Privacy NFT marketplace

Introduction

In the rapidly evolving realm of Non-Fungible Tokens (NFTs), the need for privacy in ownership transactions has never been more crucial. Existing NFT marketplaces expose sensitive details on public blockchains, compromising user privacy. Our solution, the Privacy NFT Marketplace, introduces a pioneering approach using zk-SNARKs technology to safeguard user identities during NFT transactions.

This proposal outlines a secure and decentralized platform that leverages cryptographic proofs, allowing verifiable transactions without disclosing sensitive information. By prioritizing privacy through zk-SNARKs, our proposal addresses current privacy concerns in NFT ownership, fostering trust and confidence among users. This innovation aims to redefine the NFT landscape, encouraging broader adoption and contributing to the development of a privacy-centric NFT ecosystem.

How it works

Seller

Step 1: Deposit NFT and Generate Nullifier

- The seller initiates the process by depositing their NFT into the marketplace's state Merkle tree and generates a new spending NFT nullifier associated with the deposited NFT.

Step 2: Create Sell Order

1. Public: Listing Information:

The seller creates a sell order, providing essential information. * Collection Address: The address of the NFT collection.

- Token ID: The unique identifier of the NFT.
- Amount: The desired selling price.
- Collection Address: The address of the NFT collection.
- Token ID: The unique identifier of the NFT.
- Amount: The desired selling price.
- Private: Generate Seller's ERC20 Nullifier:
- The seller generates a new spending ERC20 nullifier associated with the sale amount and a seller spending key.
- The seller generates a new spending ERC20 nullifier associated with the sale amount and a seller spending key.
- Public: Spending NFT Proofs:
- The seller creates proofs containing the collection address, token ID, spending NFT nullifier, and selling amount.
- The seller creates proofs containing the collection address, token ID, spending NFT nullifier, and selling amount.

→ (Collection address, token id, amount, new seller's ERC20 Nullifier hash, spending NFT Proofs)

Buyer

Step 1: Deposit ERC20 and Generate Nullifier

- The buyer deposits ERC20 into the marketplace's state Merkle tree and generates a new spending ERC20 nullifier associated with the deposited amount

Step 2: Create Accept-Sell Order

1. Listing Lookup and Information:
2. The buyer looks up the listing information on the public marketplace and decides to make a purchase.
3. The buyer creates an accept-sell order, providing necessary details.
4. Collection Address, Token ID, Amount.

5. Collection Address, Token ID, Amount.
6. The buyer looks up the listing information on the public marketplace and decides to make a purchase.
7. The buyer creates an accept-sell order, providing necessary details.
8. Collection Address, Token ID, Amount.
9. Collection Address, Token ID, Amount.
10. Spending ERC20 Proofs:
11. The buyer generates proofs containing the collection address, token id and lists proofs of their spending ERC20 nullifier.
12. The buyer generates proofs containing the collection address, token id and lists proofs of their spending ERC20 nullifier.
13. Generate New Buyer's Nullifiers:
14. New Buyer Spending NFT Nullifier: Associated with the collection address, token ID, and buyer spending key.
15. New Buyer Spending Left-ERC20 Nullifier: Associated with the total deposit amount minus the NFT price.
16. New Buyer Spending NFT Nullifier: Associated with the collection address, token ID, and buyer spending key.
17. New Buyer Spending Left-ERC20 Nullifier: Associated with the total deposit amount minus the NFT price.

→ (Collection address, token id, amount, new buyer's NFT Nullifier hash, new buyer's ERC20 Nullifier hash, spending ERC20 proofs)

On-chain Verifier

The on-chain verifier function receives the following parameters:

- Collection Address
- Token ID
- Amount
- New Seller Spending ERC20 Nullifier Hash
- Seller Spending NFT Proofs
- Buyer Spending ERC20 Proofs
- New Buyer Spending NFT Nullifier Hash
- New Buyer Spending Left-ERC20 Nullifier Hash

Verification Steps:

Verify all nullifiers hash are available

1. Verify Seller's ERC20 Nullifier Hash:
2. Confirm the amount in the new seller spending ERC20 nullifier hash.
3. Confirm the amount in the new seller spending ERC20 nullifier hash.
4. Verify Seller's NFT Proofs:
5. Validate the seller's spending NFT proofs.
6. Validate proof with given amount
7. Validate the seller's spending NFT proofs.
8. Validate proof with given amount
9. Verify Buyer's ERC20 Proofs and Nullifier Hash:
10. Confirm the buyer's spending ERC20 proofs and new buyer spending left-ERC20 nullifier hash.

11. Validate proof with given collection address and token id
12. Confirm the buyer’s spending ERC20 proofs and new buyer spending left-ERC20 nullifier hash.
13. Validate proof with given collection address and token id
14. Verify Collection Address and Token ID in Buyer’s NFT Nullifier Hash:
15. Ensure the correctness of the collection address and token ID in the new buyer spending NFT nullifier hash.
16. Ensure the correctness of the collection address and token ID in the new buyer spending NFT nullifier hash.

Update state:

1. Mark seller’s spending NFT nullifier hash as used
2. Mark buyer’s spending ERC20 nullifier hash as used
3. Add seller’s spending ERC20 nullifier hash to state tree
4. Add buyer’s spending NFT nullifier hash to state tree
5. Add buyer’s spending left-ERC20 nullifier to state tree

This robust verification process ensures the integrity and security of the NFT transaction on our marketplace, providing a trustworthy environment for both sellers and buyers.

Certainly! If you have any questions or need further clarification on any aspect of the process outlined above, feel free to comment