Excited to finally have our paper out that describes a protocol for improving security for users of DeFi services/DEXs through MPC/threshold signatures. Currently, using DeFi services/DEXs users have to trust people and devices to hold and secure secret keys that fully control their funds. Requiring such level of trust for day-to-day operations poses social and technical security risk, which is a significant limitation for professional users such as market makers, who often host their trading algorithms on cloud infrastructure and operate on shared accounts within trading firms. The protocol presented in our paper replaces the signing algorithm, operating on a single device with a secret key, with a client/server protocol that protects from the client holding the secret key as a single-point-of-failure. With the client/server protocol, the client has an API key and can generate pre-signatures that are then sent to the server, and the server will only finalize the signatures based on a security policy (and cannot generate signatures unilaterally either). A security policy can describe any computable property. For example, a policy can restrict an API key to trade on certain markets only and to withdraw funds only to a specific address, restrict access from a specific geolocation only, or require biometric information. In this way, users can provide restricted access to their funds, significantly limiting downside risk in the event their software or systems are compromised. The protocol can be applied to any existing DeFi service/DEX. We have deployed the protocol on Nash exchange as well as on Uniswap and 1inch.

We are looking forward to hearing your feedback, further use cases that come into your mind, and any questions you may have!

The full paper is published on arXiv: [2106.10972] Improving security for users of decentralized exchanges through multiparty computation.

The code is available on GitHub: nash-rust/mpc-wallet/nash-mpc at master · nash-io/nash-rust · GitHub.