Brief Announcement: How to Trust Strangers – Composition of Byzantine Quorum Systems

Orestis Alpos
University of Bern, Switzerland

Christian Cachin
University of Bern, Switzerland

Luca Zanolini
University of Bern, Switzerland

Abstract

Trust is the basis of any distributed, fault-tolerant, or secure system. A *trust assumption* specifies the failures that a system, such as a blockchain network, can tolerate and determines the conditions under which it operates correctly. In systems subject to Byzantine faults, the trust assumption is usually specified through sets of processes that may fail together. Trust has traditionally been *symmetric*, such that all processes in the system adhere to the same, global assumption about potential faults. Recently, *asymmetric* trust models have also been considered, especially in the context of blockchains, where every participant is free to choose who to trust.

In both cases, it is an open question how to compose trust assumptions. Consider two or more systems, run by different and possibly disjoint sets of participants, with different assumptions about faults: how can they work together? This work answers this question for the first time and offers composition rules for symmetric and for asymmetric quorum systems. These rules are static and do not require interaction or agreement on the new trust assumption among the participants. Moreover, they ensure that if the original systems allow for running a particular protocol (guaranteeing consistency and availability), then so will the joint system. At the same time, the composed system tolerates as many faults as possible, subject to the underlying consistency and availability properties.

Reaching consensus with asymmetric trust in the model of personal Byzantine quorum systems (Losa et al., DISC 2019) was shown to be impossible, if the trust assumptions of the processes diverge from each other. With asymmetric quorum systems, and by applying our composition rule, we show how consensus is actually possible, even with the combination of disjoint sets of processes.

2012 ACM Subject Classification Theory of computation → Cryptographic protocols; Software and its engineering → Distributed systems organizing principles

Keywords and phrases Byzantine quorum systems, composition of quorum systems, trust models, asymmetric trust

Digital Object Identifier 10.4230/LIPIcs.DISC.2021.44

Related Version *Full Version*: `https://arxiv.org/abs/2107.11331`

Funding This work has been funded by the Swiss National Science Foundation (SNSF) under grant agreement Nr. 200021_188443 (Advanced Consensus Protocols).

1 Extended Abstract

Secure distributed systems rely on *trust*. A security assumption defines the failures and attacks that can be tolerated and names conditions under which the system may operate. Implicitly, this determines the trust in certain components to be correct. In fault-tolerant replicated systems, trust has traditionally been expressed globally, through a *symmetric* assumption on the number or kind of faulty processes, which is shared by all processes. An example of this is the well-known threshold fault assumption: the system tolerates up to a finite and limited number of faulty processes in the system; no guarantees can be given

beyond this about the correct execution of protocols. More generally, a symmetric trust assumption is defined through a *fail-prone system*, which is a collection of subsets of processes, such that each of them contains all the processes that may at most fail together during a protocol execution.

Quorum systems [14] complement the notion of fail-prone systems and are used within distributed fault-tolerant protocols to express trust assumptions operationally.

In the classical interpretation, a quorum system is a collection of subsets of processes, called *quorums*, with two properties, formally known as *consistency* and *availability*, respectively, that any two quorums have a non-empty intersection and that in every execution, there exists a quorum made of correct processes. *Byzantine quorum systems* (BQS) have been formalized by Malkhi and Reiter [11] and generalize classical quorum systems by tolerating Byzantine failures, i.e., where faulty processes may behave arbitrarily. They are the focus of this work and allow for building secure, trustworthy systems. A BQS assumes one global shared Byzantine fail-prone system and, because of that, use the model of symmetric trust. Consistency for a BQS demands that any two quorums intersect in a set that contains at least one correct process in every execution.

Motivated by the requirements of more flexible trust models, particularly in the context of blockchain networks, new approaches to trust have been explored. It is evident that a common trust model cannot be imposed in an open and decentralized or permissionless environment. Instead, every participant in the system should be free to choose who to trust and who not to trust. Damgård *et al.* [4], and Cachin and Tackmann [2] extend Byzantine quorum systems to permit subjective trust by introducing *asymmetric* Byzantine quorum systems. They let every process specify their own fail-prone system and quorum system. Global system guarantees can be derived from these personal assumptions. Extending traditional Byzantine quorum systems that use threshold assumptions, several recent recent suggestions [7, 6, 10] have also introduced more flexible notions of trust.

In this work, we study the problem of composing trust assumptions, as expressed by symmetric and by asymmetric Byzantine quorum systems. Starting from two or more running distributed systems, each one with its own assumption, how can they be combined, so that their participant groups are joined and operate together? A simple, but not so intriguing solution could be to stop all running protocols and to redefine the trust structure from scratch, with full knowledge of all assumptions across the participants. With symmetric trust, a new global assumption that includes all participants would be defined. In the asymmetric-trust model, every process would specify new personal assumptions on all other participants. Subsequently, the composite system would have to be restarted. Although this solution can be effective, it requires that all members of each initial group express assumptions about the trustworthiness of the processes in the other groups. In realistic scenarios, this might not be possible, since the participants of one system lack knowledge about the members of other systems, and can therefore not express their trust about them. Moreover, one needs to ensure that the combined system satisfies the liveness and safety conditions, as expressed by the B^3-condition for quorum intersection. Since the assumptions are personal, it is not guaranteed, and in practice quite challenging, that the composite system will indeed satisfy the B^3-condition.

Our work, whose details appear in the full paper [1], formulates the problem of composing quorum systems and gives methods for assembling trust assumptions from different, possibly disjoint, systems to a common model. We do so by introducing composition rules for trust assumptions, in both the symmetric-trust and asymmetric-trust model. Our methods describe the resulting fail-prone systems and the corresponding quorum systems.

In a different line of work, subjective trust assumptions have also been introduced with the Stellar blockchain (`https://www.stellar.org`) [13, 8, 9], a cryptocurrency ranked in the top-20 by market capitalization today. In contrast to the original, well-understood notion of quorum systems, these works depart from the classical intersection requirement among quorums. Such systems may fork into separate *consensus clusters*, each one satisfying agreement and liveness on its own. This implies that consensus may hold only "locally", and a unique consensus across disjoint clusters is not possible. More specifically, Losa *et al.* prove [9, Lemma 4] that no quorum-based algorithm can guarantee agreement between two processes whose quorums do not intersect in their model. Our work overcomes this impossibility and shows that consensus can be reached even with disjoint sets of participants, whose trust assumptions do not intersect. Moreover, we use the established notion of quorums, which enables to run many well-understood protocols, such as consensus, reliable broadcast, emulations of shared memory, and more [2, 3].

A related form of recursive composition of (Byzantine) quorum systems has been explored and utilized in the literature. The idea is that, given two systems, each occurrence of a process in the first is *replaced* by a copy of the second system. Malkhi *et al.* [12] construct and study composite BQS, such as *recursive threshold* BQS, using this idea. Hirt and Maurer [5] use this technique to reason about multiparty computation over access structures. Our approach is orthogonal to these works, in the sense that it places the two original systems on the same level. In other words, we explore the failures that two systems can tolerate when they are joined together, as opposed when one is inserted into the other.

In summary, the contributions are as follows [1]:

1. We show how to join together two or more systems in a way where processes in one system do not need a complete knowledge of the trust assumptions of those in the other.
2. We allow processes in each system to maintain their trust assumptions within their original system.
3. We define a deterministic rule to extend the trust assumptions of each system by including the new participants.
4. Our composition rules guarantee that *consistency* and *availability* will be satisfied in the composite quorum system.

References

1 Orestis Alpos, Christian Cachin, and Luca Zanolini. How to trust strangers: Composition of byzantine quorum systems. *CoRR*, abs/2107.11331, 2021. `arXiv:2107.11331`.

2 Christian Cachin and Björn Tackmann. Asymmetric distributed trust. In *OPODIS*, volume 153 of *LIPIcs*, pages 7:1–7:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

3 Christian Cachin and Luca Zanolini. From symmetric to asymmetric asynchronous byzantine consensus. *CoRR*, abs/2005.08795v3, 2021. `arXiv:2005.08795v3`.

4 Ivan Damgård, Yvo Desmedt, Matthias Fitzi, and Jesper Buus Nielsen. Secure protocols with asymmetric trust. In *ASIACRYPT*, volume 4833 of *Lecture Notes in Computer Science*, pages 357–375. Springer, 2007.

5 Martin Hirt and Ueli M. Maurer. Player simulation and general adversary structures in perfect multiparty computation. *J. Cryptol.*, 13(1):31–60, 2000.

6 Heidi Howard, Aleksey Charapko, and Richard Mortier. Fast flexible paxos: Relaxing quorum intersection for fast paxos. In *ICDCN*, pages 186–190. ACM, 2021.

7 Shengyun Liu, Paolo Viotti, Christian Cachin, Vivien Quéma, and Marko Vukolic. XFT: practical fault tolerance beyond crashes. In *OSDI*, pages 485–500. USENIX Association, 2016.

8 Marta Lokhava, Giuliano Losa, David Mazières, Graydon Hoare, Nicolas Barry, Eli Gafni, Jonathan Jove, Rafal Malinowsky, and Jed McCaleb. Fast and secure global payments with stellar. In *SOSP*, pages 80–96. ACM, 2019.

9 Giuliano Losa, Eli Gafni, and David Mazières. Stellar consensus by instantiation. In *DISC*, volume 146 of *LIPIcs*, pages 27:1–27:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

10 Dahlia Malkhi, Kartik Nayak, and Ling Ren. Flexible byzantine fault tolerance. In *CCS*, pages 1041–1053. ACM, 2019.

11 Dahlia Malkhi and Michael K. Reiter. Byzantine quorum systems. *Distributed Comput.*, 11(4):203–213, 1998.

12 Dahlia Malkhi, Michael K. Reiter, and Avishai Wool. The load and availability of byzantine quorum systems. *SIAM J. Comput.*, 29(6):1889–1906, 2000.

13 David Mazières. The Stellar consensus protocol: A federated model for Internet-level consensus. Stellar, available online, `https://www.stellar.org/papers/stellar-consensus-protocol.pdf`, 2016.

14 Moni Naor and Avishai Wool. The load, capacity, and availability of quorum systems. *SIAM J. Comput.*, 27(2):423–447, 1998.