Bunny Notes

A Cryptographic Note Protocol

Off-Chain Transfers and Store of Value

StrawberyChocolateFudge
https://github.com/StrawberryChocolateFudge

Abstract. *A decentralized peer to peer cryptographic note protocol can allow bridging assets from blockchains to the physical world by connecting traditional contracts with smart contracts.*
Zero-knowledge proofs provide part of the solution as they can be used to verify secrets that are stored off-chain in printable PDF documents. Using QR code technology the Notes are easy to spend and verify and transferring value becomes as easy as physically giving a document to another person. The value is deposited into smart contracts and cannot be withdrawn without supplying the off-chain proof and provides a completely trustless experience.

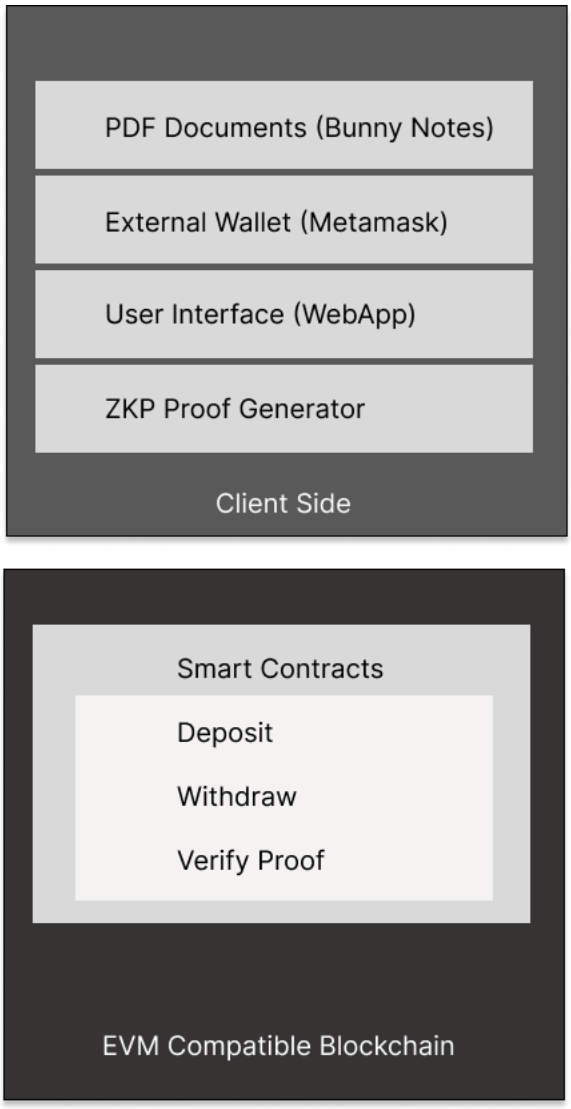
1. Introduction

Blockchain transactions inherently suffer from a few problems that make them hard for every day use. To conduct a transaction, parties need to exchange public keys so the address of the recipient is known and therefore the system cannot function similar to cash and it works more like bank transfers. The administration of financial transactions can become increasingly difficult as these transfers leave no paper trail, Travel Rule Compliance is hard and it’s nearly impossible to gift cryptocurrency in a trustless way without asking for a recipient's address, without revealing to him the intention to of gifting. Crypto cannot be stored with other physical assets in a safe without using private keys and it also cannot be spent without the spender using a wallet.

What is needed is a way to bridge a crypto asset to the traditional paper finance world and allow the creation of printable financial claims that hold value and can be exchanged or spent in person.

This could be easily implemented using zero-knowledge proofs by building on already existing frameworks and learning from the current state of the art technologies.

We propose to create a protocol using Solidity and Circom for the smart contract layer and a simple web application using React for the User Interface. The software stack also contains jsPDF for rendering the Bunny Notes as a Printable PDF.



2. Bunny Notes

We define a Bunny Note as a printable PDF document that contains a secret which can be used to withdraw value that was deposited into a smart contract.
Bunny Notes are only considered valid if they are fully backed.

Bunny Note

https://bunnynotes.finance

The smart contract on Binance Smart Chain Testnet will pay the bearer on demand the sum of 100 BNB

To verify the note, open bunnynotes.finance and scan the QR code or paste the code. You can safely share this code with others.

Terms and Conditions

The following terms and conditions apply to the use of Bunny Notes on the website bunnynotes.finance:

PURPOSE: A bunny note is a printable and verifiable claim to withdraw value that was deposited into a smart contract through the website bunnynotes.finance. It is designed to facilitate the transfer of value between parties who deposit into and withdraw funds from a smart contract.

INDEMNIFICATION: The creator of the smart contract and the website bunnynotes.finance are indemnified from any claims, damages, or losses arising from the use of Bunny Note. Parties acknowledge and agree that they use Bunny Note at their own risk and that the creator of the smart contract and the website bunnynotes.finance shall not be liable for any damages, losses, or expenses arising from the use of Bunny Note. Bunnynotes.finance offers no guarantee of any kind. Use it at your own risk.

VERIFICATION: Bunny Note can be verified on the blockchain. Parties acknowledge and agree that they are solely responsible for verifying the authenticity of the Bunny Note and ensuring that it has not been tampered with.

TRANSFERABILITY: Bunny Note is transferable and can be redeemed by the holder of the note. Parties acknowledge and agree that they are solely responsible for transferring the note to the intended recipient and that the creator of the smart contract and the website bunnynotes.finance shall not be liable for any losses or damages arising from the transfer of the note.

GOVERNING LAW AND JURISDICTION: These terms and conditions shall be governed by and construed in accordance with the laws of the jurisdiction where the parties are located. Any dispute arising between the parties shall be resolved in the courts of the jurisdiction where the parties are located without involveing bunnynotes.finance or the creator of the smart contract.

By using Bunny Note on the website bunnynotes.finance, parties acknowledge and agree to be bound by these terms and conditions. If parties do not agree to these terms and conditions, they should not use the Bunny Note and the website bunnynotes.finance.

You can add extra information about the transaction to comply with regulation.

Transfer from: Transfer to: Comment:

Use the QR Code to withdraw the value. Keep this confidential

Denomination: 100 BNB
Network: Binance Smart Chain Testnet
Token Address: Native Token
To withdraw the value scan the QR Code or paste the code into the DApp at bunnynotes.finance

https://bunnynotes.finance

The Bunny Notes contain a semantic contract and two QR codes. The first code is used only for verification while the second can be used for verification or for withdrawing value.
The First QR code may be shared publicly while the second one contains a secret that should be treated like a private key.

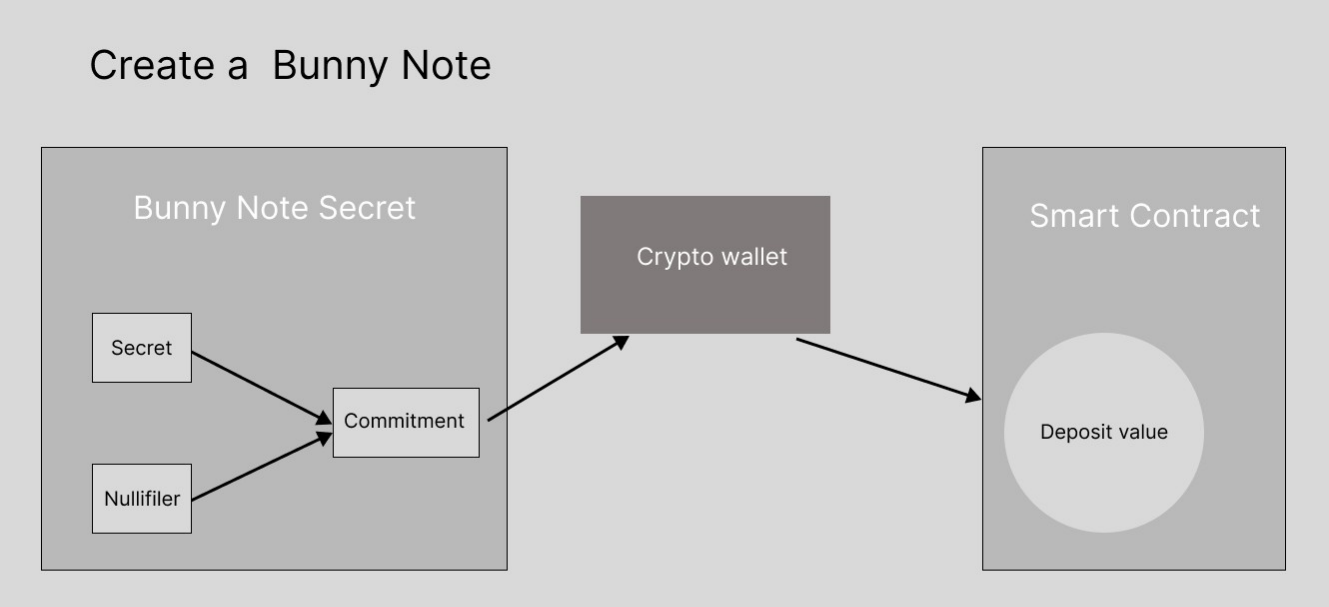
How it works?

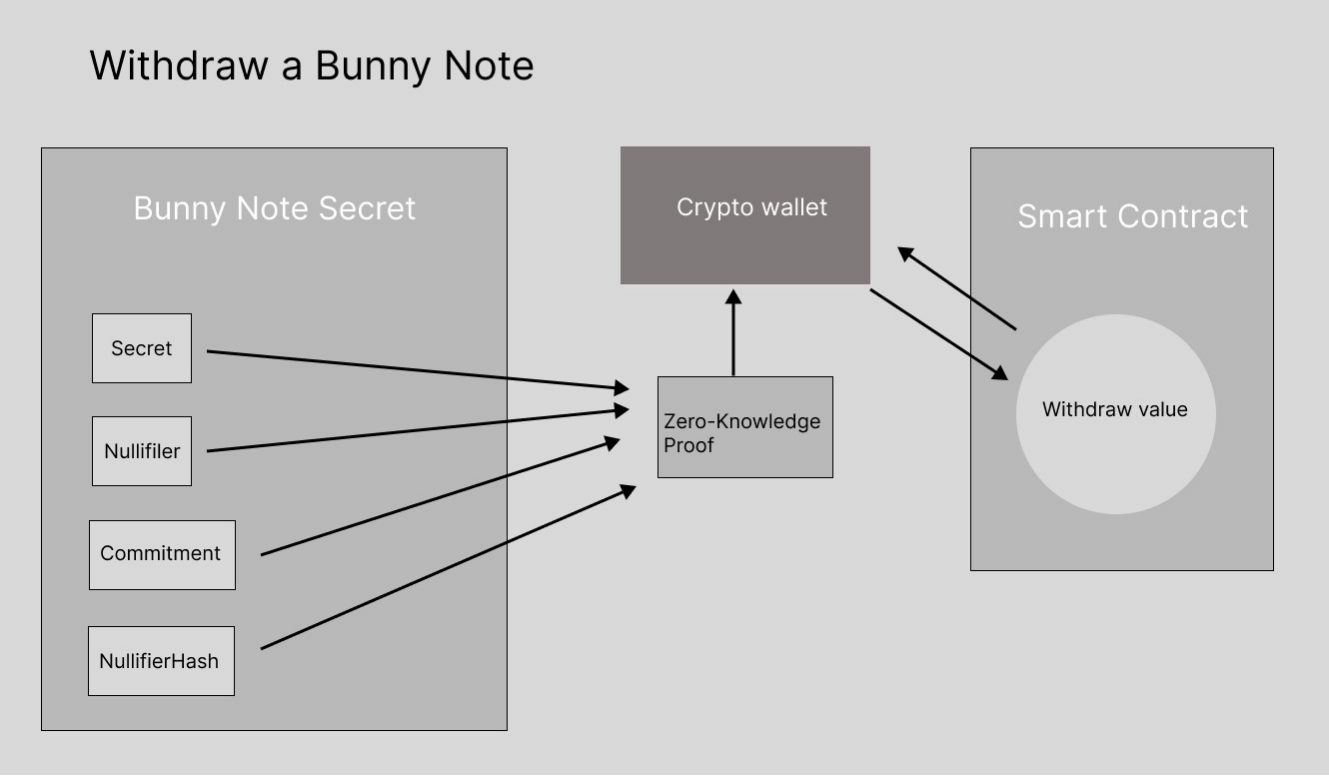
Bunny Notes are only valid if they can be verified to prove the value is still contained inside the smart contract. The notes need to be stored securely, just like private keys, however they may change holders to conduct off-chain transactions.

The secret contained in the QR code is the pre-image of a cryptographic hash that was saved in a smart contract when some value was deposited into it.
This hash is called the commitment and the Bunny Notes protocol uses Poseidon Hash to create commitments.

The creation of a Commitment requires a Secret and a Nullifier which are randomly generated big integers.

When recording the creation of the note on the blockchain, the Commitment is sent on contract interaction along ERC20 tokens or ETH.





To prove that Bob owns this note, he needs to acquire the Secret and the Nullifier from Alice, then compute the Commitment and the NullifierHash and use these 4 values to generate a Proof that can be sent to the blockchain with a contract interaction. On the blockchain the proof will be verified and it allows Bob to withdraw or perform other actions.

The NullifierHash is used on-chain to record the nullification of the note, so it may be never used again, the secret and the nullifier is never revealed.

Is it trustless?

The protocol relies on no third party however different perspectives must be considered to understand how trustless is the system. It requires no trust on the contract deployer and creator as he has zero access to the deposited funds however when sharing these Bunny Notes with third parties will allow every participant to spend them so it is recommended to cash them out immediately when receiving them.

What could go wrong?

Scenario 1 :

Alice could create a note and give it to Bob and who does not withdraw it immediately.

Alice might keep a copy of the note and withdraw it any time. In this case, Bob will be unable to withdraw it. Therefore it is recommended for Bob to never trust Alice and withdraw the funds instantly.

Scenario 2:

If Alice creates a note and then looses it, there is no way to recover deposited funds. Alice will lose the value and it will remain locked forever into the smart contract. There is no way to recover these lost deposits.

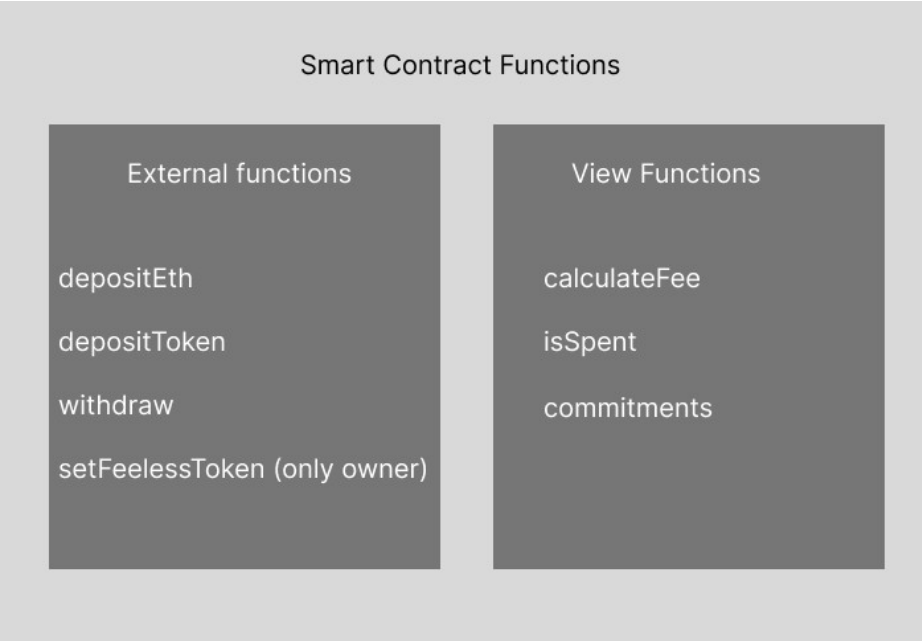
Scenario 3:

Phishing. If Alice creates a note but then visits the wrong website to withdraw it, the funds will be compromised. It is crucial that the note is never entered into a third-party website because stealing of funds does not require wallet interaction from the user!

3. Smart Contract

The Bunny Notes smart contract contains 3 functions that allow depositing value and withdrawing it. The users may deposit ERC20 tokens of any kind, ETH or other native tokens like BTT or BNB, it depends on the blockchain and a withdraw function that verifies the ZKP provided used for withdrawing the value and the function allows pull payments.

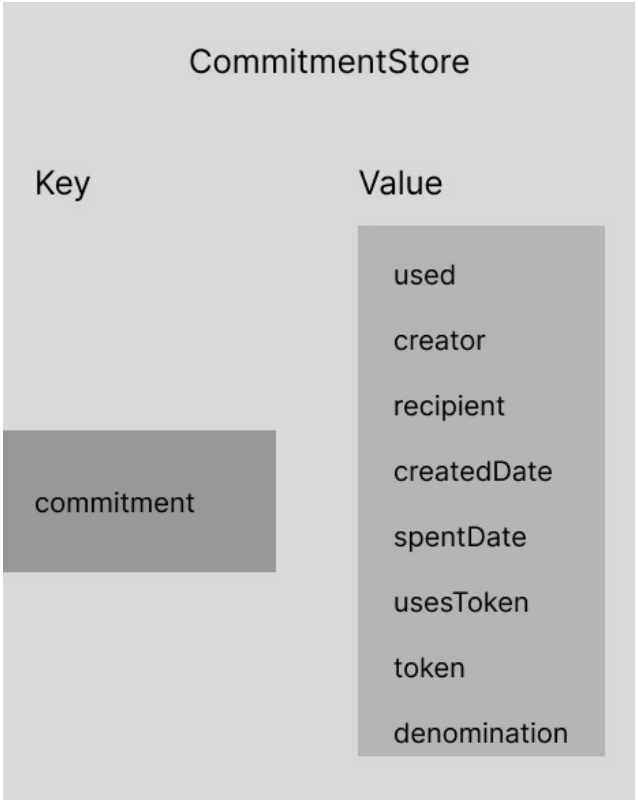
When the value is deposited a 1% fee is applied to all the deposits. This can be avoided if the deposits are made with a value transfer token ZKB specially created to use with Bunny Notes.
The owner of the smart contract has no access to the deposited value is unable to alter the underlying contract code. The contract is not pausable and requires no trust in the owner to maintain the service. The owner is only able to modify the address of the Feeless token, the ERC20 token which is used for transferring value without paying deposit fees and this was added in case the token is deployed after the main contract. The isSpend function is used to verify if a note is spent and calculateFee will return the fee, while the commitments() is used to query bunny notes.



4. Privacy

The smart contract provides an easy way to query transaction details about the creator and the spender and provides no anonymity, to help mitigate abuse and combat money laundering. The underlying technology resembles the infamous Tornado Cash in the way it uses zero-knowledge proofs for withdrawing deposits, but uses some more efficient cryptography features like updated hashing algorithm.

All the privacy features are missing to avoid misuse. The protocol cannot function as a mixer and this is an intentional feature.



The Contract contains a commitment store that is accessed with the commitment.

If we know the commitment we can access values related to the Bunny Note.
Used is a boolean, if the commitment does not exists it returns default values so this is false.
The **creator** is the address that created the bunny note and deposited the value
Recipient is the account that withdrew the value
createdDate and **spentDate** are numbers to represent the time the note was created and withdrawn
usesToken is a boolean to determine if this BunnyNote uses an ERC20 token, if true then the **token** is not zero address but a valid ERC20 address. The **denomination** is the value of the note that was deposited to back it.

5. ZKP

The protocol uses the Groth-16 cryptography proof system, which is one of the most popular zkSnark proving schemes and uses circom circuits to define how the proof verification should function.

Using an existing Powers of Tau ceremony with many contributors provides a huge benefit and so the Polygon Hermez Powers of Tau ceremony ptau files were used. This removes trust as the creator of the protocol did not have any access to sensitive secrets generated during the ceremony that could be used to break the cryptography.

The application uses a simple commitment reveal scheme, the Zero-knowledge proofs prove the ownership of the secret which is embedded inside the document and accessible via QR code, and so we may prove we own the Bunny Note by proving we know the pre-image of a hash (commitment) without ever revealing it.

6. Tokenomics.

The protocol charges fees. The deposits come with a constant 1% fee which cannot be changed however one ERC20 token can be chosen by the owner which can be used without any fees and so ZKB tokens were created. The tokens follow the ERC20 token standard and are created with the OpenZeppelin library.

100 million tokens are minted on deployment and no more shall be ever created.
The TokenSale will distribute 50% of the token supply, while the rest is allocated to the development team.
The tokens are not registered securities in the USA and are created with the sole purpose to transfer value without fees using the Bunny Notes protocol.

7. Conclusion

Using a protocol like Bunny Notes, we can effectively bridge crypto currency to the world of paper finance and unlock new ways to store and spend it. The new use-cases that can emerge from this are numerous. We believe that zero-knowledge proof technology is not only for scaling and privacy and have unexplored use cases like this. The goal to create Bunny Notes was to create something simple, effective and useful and we have successfully fulfilled it.
In the future, the developer team will aim to decentralize the protocol more by hosting the User interface on IPFS and Arweave and so it may become one more immutable , unstoppoble, uncensorable protocol built on web3.