GhostPad: Anonymous Token Launching

Simon Judd
simon@psychovirtual.io

February 25, 2025

Abstract

We present a generalized framework for anonymous token launching. This prototol can be used to launch and mint tokens anonymously with no reference to the liquidity provider.

Contents

1 Introduction

Over the last year, platforms like pump.fun demonstrated insatiable demand for token launching and minting. However, users still face significant friction points that hinder seamless token creation and distribution.

Traceable Creation

Current token launch platforms expose creator identity, eliminating privacy in the token creation process. This transparency makes it impossible to launch tokens anonymously.

Complex Privacy Tools

The available privacy-preserving solutions demand significant technical knowledge, creating a substantial barrier to entry for average users. These tools typically require:

- Understanding of advanced cryptographic concepts
- Command-line interface experience
- Knowledge of blockchain infrastructure

Identity and Proof Fragmentation

Current token launch mechanisms lack privacy-preserving identity proofs, creating significant operational friction:

- Wallet addresses permanently link creator identities to launches, compromising privacy and security

- Token locking requires manual verification through third-party platforms and public social media posts
- No standardized way to cryptographically prove social media ownership without exposing wallet addresses

These limitations force creators to choose between maintaining privacy and building credibility, while the reliance on manual verification processes increases operational overhead and security risks.

2 Notation

3 Protocol Description

3.1 Setup

Let $\mathbb{B} = \{0, 1\}$. Let $\mathcal{T}$ be a sparse Merkle tree of height 24, where each non-leaf node computes a Poseidon hash of its two children. The tree is initialized with zero values. Let $H_1 : \mathbb{B}^* \to \mathbb{Z}_p$ be a Poseidon-based commitment scheme, and $H_2 : (\mathbb{Z}_p, \mathbb{Z}_p) \to \mathbb{Z}_p$ be a MiMC permutation in sponge mode for nullifier derivation.

A trusted setup ceremony generates a Groth16 key pair $(\mathsf{pk}, \mathsf{vk})$ for the circuit $\mathcal{S}$, which proves knowledge of:

- A secret nullifier $k \in \mathbb{B}^{256}$
- A secret randomness $r \in \mathbb{B}^{256}$
- A leaf index $l \in \mathbb{B}^{24}$
- A valid Merkle path O for commitment $C = H_1(k \parallel r)$ at position l
- A hash of token metadata $D = H_1(\mathsf{name} \parallel \mathsf{supply})$ (optional)

The smart contract stores:

- The current Merkle root R and a history of the last 128 roots
- A nullifier set $\mathcal{N}$ to prevent double-minting
- A whitelist of relayers and fee parameters

3.2 Deposit

To deposit liquidity for anonymous minting:

1. The user generates $k, r \xleftarrow{\$} \mathbb{B}^{256}$ and computes $C = H_1(k \parallel r)$.
2. The user sends 1 SOL to the protocol contract, attaching C as a public commitment.
3. The contract appends C to $\mathcal{T}$, updates the Merkle root, and stores the new root in its history.

3.3 Mint

To mint tokens via a relayer:

1. The user selects a fresh Solana address A (generated anonymously) and token metadata name, supply.
2. They compute:
   - Nullifier hash $h = H_2(k)$
   - Token metadata hash $D = H_1(\mathsf{name} \parallel \mathsf{supply})$ (if applicable)
   - Merkle proof O for C under root R

3. Using pk, they generate a Groth16 proof π attesting to:

$$\mathcal{S} = \{\text{I KNOW } k, r, l, O \text{ SUCH THAT } C = H_1(k \parallel r), \\ O \text{ validates } C \text{ in } R, \\ h \notin \mathcal{N}\}$$

4. The user submits (π, R, h, A, D) to a relayer. The relayer:
   - Pays gas fees to call the protocol contract
   - Executes `mint_token` on pump.fun with metadata D
   - Sends 97% of minted tokens to A and retains 3% as fee
5. The contract verifies π with vk, checks $h \notin \mathcal{N}$, adds h to $\mathcal{N}$, and authorizes the mint.

Key Improvements over Tornado Cash(WIP):

- Sparse Merkle trees for efficient Solana state management

References

[1] Alexey Pertsev, Roman Semenov, and Roman Storm. "Tornado Cash: Privacy Solution for Ethereum." Whitepaper, Version 1.4, December 2019.

[2] Jens Groth. "On the Size of Pairing-Based Non-interactive Arguments." In *EUROCRYPT 2016*, Vol. 9666 of Lecture Notes in Computer Science, pp. 305-326. Springer, 2016.