Brief Announcement: Authenticated Consensus in Synchronous Systems with Mixed Faults

Ittai Abraham
VMware Research, Herzliya, Israel

Danny Dolev
The Hebrew University of Jerusalem, Israel

Alon Kagan
The Hebrew University of Jerusalem, Israel

Gilad Stern
The Hebrew University of Jerusalem, Israel

Abstract

Protocols solving authenticated consensus in synchronous networks with Byzantine faults have been widely researched and known to exists if and only if $n > 2f$ for f Byzantine faults. Similarly, protocols solving authenticated consensus in partially synchronous networks are known to exist if $n > 3f + 2k$ for f Byzantine faults and k crash faults. In this work we fill a natural gap in our knowledge by presenting MixSync, an authenticated consensus protocol in synchronous networks resilient to f Byzantine faults and k crash faults if $n > 2f + k$. As a basic building block, we first define and then construct a publicly verifiable crusader agreement protocol with the same resilience. The protocol uses a simple double-send round to guarantee non-equivocation, a technique later used in the MixSync protocol. We then discuss how to construct a state machine replication protocol using these ideas, and how they can be used in general to make such protocols resilient to crash faults. Finally, we prove lower bounds showing that $n > 2f + k$ is optimally resilient for consensus and state machine replication protocols.

2012 ACM Subject Classification Theory of computation → Distributed algorithms

Keywords and phrases consensus, state machine replication, mixed faults, synchrony, lower bounds

Digital Object Identifier 10.4230/LIPIcs.DISC.2022.38

Related Version *Full Version*: `https://eprint.iacr.org/2022/805`

Funding This work was supported in part by the HUJI Federmann Cyber Security Research Center in conjunction with the Israel National Cyber Directorate (INCD) in the Prime Minister's Office.

1 Introduction

In recent years there has been a surge of interest in Byzantine Fault Tolerance (BFT) and Blockchain technologies. The security of both Bitcoin and later Ethereum's proof-of-work protocols depends on a synchronous model and obtains resilience against minority corruptions [1, 2]. Following this direction there have been several academic papers that advanced *authenticated* BFT protocols and systems in the synchronous model that use more traditional membership assumptions. A major advantage of this model is that it can obtain resilience as long as $n > 2f$ which is qualitatively much better than protocols that assume partial synchrony (or asynchrony) that can only obtain resilience of $n > 3f$.

In this paper we continue this line of research into authenticated BFT protocols and merge it with yet another long line of research around *mixed-faults*. In the mixed-faults model that we study in this paper, the adversary can corrupt up to f parties in a malicious manner and can crash up to k additional parties. One motivation behind this assumption is that it allows us to model a real world case where the non-faulty parties can detect some of

the corrupted parties (via some side broadcast channel, say the internet or a secure messaging application) and publicly mark them as faulty, hence simulate a crash. Another motivation is to model the case that there is some trusted hardware that may cause some of the parties to crash if they become compromised. Another motivation, for a large set of parties, is that some of the honest parties may be offline for large periods of time, and hence can be modeled as crashed. Finally, there may be a need to model crash failures (due to hardware failures) as a separate parameter from Byzantine failures (due to an adversary).

To the best of our knowledge, this problem of authenticated BFT in synchrony with mixed-faults has not been systematically studied. It is well known that the best one can hope for in partial synchrony (or asynchrony) is security when $n > 3f + 2k$. This is where authenticated BFT in synchrony gives a major resilience advantage. The main result of this paper is that it is possible to get security *if and only if* $n > 2f + k$. We note that we do not find this bound very surprising, but we believe getting to this bound and proving tight upper and lower bounds provides new insights into how to design authenticated BFT protocols in the synchronous model.

Our Contributions – Upper Bounds. The main contribution of our paper is MixSync, an authenticated consensus protocol in a synchronous network resilient to f Byzantine faults and k crash faults if $n > 2f + k$. MixSync uses a simple technique for achieving authenticated non-equivocation in synchronous networks with mixed-faults. As far as we know, this is the first authenticated consensus protocol in a mixed-faults setting achieving a resilience of $n > 2f + k$ without limiting the power of the adversary. This is made possible by solving the task in a synchronous setting, as opposed to the partially synchronous setting which requires at least $n > 3f + 2k$ replicas. The protocol is oblivious to the number of crash faults, so it is possible to use it as long as some bound is known on the number of Byzantine replicas and at least $f + 1$ honest replicas are guaranteed to stay online.

To construct our new authenticated consensus protocol we decompose it into an outer protocol and an inner building block. We call this inner building block *Publicly Verifiable Crusader Agreement* (PVCA). We show how to construct a simple and efficient $PVCA$ protocol in a network with mixed faults. In this task, there is a commonly known sender with an input x, and replicas are required to output some value v and a proof π. If the sender is honest, every honest replica that completes the protocol outputs x and a proof π, showing that it is their actual output from the protocol. If the sender is faulty, then there exists some value v such that every honest replica either outputs v or $\perp$ with an appropriate proof. Using these proofs, replicas can convince each other that the value they received is correct, or alternatively that the sender was faulty by producing a proof for $\perp$. This task formalizes a rather strong notion of a non-equivocation round. By the end of the round, every replica that hears a message from the sender, hears the same message, in addition to proving that a given value was actually received from the sender (or that the sender was faulty). The $PVCA$ protocol relies on a simple technique: forwarding a received message to all replicas, and then sending a second message immediately after that. A replica receiving the second message knows that if its sender was non-Byzantine, the first message has already been sent to all replicas.

Using the simple idea of a double-send, we then construct the MixSync protocol. The protocol is based on the Sync HotStuff protocol, with slight adaptations made for it to solve the task of single-shot consensus. Our protocol is view and leader based, and just like Sync HotStuff, each view consists of a view change and a non-equivocation round. In order to make our protocol resilient to k crash faults, all that is needed is using the double-send

technique from our $PVCA$ protocol. In fact, we could use the $PVCA$ protocol as a blackbox inside the Sync HotStuff protocol, only requiring the addition of a view-change protocol, but we open the blackbox in order to optimize the protocol. Finally, we discuss how to create a State Machine Replication (SMR) protocol using the same double-send idea. This can either be done generically by using our consensus protocol, or by adapting optimized protocols such the Sync HotStuff protocol. Thankfully, in the Sync HotStuff protocol replicas already send two messages to all replicas after receiving a value from the leader, meaning that the only change required is making sure that replicas send them in a specific order.

These constructions suggest a possible general approach to constructing consensus and SMR protocols in the authenticated synchronous mixed-fault scenario. If a protocol mainly consists of a non-equivocation round and a view change protocol, replace the non-equivocation round with our $PVCA$ or simply a double-send, yielding a crash-resilient protocol. Specific protocols might also require adapting other parts of the protocol if they rely on specific properties of the non-equivocation round.

Our Contributions – Lower Bounds. In order to complete the picture, we also provide tight lower bounds for consensus tasks in the presence of mixed-faults in the synchronous model. First, we show that consensus is impossible in a system with f Byzantine faults and k crash faults if $n \leq 2f + k$. Secondly, in order to prove a similar lower bound for the task of SMR, we first formalize the notion of a Write-Once Register. A Write-Once Register is a shared memory object to which clients can write only once. That means that once a client manages to write a value to the register, this value is final and cannot be changed. Unlike registers that allow clients to overwrite previous values, the Write-Once Register captures a specific idea of finality that is shared with consensus protocols. In SMR protocols, replicas are required to commit to a value v_s for each log position $s \in \mathbb{N}$, which clients can read by asking replicas to send their committed values. This means that SMR protocols actually implement infinitely many Write-Once Registers. As such, we show that no protocol virtualizing a Write-Once Register exists in a system with f Byzantine faults and k crash faults if $n \leq 2f + k$, yielding a lower bound on SMR protocols as well.

References

1 Juan Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 281–310. Springer, 2015.

2 Rafael Pass, Lior Seeman, and Abhi Shelat. Analysis of the blockchain protocol in asynchronous networks. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 643–673. Springer, 2017.