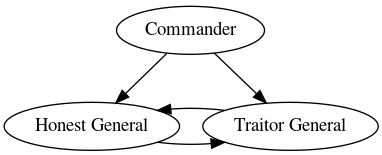
The Difference Between Hard and Impossible

Sarah Jamie Lewis

2021-07-22

The Art of Reaching Consensus



I'm not going to bore you by starting with an introduction to the Byzantine Generals Problem [1] so to jump right in, there are well established impossibility results which state that consensus cannot be reached in the case of fewer than `3m+1` generals where `m` is the number of traitors in solutions where we allow traitors to forge messages. Notably, if we remove the ability of traitors to forge messages then a 3-general solution does exist.

Since we live in a world with cryptography, we can safely assume that we can create unforgeable messages.

However for such solutions to hold, we have to also assume some properties of the underlying communication network that connects the generals. Roughly, we have to assume that the graph of honest generals cannot be significantly partitioned by a traitor [2].

Additionally, an actual majority is required to reach a decision, and we can only assert that majority with an assumption regarding how many entities are voting.

Easy enough, right?

[1] See Lamport et al. (1982) for a refresher.

[2] Remember, the *General* can also be a **traitor**. If the General has the ability partition the graph of *Honest Generals* graph into 2 connected components then they can trivially provide different instructions to each group

Crumbling Assumptions

Let's start with the network, can we assume that the network of Honest Generals is connected, or at least, sufficiently connected?

No.

For a practical example check out the wonderful world of Eclipse attacks [3]. They exist, they happen, and not just from flaws in node software, but from actual network-level adversaries too [4].

Needless to say, you **cannot** assume that your honest nodes are strongly connected in a nice safe little subgraph.

But don't worry, there is more bad news, you don't actually know how many honest nodes there are either! Your network is *open* and *decentralized* right? Anyone can join and leave as they please? There is no bouncer at the door who would be a de-facto censor. Congratulations, you now have to defend against an unbounded number adversarial nodes and you have no idea how many of them might be honest.

[3] See Heilman et al. (2015) for a cryptocurrency-oriented introduction

[4] See Apostolaki et al. (2017)

Impossible Things Before Breakfast

I'll cut to the chase. Byzantine fault tolerance cannot be all of **secure**, **decentralized** and **resource efficient** [5].

And what I just said is definitely controversial in some circles. Everyone wants to claim their baby-protocol is secure and decentralized [6] (and energy-efficiency is now in-vogue) but for the purposes of my argument we can simplify these definitions:

- **To achieve decentralization** the power to make decisions in the system must not be centralized around, or extended from, a single entity [7]. Fundamentally this means that consensus needs to originate from all participants in the protocol, rather than a select few. That means no distribution lists of approved nodes, no trusted authority lists, no proof-by-authority, no trusted consensus oracles, and no trusted witnesses [8].
- **To achieve security** the system must present sound strategies to thwart the impact of network partitions and unbounded adversarial nodes that result in a globally recognized consensus [9]. Critically, to maintain security and decentralization, *all entities in the system should be bound by these strategies*. [10]
- **To achieve resource-efficiency** the systems strategies for decentralization and/or security should not rely on access to significant quantity of a scarce resource e.g. energy. This one is fairly self-explanatory. Whether a system is resource-efficient or follows directly as a consequence of the selection of strategies for security and/or decentralization.

To be clear, at best you get to pick 2. Be secure and decentralized but use a lot of energy (Bitcoin), be secure and energy-efficient but heavily centralized (Visa) or be decentralized and energy efficient but be ridiculously insecure [11].

These properties are fundamentally intertwined such that it is trivial to sacrifice one to obtain another. Security is essential [12] and so systems must inevitably choose between decentralization or efficiency.

But what makes decentralization so expensive?

The Cost of Decentralization

Recall that you don't know how many honest nodes you have?

Bitcoin [13], for all of its woes, presented to the world an elegant solution to problem of unbounded adversarial nodes [14] - energy consumption! [15]

The *thing* about energy consumption that makes it so great is that it is hard to universally centralize. While nation states may claim control over various resources no one state, nor even a collection of states, controls **all** resources. Like it or loathe it, that is decentralization [16].

[5] oh no a trilemma!

[6] Not a trivial property to define See: Walch (2019)

[7] the inclined might want to reduce this to "hard forks must be possible".

[8] Trusted witnesses are nodes that are known to have greater connectivity, or are otherwise trusted to maintain the canonical global state

[9] Some people in the "*biz*" like to pretend that "safety" and "finality" are two separate properties that can be individually optimized, in reality there is only safety - if two honest nodes can disagree it's already over.

[10] If a single entity can propose a change to consensus and have that be accepted by the network with overwhelming probability then the system is not secure against a malicious or compromised form of that entity, and as such is neither centralized nor secure

[11] insert your favorite cryptocurrency here

[12] and yet some projects do trade off security though rarely intentionally

[13] See Nakamoto (2008)

[14] otherwise known as Sybil attacks Douceur (2002)

[15] Put too simply, proof of work is open, simple and equitable. There are no malicious entities under proof of work, only entities with more chance to shape consensus v.s. those with less chance.

[16] "the thing you are supposed to be decentralizing is power" \s

True decentralization is expensive **because** true decentralization is about **distributing power**. The sole purpose of distributing power is to **make it hard to make decisions**.

For many this is, in itself, a security argument. Decentralization is inseparable from Security (and thus security **requires** energy expenditure). Others are more trusting, literally, and so are willing to sacrifice decentralization for the sake of efficiency. [17]

[17] Dare I repeat myself again?

The cost associated with proof-of-work is designed to act as an incentive against bad-behavior. To remove that cost is to remove the incentive. As such any replacement must, by definition, be as costly to malicious actors.

Modern proposals have attempted to circumvent this issue by restricting costs to *only* bad actors requiring ever more sophisticated methods of surveillance [18] of block producers such that punishments can be applied, but more often than not punishing otherwise honest actors for failing to maintain a perfect setup [19].

[18] which also now relies on having a complete, and secure, view of the network - some might define this as circular reasoning...

[19] See "Slashing Risks and Validator Diligence" Staked (2019)

Lest this become an apologetica let me specify that there are obvious issues with proof of work that are not addressed here, and there may be other resource-intensive mechanisms which achieve similar goals - however they must be resource intensive [20].

[20] if there is no cost associated with decision making, then decision making is trivially influenced (by definition).

Those who would sacrifice security...

Achieving decentralization without resource expenditure is a contradiction in terms. Those who believe they have done so have tricked themselves in one of two fundamental ways (often both at the same time):

1. The system is actually centralized i.e. the security rests on an assumption that some entity within the system is not malicious.
2. The protocol is simply not secure i.e. the system is trivially broken via a malicious entity partitioning the network of honest participants or influencing the vote via cheap participation [21].

[21] if it is cheap to participate then there is no bound on the number of adversarial nodes (by definition)..

It's very easy to design an insecure protocol. Everyone who has worked in distributed systems has designed an insecure protocol.

It's easy because there is no siren that sounds when you deploy an insecure solution.

secure, **decentralized** and **resource efficient** It's not hard to do all 3. It is impossible. That's the difference.

References

Maria Apostolaki, Aviv Zohar, and Laurent Vanbever. Hijacking bitcoin: Routing attacks on cryptocurrencies. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 375–392. IEEE, 2017.

John R Douceur. The sybil attack. In *International workshop on peer-to-peer systems*, pages 251–260. Springer, 2002.

Ethan Heilman, Alison Kendler, Aviv Zohar, and Sharon Goldberg. Eclipse attacks on bitcoin's peer-to-peer network. In *24th {USENIX} Security Symposium ({USENIX} Security 15)*, pages 129–144, 2015.

Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, pages 382–401, July 1982. URL https://www.microsoft.com/en-us/research/publication/byzantine-generals-problem/.

Satoshi Nakamoto. Bitcoin whitepaper. *URL: https://bitcoin.org/bitcoin.pdf*, 2008.

Staked. Slashing risks and validator diligence. *URL: https://medium.com/@staked/slashing-risks-and-validator-diligence-f6901cc9622a*, 2019.

Angela Walch. Deconstructing'decentralization': Exploring the core claim of crypto systems. *Crypto Assets: Legal and Monetary Perspectives (OUP, Forthcoming)*, 2019.