Brief Announcement: Scalable Agreement Protocols with Optimal Optimistic Efficiency

Yuval Gelles
Hebrew University of Jerusalem, Israel

Ilan Komargodski
Hebrew University of Jerusalem, Israel
NTT Research, Sunnyvale, CA, USA

Abstract

Designing efficient distributed protocols for various agreement tasks such as Byzantine Agreement, Broadcast, and Committee Election is a fundamental problem. We are interested in *scalable* protocols for these tasks, where each (honest) party communicates a number of bits which is sublinear in n, the number of parties. The first major step towards this goal is due to King et al. (SODA 2006) who showed a protocol where each party sends only $\tilde{O}(1)$[1] bits throughout $\tilde{O}(1)$ rounds, but guarantees only that $1 - o(1)$ fraction of honest parties end up agreeing on a consistent output, assuming constant $< 1/3$ fraction of static corruptions. Few years later, King et al. (ICDCN 2011) managed to get a full agreement protocol in the same model but where each party sends $\tilde{O}(\sqrt{n})$ bits throughout $\tilde{O}(1)$ rounds. Getting a full agreement protocol with $o(\sqrt{n})$ communication per party has been a major challenge ever since.

In light of this barrier, we propose a new framework for designing efficient agreement protocols. Specifically, we design $\tilde{O}(1)$-round protocols for all of the above tasks (assuming constant $< 1/3$ fraction of static corruptions) with optimistic and pessimistic guarantees:

- **Optimistic complexity**: In an honest execution, all parties send only $\tilde{O}(1)$ bits.
- **Pessimistic complexity**: In any other case, (honest) parties send $\tilde{O}(\sqrt{n})$ bits.

Thus, all an adversary can gain from deviating from the honest execution is that honest parties will need to work harder (i.e., transmit more bits) to reach agreement and terminate. Besides the above agreement tasks, we also use our new framework to get a scalable secure multiparty computation (MPC) protocol with optimistic and pessimistic complexities.

Technically, we identify a relaxation of Byzantine Agreement (of independent interest) that allows us to fall-back to a pessimistic execution in a coordinated way by all parties. We implement this relaxation with $\tilde{O}(1)$ communication bits per party and within $\tilde{O}(1)$ rounds.

2012 ACM Subject Classification Theory of computation → Distributed algorithms

Keywords and phrases Byzantine Agreement, Consensus, Optimistic-Pessimistic, Secure Multi-Party Computation

Digital Object Identifier 10.4230/LIPIcs.DISC.2023.42

Related Version *Full Version*: `https://eprint.iacr.org/2023/751` [12]

Funding This research is supported in part by an Alon Young Faculty Fellowship, by a grant from the Israel Science Foundation (ISF Grant No. 1774/20), and by a grant from the US-Israel Binational Science Foundation and the US National Science Foundation (BSF-NSF Grant No. 2020643).
Ilan Komargodski: Incumbent of the Harry & Abe Sherman Senior Lectureship at the School of Computer Science and Engineering at the Hebrew University.

[1] We use the notation $\tilde{O}(\cdot), \tilde{\Omega}(\cdot)$ to hide poly-logarithmic factors in n.

1 Introduction and Results

We propose a new framework for designing efficient fault-tolerant distributed algorithms for large scale networks. Fault-tolerance means that the functionality of the protocol should not be compromised even if some of the participants in the protocol collude and arbitrarily deviate from the protocol's specification. Basic and well-studied abstractions in this context are Byzantine Agreement (BA) [20, 19], Broadcast, and Committee Election. These serve as building blocks in many distributed protocols. Perhaps most notably, these protocols underlie essentially all secure (cryptographic) multiparty computation protocols [2, 13, 5].

We are interested in designing protocols for all of the above tasks that can be used in large scale settings, where the number of parties could potentially be really large. Let n be the number of parties involved. For protocols to be scalable, we want the amount of work or communication required by each party to be sub-linear in n. Even for the very basic tasks, such as BA, this question was out of reach for many years. The situation changed with the groundbreaking work of [18] that gave a technique for "almost everywhere" *scalable* agreement. Using their technique, it is possible to solve BA, Broadcast, and Committee Election in a scalable fashion with the caveat that only $1 - o(1)$ fraction of the honest parties agree on the output of the protocol. In terms of efficiency though, their protocol is essentially optimal: it requires only $\tilde{O}(1)$ communication rounds and $\tilde{O}(1)$ communication per party.

The model that is considered in [18] is point-to-point synchronous communication and with a computationally unbounded Byzantine (a.k.a malicious) adversary that controls a $(1/3 - \epsilon)$-fraction of the parties for any $\epsilon > 0$. Corruptions occur statically, namely, before the protocol begins (but after it is specified). Further, the adversary has full-information, namely, it sees all messages sent, even messages sent between two honest party, and it is rushing, namely, it gets to send its messages after seeing the honest party's messages for that round. This is the model that we consider in this work as well.

Since [18]'s work, there has been an effort to boost their almost everywhere agreement property into full agreement in various different settings [16, 17, 15, 3]. In the classical information theoretic setting, the work of [15] managed to get full agreement protocols for all of the above tasks at the cost of increased overhead: each party's communication is $\tilde{O}(\sqrt{n})$ bits (the round complexity remains $\tilde{O}(1)$). More than a decade since this work, it is still essentially the state of the art (excluding protocols relying on cryptographic assumptions). The work of [14] proved a lower bound saying that $\tilde{\Omega}(\sqrt[3]{n})$ communication is necessary for at least one party for a certain (non-trivial) class of protocols. See Section 2 for more details. Thus, it is still major open problem to get a full agreement protocol with $o(\sqrt{n})$ communication per party.

Optimistic/pessimistic efficiency

Given the lack of progress in getting better agreement protocols in the worst-case, we suggest a new "beyond worst-case" approach for designing protocols. Specifically, we consider protocols that could have different complexities, depending on the attacker's actions. It is important to emphasize that we want protocols that are correct/secure no matter what; i.e., all honest parties terminate and full agreement is reached. Thus, all an adversary can gain from deviating from the honest execution is that honest parties will need to work harder (i.e., transmit more bits) to reach agreement and terminate. So, it is conceivable to assume that in some applications an attacker will have no incentive to execute an attack. That is, if all the adversary cares about is breaking correctness/security (and gains nothing from a delay), then there is no point to deviate from the protocol and much better efficiency is obtained in this case. We state our main result next.

▶ **Theorem 1** (Scalable agreement compiler). *Let* $X \in \{$*ByzantineAgreement*, *Broadcast*, *CommitteeElection*$\}$ *be a task. Assume that there is a protocol* Π *for* X *tolerating* $1/3 - \epsilon$ *fraction of static corruptions for any* $\epsilon > 0$*. Then, there is another protocol* Π' *for* X *(tolerating* $1/3 - \epsilon$ *fraction of static corruptions for any* $\epsilon > 0$*) with the following features:*

Optimistic Complexity: *In an honest execution, each party sends* $\tilde{O}(1)$ *bits and the protocol terminates within* $\tilde{O}(1)$ *rounds.*

Pessimistic Complexity: *In the worst-case, the communication and round complexities of* Π' *is the same as that of* Π *plus* $\tilde{O}(1)$*.*

We remark that while we stated above that our protocols obtain optimistic efficiency ($\tilde{O}(1)$ communication overall per party) in honest executions, in fact, this is the case even if crash failures are allowed (i.e., an adversary can crash corrupt nodes).

As a corollary of Theorem 1, using the protocols of [18, 15] as Π, we get the following result.

▶ **Corollary 2** (Scalable agreement instantiation). *There are* $\tilde{O}(1)$*-round Byzantine Agreement, Committee Election, and Broadcast protocols tolerating* $1/3 - \epsilon$ *fraction of static corruptions for any* $\epsilon > 0$ *with the following features:*

Optimistic Complexity: *In an honest execution, each party sends* $\tilde{O}(1)$ *bits.*

Pessimistic Complexity: *In the worst-case, each honest party sends* $\tilde{O}(\sqrt{n})$ *bits.*

The $\tilde{O}(\sqrt{n})$ term in the pessimistic complexity bullet comes from the cost of [15]'s protocol. Any improvement on the latter will immediately translate to improved pessimistic complexity.

Technical highlight: agreement with error detection. The main building block of the above theorem is a new protocol for a relaxed variant of an agreement functionality. Recall that in an agreement functionality the goal is to guarantee that all parties terminate and agree on some specified value. We introduce a relaxation of the above requirement by requiring that (1) in an honest execution, indeed all parties terminate and reach agreement, but (2) in all other cases, all parties should agree on a special Fail symbol. With this abstraction, we can identify failure, fallback and invoke another (expensive) agreement protocol. To get Theorem 1, we implement such a relaxed agreement protocol using a protocol with $\tilde{O}(1)$ rounds and per-party communication.

Generic optimistic/pessimistic framework. More generally, we use our agreement with error detection protocol to obtain a generic framework for combining efficient optimistic protocols with less efficient pessimistic protocols. Specifically, we manage to combine an efficient protocol Π_{light} (solving a given task X) where it is guaranteed that a failure was noticed by at least one party with a less efficient protocol Π_{heavy} (solving the same task X) that is executed only if everyone knows that some failure occurred. The complexity of the combined protocol is inherited from Π_{light} in an honest execution and from Π_{heavy} in any other case.

Application to Scalable MPC

The above agreement tasks can be thought of as special cases of secure multi-party computation (MPC) [13, 2, 5]. MPC protocols enable a set of mutually distrusting parties to compute a function on their private inputs, while guaranteeing various properties such as correctness, privacy, independence of inputs, and more. We consider the problem of scalable MPC in the peer-to-peer synchronous communication model with private channels. (We emphasize that only this result relies on private channels.)

Feasibility results for (non-scalable) MPC have been long known, e.g., the BGW [2] protocol gives a method for computing an arbitrary function with communication cost that grows multiplicatively with the circuit size of the function and some polynomial in the number of parties. That is, the communication complexity of each of the n parties is $s \cdot \mathsf{poly}(n)$ when computing a function represented as a circuit of size s.[2] The question of scalable MPC, i.e., protocols where the dominant term in the complexity is just the circuit size, is still an active topic and modern results achieve MPC protocols with strong security guarantees and communication complexity $\tilde{O}(s + \mathsf{poly}(n))$ (see Section 2 for an overview and references).

We use our scalable agreement protocols to obtain a new generic scalable MPC. The properties of the resulting protocol are summarized in the next theorem.

▶ **Theorem 3** (Scalable MPC). *There is a statistically maliciously secure MPC protocol tolerating $1/3 - \epsilon$ fraction of static corruptions for any $\epsilon > 0$ with the following features. Given a circuit of size s and depth d over n inputs, the protocol has the following complexity:*
Optimistic Complexity: *In an honest execution, each party sends $\tilde{O}(s/n)$ bits.*
Pessimistic Complexity: *In the worst-case, each party sends $\tilde{O}(s/n + \sqrt{n})$ bits.*
Round Complexity: *All parties terminate after $\tilde{O}(d)$ rounds.*

The additive $\tilde{O}(\sqrt{n})$ term in the pessimistic complexity bullet comes generically from Theorem 1. At a high level, the above MPC is obtained by using our agreement protocol to generate a *quorum*: assign to each party its own representative (small and balanced) committee where there is a strong majority of honest parties. Then, we distribute the gates of the circuit to these committees. Each gate is evaluated by its assigned committees using some standard MPC (e.g., BGW). We emphasize that our protocol has the appealing feature that in an honest execution the total communication complexity is essentially equal to the circuit size, for any circuit, and it is split in a balanced manner across parties. This idea largely appeared in the scalable MPC protocol of [9]. The main difference is that we obtain optimistic/pessimistic complexity, while they only had pessimistic complexity, and this is due to the use of Theorem 1 instead of the protocol of [15].

2 Background and Related Work

Scalable agreement

The BA problem was introduced in the landmark work of Lamport, Shostak, and Pease [19]. In the following couple of decades, several protocols were presented (e.g., [10]) but they all had quadratic total overhead, that is, every party had to essentially communicate with every other party. The protocol of [18] was the first to break this barrier, but it had the caveat of almost-everywhere agreement (rather than full agreement). Extending their almost-everywhere agreement to full agreement in a scalable manner and with minimal cost is still an exciting challenge. We mention some of the key papers addressing this challenge.

First, [16] presented a protocol that satisfies full agreement but it is not balanced. That is, while most parties do communicate a sublinear amount of bits in n overall, there are few parties that communicate essentially with everyone. Several follow up works (e.g., [4, 1]) suffer from the same issue. Then, [15] presented a protocol that satisfies full agreement and it is balanced, but this comes with an extra $\tilde{O}(\sqrt{n})$ term in the communication cost.

[2] Interestingly, in BGW it was already observed that their protocol has an optimistic/pessimistic flavor where in the former the polynomial in n is slightly better than in the pessimistic case.

The extra cost in efficiency is partially explained by an $\Omega(\sqrt[3]{n})$ lower bound on the communication complexity of at least one party in any BA protocol with full agreement, due to [14]. This lower bound, however, applies only to protocols with *static filtering*. In static filtering, every party decides on the set of parties it will listen to before the beginning of each round (as a function of its internal view at the end of the previous round). It is an intriguing open problem to extend the lower bound beyond protocol with static filtering rules.

Lastly, we mention a work of [3]. They assume cryptographic and trusted setup assumptions and further that the adversary is computationally bounded. Also, they assume dynamic filter – namely, the decision of which message is received can be based on the content of received messages (in their case, every message is checked if it contains a valid digital signature). With these relaxations of the model, no lower bound is known. They showed a communication-optimal protocol: only $\tilde{O}(1)$ bits of communication per party are needed to reach full agreement.[3]

We remark that all of the above works, as well as ours, assume near-optimal resilience, i.e., up to $(1/3-\epsilon)$ fraction of corruptions (i.e., near-optimal resilience range). Less than $n/3-1$ corruptions is strictly necessary due to lower bounds of [19, 11] (unless further assumptions are made such as a trusted setup or a computationally bounded adversary).

Scalable MPC

There has been a rich line of work on scalable MPC protocols. The main goal is to design protocols where the total communication complexity scales like $O(s+\mathsf{poly}(n))$ for securely computing a size s circuit by n parties. This was studied in the context of perfect or statistical security and optimal resilience (up to $n/3-1$ or $n/2-1$ corrupted parties) e.g., [2, 13], or with perfect or statistical security and near-optimal resilience (up to $(1/3-\epsilon)$ or $(1/2-\epsilon)$ fraction of corrupted parties) e.g., [9, 6]. In all of these works a broadcast channel is assumed but its usage is limited to a number of times which is independent of the circuit size. All of these works obtain somewhat stronger notions of security than what we obtain in Theorem 3 (e.g., they often tolerate adaptive corruptions while we handle only static ones). Note that we can use our broadcast protocol from Theorem 1 to instantiate the broadcast channel in the above works, achieving statistical security for $(1/3-\epsilon)$ fraction of static corruptions.

Most related to us are the works [7, 9]. In these works the authors used the full agreement protocol of [15] to get a scalable MPC with complexity $\tilde{O}(s/n+\sqrt{n})$ to compute a size s circuit by n parties, per party. The corruption model is $(1/3-\epsilon)$ fraction static corruptions, same as ours. Our MPC prootcol is very similar to theirs (associating the wires of the circuit to quorum members); but, our description is somewhat simpler because we use generic maliciously secure MPC as black-box whereas they sometime use internal building blocks such as verifiable secret sharing. The optimistic/pessimistic aspect is new to our work. Lastly, we mention the work of [8] who studied scalable MPC protocols in an asynchronous setting.

[3] [3] have two variants presenting tradeoffs between the cryptographic assumptions and the trusted setup assumptions. Either a weaker trusted setup assumption (a public-key infrastructure and a common random string) and a stronger cryptographic assumption (SNARKs with linear-time extraction and a collision resistant hash).

References

1 Ittai Abraham, T.-H. Hubert Chan, Danny Dolev, Kartik Nayak, Rafael Pass, Ling Ren, and Elaine Shi. Communication complexity of byzantine agreement, revisited. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC*, pages 317–326, 2019.

2 Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, STOC*, pages 1–10, 1988.

3 Elette Boyle, Ran Cohen, and Aarushi Goel. Breaking the $O(\sqrt{n})$-bit barrier: Byzantine agreement with polylog bits per party. In *ACM Symposium on Principles of Distributed Computing, PODC*, pages 319–330, 2021.

4 Nicolas Braud-Santoni, Rachid Guerraoui, and Florian Huc. Fast byzantine agreement. In *ACM Symposium on Principles of Distributed Computing, PODC*, pages 57–64, 2013.

5 David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols (abstract). In *Advances in Cryptology - CRYPTO*, volume 293, page 462, 1987.

6 Ivan Damgård, Yuval Ishai, Mikkel Krøigaard, Jesper Buus Nielsen, and Adam D. Smith. Scalable multiparty computation with nearly optimal work and resilience. In *Advances in Cryptology - CRYPTO*, pages 241–261, 2008.

7 Varsha Dani, Valerie King, Mahnush Movahedi, and Jared Saia. Breaking the $O(mn)$ bit barrier: Secure multiparty computation with a static adversary. In *8th Student Conference*, page 64, 2012.

8 Varsha Dani, Valerie King, Mahnush Movahedi, and Jared Saia. Quorums quicken queries: Efficient asynchronous secure multiparty computation. In *Distributed Computing and Networking - ICDCN*, pages 242–256, 2014.

9 Varsha Dani, Valerie King, Mahnush Movahedi, Jared Saia, and Mahdi Zamani. Secure multi-party computation in large networks. *Distributed Computing*, 30:193–229, 2017.

10 Danny Dolev and H. Raymond Strong. Authenticated algorithms for byzantine agreement. *SIAM J. Comput.*, 12(4):656–666, 1983.

11 Michael J. Fischer, Nancy A. Lynch, and Michael Merritt. Easy impossibility proofs for distributed consensus problems. *Distributed Comput.*, 1(1):26–39, 1986.

12 Yuval Gelles and Ilan Komargodski. Scalable agreement protocols with optimal optimistic efficiency. Cryptology ePrint Archive, Paper 2023/751, 2023.

13 Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, STOC*, pages 218–229, 1987.

14 Dan Holtby, Bruce M. Kapron, and Valerie King. Lower bound for scalable byzantine agreement. *Distributed Comput.*, 21(4):239–248, 2008.

15 Valerie King, Steven Lonargan, Jared Saia, and Amitabh Trehan. Load balanced scalable byzantine agreement through quorum building, with full information. In *Distributed Computing and Networking - ICDCN*, pages 203–214, 2011.

16 Valerie King and Jared Saia. From almost everywhere to everywhere: Byzantine agreement with $\tilde{o}(n^{3/2})$ bits. In *Distributed Computing, 23rd International Symposium, DISC*, pages 464–478, 2009.

17 Valerie King and Jared Saia. 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*, pages 420–429, 2010.

18 Valerie King, Jared Saia, Vishal Sanwalani, and Erik Vee. Scalable leader election. In *17th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 990–999, 2006.

19 Leslie Lamport, Robert E. Shostak, and Marshall C. Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982.

20 Marshall C. Pease, Robert E. Shostak, and Leslie Lamport. Reaching agreement in the presence of faults. *J. ACM*, 27(2):228–234, 1980.