enabled by Sidechains do not enjoy the same security guarantee provided by Bitcoin, and micro-payment channels only work for a few applications. ELASTICO, however, allows scaling up the underlying blockchain protocol without degrading any security property.

Buterin *et al.* also address the scalability problem in blockchain with sampling and challenging techniques [125]. Similar to ELASTICO, the paper's approach is to use sharding. However, the protocol "randomly" samples verifiers to verify others' updates, and allows users to challenge others' verification results if they ever detect an invalid update. The solution relies on a random seed, for which the paper does not provide any security analysis. Further, the paper does not consider byzantine adversaries but rational ones in a "cryptoeconomic" threat model, which is different from the threat model that we consider in this paper.

Recent non-peer-reviewed proposals including Stellar [106], Ripple [50], and Tendermint [84] claim to support high transaction rate, but either have weaker threat models or are not as scalable as ELASTICO. Specifically, Tendermint assumes all identities are known before the protocol starts, thus is not applicable in decentralized environments like cryptocurrencies. Besides, Tendermint is essentially a variant of PBFT [4], which has its own scalability limitation if the network size grows as we discussed in Section 5.1. Plus, the network nodes in Ripple and Stellar are permissioned, hence it faces no challenges of establishing identities. For instance, identities in Stellar need financial agreements or reputations of others to form their "slices" (or committees). In Elastico, these have to be chosen randomly based on computational assumptions.

5.5.3 Prior Byzantine Consensus Protocols

There have been significant efforts devoted to developing scalable communication-efficient consensus protocols. The idea of dividing the users into committees (as we do in this paper) is prevalent in the existing literature; first introduced by Bracha [38].

If the users are honest, but crash prone, there exists an optimal algorithm with $\Theta(n)$

[4] http://tendermint.com/posts/tendermint-vs-pbft/