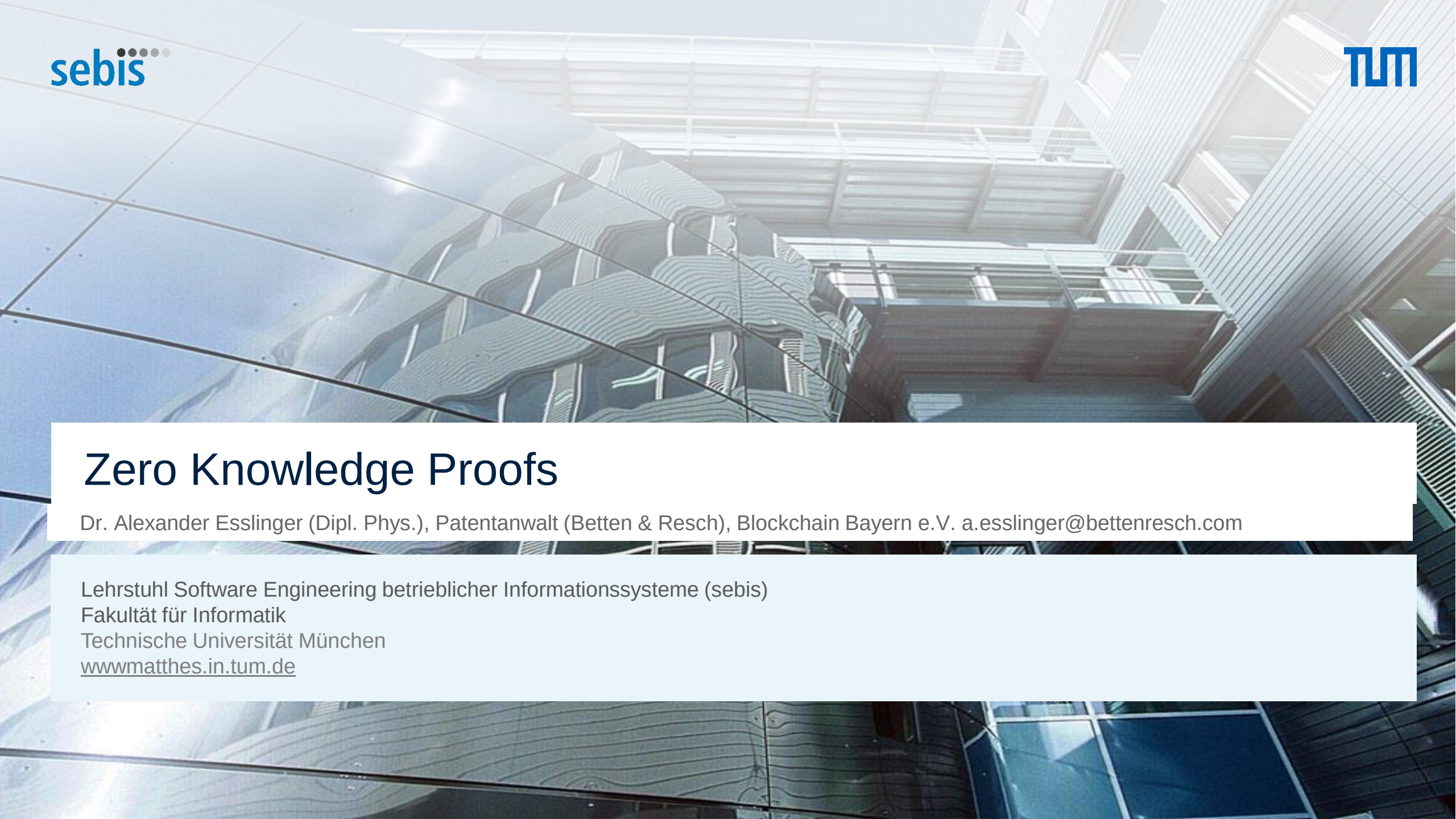


Zero Knowledge Proofs

Dr. Alexander Esslinger (Dipl. Phys.), Patentanwalt (Betten & Resch), Blockchain Bayern e.V. a.esslinger@bettenresch.com

Lehrstuhl Software Engineering betrieblicher Informationssysteme (sebis)
Fakultät für Informatik
Technische Universität München
wwwmatthes.in.tum.de

Zero Knowledge Proofs: Outline

1. Introduction

   Definition, Example, Motivation, History
2. How do they work ?

   The PCP Theorem, Computational Integrity Proofs, Expansion Codes, Illustrative Example, (Non) Interactive proofs, Trusted Setup, zk-SNARKS, Blockchain scaling, Trade-Offs
3. Applications

   Privacy, Scalability, Business Applications
4. Related Topics

   Verifiable Delay Function, Multi Party Computation, Fully Homomorphic Encryption, Stateless Blockchains
5. Resources and References

Zero Knowledge Proofs: Introduction: Definition

Zero Knowledge Proofs are a particular type of cryptographic systems.

Definition

A zero-knowledge proof is a method by which one party (the prover) can prove to another party (the verifier) that the prover knows a value x, without conveying any further information apart from the fact that they know the value x.

The essence of zero-knowledge proofs is that it is trivial to prove that one possesses knowledge of certain information by simply revealing it; the challenge is to prove such possession without revealing the information itself or any additional information.

Zero Knowledge Proofs: Introduction: Example

Digital signatures based on PKI (RSA-Algorithm or ECC, Cryptographic Basics, slide 36) are a special example of Zero Knowledge Proofs:

The person holding the private key can convince any public key holder of his/her knowledge of the private key without revealing the private key itself.

Digital signatures, however, can only sign a small amount of data.

ZKPs and, more generally, Probabilistically Checkable Proofs (PCPs), can verify arbitrary amounts of data.

Zero Knowledge Proofs: Introduction: Definition

Properties of ZKPs

Completeness: A statement is true if an honest prover can convince an honest verifier.

Soundness: If the prover is dishonest, they can't fool the verifier.

Zero Knowledge: No information is revealed to the verifier.

Zero Knowledge Proofs: Introduction: Motivation (1)

Why are Zero Knowledge Proofs (ZKPs) interesting in the Blockchain world ?

ZKPs and PCPs can help solving two major problems occurring in Blockchain networks:

- Blockchain Trilemma: Scalability – Security – Decentralization
- Privacy dilemma: Transparency – Privacy

Zero Knowledge Proofs: Introduction: Motivation (2)

Traditional trust model: Trust in person/organization of Authority.

„I, Person of Authority, hereby verify that account xyz at bank ABC holds EUR 1.900.000.000 per 31 December 2019.“

Blockchain trust model: Don´t trust, verify. Distributed/Inclusive accountability

The blockchain trust model is based on both cryptography and game theoretical incentives. ZKPs shift some of the burden away from game theory to cryptography.

Zero Knowledge Proofs: Introduction: History

1985: Zero Knowledge Proofs were first published by Goldwasser, Micali, Rackoff under the title: "*The Knowledge Complexity of Interactive Proof Systems*". (Goldwasser and Micali won the Turing Award in 2012).

1992: PCPs were introduced by Arora and Safra and other authors

May 2013: Eli Ben Sasson proposes use of ZKPs for Bitcoin at Bitcoin conference

October 2016: Privacy Coin Zcash based on ZKPs goes life

2019/2020: "Cambrian Explosion" of new ZKP-Systems, ZKP-based scaling solutions go life

Zero Knowledge Proofs: How do they work?: The PCP Theorem

The PCP theorem

Every decision problem in the NP complexity class has probabilistically checkable proofs of constant query complexity and logarithmic randomness complexity.

On an intuitive level, the PCP theorem says that there exists a probabilistic verifier for proofs of mathematical assertions which can look only at a constant number of bit positions at the proof and yet with some positive probability catch any mistake made in a fallacious argument. (Shafira Goldwasser)

Quantitively: For an error probability of 2^{-k}, 3k bit positions have to be verified.

(Example: Soundness error probability of $2^{-40} \approx 10^{-12}$ with 120 bit verification)

Zero Knowledge Proofs: How do they work?: Computational Integrity Proof

Computational Integrity Proofs are probabilistic proof systems based on the PCP theorem allowing a prover to convince a verifier of the correctness of an arbitrary computation with an efficiency that is exponentially faster than naïve checking of the computation.

Properties:

Completeness: A statement is true if an honest prover can convince an honest verifier.

Soundness: If the prover is dishonest, they can't fool the verifier.

Succinctness: Exponentially faster verification than naïve checking of the computation.

Zero Knowledge (Bonus): No information is revealed to the verifier.

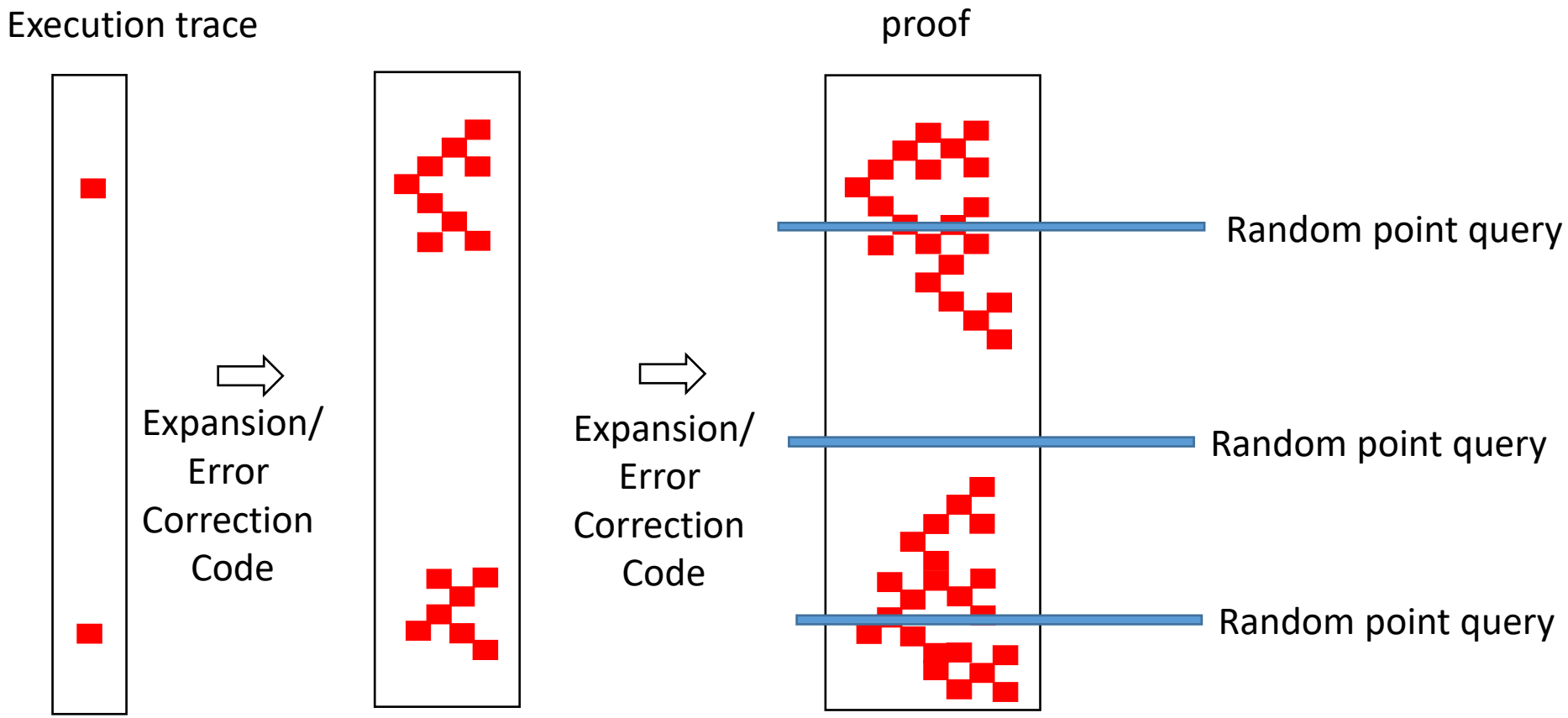
Zero Knowledge Proofs: How do they work?: Expansion codes (1)

A transcript of the original computation is expanded into a proof using an error correcting code (e.g. a Reed-Solomon-Code, polynomial commitment), which spreads any errors within the proof.

A low number (e.g. 3) of stochastical queries by the verifier are sufficient to prove the correctness of the computation with high probability.

Zero Knowledge Proofs: How do they work?: Expansion codes (2)

Illustration:



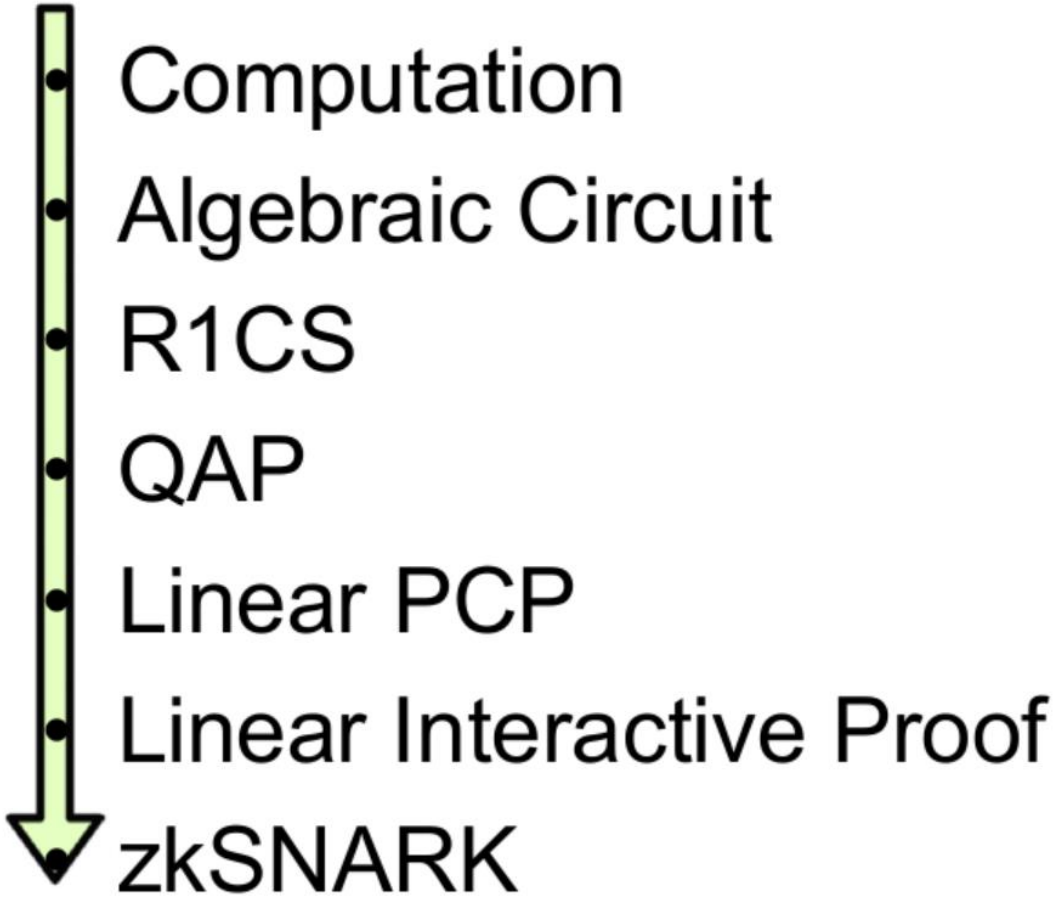
Zero Knowledge Proofs: How do they work?: Illustrative Example

						SUM	
	0	0	0	0	0	0	
	75	11	25	56	50	217	
	75	11	25	56	50	217	
	22	48	99	88	33	290	
	97	59	124	144	83	507	
	80	58	91	34	91	354	
	177	117	215	178	174	861	A
	45	62	33	28	99	267	B
	222	179	248	206	273	1128	C=A+B?
	8	63	50	51	25	197	
	230	242	298	257	298	1325	
	51	75	56	8	11	201	
A	281	317	354	265	309	1526	
B	28	22	88	45	48	231	
C=A+B?	309	339	442	310	357	1757	
	34	80	34	80	58	286	
	343	419	476	390	415	2043	
	88	45	28	22	62	245	
	431	464	504	412	477	2288	
	56	8	51	75	63	253	
	487	472	555	487	540	2541	
	50	51	8	63	75	247	
	537	523	563	550	615	2788	A
	33	28	45	62	22	190	B
	570	551	608	612	637	2978	C=A+B?
	91	34	80	58	80	343	
	661	585	688	670	717	3321	
	99	88	22	48	45	302	
	760	673	710	718	762	3623	
	25	56	75	11	8	175	
	785	729	785	729	770	3798	
	11	50	63	25	51	200	
	796	779	848	754	821	3998	
	48	33	62	99	28	270	
	844	812	910	853	849	4268	
	58	91	58	91	34	332	
	902	903	968	944	883	4600	
	62	99	48	33	88	330	
	964	1002	1016	977	971	4930	
	63	25	11	50	56	205	
	1027	1027	1027	1027	1027	5135	

						SUM	
	0	0	0	0	0	0	
	75	11	25	56	50	217	
	75	11	25	56	50	217	
	22	48	99	88	33	290	
	97	59	124	144	83	507	
	80	58	91	34	91	354	
	177	117	215	178	174	861	A
	45	62	33	28	99	267	B
	222	179	248	207	273	1129	C=A+B?
	8	63	50	51	25	197	
	230	242	298	258	298	1326	
	51	75	56	8	11	201	
A	281	317	354	266	309	1527	A
B	28	22	88	45	48	231	B
C=A+B?	310	339	442	311	357	1759	C=A+B?
	34	80	34	80	58	286	
	344	419	476	391	415	2045	A
	88	45	28	22	62	245	B
	432	464	505	413	477	2291	C=A+B?
	56	8	51	75	63	253	
	488	472	556	488	540	2544	
	50	51	8	63	75	247	
	538	523	564	551	615	2791	A
	33	28	45	62	22	190	B
	571	552	609	613	637	2982	C=A+B?
	91	34	80	58	80	343	
	662	586	689	671	717	3325	
	99	88	22	48	45	302	
	761	674	711	719	762	3627	
	25	56	75	11	8	175	
	786	730	786	730	770	3802	
	11	50	63	25	51	200	
	797	780	849	755	821	4002	A
	48	33	62	99	28	270	B
	845	813	911	854	850	4273	C=A+B?
	58	91	58	91	34	332	
	903	904	969	945	884	4605	
	62	99	48	33	88	330	
	965	1003	1017	978	972	4935	
	63	25	11	50	56	205	
	1028	1028	1028	1028	1028	5140	

14 Zero Knowledge Proofs - Esslinger, Alexander “Blockchain-based Systems Engineering”

Zero Knowledge Proofs: How do they work?: zk-SNARKS (2)



R1CS: Rank-1 constraint system

QAP: Quadratic Arithmetic Program

The difficult part is to generate a zkSNARK with reasonable resources for real-world applications.

Zero Knowledge Proofs: How do they work?: zk-SNARKS (1)

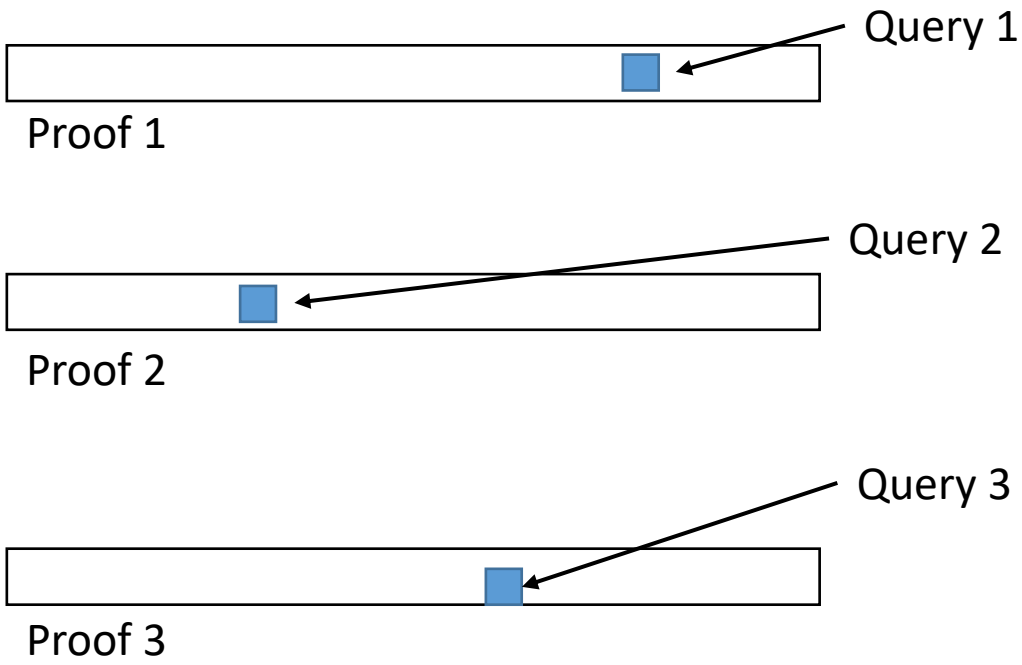
zk-SNARKS: Succinct Non-Interactive Arguments of Knowledge

zk-STARKS: Scalable Transparent Arguments of Knowledge

Zero Knowledge Proofs: How do they work?: (Non) Interactive Proofs

Interactive Proof

Prover Verifier

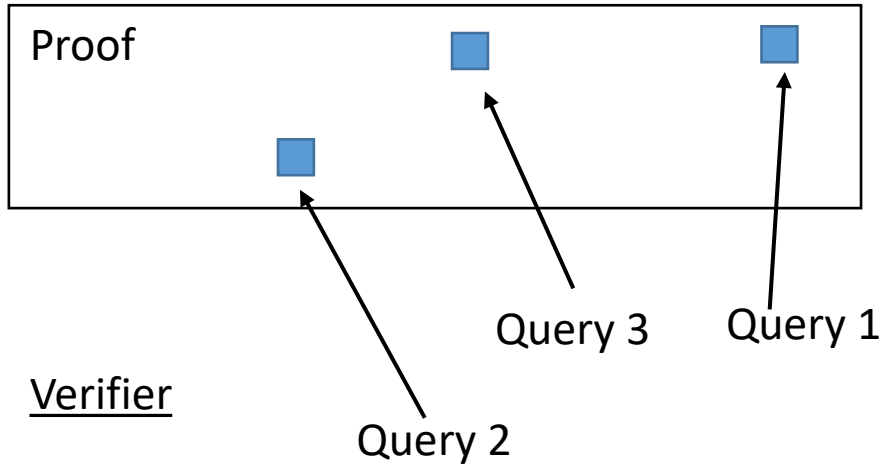


Non-Interactive Proof

Fiat-Shamir Heuristic: Interactive dialogue between prover and verifier is replaced by randomnes)

Prover:

Query locations are predefined by randomness which cannot be influenced by the prover



Zero Knowledge Proofs: How do they work?: Trusted Setup

ZK-SNARKS are not transparent and require a so-called trusted setup to secure the randomness of the parameters determining the selection of queries in the verification procedure.

The random parameters are e.g. generated using a Hash-Function. The knowledge of the original values fed into the Hash function ("*toxic waste*"), however, has to be kept secret. Otherwise, it would be possible to retrieve the random parameters and manipulate the selection of verification queries.

A trusted setup is a Multi Party Computation (MPC;"*ceremony*") together recursively generating the random parameters. If at least one party is honest and forgets its input values, the randomness of the trusted setup is safe.

Universal trusted setups can be generated once and can be re-used for other applications as well, non-universal ones are tied to the specific circuit.

Zero Knowledge Proofs: How do they work?: Blockchain scaling – zk-Rollup

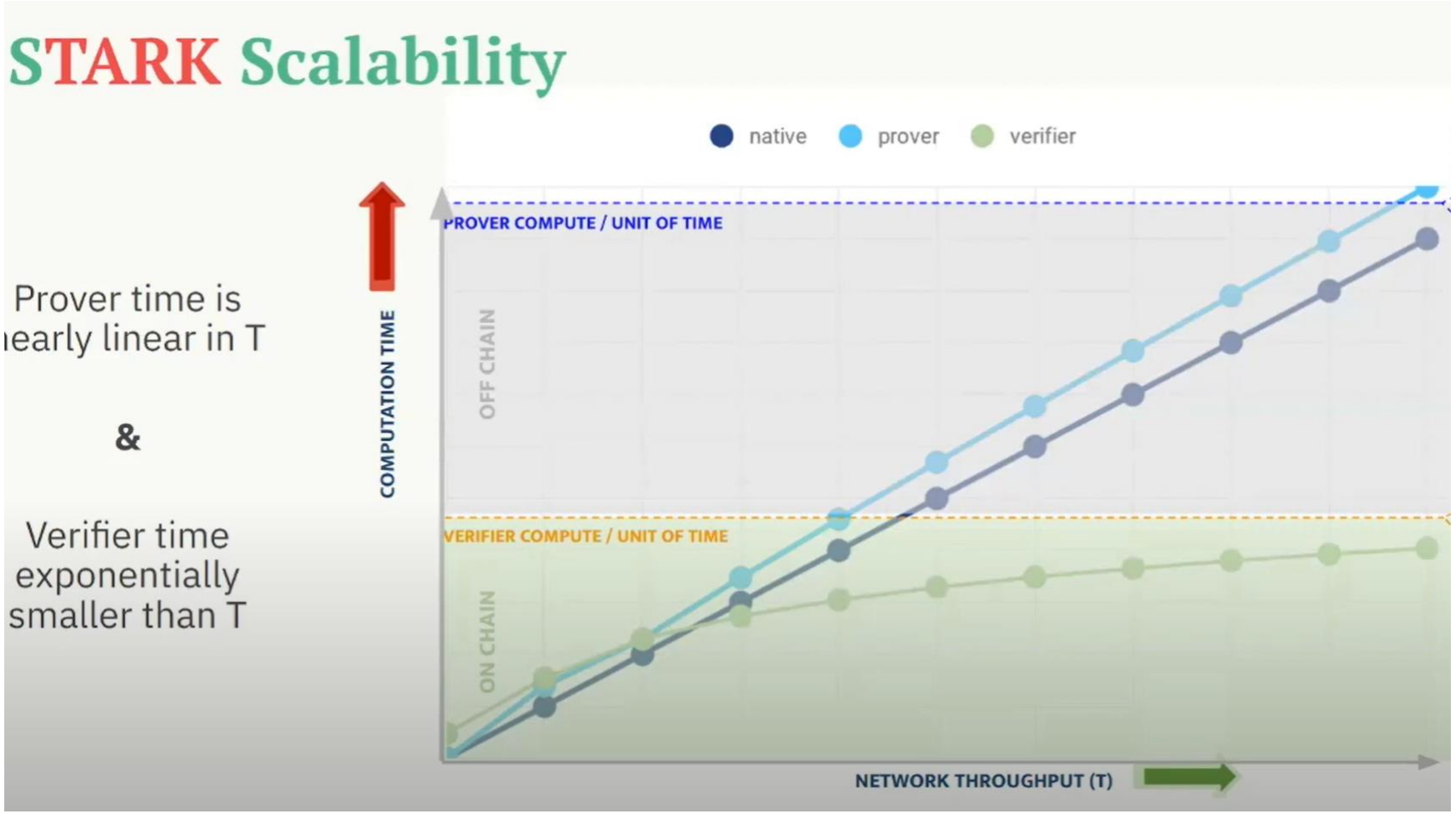
The computational cost of the independent verification of every single transaction by a large number of distributed nodes having limited resources is a main bottleneck for the number of transactions a blockchain can perform in a given time interval.

ZKPs/CIPs can solve this problem by generating a single probabilistic proof allowing the verification of thousands of transactions in one step instead of verifying the correctness of each transaction separately (zk-Rollup).

As the computational cost of verification grows only logarithmically (or less) with the number of transactions per proof, the verification cost per transaction is reduced dramatically.

Computational cost can be expended for prove generation off-chain (where it is abundant, because not part of a consensus mechanism) and traded in for less computational cost for verification on-chain.

Zero Knowledge Proofs: How do they work?: Blockchain scaling (2)



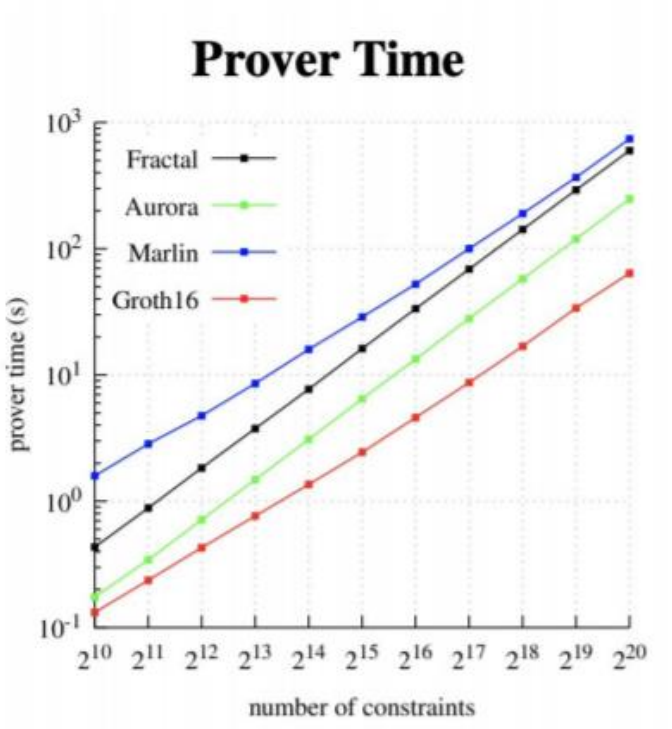
Eli Ben Sasson: "Blockchain Scalability and Privacy via Zero-Knowledge Proofs", Warwick CS Colloquium, July 6, 2020

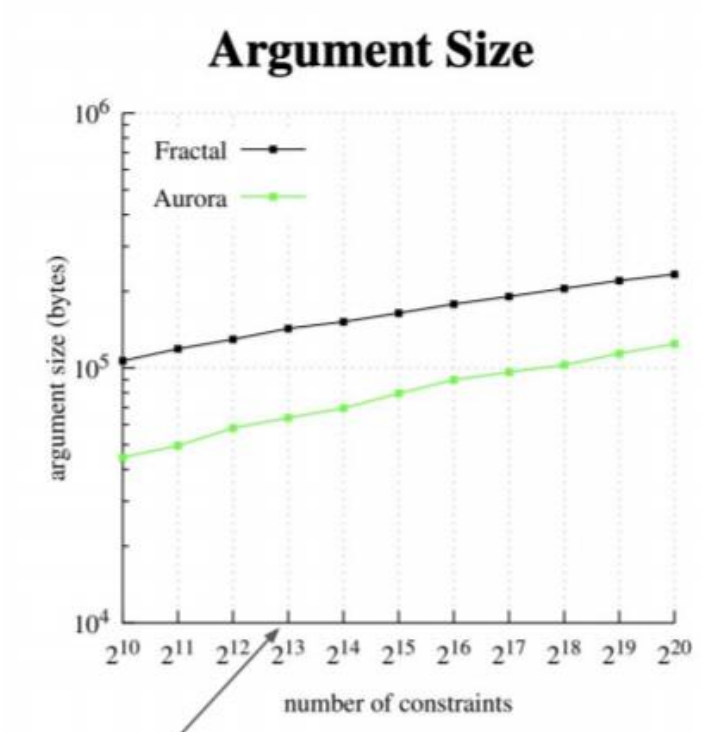
Zero Knowledge Proofs: How do they work?: Trade-Offs (2)

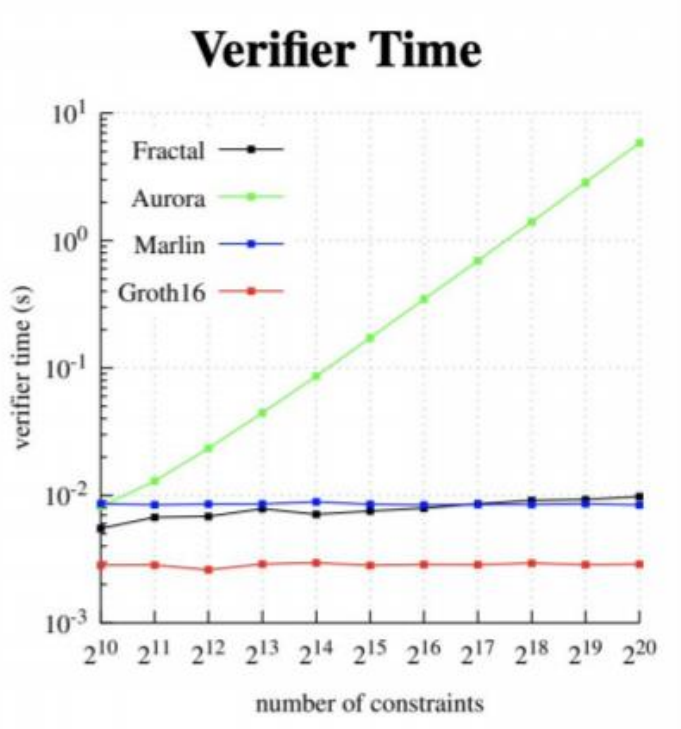
Trade-offs between different properties of different proof systems

- Non-Interactive (all zk-SNARKs, zk-STARKs)
- Transparent, universal trusted setup
- Recursive (Halo)
- Verfication time: const., … ,polylog(n); 2ms, …. , 250ms.
- Prover time: n, … , nlog(n); 1s, … , 100s.
- Proof size: log(n), … $\log^2(n)$; 200 bytes, … , 250 kbytes.
- Quantum security

Zero Knowledge Proofs: How doe they work?: Trade-Offs (3)







Gro16 + Marlin too small to be shown

https://eprint.iacr.org/2019/1076.pdf

Zero Knowledge Proofs: How do they work?: Trade-Offs (1)

Cryptographic Primitives:

- Collision-resistant Hash Function (quantum secure): *STARK, Fractal, Aurora*
- Elliptic Curve Cryptography: *Bullet Proofs, Halo*
- Knowledge of Exponent/Pairing groups: *Groth16*, *Sonic, Marlin, PLONK*
- Groups of Unknown Order: *SuperSonic*
- Lattice-based Cryptography (quantum secure): under development

Zero Knowledge Proofs: Applications: Privacy (1)

Zcash (October 2016)

Purpose:

A trust-minimized payment system like Bitcoin (PoW, UTXO), but allowing private transactions.

Use of ZKPs/PCPs:

Groth16-zk-SNARKs are used for decentralized validation of private (shielded) transactions.

Zero Knowledge Proofs: Applications: Privacy (2)

Aztec (February 2020)

Purpose:

A privacy engine for Ethereum, enabling to hide the value of transactions over the Ethereum public mainnet.

Use of ZKPs/PCPs:

PLONK-zk-SNARKs embedded in an Ethereum smart contract are used to encrypt the values of ERC-20 tokens transferred on Ethereum. Different from ZCash, Aztec does not conceal the sender and receiver of the transaction.

Zero Knowledge Proofs: Applications: Privacy (3)

Tornado Cash (April 2020)

Purpose:

An Ethereum mixing service, which provides non-custodial anonymity to ETH- and ERC-20 token transactions on Ethereum.

Use of ZKPs/PCPs:

The user deposits funds to a smart contract, receives a note as sort of a proof of ownership for the deposited amount and the contract adds them to its list of commitments. Later, when the withdrawal is requested, the user inputs the same note, the smart contract allows withdrawal to a new address. A Groth16-SNARK then ensures that the withdrawal is made from unspent commitments without revealing the source of the funds.

Zero Knowledge Proofs: Applications: Scalability-zk-Rollups (1)

ZK Sync (June 2020)

Purpose:

A layer-2(L2) scaling solution on top of the Ethereum mainnet increasing the transaction throughput of the Ethereum blockchain.

Use of ZKPs/PCPs:

A PLONK-zk-SNARK for batch-verification of several thousand transactions gradually enables a thoughput of 2.000 transactions per second (TPS). Transaction privacy may be added at a later stage.

Zero Knowledge Proofs: Applications: Scalability-zk-Rollups (2)

Loopring (March 2020)

Purpose:

Decentralized non-custodial cryptocurrency exchange and payment solution implemented as smart contracts on the Ethereum blockchain. Part of the DeFi-Ecosystem.

Use of ZKPs/PCPs:

Loopring exchange uses zkSNARKs enabling up to 2.500 trades per second. User balances are stored in Merkle trees.

Loopring Pay (since June 2020) provides a free-of-charge, instant ETH- and ERC-20 payment solution (5.000 TPS). Because all transfers happen on the Loopring zk-Rollup, users must be on the zkRollup to send or receive payments. Registering an account means making one initializing on-chain transaction that ties an Ethereum address to a “slot” (Account ID) in the off-chain account system (Merkle tree) of the zk-Rollup.

Zero Knowledge Proofs: Applications: Scalability-zk-Rollups (3)

DiversiFi (June 2020)

Purpose:

Decentralized non-custodial cryptocurrency exchange implemented as smart contract on the Ethereum blockchain. Part of the DeFi-Ecosystem.

Use of ZKPs/PCPs:

DiversiFi uses a zk-STARK enabling up to 9.000 private high-speed exchange transactions per second.

Zero Knowledge Proofs: Applications: Scalability (3)

Coda (Testnet)

Purpose:

Lightweight proof-of-stake cryptocurrency blockchain network having constant block size.

Use of ZKPs/PCPs:

Coda protocol stores on the blockchain recursive zk-SNARKs of transactions instead of transactions and signatures itself in order to reduce the block size to appr. 25kB.

Zero Knowledge Proofs: Applications: Business Applications

Baseline Protocol (under development)

Purpose:

Open Source protocol on top of the public Ethereum blockchain enabling confidential and trust-minimized collaboration between enterprise software systems (SAP etc.) of different companies, e.g. for supply chain management (cooperation of EY, Microsoft, and ConsenSys).

Use of ZKPs/PCPs:

Zk-SNARKs (Groth16) in Ethereum smart contracts are used for confidentiality as well as scalability.

Zero Knowledge Proofs: Applications: Business Applications

Findora (Testnet)

Purpose:

Public blockchain network for encrypted transfer and verification of financial assets (Co-founder Benedikt Bünz from Hamburg).

Use of ZKPs/PCPs:

Fast Supersonic zk-SNARKs are used for confidential data verification.

Zero Knowledge Proofs: Related Topics: Verifiable Delay Functions

Verifiable Delay Functions (June 2018)

A verifiable delay function (VDF) is a function f : X → Y that takes a prescribed minimum time to compute, even on a parallel computer. Once computed, however, the output can be quickly verified by anyone.

Non-parallelizable proof of work, e.g. using recursive Hash-Functions.

Application: to prevent fraud or frontrunning on exchanges, online auctions, games, or prediction markets.

Zero Knowledge Proofs: Related Topics: Multi Party Computation

Multi Party Computation (MPC)

Methods for parties to jointly compute a function over their inputs while keeping those inputs private. Unlike traditional cryptographic tasks, where cryptography assures security and integrity of communication or storage and the adversary is outside the system of participants, the cryptography in MPCs protects participants' privacy from each other.

Application: A trusted setup ceremony is a secure MPC

Zero Knowledge Proofs: Related Topics: Fully Homomorphic Encryption

Fully Homomorphic Encryption (FHE)

The “holy grail of cryptography”

Fully Homomorphic Encryption allows arbitrary mathematical operations on encrypted data. For every f: y = f(x) -> Encrypted y’ = f(x’)

In Blockchain context, the “*Mimblewimble*” protocol, on which the privacy-focused cryptocurrency networks (PoW, UTXO) *Grin* and *Beam* run, implements additively homomorphic encryption.

Zero Knowledge Proofs: Related Topics: Stateless Blockchains

Aggregatable Subvector Commitments for Stateless Cryptocurrencies (May 2020)

In a stateless cryptocurrency, neither miners nor cryptocurrency users need to store the full blockchain. Instead, the blockchain state consisting of users' account balances is authenticated using commitments. Miners only store a succinct digest of the blockchain state. Miners validate user transactions including proofs that they have sufficient balance. Users synchronize or update their proofs as new blocks get published.

The paper suggests such scalable, updatable proofs using aggregatable subvector commitments.

Authors include Vitalik Buterin, Justin Drake (Ethereum 3.0 ?)

Zero Knowledge Proofs: Resources and References

Resources to stay up-to-date in the ZK-community

ZK-Podcast and YouTube-channel

StarkWare YouTube-channel

Standardization effort for ZKPs: ZKProof Community Reference

Zero Knowledge Proofs: Resources and References

References

- Goldwasser, Micali, Rackoff: "The Knowledge Complexity of Interactive Proof Systems", 1985
- Ryan O'Donnell: „A history of the PCP Theorem", 2005
- Gina Rubino: "A (not so) gentle Introduction to the PCP Theorem – part 1", April 19, 2019
- Vitalik Buterin: "Quadratic Arithmetic Programs: from Zero to Hero", December 12, 2016
- Eli Ben Sasson: "The Sounds of Soundness", STARK @ Home 5, April 30, 2020
- Eli Ben Sasson: "Blockchain Scalability and Privacy via Zero-Knowledge Proofs", Warwick CS Colloquium, July 6, 2020
- Justin Drake: "Polynomial Commitments with Justin Drake", ZKStudyClub Part 1, December 3, 2019
- Roland Mannak: "Comparing General Purpose zk-SNARKs", November 11, 2019
- Kimi Wu: "Revealing The All Mysterious zk-STARKs", December 23, 2019
- Alex Gluchowski: "Optimistic vs. ZK Rollup: Deep Dive", November 4, 2019
- Chiesa, Ojha, Spooner: "FRACTAL: Post-Quantum and Transparent Recursive Proofs from Holography". July 15, 2020
- Boneh, Bonneau, Bünz, Fisch: "Verifiable Delay Functions", June 26, 2019
- Tomescu, Abraham, Buterin, Drake , Feist, Khovratovich: „Aggregatable Subvector Commitments for Stateless Cryptocurrencies", May 5, 2020