MPC generally has two flavors of implementations - based on Garbled Circuits (the original construction by Yao, abbreviated as GC) and based on Secret Sharing (usually either additive or threshold, e.g., Shamir).

Most research into scalable, efficient MPC in the last decade+ focused on improving protocols based on secret sharing. This is because secret sharing protocols generally require information-theoretic primitives, which are blazing fast compared to cryptographic operations. Also, with improvements like SPDZ, the overhead of adding more parties became linear, whereas in the GC case, the computational overhead scales quadratically. It's important to note that for this reason, most research on efficient 2PC protocols kept using the GC approach, as opposed to secret-sharing ones.

Although secret-sharing (SS) schemes are cheap computationally, and scale relatively well with the number of parties, they scale poorly with the depth d

of the circuit being computed. This is because the number of network round-trips is O(d)

, which is not too bad in a LAN setting, but becomes an issue in an internet-like setting where nodes are located everywhere around the globe.

This is why we've been seeing a trend shift in the past couple of years. More and more efficient protocols are being based on top of the BMR protocol, which in itself is a generalization of the original GC approach, that also utilizes SS. For that reason, BMR in many ways feels like a hybrid between GC and SS-based MPC.

So why go back to BMR again? After all, the protocol is almost 30 years old. The primary reason is that BMR is a constant-round protocol

. This means that regardless of the computation, the network will only have to sync a few times before successfully obtaining a result. Just for reference, a simple division of two secret fixed-point numbers in the SS approach (with arithmetic circuits) requires roughly a couple of dozen rounds (in my thesis I was able to reasonably achieve 11 rounds for 32-64 bit numbers, which is state-of-the-art). As mentioned above, having a minimal amount of rounds is potentially the most important factor for a global, open network, which Enigma strives to become.

Given the hybrid nature of BMR, it can actually serve as a kind of wrapper protocol that enjoys a lot of the advancements in the space obtained in recent years (e.g., SPDZ, TinyOT, fast OT extensions). It therefore makes sense to revisit its applicability for internet-scale MPC, as has recently been done in work by Wang et al. and work by Hazay et al.. To be clear, we're still quite far away from MPC being able to replace normal computation, but for many secret contracts

, who already run in a slower environment (blockchain), it could suffice.