Category 3: Polkadot PoCs, Papers & More

We seek proof-of-concept cryptographic back ends, or interesting student papers or pre-specs, which should be Polkadot flavored in the broadest possible sense.

What is Polkadot?

Polkadot is heterogeneous sharded blockchain, which means two things:

First, the shards called parachains each run different code, controlled by their own governance.

Second, Polkadot itself runs an interactive cut-n-choose proof of correctness for the parachains, which remains secure under conventional Byzantine assumptions, so parachains interact easily without the extra complexity, latency, and poor security assumptions required by bridges.

What cryptography works on Polkadot?

As parachains' block verification code is provided in WASM, almost all cryptographic primitives should typically work out-of-the-box on Polkadot, albeit with performance penalties similar to browsers. Arkworks provides a wonderfully powerful environment for doing zero-knowledge proofs on parachains, especially since Polkadot tooling is largely in Rust.

We do need deterministic verification like any blockchain, so on-chain verifiers cannot employ system randomness or secret keys. Yet, parachains do control the native code for their own collators aka block producers, meaning native execution speed for MPC protocols or similar off-chain protocols.

We do not waste cycles doing gas accounting, but each parablock recieved only about 2 seconds of CPU time.

What counts as Polkadot flavored?

As proof-of-concept code…

Anything novel goes, provided the prover and verifier use Arkworks or Rust more generally. Avoid smart contract languages or higher-level platform specific zk languages.

A few ideas include…

Games: Any zero-knowledge circut or other cryptographic work which provides a back-end upon which one could build games. An *on-chain game* proceeds by blockchain transactions, while an *off-chain game* proceeds by state channel updates, but permits registering an ending state on-chain. An *open world* game potentially involves most other parachain users somehow, perhaps via rooms if using state channels.

Batch verifiers: We have three entry points when verifying parachain blocks, `on_initialize`, individual transaction verification, and `on_finalize`. Assume arbitrary data could be passed between these, so partial aggregation schemes like SnarkPack or Schnorr half-aggregation work well. What partial aggregation does your favorite zero-knowledge proof system favor?

Avoid full ZK roll ups though: ZK roll up parachains could strengthen Polkadot's Byzantine assumptions to cryptographic assumptions. Yet, realistically zk roll ups shall never have performance competitive with the "cut-n-choose roll up" already running on Polkadot. Also, roll ups have "differently cryptographic" concerns, like availability, which makes them overly too complex for student projects.

It's suffices to implement, document, and explain interesting cryptography well. Actually building a working parachain on cumulus involves non-cryptographic tasks, like inventing transaction weights, which feels out-of-scope for a short hackathon.

As student papers or pre-specs…

Any serious multi-party computation proposal likely requires networking protocols beyond the gossiped memepool framework typical of blockchains. We found MPC crates remain less developed than ZKP crates in Arkworks and elsewhere.

We thus feel multi-party computation (MPC) submissions could omit proof-of-concept code, as otherwise students risk being bogged down in too many non-cryptographic tasks, but proof-of-concept code is still appreciated.

A few vague ideas include…

User-collator MPC: What MPC protocols make sense between users and block producers? Games employing MPC? A zero-knowledge liquidity pool?

Exploit user-collator anonymity: We're building a secret single-leader election (SSLE) scheme called sassafras for polkadot and its parachains. At some point in the future, sassafras shall support dialog between user agents and upcoming block producers, with both sides remaining anonymous, just imagine a separate Tor `.onion` address for each upcoming block production slot. How would you use sassafras' anonymous channels?

Inter-collator MPC: What MPC protocols make sense between different block producers? In sassafras, we no longer waste resources on mempools, but instead each block producer knows only a fragment of upcoming transactions. Is an automatic market maker possible over MPC? Linear program solvers?

Always…

Cite your sources and your dependencies' source code. If you write code then retain or at least reference the original git history for your dependencies.

Prizes:

1st Prize: $5k
2nd Prize: $3k
3rd Prize: $2k

Plus - Winners are nominated for admission to the Polkadot Blockchain Academy being held this summer at Berkeley; more details here https://polkadot.network/development/academy/