ENCOINS: a Private Transactions System on Cardano

Version 0.1

team@encoins.io

1 Introduction

Public blockchains are designed to store the information in them forever. Their transparency has many advantages over private and/or centralized databases and ledgers. However, such systems cannot be self-sufficient without the tools for private transactions and storage of value. Indeed, users will inevitably seek such tools elsewhere whenever they need them. Some even argue that privacy-by-default (as in ZCash [3] or Monero [2]) should be the standard. At the same time, smart contract functionality is crucial for real-world financial applications. As we do not have both smart contracts and privacy-by-default on any currently operating blockchain, building composable privacy protocols on public smart contract blockchains presents a reasonable compromise.

Cardano is a general-purpose smart-contract platform that strives to become the financial operating system of the world. A key feature of Cardano is the concept of a native asset: such assets can be minted and burned using minting policy scripts provided by the users. After minting, native assets can be transacted in the same way as ADA, Cardano's primary asset.

Cardano has also pioneered the eUTXO model of transaction data storage, where the system's state is contained in unspent transaction outputs (UTXOs) that can carry value and arbitrary data (albeit in very limited quantities). These UTXOs can be locked either by the user's private keys or by scripts with complex validation logic (validator scripts). The purpose of attaching data to UTXOs is usually to perform some computations with it *in the future*, while native tokens can represent proof that something has happened *in the past*. As such, native tokens present a great vessel for holding a private, only known to the owner value: the owner can prove that it took them exactly this value to mint the token. We named such tokens *encrypted coins*.

ENCOINS, the private transaction system we propose, is a modular, highly customizable, and easily upgradable set of protocols that utilize encrypted coins and could be deployed on the Cardano blockchain. It utilizes zero-knowledge proofs to protect the user's privacy.

Currently, many zero-knowledge proving systems are available (e.g., [5, 8, 7, 4, 10, 12]). These systems found many applications in private transactions systems and scalability solutions (e.g., [2, 3, 9, 11, 1, 6]). ENCOINS is based on Bulletproofs [4], a proving system that allows the construction of aggregated proofs with inputs from multiple parties. The key operation in the Bulletproof verification algorithm is group exponentiation. This operation implemented for certain elliptic curve groups is expected to soon become a Plutus built-in function. This, in turn, should make Bulletproof verification viable on-chain.

The rest of the paper is structured as follows. Sections 2 and 3 introduce ENCOINS Core and ENCOINS Ledger, respectively. These are the two essential components of ENCOINS. The core functionality will be accessed by a user through the Minting Portal, while operations on the ledger become convenient with ENCOINS Wallet. In Section 4, we overview the governance of ENCOINS and tokenomics of ENCS, the project's token. Finally, possible directions for future ENCOINS development are discussed in Section 5.

2 ENCOINS Core

ENCOINS Core consists of the Encrypted Coins Protocol, i.e., the rules of minting and burning of encrypted coins and the software components for interaction with it.

2.1 Encrypted Coins Protocol

Every native asset on Cardano is characterized by a minting policy and a token name. The minting policy determines the rules of creating (minting) and destroying (burning) the native asset, i.e. its tokenomics. Developers can choose token names either for aesthetic reasons or for the purpose of encoding some useful information in them.

In the Encrypted Coins Protocol, each token (encrypted coin) has the same minting policy but a unique token name. In other words, each encrypted coin is an NFT. Token names contain information about the redeeming ADA value of encrypted coins. As the name suggests, this information is encrypted: it is only known to the "owner" of a particular coin. We will refer to the private information used to generate an encrypted coin as its *minting key*. Each minting key consists of the redeeming ADA value and a big randomly generated number (nonce).

On a high level, a transaction that mints and burns some encrypted coins is accepted if and only if the following conditions are satisfied:

1. Proof of knowledge of minting keys for all minted and burned coins is provided;
2. The difference (in ADA) between the redeeming values of minted and burned coins is locked up in the validator script if positive or withdrawn to the user-supplied address if negative.

Let us further elaborate on these rules. Each proof of knowledge is a cryptographic zero-knowledge proof: it does not reveal any information about the minting keys or the redeeming values of the coins. The withdrawal address for ADA is encoded in the proof. This is important when the relayer submits a transaction on behalf of the user: the relayer cannot simply alter the address for redeemed ADA, as this would require new proof from the owner.

ENCOINS Minting Portal. Users will be able to mint, send, re-mint, and burn encrypted coins using the minting portal. Integrations with popular Cardano wallets are also possible and would be appreciated.

2.2 Use-Cases

The Encrypted Coins Protocol is envisioned as the foundation for different tools that satisfy various privacy demands. However, even on its own, it can be used in several different scenarios.

Safe storage. The ability to self-custody your assets of value is one of the original appeals of cryptocurrency. Notably, it puts the responsibility for asset safety on the user. So-called cold wallets provide a reasonable defense against a wide range of hacking attacks. However, their protection against physical attacks is limited. This is a consideration for people owning significant sums of cryptocurrency as it is very difficult to keep the identity behind the blockchain address completely private. Multi-signature wallets are considered the most secure form of self-custodial cryptocurrency storage. They require several signing keys to move the assets. Those keys are usually stored at different geographical locations, so obtaining a sufficient number of them is extremely difficult for a potential attacker. Encrypted coins could be used for safe ADA storage, too, as they turn the value of one's ADA possessions private, thus making the owner a significantly less appealing target of an attack.

Transactions with private amounts. Let us consider an example of how the Encrypted Coins Protocol enables users to make ADA transfers with private amounts. Suppose Alice wants to send 147 ADA to Bob without revealing the amount publicly and store the rest of her 1000 ADA privately. First, she mints two encrypted coins with a total redeeming value of 1000 ADA (Figure 1).

Alice then sends the 147 ADA coin to Bob and communicates to him the minting key for this coin (Figure 2). The key communication can be done either off-chain, e.g., with a secure messenger, or on-chain.

Assume Alice now wants to make two more payments of 350 and 490 ADA, respectively. She can burn her 853 ADA coin and mint two new coins withdrawing the change as ADA (Figure 3).

How can Bob keep the value Alice sent to him private if he wants to convert it to ADA? He could, for example, burn his coin and mint a 1 ADA coin withdrawing the remaining 146 ADA from the validator script. In case Bob collects payments from different sources, it would make sense to burn and re-mint those coins *simultaneously* to protect everyone's privacy.

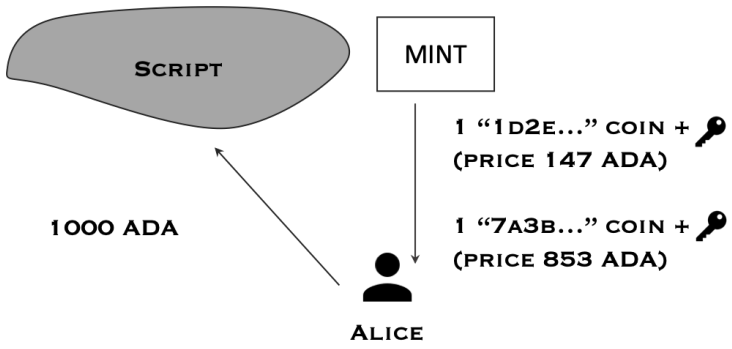

Figure 1: Alice mints two encrypted coins.

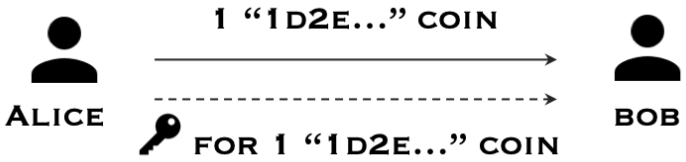

Figure 2: Alice sends one of her coins to Bob.

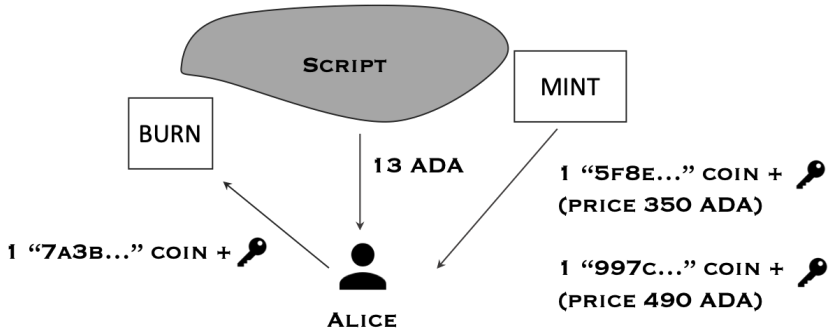

Figure 3: Alice re-mints her old coin into two new coins plus 13 ADA change.

3 ENCOINS Ledger

ENCOINS Core gives privacy with respect to transaction amounts. However, all transfers of ENCOINS are visible on the blockchain. Who (which address) owned which encrypted coin is public information.

The information about intermediate transacting parties could be made completely private if there is a way to construct a private transaction sub-ledger on the Cardano blockchain. With ENCOINS Ledger, it is indeed possible!

ENCOINS Ledger can be implemented as a Plutus validator script that locks the encrypted coins of all participants. The condition for spending the coins locked by this script is very simple: encrypted coins can be moved or re-minted while on the ledger, but the value can be withdrawn from ENCOINS Ledger only by burning encrypted coins.

ENCOINS Relayer. Privacy of transactions on ENCOINS Ledger is facilitated by *relayers*, a decentralized network of independent servers that sign transactions on behalf of users and collect the *relayer fee*. Zero-knowledge proof sent by a user allows a relayer to mint and burn only the coins requested by the user. Additionally, the proof depends on the provided withdrawal address (if any): a relayer will not be able to alter it when submitting the transaction as it would require new proof. ENCOINS Relayer currently requires 2 GHz processor with at least two cores, 200 GB of disk space,

and 16 GB of RAM.

ENCOINS Wallet. To make it easier for users to keep track of their encrypted coins locked in ENCOINS Ledger, we propose to build a special wallet app that will keep track of all coins known to the user. With this app, users will be able to add, send, re-mint, and withdraw ADA from ENCOINS Ledger.

Example. Let us consider a typical transaction on ENCOINS Ledger. Suppose Alice has two coins in the ledger, i.e., she knows the minting keys for those coins. She wants to send one of them to Bob. To do so, she simply sends Bob the minting key for the coin over a secured communication channel (Figure 4).

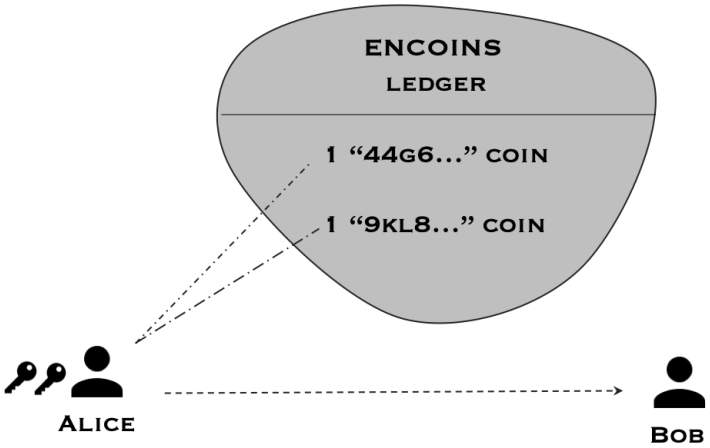

Figure 4: Alice makes a private transaction on ENCOINS Ledger by communicating a minting key to Bob.

Now the coin is in their shared possession. To finish the transaction, Bob sends the proof of knowledge of the coin to a relayer and asks them to re-mint it (Figure 5).

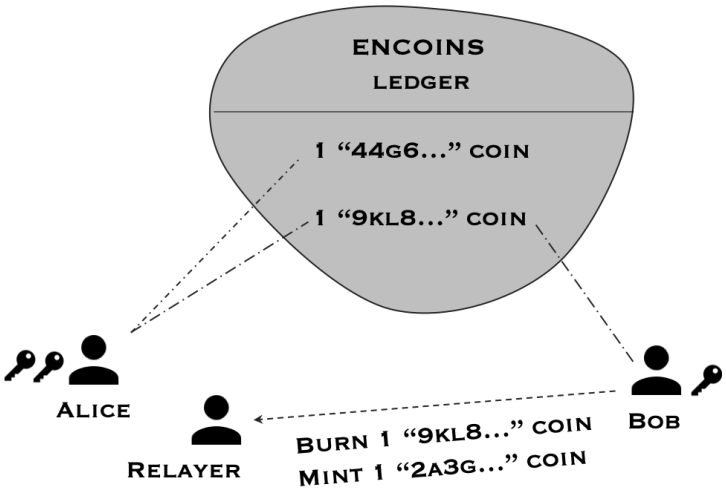

Figure 5: Both Alice and Bob share a coin. Bob re-mints the coin to finalize.

After minting the new coin, the value sent by Alice is in sole Bob's possession. From an external observer's perspective, the above example looks like somebody just re-minted a coin (Figure 6).

Only when a coin is withdrawn from ENCOINS Ledger, its ownership (the withdrawing user's address) is revealed. However, no information about the intermediate owners is leaked. In other words, an ENCOINS Ledger user only needs to "claim ownership" of the coin that they want to redeem and only just before redeeming it.

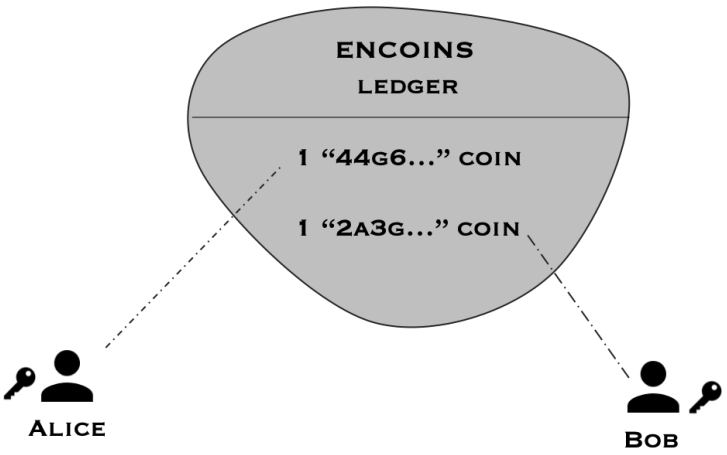

Figure 6: Bob is now the sole owner of the transferred coin.

4 Protocol Governance and Tokenomics

In this section, we describe the ENCOINS governance. We also introduce the ENCS token, which is a governance/utility token of the ecosystem.

4.1 ENCOINS DAO

ENCOINS ecosystem is being built in a decentralized way from the start. The project treasury will be replenished with the profits captured by the Encrypted Coins Protocol. The treasury funds may be directed towards different initiatives that help boost the project's scope, scale, and adoption. These initiatives may include (but are not limited to)

- Supporting the development and maintenance of the ecosystem tools;
- Subsidizing anonymity mining on ENCOINS Ledger;
- Incentivizing ENCS token holders;
- Executing promotions campaigns.

ENCOINS DAO will make decisions on treasury fund distribution. This DAO will use ENCS as a governance token. The work on ENCOINS DAO Portal will commence once all ENCS tokens are distributed.

4.2 ENCS token

This subsection discusses the utility, allocation, and launch of the ENCS token.

4.2.1 Utility

ENCS token is a hybrid governance/utility token. The governance aspect of it was discussed above. Here we present the utility use-cases of ENCS in the ENCOINS ecosystem.

The key idea is that a significant ENCS holding proves "an interest" in the project. This can be used to create lists or rank reliable participants in our protocols. In particular, ENCOINS Ledger will utilize relayers for re-minting transactions like the one showcased in the previous section. The list of relayers will be formed based on their ENCS stake. For transactions on ENCOINS Ledger, a user will connect to one of the relayers from the list. For this list, the initial stake amount we propose is 100 000 ENCS. ENCOINS DAO can later adjust this parameter.

It should be noted that, as open-source software, the Relayer App can be run by anyone (with some technical knowledge). Connecting to a custom relayer (even not from the list) will be technically possible.. However, this is not generally advised, as choosing a relayer at random provides better privacy guarantees.

4.2.2 Initial token distribution

The ENCS token has a fixed supply of 15 000 000 tokens, minted after ENCOINS Core launch. The initial distribution of the token is summarized in the following pie chart below (Figure 7).

Below we give the description of each category.

- 10 000 000 ENCS are allocated for distribution to the ISPO participants.
- 3 000 000 ENCS will make the initial protocol's treasury.
- 2 000 000 ENCS are reserved for the original development team.

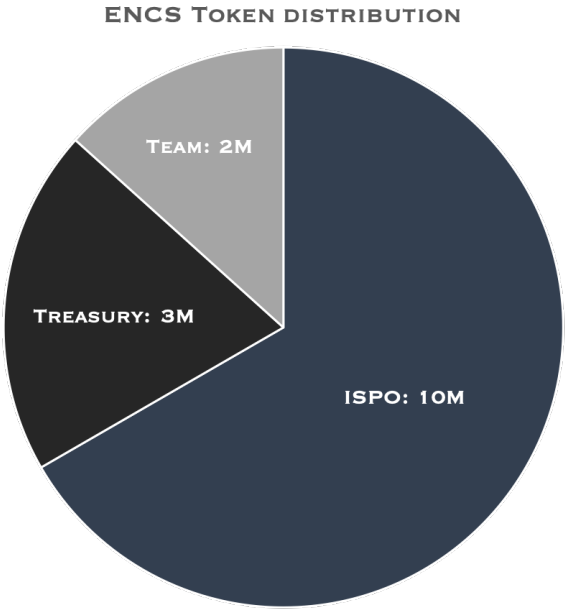

Figure 7: Initial token distribution of ENCS.

ENCS Minting and Distribution Event will happen once the ISPO is completed. We have prepared an algorithm to do the distribution in a completely decentralized way. The details will be shared shortly before the event.

5 ENCOINS Ecosystem

In this section, we outline a possible future direction of protocol development.

5.1 DeFi Interface

ENCOINS can be potentially used in many DeFi applications in place of the ADA value they represent. Sometimes, this can be done directly from ENCOINS Ledger.

For this reason, we propose implementing an API interface to perform common operations. The interface will consist of two parts: on-chain and off-chain. Depending on a particular use case, either one may be used.

For ENCOINS to be useful in DeFi, the common functions needed include

- Send a coin along with its minting key.
- Build and share proof that a coin has a certain redeeming value or a value in a certain range.
- Build and share proof that allows a DeFi counter-party to re-mint a coin with their own minting keys.
- Verify the provided proofs or minting keys.

5.1.1 ENCOINS as a payment option

Many companies and projects that utilize the Cardano blockchain in some capacity accept payments in ADA. One of the most obvious use-cases for ENCOINS Ledger is to be used as an alternative payment method whenever a simple ADA transfer is an option. In this scenario, a customer would just need to enter a minting key for one of their coins to make the payment (assuming they already minted a coin with the appropriate value). The seller could then use our API toolkit to finalize such payments.

5.2 Additional Privacy Enhancements

In this subsection, let us consider certain protocol extensions that might be implemented by the community to further enhance the privacy of the users.

5.2.1 Collaborative transactions

If a significant number of participants use ENCOINS Ledger for private transactions between each other (as intended), then non-transacting participants may also benefit from this activity. By emulating private transactions, i.e., by re-minting some of their coins a few times, they can achieve reasonable *forward privacy*. It means that there is a significant chance that a person that redeems a coin is not the same as the person that deposited *the coin it was minted from* into the ledger. With more time passed and more re-mints, this probability naturally increases.

However, it is possible to increase the uncertainty even more dramatically. One such option would be to implement a version of CoinJoin protocol on top of ENCOINS Ledger. Instead of just transmitting the proof of knowledge for a re-mint, users can participate in a very simple MPC scheme described below.

1. Several users agree on which coins will be minted and burned in the re-minting transaction. This allows them to calculate the *challenges* in the Bulletproof algorithm.

2. Each user can then submit their part of the proof (either directly to the relayer or to a participant chosen to be a leader).

3. The relayer builds the aggregate proof, signs the transaction, and submits it to the network.

The coins minted in this transaction have all burned coins as their predecessors. Doing several rounds of such transactions, therefore, gives users very high forward privacy. Importantly, this privacy protocol does not require users to wait or submit similar amounts as inputs to these transactions.

5.2.2 ENCOINS P2P Exchange

Another protocol for forward privacy that can be built on top of ENCOINS Ledger is a peer-to-peer exchange of ENCOINS and ADA.

ENCOINS P2P could work like this. A buyer sends ADA to the seller and asks to mint a specific coin. The seller re-mints some of their coins on ENCOINS Ledger into the requested coin plus change. This does not create a public link between the ADA sent by the buyer and the minted coin on the ledger.

In general, peer-to-peer exchanges rely on trust in the sellers. Such exchanges usually use several metrics to inform the buyers about the reliability of a particular seller. Oftentimes, these include the number of transactions and the percentage of satisfied buyers. Besides those two metrics, ENCOINS P2P could have a system where a buyer would be able to prove that the seller has not minted the desired coin if it ever occurs. This would obviously damage the reputation of a dishonest seller, strongly discouraging such behavior.

5.3 Extension to Other Native Assets

Finally, we point out that all ENCOINS protocols can be implemented for any native asset on Cardano, not just ADA. Extension of ENCOINS protocols to other native assets, among other things, could have the following applications.

- **Enhance over-the-counter (OTC) trading on Cardano**. As public blockchains are analyzed for any kind of financial data, a movement of a large quantity of a particular token may be picked up by trading algorithms to extract value from other market participants. Private transactions involving ENCOINS do not reveal the number of tokens moved. When a transaction happens on ENCOINS Ledger, it is not possible to tell whether the ownership of the tokens actually changed or not.
- **Enable private NFT ownership**. For collectors who own expensive art NFTs, it is particularly difficult to maintain privacy. One of the ways how a user can own an NFT while not exposing any of their wallets is by purchasing its respective encrypted coin (sitting on ENCOINS Ledger). Optionally, they can burn the coin to withdraw the NFT into a fresh wallet for display.
- **Privacy-focused DEX**. When both native assets have their respective encrypted coins, it would be possible to organize the exchange of one type of ENCOINS for another (e.g., ADA ENCOINS for Djed ENCOINS) directly on ENCOINS Ledger.

In principle, a single encrypted coin could even represent a basket of assets. However, the practical efficiency of such a solution needs further investigation as it may be too expensive to mint this kind of encrypted coins given the current capabilities of the Cardano blockchain.

References

[1] Amitabh Saxena Alexander Chepurnoy. Zerojoin: Combining zerocoin and coinjoin, 2020.

[2] Kurt M. Alonso and Jordi Herrera Joancomartí. Monero: Privacy in the blockchain, 2018. `https://eprint.iacr.org/2018/535.pdf`.

[3] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin, 2014. `http://zerocash-project.org/media/pdf/zerocash-extended-20140518.pdf`.

[4] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 315–334, 2018.

[5] Ivan Damgård. On σ-protocols. CPT 2010, v.2, 2010.

[6] Daniel Firth, Lukasz Golębiewski, Mason Mackaman, Ryan Matovu, Pawel Szulc, and Morgan Thomas. Orbis 1.0: A general-purpose layer 2 zero-knowledge rollup protocol. White paper, 2022. `https://papers.orbisprotocol.com/whitepaper.pdf`.

[7] Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016*, pages 305–326, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[8] Jens Groth and Markulf Kohlweiss. One-out-of-many proofs: Or how to leak a secret and spend a coin. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015*, pages 253–280, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[9] Aram Jivanyan. Lelantus: A new design for anonymous and confidential cryptocurrencies. Cryptology ePrint Archive, Report 2019/373, 2019. `https://ia.cr/2019/373`.

[10] Mary Maller, Sean Bowe, Markulf Kohlweiss, and Sarah Meiklejohn. Sonic: Zero-knowledge snarks from linear-size universal and updateable structured reference strings. Cryptology ePrint Archive, Paper 2019/099, 2019. `https://eprint.iacr.org/2019/099`.

[11] Alexey Pertsev, Roman Semenov, and Roman Storm. Tornado cash privacy solution, 2019. `https://tornado.cash/Tornado.cash_whitepaper_v1.4.pdf`.

[12] Srinath Setty. Spartan: Efficient and general-purpose zksnarks without trusted setup. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020*, pages 704–737, Cham, 2020. Springer International Publishing.