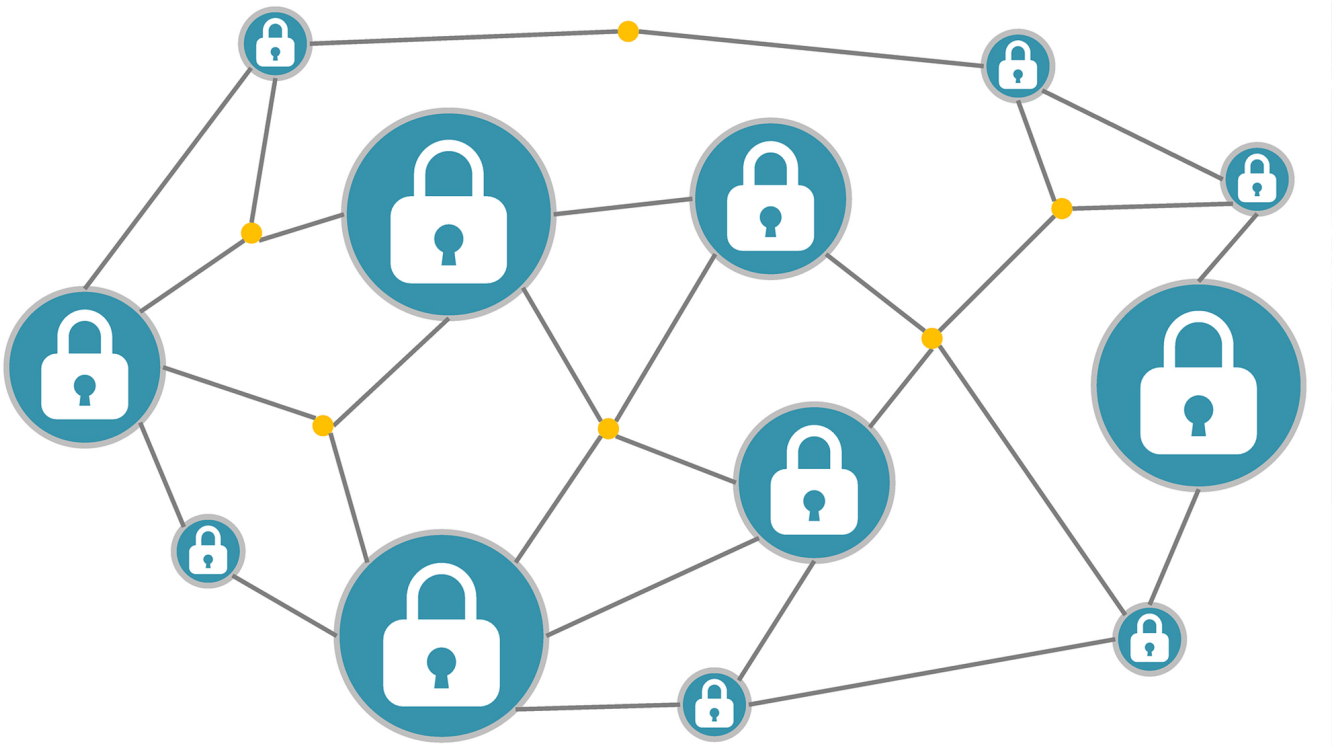
zkDAI

Daniel von Ahn
University of Applied Science
Flensburg, Germany
daniel.von-ahn@stud.hs-flensburg.de

Thomas Dethlefsen
University of Applied Science
Flensburg, Germany
thomas.dethlefsen@stud.hs-flensburg.de



Abstract

This paper discusses the use of zero-knowledge proof systems in blockchain-based applications. In particular, it discusses the creation of an anonymized cryptocurrency that uses zero knowledge techniques to hide all transaction details.

Keywords: crypto currency, zero knowledge, blockchain

ACM Reference Format:
Daniel von Ahn and Thomas Dethlefsen. 2021. zkDAI. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Motivation

For many blockchain applications, particularly cryptocurrencies, anonymity is a critical property expected from the underlying platform. It does not only obscure users' identity, but it also ensures fairness, without which users may opt-out of participation due to plausible fear of unfair treatment. Inadequate anonymity guarantees can also result in malicious parties focusing their efforts on de-anonymized high-value targets, leaking business information, and undermining the negotiation position. Besides, inefficient anonymity protection can lead to targeted denial of service which can decrease the fungibility of the affected cryptocurrency and further cripple its efficacy as a currency. [2]

Moreover, different privacy enhancement technologies may offer the same (or similar) level of user anonymity, but their characteristics, like footprint, and computational effort and time, have direct repercussions on the adopt-ability of the cryptocurrency, its scalability, and even its transaction fees. In the past decade, Bitcoin, altcoins, and many other decentralized blockchain-based applications have been hot research topics. As such, there are numerous surveys on blockchain and cryptocurrencies. [1]

2 Solution approaches

The literature review showed in the previous section shows that there is a lot of research related to zero-knowledge proofs. Many solutions or companies have been created from these investigations. In this section a list of some of those ones that use this cryptographic technique somehow will be provided. There are multiple solutions that use zero-knowledge proofs in the blockchain ecosystem but the ones listed below are those that are on Ethereum or use its standards.

2.1 Aztec

Functionality: Protocol
Website:https://www.aztecprotocol.com

The Anonymous Zero-knowledge Transactions with Efficient Communication protocol commonly known as AZTEC, is implemented on the Ethereum public blockchain and describes a set of zero-knowledge proofs defining a confidential transaction protocol that can be used on blockchains that supports Turing-complete computation, such as Ethereum.[3]

2.2 Zokrates

Functionality: Toolbox
Website:https://zokrates.github.io

Zokrates is a toolbox for using zk-SNARKs on the Ethereum network. ZoKrates allows developers to verify proofs in Solidity from the specification of an off-chain program coded in a high level language, giving the possibility of using verifiable computation in the decentralized application (DApp) by linking them to the Ethereum blockchain. ZoKrates allows developers to create and verify zero-knowledge proofs using Solidity contracts. In the Byzantium hard fork there were introduced some enhancing cryptography changes on Ethereum, such as curve addition, scalar multiplication and pairing checks on the elliptic curve alt bn128 in order to perform zk-SNARK varication.

Once ZoKrates is installed you will be able to use all the CLI commands ZoKrates provide, which are the following (or you can see them by running: $ zokrates –help):

- **compile**: Compiles a file written in ZoKrates high level language (.zok) into an arithmetic circuit representation. You need to set the path to the file with the ag -i.
- **compute-witness**: Generates a witness for the compiled constraint system produced by the previous command from public and private arguments provided by the Prover using the ag -a arg1 arg2.
- **export-verifier**: Using a verifying key (verification.key) generates a Solidity smart contract (Verifier.sol) which contains a public function (verifyTx) that accepts a proof and public inputs. The verifying key is hard-coded in the function verifyingKey().
- **generate-proof**: Generates a proof in JSON format for the compiled constraint system and witness using a proving key (proving.key).
- **print-proof**: Prints the proof in JSON or Remix format.
- **setup**: Performs a trusted setup for the compiled constraint system and creates the proving (proving.key) and verifying (verification.key) keys derived from the toxic waste.

As of today (Feb 2021) ZoKrates is still under development and there are some considerations to take into account when you use it, besides that some details may be subject to change in the future.

2.3 EY Nightfall

Functionality: Tools Suite
Website:https://github.com/EYBlockchain/nightfall

Nightfall is an open source suite of tools that combine ZoKrates and a set of smart contracts, specifically the ERC-20 and ERC-721 Ethereum token standards, designed for a fully privacy token transactions over the Ethereum public blockchain. Other standards could be added to the specification in the future if their use is extended.

2.4 Loopring

Functionality: DEX Protocol
Website:https://loopring.org

An open source protocol for decentralized exchange over the Ethereum network based on the interoperability among decentralized applications with exchange funcionalities allowing their users to trade their assets. In the third version of the protocol, the use of zk-SNARKs was included to improve the throughput.[3]

2.5 Tornado

Functionality: Mixer
Website:https://tornado.cash

Tornado is an Ethereum privacy solution based on zk-SNARKs to ensure privacy transaction, acting like a mixer or a proxy because it uses a smart contract, which accepts Ether or ERC-20 tokens allowing the withdraw to a different address, removing the reference between the withdrawal and the deposit of the original transaction as a privacy preserving mechanism.[3]

3 Technical Background

3.1 What is a Blockchain?

A blockchain is a continuously expandable list of data records, called blocks, which are chained together using cryptographic algorithms. Each block typically contains a cryptographically secure hash of the previous block, a timestamp and transaction data. The term blockchain is also used when an accounting system is maintained in a decentralized manner and the correct state in each case must be documented because many participants are involved in the accounting. This concept is called distributed ledger technology (DLT). What will be documented is irrelevant to the concept of blockchain. What matters is that later transactions are based on earlier transactions and confirm them as correct by proving knowledge of the earlier transactions. This makes it impossible to tamper with the existence or content of the earlier transactions without simultaneously destroying all later transactions as well. Other participants in the decentralized ledger who still have knowledge of the later transactions would recognize a manipulated copy of the blockchain by the fact that it has inconsistencies in the calculations. The process of cryptographic chaining in a decentralized ledger system is the technical basis for cryptocurrencies, but it can also help improve or simplify transaction security in distributed systems compared to centralized systems. One of the first applications of blockchain is the cryptocurrency Bitcoin.

3.2 Smart Contracts

Smart contracts are computer protocols that can map or check contracts or provide technical support for negotiating or processing a contract. An infrastructure for smart contracts can be implemented through a replicated asset register and contract execution via cryptographic hash chains and fault-tolerant replication.

3.3 Blockchain Transactions

Each node in a peer-to-peer network acts as a register and trustee who carries out changes of ownership and automatically maps verifiable rules about these transactions. All transactions are always audited by all other nodes. Askemos implemented this approach in 2002 with Scheme and in later versions SQL as the contract description language. Askemos only describes a concept for the mutual audit of a number of independent registers of any assets (values) without coupling these registers to a money-like resource Cryptocurrencies like Bitcoin have implemented special cases of such registers; there the asset is money. Bitcoin and many of its offshoots contain mechanisms that enable the management of more general assets and contracts. If a participant now wants to transfer an amount to an account, he creates a transfer order with the amount and the public key of the target account and signs this order with his secret key. This order is published via the P2P network. It must now be checked and certified and archived as a transaction in the joint accounting.

The steps in the operation of a decentralized cryptocurrency are:

1. New transactions are signed and sent to all nodes.
2. Each node collects new transactions in a block.
3. Each node looks for the nonce that validates its block.
4. When a node finds a valid block, it sends the block to all other nodes.
5. The nodes only accept the block if it is valid according to the rules:
   - The hash value of the block must correspond to the current level of difficulty.
   - All transactions must be correctly signed.
   - The transactions must be covered in accordance with the previous blocks (no double spending).
   - New issue and transaction fees must conform to the accepted rules.
6. The nodes express their acceptance of the block by adopting its hash value in their new blocks.

3.4 Nonces

In cryptography, the term nonce was taken up to designate a combination of numbers or letters that is only used once in the respective context. Typical ways of generating a nonce are the use of (cryptographically secure) random values that are sufficiently large that the probability of double use is negligible (see birthday paradox).

3.5 Smart bond

A smart bond is a special type of automated bond contract that leverages the capabilities of blockchain databases that

can function as cryptographically secure, yet open and transparent ledgers. It belongs to a class of financial instruments known as a smart contract, "a computerized transaction log that executes the terms of a contract".

3.6 Stablecoins

Stablecoins are cryptocurrencies, the price of which is controlled through active or automatic monetary policy with the aim of low volatility in relation to a national currency, a currency basket or other assets.

3.7 Wallet

A crypto money wallet (translated: wallet) enables interaction with the data storage of the crypto money. Sometimes a wallet is described as a wallet that holds cryptocurrency. This can easily be misunderstood: The crypto money is not stored in the wallet, but in the crypto money data storage, i.e. mostly in a distributed blockchain. The wallet enables access to the crypto money stored in the blockchain. If you lose the wallet, the cryptocurrency is not lost. The crypto money is only lost if you lose your private key.

3.8 Oracle

Smart contracts can execute program code and calculations and store the results in the blockchain. But smart contracts cannot communicate with the outside world. For some smart contracts, however, queries have to be carried out in order to be able to check certain "real-world" conditions, for example stock exchange prices, flight delays or weather data. So-called oracles are programmed as intermediaries ("data feed") (e.g. in Solidity) and added to the blockchain. A smart contract can query the Oracle, and the Oracle can call up external interfaces e.g. via REST / JSON and transmit the result to the smart contract via callback.

3.9 DApp and DAO

Dapps (decentralized applications) add a user interface to smart contracts, for example a website. The smart contract can be called up and used via this. DAOs ("Decentralized Autonomous Organizations") can be set up via Ethereum. These are decentralized investment companies in which not a central management, but instead smart contracts regulate all processes. Depending on the implemented smart contracts, all participants / token owners can be on an equal footing and make decisions together.

3.10 DAI

The DAI Coin is a crypto currency that represents the equivalent of exactly 1 US dollar. Other crypto currencies such as Bitcoin, Ether Co. have their own value that increases or decreases in value compared to a normal currency such as US dollars or euros. Technologically speaking, DAI runs on the Ethereum blockchain. This coin therefore also represents a decentralized cryptocurrency. There are several useful reasons to use DAI instead of the US dollar:

1. The almost 2 billion people in the world without access to the banking system can participate in business life with the help of this cryptocurrency. In addition, this currency serves as an inflation protection for citizens of a country with high inflation. To do this, they simply invest the local currency in DAI.
2. You can use DAI for smart contracts and thus carry out automated transactions.
3. You can send the DAI Coin very quickly and cheaply around the world. In contrast to conventional international transfers, you save a lot of time and money.
4. Since the cryptocurrency runs decentrally on the blockchain, you don't need to trust a bank or other central authority. The coin is distributed decentrally in the network.

3.11 zCash

Zcash is a cryptocurrency that aims to use cryptography to improve privacy for its users compared to other cryptocurrencies like Bitcoin. Like Bitcoin, Zcash has a total volume of 21 million coins. Transactions can be "transparent" and similar to Bitcoin transactions, in which case they are controlled by a t-addr, or they can be some kind of zero-knowledge evidence called zk-SNARKs; the transactions are then considered "shielded" and are controlled by a z-addr. Zcash coins are either in a transparent pool or in a screened pool; As of December 2017, only about 4% of Zcash coins were in the shielded pool, and at that point most crypto wallets, as well as web-based wallets, did not support z-addrs. The shielded pool of zCash coins was further analyzed for security, and it was found that heuristics-based identifiable usage patterns can significantly shrink the anonymity rate.
Zcash offers private transaction partners the option of "selective disclosure", which enables a user to prove the payment for verification purposes. One such reason is to give private funders the choice of complying with anti-money laundering or tax regulations. "Transactions are auditable, but disclosure is under the control of the participant.
The basic principle is easy to explain: if Alice has secret information, such as the combination for opening a safe, and Bob is supposed to verify that she has this information without receiving the information himself, she will open the safe without Bob to see the number combination and then close it again. In this scenario Alice is the prover and Bob is the verifier. In the case of Zcash, the process is implemented in the form of so-called zk-SNARKs. The acronym stands for Zero-Knowledge Succinct Non-Interactive Argument of Knowledge, where Zero-Knowledge is the proof procedure, Succinct means "scarce" and refers to the quick implementation and Non-Interactive means that the verifier does not

have to do anything himself only needs to receive the argument of knowledge from the prover once to carry out the procedure.

With a cryptocurrency like Zcash, the evidence to be enforced is that the prover actually made the transaction in question (instead of just pretending to do so and actually holding the money). The key information such as the name of the account, amount and destination of the transfer should remain secret. A zero-knowledge procedure now makes it possible for the verifier to read from the message from the prover that everything is in order without being able to see the details of the transaction.

In terms of performance, the picture is mixed: Zerocoin was considered notoriously slow. Zcash has taken on this problem and enables swift transactions, but requires huge gigabyte-sized databases for comparison in order to use zk-SNARKs. The additional anonymity also seems to have an adverse effect on security. At least the forerunner Zerocoin suffered from attacks, for example in 2017 when 370,000 fake XZCs were generated by exploiting a bug in the software and exchanged for the impressive sum of 400 Bitcoins. Incidentally, the concept will meet with little love from control authorities, because the degree of anonymity achieved by Zerocoin and Zcash makes these procedures candidates for money laundering and illegal money transfers.

Zcash was created through a spin-off of Bitcoin. This fact alone results in various similarities between the two cryptocurrencies. There are two different types of transactions for Zcash: private and public (similar to BTC). When downloading Zcash Wallet, you will notice that it has two different addresses available: t-addr, your public address, and z-addr, your private address.

3.12 Zero Knowledge Proofs

Generally speaking, a zero-knowledge proof or protocol (zkp) is a cryptographic method defined as an interaction between two parties where one - the Prover - works to convince to another party - the Verifier - that some statement is true or some information is known without revealing the statement or the information to the Verifier, that it will not learn anything. The implementation of zero knowledge proofs in cryptocurrencies works, in simplified terms, as follows. The sender of the transaction (the prover) encrypts his transaction note, consisting of his public key and the transmitted value, using a SHA256 hash algorithm. Before he puts the note on the blockchain, he generates a proof that he knows which parameters are behind the encrypted note. This way, he can prove to the blockchain network that his transaction is authentic without revealing any details, and the transaction will be performed.

4 Solution

With zkDAI you can shield the sender, recipient and the amount of the transaction.

4.1 Implementation

Zero-knowledge protocols are used, among other things of authentication. With some crypto currencies such as Zcash or mobile payment services such as Bluecode, they increase the anonymity of payment transactions. For the zero knowledge proof generation in zkDAI, we used the ZoKrates, which we introduced earlier. A zero proof of knowledge proves possession of a note, not the sender of the transaction. The recipient is always hidden because this information is encoded in the hash note.

zkDAI has the concept of a secret note. A note is identified by a tuple, which is a combination of two elements - the public key of the note owner (public key) and the value of the note in DAI (v).

The zkDAI notes are output like UTXOs (Unspent Transaction Output = cache of unsaved transactions). In order to transfer a certain value to a recipient, one selects some secret notes, the net value of which is at least equal to the value with which one wants to carry out the transactions. This value is sent to the recipient in the form of a new zkDAI note, which the recipient can then redeem. Note that this transaction hides the transaction details. The sender is hidden in the sense that one could use a new eth-address each time to perform a transaction that the zero knowledge proof sends to the chain. One only has to prove knowledge of "Private Key", which is the secret key corresponding to the public key that the note belongs to. This is done by a smart contract, called "verifier.sol", which is also exported by ZoKrates when compiling the zero knowledge circuit. To create a possibility to interact with this contract, we used another smart contract ("SecretNote.sol"), which was designed by a singaporean blockchain developer, who already implemented zkDAI in 2018. Unfortunately, most of his code is deprecated and not usable anymore, so we had to build everything from scratch (except the SecretNote contract). The SecretNote contract offers some public methods to interact with the verifying contract from ZoKrates, the specified test network and a DAI faucet. Using these interfaces you can run zero knowledge transactions on a test network like Ropsten or Rinkeby. Visit our GitHub repository to try it yourself.

5 Conclusion

Benefits of confidential transactions:

- The sender is not published.
- The recipient is not published.
- The value of the transaction is not published.

Confidential transactions are therefore particularly attractive for companies, as they strive to keep information about their supply chain secret from competitors. But also private users

who do not want to make their payment information public benefit from these systems.

But what about any disadvantages? Anonymize cryptocurrency uses a new form of cryptography. Since this has not been used for too long, no one can guarantee that it will always work properly. Until this approach has been tested for vulnerabilities for a number of years, it cannot be ruled out that anonymous transactions could be discovered or that errors could occur in transaction totals. If someone creates additional anonymize cryptocurrency coins through a loophole in the code, this cannot be recognized immediately, as the current total amount of coins in circulation cannot be cross-checked against the confidential transaction sums. The blockchain and the network keep track of the amount of protected coins, but ignore when a user reverses part of it. The coins go into a large pot and if part of it is later withdrawn, it is impossible to understand how much has gone into the pot in total.

References

[1] D. Boneh A. Poelstra P. Wuille B. B unz, J. Bootle and G. Maxwell. 2018. *Bulletproofs: Short proofs for confidential transactions and more.*

[2] Greg Maxwell. 2021. *Confidential transactions.* https://people.xiph.org/greg/confidentialvalues.txt

[3] Alberto Ballesteros Rodriguez. 2019. *Master's thesis report for the studies of the Inter-university Master's Degree in Security of Information and Communication Technologies.*