ANALOG

By: Herman Pardo

ANALOG

TOPIC FOCUS

What are the different variations of the ZKPs and which one of them is better?

What are ways that ZK-proofs can be used for scalability?

INTRODUCTION

When the internet first started, you can easily browse the internet using HTTP though it allowed data to be seen by outsiders. Private communications to businesses, banks, government, etc., wouldn't have succeeded until the release of HTTPS which enabled an encrypted security connection.

We are now on the same fork on the road in the blockchain space. The problem with HTTPS is that it only works on a point-to-point connection, not on the architecture that blockchains use because of decentralization. Here is where ZKPs come to play. Zero Knowledge Proofs aim to be the standard of security on blockchains by not only creating a secure transaction, but also through keeping data confidential on both sides so no personal data is shared. These are mathematical algorithms created to prove TRUST while not revealing a user's data.

Two Main Types of proofs:

- Interactive zero-knowledge proofs: The prover and the verifier interact several times. The verifier challenges the prover who provides replies to these challenges until the verifier is convinced.

- Non-interactive zero-knowledge proofs: Proof delivered by the prover can be verified by the verifier only once at any time. This type of ZKPs requires more computational power than interactive ZKPs.

A great example future use of this technology would be a person applying for a mortgage at a bank that's on a blockchain. Instead of the bank pulling all your credit scores, employment information, and other personal information, a ZPK proving system is used to let the bank know that you have all the required qualifications without them seeing any of your data.

ANALOG

TOPIC 1

What are the different variations of the ZKPs and which one of them is better?

VARIATIONS

Usually named with the combination of the last names of the team's members, the following are the variations of security algorithms :

- GGPR13
  - Pinocchio (PGHR13)
  - BCGTV13
  - Geppetto (CFHKKNPZ14)
  - BCTV14a
- BCTV14b
  - Coda (MS18)
- CTV15
- ZKBoo (GMO16)
- Groth16
  - GM17
  - BG18
  - DIZK (WZCPS18)
- BCCGP16
  - Bulletproofs (BBBPWM17)
- Hybrid Interactive ZK (CCM16)
- ZKB++ / Picnic (CDGORRSZ17)
- Ligero (AHIV17)
- Hyrax (WTSTW17)
- zk-STARKs (BBHR18)
- Updatable Universal CRSs (GKMMM18)
  - Sonic (MBKM19)
- Hybrid NIZK (ACM18)
- Aurora (BCRSVW18)
- Libra (XZZPS19)

ANALOG

TOPIC 1
Cont.

OPERATIONS OF PROVING SYSTEMS

Name	Language	Curves	Proving systems
libsnark	C++	BN254	Groth16, BCTV 14a, BCTV14b, CTV15
bellman	Rust	BLS12-381	Groth16
dalek bulletproofs	Rust	ristretto255	BBBPWM17
adjoint-io bulletproofs	Haskell	secp256k1	BBBPWM17
DIZK	Java	BN254	Groth16
snarkjs	JavaScript	BN254	Groth16, BCTV 14a
websnark	WebAssembly	BN254	Groth16

Best ZKP

From data, examples, and technology point of view, the best (at the moment of this writing) proving system would be Groth16 based on the following:

- Distributed implementation of Groth16 allows it to be programmed in more languages than others (C++, RUST, Java, WebAssembly)
- Enables zkSNARK computations of up to billions of logical gates (100x larger than prior art) at a cost of 10µs per gate (100x faster than prior art)
- Implements distributed polynomial evaluation/interpolation, distributed Lagrange polynomial computations, and distributed multi-scalar multiplication.

ANALOG

TOPIC 2

What are ways that ZK-proofs can be used for scalability?

ZK ROLLUPS

Transactions on a chain are timely and costly factors that limit its scaling abilities. For this reason, there are layer 2 solutions that run off chain and report to the main chain periodically. For example, rather than having hundreds of cars drive people (transactions) from one city to the next, they can all just take one plane. It's cheaper, faster, and allows the mass scaling of people traveling from city to city.

How is this scaling done in the blockchain that adds security?

ZK Rollups. Layer two solutions that aggregate transactions to speed up the network and transactions. They do the verifying for validators without having complete knowledge of the transaction.

Great example is the current development of ETH 2.0. The current ETH network supports around 30 transactions per second, which is why transaction fees are sometimes more than the transaction itself. When ZK rollups are ready on ETH 2.0, that number jumps to a minimum of 1,000 TPS with ability to go around 100K TPS! All while adding the security of making sure that your data isn't shared with anyone else.

Real life use cases that will need and use this security scaling tech:

- Buying a car
- Government transactions
- Applying for a mortgage and buying a home
- Any purchases on blockchains the require anonymity

The End