathcal{T}_{\text{edges}} \subset \mathcal{E}$, in the network such that $|\mathcal{T}_{\text{nodes}}| \leq t_n$ and $|\mathcal{T}_{\text{edges}}| \leq t_e$. $\mathcal{A}$ has unbounded computational power and can corrupt both nodes and edges *adaptively* (that is, the adversary can decide which nodes and edges to corrupt on the fly during the course of the protocol, after observing the messages from honest parties). Furthermore, $\mathcal{A}$ is *rushing*, meaning that it can decide the messages to be sent by adversarial parties (or on adversarial edges) in a particular round after observing the messages sent by honest parties in the same round.

Almost-everywhere agreement. The problem of *almost-everywhere agreement* ("a.e. agreement" for short) was introduced by Dwork, Peleg, Pippenger and Upfal [5] in the (traditional) context of node corruptions. A.e. agreement "gives up" some of the non-faulty nodes in the network from reaching agreement, which

is unavoidable due to their poor connectivity with other non-faulty nodes. We refer to the given-up nodes as *doomed* nodes; the honest nodes for which we guarantee agreement are referred to as *privileged* nodes. Let the set of doomed nodes be denoted by $\mathcal{X}$ and the set of privileged nodes by $\mathcal{P}$; note that the sets $\mathcal{P}$ and $\mathcal{X}$ are a function of the set of corrupted nodes ($\mathcal{T}_{\text{nodes}}$) and the underlying graph. Let $|\mathcal{X}| = x$ and $|\mathcal{P}| = p$. Clearly, we have $p + x + t = n$. We present the formal definition of a.e. agreement, and state the relevant results from Dwork *et al.* [5], in the full version.

Differential agreement. We now present a tool that will be used in our recursive construction in Section 3.2. Fitzi and Garay [6] introduced the problem of δ-*differential agreement* (also, "consensus") developing on the so-called "strong consensus" problem [11], in which every party begins with an input v from a domain $\mathcal{D}$.[3] We describe the problem below and state the results from [6]. In the standard Byzantine agreement problem [12,10], n parties attempt to reach agreement on some value v (for simplicity, we assume $v \in \{0, 1\}$). Let c_v denote the number of honest parties whose initial value is v, and δ be a non-negative integer. δ-*differential agreement* is defined as follows:

Definition 1. *A protocol for parties* $\{P_1, P_2, \cdots, P_n\}$*, each holding initial value* v_i*, is a* δ-differential agreement *protocol if the following conditions hold for any adversary* $\mathcal{A}$ *that corrupts a set* $\mathcal{T}_{\text{nodes}}$ *of parties with* $|\mathcal{T}_{\text{nodes}}| \leq t_n$*:*

- Agreement: *All honest parties output the same value.*
- δ-Differential Validity: *If the honest parties output* v*, then* $c_v + \delta \geq c_{\bar{v}}$*.*

Theorem 1. [6] *In a synchronous, fully connected network,* δ*-differential agreement is impossible if* $n \leq 3t_n$ *or* $\delta < t_n$*. On the other hand, there exists an efficient (i.e., polynomial-time) protocol that achieves* t_n*-differential agreement for* $n > 3t_n$ *in* $t_n + 1$ *rounds.*

We will use $\text{DA}(n, t_n, \delta)$ to denote a δ-differential agreement protocol for a fully connected network tolerating up to t_n faulty processors.

The edge corruption model. In this work we additionally allow the adversary to corrupt edges on the network graph—the set $\mathcal{T}_{\text{edges}} \subset \mathcal{E}$, $|\mathcal{T}_{\text{edges}}| \leq t_e$. We will bound this quantity, as well as the total number of nodes that the adversary can corrupt, and attempt to construct a network graph $\mathcal{G}$ of small (sublinear) degree on which a significant number of honest nodes can still reach agreement. We now give some definitions and make some remarks about a.e. agreement and a.e. secure computation for this setting.

We first observe that since we are working with (asymptotically) regular graphs, obtaining an a.e. (agreement, MPC) protocol in the presence of a constant fraction of corrupted edges will also imply a protocol in the presence of a

[3] In contrast to standard Byzantine agreement, the validity condition in the strong consensus problem states that the output value v must have been the input of some honest party P_i (which is implicit in the case of binary Byzantine agreement).

constant fraction of corrupted edges and a constant fraction of corrupted nodes nodes, as every corrupted node can be "simulated" by corrupting all the edges incident on this node. Thus, we will henceforth consider only adversarial edges and assume that all the nodes are honest.

As in the case of a.e. agreement on sparse networks in the presence of adversarial nodes, a.e. agreement in the presence of adversarial edges also "gives up" certain honest nodes in the network, which, as argued before, is unavoidable due to their poor connectivity with other honest nodes. Let the set of such doomed nodes be denoted by $\mathcal{X}$ and the set of privileged nodes by $\mathcal{P}$. Note that the sets $\mathcal{P}$ and $\mathcal{X}$ are a function of both the set of corrupted edges ($\mathcal{T}_{\text{edges}}$) and the underlying graph. Let $|\mathcal{X}| = x$ and $|\mathcal{P}| = p$; we let the fraction of corrupt edges be α_e. The definition of a.e. agreement with corrupted edges, in particular, now readily follows in the same manner.

Next, we remark that the problem of a.e. agreement for edge corruptions also reduces to that of constructing a reliable message transmission protocol between any two nodes $u, v \in \mathcal{P}$, in particular those which are not directly connected by an edge in $\mathcal{E}$. Furthermore, Garay and Ostrovsky showed that, given such a channel between two nodes u and $v \in \mathcal{P}$, plus some additional paths, most of which (i.e., all but one) might be corrupted, it is possible to construct a (unidirectional) *secure* (i.e., private and reliable) channel between them. The construction is via a protocol known as *secure message transmission by public discussion* (SMT-PD) [8,7]. In turn, from the protocol for a secure channel, an a.e. MPC protocol amongst the nodes in $\mathcal{P}$, satisfying the same notion of security as in [8], readily follows We refer the reader to the full version for details on the definitions of these primitives in the presence of edge corruptions and a.e. MPC (for node-corruptions).

Finally, we remark that one can define the notion of *a.e.* differential agreement (for edge corruptions) in the same manner as a.e. agreement by replacing the set of honest parties with the set of privileged parties in Definition 1 (i.e., by treating doomed parties also as adversarial). Furthermore, note that one can also obtain an a.e. differential agreement protocol (for edge corruptions) from the construction of a reliable message transmission protocol between any two nodes $u, v \in \mathcal{P}$: simply, execute a standard differential agreement protocol and replace every communication between nodes with an execution of the message transmission protocol. We will use $\mathsf{AE\text{-}DA}(n, t_n, \delta)$ to denote an a.e. δ-differential agreement protocol for a partially connected network where the number of privileged parties is $n - t_n$.

3 A.E. Agreement on Low-Degree Networks

In this section we construct a graph in which the maximum degree of any node is low, and yet, there exists a set of nodes (of size a constant times the total number of nodes), such that all nodes in this set can reach agreement even when a constant fraction of the edges in the graph are corrupted. First, our goal will be to construct a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ on n nodes with maximum degree d, and a protocol for reliable message transmission scheme, $\mathsf{TS}^{\mathcal{G}}_{u,v}(m)$, with the following

properties. Let the set of edges that are corrupted by an adversary be denoted by $\mathcal{T}_{\text{edges}} \subset \mathcal{E}, |\mathcal{T}_{\text{edges}}| \leq \alpha nd$. We shall show that there exists a set of nodes $\mathcal{P} \subseteq \mathcal{V}$, such that $|\mathcal{P}| \geq \gamma n$, and any two nodes $u, v \in \mathcal{P}$ can communicate using $\mathsf{TS}^{\mathcal{G}}_{u,v}(m)$. This will then be sufficient to obtain a protocol for a.e. agreement (as in the work of Dwork *et al.* [5]), as well as a.e. secure computation (using the techniques from Garay and Ostrovsky [8]). Our graph will have maximum degree $\mathcal{O}(n^{\epsilon})$, for arbitrary constants $0 < \epsilon < 1$, such that $|\mathcal{P}| \geq \gamma n$, for constant $0 < \gamma < 1$.

We begin this section by constructing such a message transmission scheme on a graph of larger degree, $\mathcal{O}(\sqrt{n} \log n)$, and then show how to use that construction to obtain a scheme on a graph of maximum degree $\mathcal{O}(n^{\epsilon})$.

3.1 A.e. Agreement on $\mathcal{O}(\sqrt{n} \log n)$-Degree Graphs

We now show how to construct a graph of maximum degree $\mathcal{O}(\sqrt{n} \log n)$, and then present a protocol for a remote message transmission protocol between any two nodes $u, v \in \mathcal{P}$, tolerating a constant fraction of corrupted edges. For simplicity, we will assume that all messages in our protocols are binary. We remark that this restriction can be easily removed.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a graph on n nodes, d_v the degree of vertex $v \in \mathcal{V}$, and $\mathsf{Paths}_2(u, v)$ the set of all paths between any two vertices $u, v \in \mathcal{V}$ of length exactly 2. Let $\mathcal{G}$ satisfy the following two properties:

1. $\frac{\sqrt{n} \log n}{2} \leq d_v \leq 2\sqrt{n} \log n$ for all $v \in \mathcal{V}$; and
2. $|\mathsf{Paths}_2(u, v)| \geq \frac{\log^2 n}{2}$ for all $u, v \in \mathcal{V}$.

We will construct our transmission scheme on any graph $\mathcal{G}$ satisfying the above properties. We first observe that such a graph is easy to construct probabilistically. Consider the Erdős-Renyi random graph $\mathcal{G}(n, p)$, with $p = \frac{\log n}{\sqrt{n}}$; that is, construct the graph $\mathcal{G}$ such that there is an edge between every pair of nodes u and v, independently with probability $p = \frac{\log n}{\sqrt{n}}$ (for simplicity, we allow self-edges). Then, except with negligible (in n) probability, $\mathcal{G}(n, p)$ satisfies the conditions that we require of graph $\mathcal{G}$. (For completeness, we provide the proof of this in the full version.) We sometimes denote this process by $\mathcal{G} \leftarrow G(n, p)$. We now present two lemmas for graph $\mathcal{G}$ satisfying the two properties above.

Lemma 1. *In graph $\mathcal{G}$, no edge participates in more than $4\sqrt{n} \log n$ paths of length exactly 2 ($\mathsf{Paths}_2(u, v)$) between any two vertices $u, v \in \mathcal{V}$.*

Let $0 < \alpha_e, \alpha_n < 1$ be constants denoting the fraction of corrupt edges and corrupt nodes in the graph, respectively. Note that if we are able to design a protocol that can tolerate $\alpha_e \sqrt{n}(n-1) \log n + 2\alpha_n \sqrt{n}(n-1) \log n$ edge corruptions, then we will automatically get a protocol that can tolerate an α_e fraction of corrupt edges and an α_n fraction of corrupt nodes. Hence, let $\alpha = \alpha_e + 2\alpha_n$; we will construct a protocol that can tolerate an α fraction of corrupt edges (and no corrupt nodes). The next lemma bounds the number of nodes in $\mathcal{G}$ with poor connectivity.

Lemma 2. *Let $\mathcal{Y}_u$ denote the set of nodes v such that the fraction of paths in* $\mathsf{Paths}_2(u, v)$ *with no corrupt edges is* $\leq \frac{1}{2}$*. We say that a node $u \in \mathcal{V}$ is* doomed *if $|\mathcal{Y}_u| \geq \frac{n}{4}$. Then, in graph $\mathcal{G}$, at most $64\alpha n$ nodes are doomed.*

The set of privileged nodes $\mathcal{P}$ in $\mathcal{G}$ will simply be the nodes that are not doomed. By Lemma 2 above, we have that $|\mathcal{P}| \geq (1 - 64\alpha)n = \gamma n$ (for some constant $0 < \gamma < 1$). We now present the construction of a message transmission protocol between any two nodes $u, v \in \mathcal{P}$:

$\underline{\mathsf{TS}^{\mathcal{G}}_{u,v}(m)}$

1. For every node $w \in \mathcal{V}$, u sends m over all paths in $\mathsf{Paths}_2(u, w)$.
2. Every node $w \in \mathcal{V}$, upon receiving m over the different paths, takes the majority of the values received, and sends this value to v over all paths in $\mathsf{Paths}_2(w, v)$.
3. For every w, v takes the majority value of all messages received over $\mathsf{Paths}_2(w, v)$ as the message received from w. Then, v takes the majority (over all w) of the received values as the value sent by u.

We now show that if nodes u and v are not doomed, then the protocol described above is a reliable message transmission protocol.

Lemma 3. *Let $u, v \in \mathcal{P}$ (i.e., any two nodes in $\mathcal{G}$ that are not doomed), Then, after an execution of $\mathsf{TS}^{\mathcal{G}}_{u,v}(m)$, v outputs m with probability* 1*.*

3.2 A.e. Agreement on $\mathcal{O}(n^{\epsilon})$-Degree Graphs

In this section we present our main technical result: we show how to recursively increase the number of nodes in graph $\mathcal{G}$ from the previous section, while not increasing its degree (asymptotically), and show how to implement a reliable message transmission on such graphs (this will in turn lead to a.e. agreement protocols on such graphs with the same parameters). We will do this in two steps. Let $\gamma = (1 - 64\alpha)$. We will first show the following:

Lemma 4. *Let $\mathcal{G}$ be a graph on n nodes with maximum degree d. Furthermore, let $\mathcal{G}$ be such that it admits a reliable message transmission protocol, $\mathsf{TS}^{\mathcal{G}}_{u,v}(\cdot)$, between any two nodes u and v from a set of size at least γn nodes even in the presence of αnd corrupt edges. Then, there exists a graph $\mathcal{G}'$ on n^2 nodes of maximum degree $2d$, such that $\mathcal{G}'$ admits a reliable message transmission protocol between any two nodes u and v from a set of size at least $\gamma^2 n^2$ nodes even in the presence of $\alpha^2 n^2 d$ corrupt edges.*

In the full version [2], we show how to apply the $\mathcal{G}'$ construction from $\mathcal{G}$ recursively to obtain the desired result on graphs of degree $\mathcal{O}(n^{\epsilon})$.

Construction of $\mathcal{G}'$. We construct $\mathcal{G}'$ as follows. Take n copies of graph $\mathcal{G}$; we will call each copy a *cloud*, and denote them $C_1, \cdots, C_n$. Connect the n clouds using another copy of graph $\mathcal{G}$. We do this by connecting the i^{th} node in cloud C_j to the i^{th} node in cloud C_k by an edge, whenever there is an edge between j and k in $\mathcal{G}$. We will call such a collection of edges between clouds C_j and C_k as a *cloud-edge*. Note that the maximum degree of any node in $\mathcal{G}'$ is $2d$.

We now describe how a node u in cloud C_j will communicate with a node v in cloud C_k—call this protocol $\mathsf{TS}^{\mathcal{G}'}_{u,v}(m)$. To do this, we will first describe how two clouds that share a cloud-edge will communicate. Let every node $i \in C_j$ hold a value m_i as input (note that every node need not hold the same value m_i) and assume cloud C_j wishes to communicate with cloud C_k. We describe a protocol such that, assuming a large-enough fraction of nodes in C_j hold the same input value, say m, then at the end of this protocol's execution a large-enough fraction of nodes in cloud C_k will output m. We call this protocol $\mathsf{CloudTransmit}_{C_j,C_k}(m_i)$. Let δ be such that $64\alpha n < \delta < (\gamma - 130\alpha)n$.

$\underline{\mathsf{CloudTransmit}_{C_j,C_k}(m_i)}$

1. For every node $1 \leq i \leq n$, the i^{th} node in C_j sends m to the i^{th} node in C_k through the edge connecting these two nodes.
2. The nodes in C_k execute a.e. differential agreement protocol $\mathsf{AE\text{-}DA}(n, 64\alpha n, \delta)$ using the value they received from their counterpart node in C_j as input. (Recall that the existence of protocol $\mathsf{TS}^{\mathcal{G}}_{u,v}(m)$ between privileged nodes in $\mathcal{G}$ guarantees that one can construct an a.e. differential agreement protocol; see the full version for more details on constructing an a.e. agreement protocol on $\mathcal{G}$.)
3. Each node takes the output of protocol $\mathsf{AE\text{-}DA}(n, 64\alpha n, \delta)$ as its output in this protocol.

We are now ready to describe $\mathsf{TS}^{\mathcal{G}'}_{u,v}(m)$:

$\underline{\mathsf{TS}^{\mathcal{G}'}_{u,v}(m)}$

1. u sends m to i for all nodes i in cloud C_j using $\mathsf{TS}^{\mathcal{G}}_{u,i}(m)$ from Section 3.1. The i^{th} node in C_j receives message m_i.
2. Clouds C_j and C_k now execute protocol $\mathsf{TS}^{\mathcal{G}}_{C_j,C_k}(m_i)$ over the graph $\mathcal{G}$ connecting the n clouds[4]. Whenever cloud C_w is supposed to send a message to C_z according to the protocol, they use protocol $\mathsf{CloudTransmit}_{C_w,C_z}(\cdot)$ over the cloud-edge connecting C_w and C_z.
3. Node $v \in C_k$ takes its output in the protocol $\mathsf{TS}^{\mathcal{G}}_{C_j,C_k}(m_i)$ as the value sent by u.

We prove the correctness of the transmission scheme above through a series of lemmas. At a high level, our proof goes as follows. We will call a cloud C_j as good if it does not have too many corrupt edges within it (that is, corrupt edges of the form (u, v) with both u and v in C_j); otherwise we will call the cloud, bad. We first show that an adversary cannot create too many bad clouds. Next, we define what it means for a cloud-edge between two clouds C_j and C_k to be good (informally, the cloud-edge is good if both C_j and C_k are good clouds and there are sufficient number of edges connecting privileged nodes in C_j and C_k). We then show that the adversary cannot create too many bad cloud-edges.

[4] We again use m_i as the input argument, since the input values to nodes in C_j might be different.

Next, we show that two good clouds can communicate reliably across a good cloud-edge. Finally, we show that there exists a large set of clouds such that any two privileged nodes in any two clouds from this set, can communicate reliably. From this, the proof of Lemma 4 readily follows. We refer to the full version [2] of the paper for the complete proof of correctness of the transmission scheme.

We now arrive at our main result by applying the construction of $\mathcal{G}'$ from $\mathcal{G}$ recursively, a constant number of times, beginning with graph $\mathcal{G} \leftarrow G(n, \sqrt{n}\log n)$. That is, we obtain a transmission scheme on $\mathcal{O}(n^\epsilon)$ degree graphs and then show obtain our a.e. agreement protocol as well as the a.e. MPC protocol on the same graph. That is, we show:

Theorem 2. *For all sufficiently large n and all constant $0 < \epsilon < 1$, there exists a graph $\mathcal{G}_{main} = (\mathcal{V}, \mathcal{E})$ with maximum degree $\mathcal{O}(n^\epsilon)$, and a set of nodes $\mathcal{P} \subseteq \mathcal{V}$, with $|\mathcal{P}| \geq \mu n$ (for constant $0 < \mu < 1$) such that the nodes in $\mathcal{P}$ can execute an a.e. agreement protocol and a secure multi-party computation protocol (satisfying the security definition of* [8]*), even in the presence of of an α fraction of edge corruptions in $\mathcal{G}_{main}$ (for some constant $0 < \alpha < 1$).*

References

1. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation. In: STOC 1988 (1988)
2. Chandran, N., Garay, J., Ostrovsky, R.: Almost-everywhere secure computation with edge corruptions. Cryptology ePrint Archive, Report 2012/221, `http://eprint.iacr.org/`
3. 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, Part II. LNCS, vol. 6199, pp. 249–260. Springer, Heidelberg (2010)
4. Chaum, D., Crépeau, C., Damgård, I.: Multiparty unconditionally secure protocols (abstract). In: STOC 1988 (1988)
5. Dwork, C., Peleg, D., Pippenger, N., Upfal, E.: Fault tolerance in networks of bounded degree (preliminary version). In: STOC 1986 (1986)
6. Fitzi, M., Garay, J.: Efficient player-optimal protocols for strong and differential consensus. In: PODC 2003 (2003)
7. Garay, J., Givens, C., Ostrovsky, R.: Secure Message Transmission by Public Discussion: A Brief Survey. In: Chee, Y.M., Guo, Z., Ling, S., Shao, F., Tang, Y., Wang, H., Xing, C. (eds.) IWCC 2011. LNCS, vol. 6639, pp. 126–141. Springer, Heidelberg (2011)
8. Garay, J.A., Ostrovsky, R.: Almost-Everywhere Secure Computation. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 307–323. Springer, Heidelberg (2008)
9. King, V., Saia, J., Sanwalani, V., Vee, E.: Towards secure and scalable computation in peer-to-peer networks. In: FOCS 2006 (2006)
10. Lamport, L., Shostak, R., Pease, M.: The Byzantine generals problem. ACM Transactions on Programming Languages and Systems 4(3) (1982)
11. Neiger, G.: Distributed consensus revisited. Information Processing Letters 49(4), 195–201 (1994)
12. Pease, M., Shostak, R., Lamport, L.: Reaching agreement in the presence of faults. Journal of the ACM (1980)
13. Upfal, E.: Tolerating linear number of faults in networks of bounded degree. In: PODC 1992 (1992)