Threshold signatures allow you to split a key into N shares, such that M out of them are required to construct a valid signature. Essentially, these signatures are like multi-sig, but there's only a single key (that's split) and therefore each threshold-signed transaction only produces a single signature (as opposed to N).

To sign a transaction, M of these shares are used in an MPC protocol that ensures that each party only ever sees a single share. Similarly, the shares are generated in a distributed fashion. Both of these ensure that the key never lives in a single location, where it can be attacked.

Why is this important?

- Cheaper gas costs - N parties can now produce a single signature off-chain instead of sending N separate ones. This is huge reduction is gas costs.

- Added privacy - the signed transaction looks like it was produced by a single party. Not incidentally, this idea can be used as a building block to building mixers as well.

The first reason is especially appealing for Enigma. In our network, after a task is completed, a signed payload, signaling a proof of correct execution, is sent alongside other meta data to the Enigma Contract on Ethereum, which verifies that signature comes from a proper TEE. Currently, a single worker is sampled per computation, but in the future, the idea is to have multiple workers jointly work on a single task (this adds an additional layer of security beyond the TEE, and also ensures higher availability).

The problem is that by doing so the number of txs that needs to be sent and verified on-chain increase from 1 --> N. Threshold signatures solve that problem, and make increasing the number of workers as cheap (on-chain) as a single worker is.

Threshold signatures have been extremely practical for the past couple of years, but one problem that was critical to our network and somewhat limited their applicability was the lack of cheater identification. Essentially, if someone in the quorum cheats, the signature generation fails and either everyone needs to lose money (unfairly) or no one will. This is a big problem in tBTC proposal as well - which I commented on a while ago (https://twitter.com/GuyZys/status/1162736363730558976).

Yesterday, in a presentation by Goldfeder (same researcher that built a previous state of the art threshold signature system) they claim to solve that problem efficiently. It's important to note that we've known how to solve it before - even in my thesis I had to deal with cheater identification for the general MPC case. That said, cheater identification is EXTREMELY expensive for general computations, so that was by far the most prohibitive part of my work, but Goldfeder claims that for the specific case of threshold signatures, they have a very efficient scheme (which makes perfect sense!).

The details are still nebulous, but this is an exciting development that makes TSS practical for Enigma.

Another added benefit for this scheme is that signing can be done non-interactively. This is the second proposal in 2019 to enable this. Why is non-interactivity important? Well, for the most part, this is why a lot of proposals opted to use threshold Schnorr signatures in Bitcoin and Ethereum (e.g., ChainLink instead of threshold ECDSA. With this new advancement, there's no longer a material reason to use Schnorr, since validating ECDSA sigs is much cheaper, and I'd argue that this benefit outweighs any added off-chain complexity.