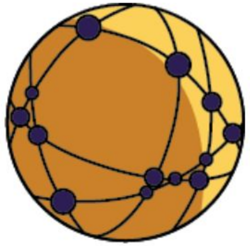
ZCASH

Overview of Zcash (ZEC)

ZCash: Fork of Bitcoin protocol



A decentralized and open-source cryptocurrency that provides strong privacy protections

Zcash is a P2P Cryptocurrency

Forked out of Bitcoin

Inherits basic properties like $21M mining limit with minor changes in incentive structure discussed later

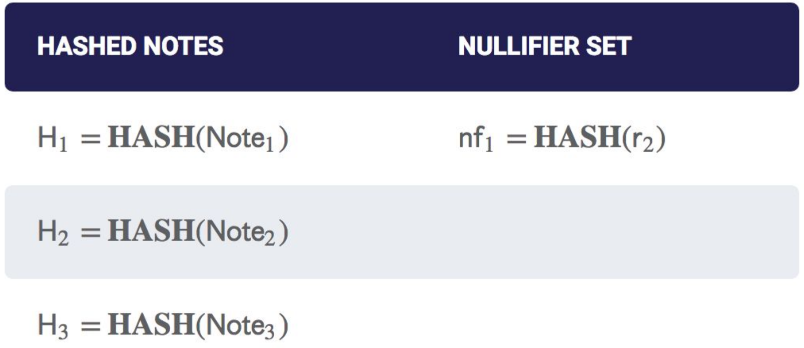
ZCash: Sole focus on privacy & anonymity

Shielded transactions hide the sender, recipient, and value on the blockchain

Its based on special cryptographic **zero-knowledge proof** (zk-SNARK) protocol that ensures privacy, anonymity, and fungibility of Zcash coins.

However they have to be opt-in.

zk-Snark: Look at Unspent Txn output (UTXO)



HASHED NOTES	NULLIFIER SET
$H_1 = \mathbf{HASH}(\text{Note}_1)$	$nf_1 = \mathbf{HASH}(r_2)$
$H_2 = \mathbf{HASH}(\text{Note}_2)$	
$H_3 = \mathbf{HASH}(\text{Note}_3)$	

Typically it contains

(PK1, balance) => Note

Add a random number to improve privacy

Note-1 = (PK1, balance, *r1*)

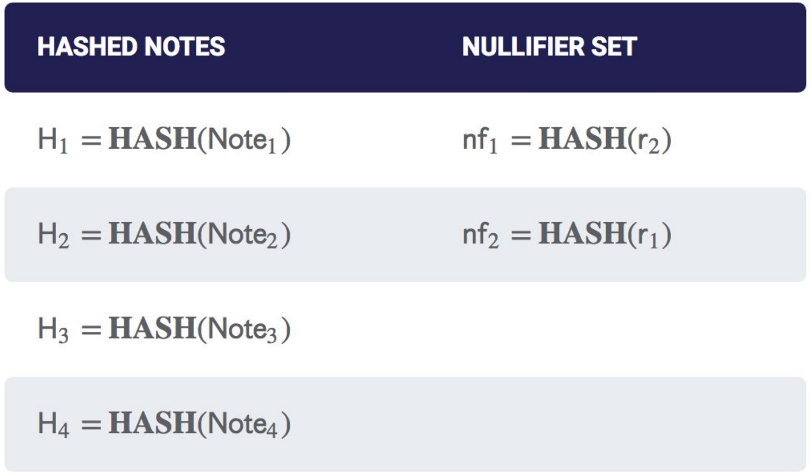
Hashed Notes:

H1 = HASH(Note-1)

Nullifier Sets:

nf1 = HASH(r1)

zk-Snark: Alice sends 1 ZEC to Bob



HASHED NOTES	NULLIFIER SET
$H_1 = \mathbf{HASH}(\text{Note}_1)$	$\text{nf}_1 = \mathbf{HASH}(r_2)$
$H_2 = \mathbf{HASH}(\text{Note}_2)$	$\text{nf}_2 = \mathbf{HASH}(r_1)$
$H_3 = \mathbf{HASH}(\text{Note}_3)$	
$H_4 = \mathbf{HASH}(\text{Note}_4)$	

* Hashed notes are stored as merkle tree

Alice randomly chooses a new serial number r4 and defines the new note

Note4 = (PK4, r4)

Assume She sends **Note4** to Bob privately.

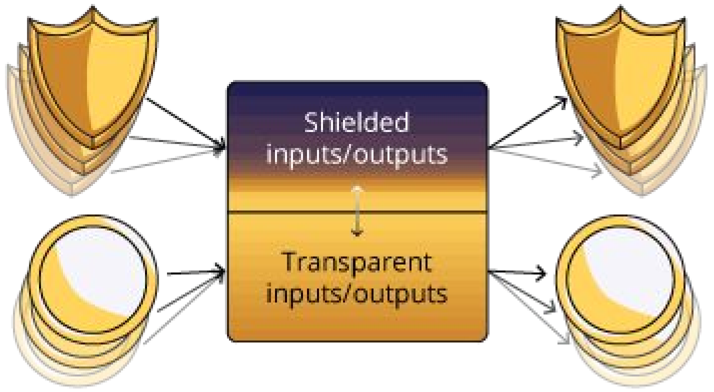
She sends the nullifier of **Note-1**

nf2 = HASH(r1) to all nodes

She sends the hash of the new note

H4 = HASH(Note4) to all nodes.

ZCash: Interoperability



With the support for both shielded and transparent addresses, users can choose to send Zcash privately or publicly (like Bitcoin).

Zcash payments sent from a shielded address to a transparent address reveal the received balance, while payments from a transparent address to a shielded address protect the receiving value.

19.9% of the transactions that happens today are shielded (>8.31% by volume)

Note: Default transaction is transparent

ZCash Team & performance

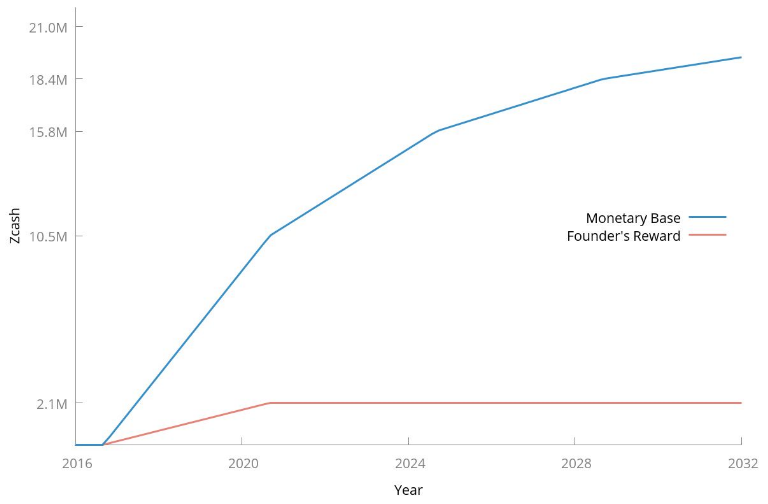
Credible team of Crypto PhD held Scientists, University profs & Developers

ZCash is completely open-source but backed by an official company

zk-SNARK protocol was developed in 2014 by scientists and research cryptographers at Johns Hopkins University, Massachusetts Institute of Technology (MIT), Technion-Israel Institute of Technology, and Tel Aviv University.

Transaction time is **~40 seconds** & Over **15,000 transactions** per day

Zcash: Mining + founders reward

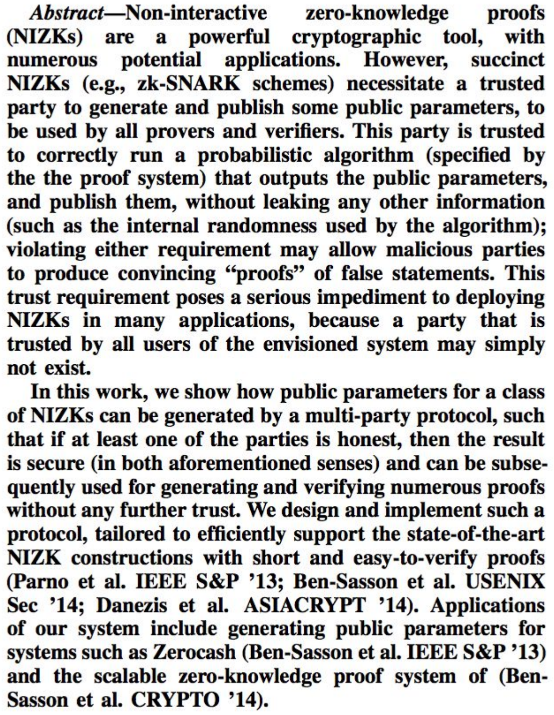


$1M funded by closed investor group - No ICO!

In return, these investors are promised a 10% reward of the total supply in an incremental way over the first 4 year period.

Divum Labs

zkSNARK: Multiparty parameter generation



Abstract—Non-interactive zero-knowledge proofs (NIZKs) are a powerful cryptographic tool, with numerous potential applications. However, succinct NIZKs (e.g., zk-SNARK schemes) necessitate a trusted party to generate and publish some public parameters, to be used by all provers and verifiers. This party is trusted to correctly run a probabilistic algorithm (specified by the the proof system) that outputs the public parameters, and publish them, without leaking any other information (such as the internal randomness used by the algorithm); violating either requirement may allow malicious parties to produce convincing "proofs" of false statements. This trust requirement poses a serious impediment to deploying NIZKs in many applications, because a party that is trusted by all users of the envisioned system may simply not exist.

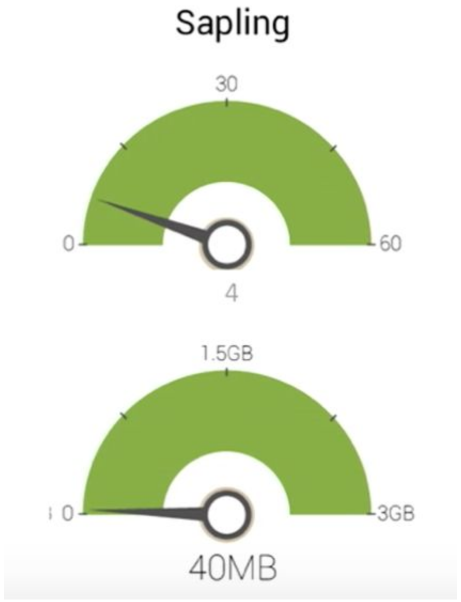
In this work, we show how public parameters for a class of NIZKs can be generated by a multi-party protocol, such that if at least one of the parties is honest, then the result is secure (in both aforementioned senses) and can be subsequently used for generating and verifying numerous proofs without any further trust. We design and implement such a protocol, tailored to efficiently support the state-of-the-art NIZK constructions with short and easy-to-verify proofs (Parno et al. IEEE S&P '13; Ben-Sasson et al. USENIX Sec '14; Danezis et al. ASIACRYPT '14). Applications of our system include generating public parameters for systems such as Zerocash (Ben-Sasson et al. IEEE S&P '13) and the scalable zero-knowledge proof system of (Ben-Sasson et al. CRYPTO '14).

Require trusted 6 parties to ***generate parameters*** for the proof system

…and they **destroy** the parameters once shared.

If they **fail** to destroy, they could produce convincing "proofs" of false statements.

Zcash: Future



Proving time improvements from 40 sec to 4 sec

Reduction in RAM usage from 3GB to few 10s of MB

Improvement MPC protocol called “**Powers of Tau**”, which allows dynamically choose parties on the network

Powers of Tau can be applied on other apps

Zcash: Distant Future

Improved Privacy on Smart Contracts

Hides the code being circulated & executed

Thank You
in/hivaidee