This is a cross post of this

Quantification of properties allows us to measure how well a system might perform. Quantification also allows us to effectively optimize systems, because if we can quantify something, we can know if things are getting better or worse.

In 2017 Balaji S. Srinivasan, described how it's possible to quantify decentralization of a project based on mining, client usage, dev commits, exchange volume, liveliness of nodes & ownership of capital by crypto address.

Decentralization is not however the goal of a particular blockchain project, I think the decentralization metric is great to have as an overview of how power is distributed within a blockchain project. The use of decentralization (transfer of authority from a single institution to multiple local authorities) is to achieve trust-minimization.

"Decentralization is what allows Bitcoin to substitute an army of computers for an army of accountants, investigators, and lawyer's" - Nick Szabo

Bitcoin is basically designed around payments. So decentralization of the bitcoin network vaguely guarantees that the payment mechanism will be trust-minimized at the same security level.

One thing with smart-contract platforms like Ethereum is that, they are plain platforms without a specification for it's applications.

For this reason, the security mechanisms of the platform like Ethereum are detached from the security mechanisms of the applications or "dapps" layer.

This is why the Ethereum network can be decentralized & trust-minimized but the dapps on it may not be trust-minimized.

What trust-minimization is

Trust minimization is basically the ability for one to mitigate a specific class of harmful agents (also known as defectors).

Liars

Liars are agents that feed the system with false information.

Thieves

These take assets from other agents, without the original owners consent.

Cheats

Cheats are people that break rules of a game.

Most people mistakenly regard trust-minimization as a binary circumstance where a dapp is either centralized (you are relying on the kindness of strangers to not rip you off) or trust-minimized (complete protection from defectors via cryptography, game-theory or both).

But trust-minimization is a spectrum & we need ways of measuring it. This is whatZack Hess is trying to develop with trust theory.

It goes as follows;

- First we pick a dapp whose trust-minimization we want to quantify.
- Then, we outline the mechanisms that make it up.
- Each mechanisms is given a scale, the smaller the number the better;

Level 1

These are provably secure cryptographic protocols often relying on entropy as a security mechanism. Like hashing algorithms & public key cryptography. In level 1, it doesn't matter how much resources an attacker has as it is practically impossible to guess the source of randomness since the numbers are astronomically large.

Level 1 mechanisms can be considered fully trust-less.

Level 2

Mechanisms where the costs of breaking a system are are more than the payoff an attacker gets from behaving maliciously.

Basically, these are mechanisms where it's possible to cheat, but it will be unprofitable to do so. One thing worth noting is that here, we should have attribution. i.e On top of it being unprofitable to cheat, lie or steal. We can know who behaved maliciously & can act accordingly by punishing the defector (e.g forking, slashing, reduce reputation)

Level 3

In level 3 it is possible to cheat, & the pay off from behaving maliciously is either profitable or net.

Here we can also know who did what & can punish them accordingly like in level 2

Level 4

This is the worst security level (completely centralized). Here, it is possible & profitable to cheat. Plus no one can know who did what for a significant period.

Example of usage

If we have a crypto-economic primitive likehash-time-locks.

Hash time locks are crypto economic primitives that allow execution of instructions only when two conditions are satisfied.

- The first condition is that the number of mined blocks or “block-time” should be greater than a given value.
- The second condition is that the user must submit a unique secret piece of data

For simplicity, lets say, hashlocks rely on only two things to achieve this;

a) Time locks (block headers)

Time-locks only execute instructions when the number of blocks are beyond a given number. Sinceblock-time is not based on cryptography but game theory (i.e can be attacked via 51% attack) and would probably be unprofitable for the attacker, we put this at level 2.

b) Hash locks (cryptography)

Hashlocks only execute instructions when a unique piece of data/secret is submitted. If a hash-lock is secure, that is, collusion resistant hashing algorithm, we can say it’s protected by entropy & hence put it in the level 1

Sub-levels of security

So now, we have a crypto economic mechanism with two separate levels of security. To have a metric that represents levels of security

for both mechanisms. We sum up both levels of security.

For example, time-locks would be put in level 2 and hash-locks have a trust security level of 1.

This gives us a combined trust level of 2.1, “.” is an indication of sub-levels in the crypto-economic mechanism & 1 to indicate it also has a component of being purely cryptographic.

When adding different trust levels, it’s important that the bigger number comes first, and the smaller number comes second.

Trust theory, clarifies the spectrum of risks that come with a dapp instead of every one having to read a white-paper, they may not fully understand. On the other hand several, dapp rating agencies using Hess’s trust theory would also to used rate dapps.