BLENDER

@Deface, GenesisLab
Contact: blender4pepu@proton.me

Version 0.1 — March 25, 2025

Contents

1 Abstract

The **BLENDER** Protocol implements a privacy-enhanced transaction layer using cryptographic commitments and a dual-token system. This paper formally specifies the protocol's mathematical foundations, security model, and operational parameters. The **BLENDER** token is deflationary with a total supply of 1,000,000,000 tokens. Each time tokens are mixed, a fixed commission of 1000 **BLENDER** will be paid to the smart contract. This commission structure is designed to ensure the sustainability and growth of the **BLENDER** ecosystem.

2 Protocol Overview

2.1 Core Components

$$\Psi = (P, B, D, W, R) \quad \text{where:} \tag{1}$$

P	**PEPU** (Native gas token, $\mathbb{N}$-denominated)
B	**BLENDER** (ERC-20 utility token, fixed supply of 1B tokens)
D	Deposit structure $\langle amount, h(s), t, spent\rangle$
W	Withdrawal proof $\pi = (s, r)$
R	Refund condition $t > t_{deposit} + 30d$

3 Technical Specification

3.1 Deposit Mechanics

$$\text{Valid Deposit} \iff \exists d \in \{100, 10^3, 10^4, 10^5, 10^6\} \subset \mathbb{N} \tag{2}$$

Fee structure:

$$\phi = 1000 \times 10^{18} \text{ wei}_B \quad (\text{Fixed in } \textbf{BLENDER}) \tag{3}$$

Cryptographic requirements:

$$\begin{aligned} s &\xleftarrow{\$} \{0,1\}^{256} \quad (\text{32-byte secret}) \\ h(s) &= \text{keccak256}(s) \\ \text{Commitment} &= (d_P, h(s)) \in \mathbb{N} \times \{0,1\}^{256} \end{aligned}$$

3.2 Withdrawal Protocol

Withdrawal validity predicate:

$$\text{Verify}(s, C) = \begin{cases} 1 & \text{if } h(s) = C.h(s) \wedge C.spent = 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Funds transfer:

$$\Delta P = \begin{cases} d_P \to \text{Recipient} & \text{if Verify}(s, C) = 1 \\ d_P \to \text{Depositor} & \text{if } t > t_0 + 30d \end{cases} \tag{5}$$

4 Contract Methods

Below are the key methods within the **BLENDER** contract and a simple explanation of their functionality:

4.1 deposit(bytes32 hashedSecret)

This function allows a user to make a deposit into the system. The user must provide a hashed secret, which is used to uniquely identify the deposit later. The deposit amount can only be one of the specified amounts (100, 1000, 10000, 100000, or 1000000 ether). The user also needs to pay a fee of 1000 **BLENDER** tokens to initiate the deposit. Once the deposit is made, the system stores the details of the deposit in the ‘deposits‘ mapping, making the deposit available for future withdrawal.

Payout: - User provides an amount and a secret hash. - 1000 **BLENDER** fee is deducted.

4.2 withdraw(bytes calldata secret, address recipient)

This function allows a user to withdraw their deposit. The user provides the original secret, which is hashed to match the corresponding deposit. If the secret matches the one stored in the contract, and the deposit has not already been withdrawn, the specified amount is sent to the recipient’s address. This operation marks the deposit as ”spent” to prevent double withdrawals.

Payout: - User provides the original secret. - If valid, the deposited amount is transferred to the recipient.

4.3 refund(bytes32 hashedSecret)

If 30 days have passed since the deposit was made and the funds were not withdrawn, the user can request a refund. This function allows the user to reclaim their deposit. The contract ensures that the refund is only allowed after the 30-day period, and once refunded, the deposit is marked as ”spent.”

Payout: - User requests a refund after 30 days. - The deposit is returned to the user.

5 Security Model

5.1 Cryptographic Guarantees

Component	Type	Security Property
$h(s)$	keccak256	Collision resistance
s	256-bit	Brute-force resistance: $\mathcal{O}(2^{128})$
ECDSA	secp256k1	EUF-CMA secure

5.2 Economic Safeguards

$$\mathcal{F}_{fee} = \phi \times N_d \quad \text{(Fee accumulation)} \tag{6}$$

Where N_d is the number of deposits. Fee destruction schedule:

$$B_{burn} = \mathcal{F}_{fee} \times \eta \quad \text{where } 0 \leq \eta \leq 1 \tag{7}$$

6 Conclusion

The **BLENDER** Protocol establishes a new paradigm in blockchain transaction privacy through its innovative combination of:

- Dual-token economic model (**PEPU** + **BLENDER**)
- Cryptographic commitment scheme
- Time-locked refund mechanism
- Fixed-fee structure with deflationary pressure

The protocol’s non-custodial architecture and mathematically verifiable security properties position it as fundamental infrastructure for private value transfer in Web3 ecosystems.

References

[1] Project GitHub Repository
`https://github.com/blender4pepu`

[2] Keccak Specifications
`https://keccak.team/keccak.html`

[3] SEC 2: Recommended Elliptic Curve Domain Parameters
`https://www.secg.org/sec2-v2.pdf`

[4] ERC-20 Token Standard
`https://eips.ethereum.org/EIPS/eip-20`