Merkle Tree Implementation in Rust

License MIT

Rust 1.60+

This project provides a **Merkle Tree** implementation in Rust, a cryptographic data structure that allows efficient and secure verification of the contents of a dataset. The Merkle Tree is built using the **BLS12-381 elliptic curve** and the **SHA-256 hash function**.

Table of Contents

Overview

A **Merkle Tree** is a tree in which every leaf node is labeled with the hash of a data block, and every non-leaf node is labeled with the hash of its child nodes. This structure allows efficient and secure verification of the contents of large datasets.

This implementation:

- Uses the **BLS12-381 elliptic curve** for scalar field operations.
- Uses the **SHA-256 hash function** for hashing.
- Provides methods for:
  - Building a Merkle Tree.
  - Generating Merkle proofs.
  - Verifying Merkle proofs.

Features

- **Efficient Proof Generation**: Generate proofs for specific leaves in `O(log n)` time.
- **Secure Verification**: Verify proofs using the root hash of the tree.
- **Benchmarking**: Includes benchmarks for tree construction, proof generation, and proof verification.

Installation

1. **Clone the repository**:

   ```
   git clone https://github.com/nixkitax/proof-craft.git
   cd proof-craft/merkletree
   ```

2. **Build the project**:

   ```
   cargo build
   ```

3. **Run the example**:

   ```
   cargo run
   ```

Benchmarks

The project includes benchmarks to measure the performance of the Merkle Tree operations. To run the benchmarks, use:

```
cargo bench
```

Benchmark Results

Here are the benchmark results for a Merkle Tree with 1000 leaves:

Operation	**Time Complexity**	**Benchmark Result (1000 leaves)**
Tree Construction	O(n)	556.37 µs
Proof Generation	O(log n)	443.36 µs
Proof Verification	O(log n)	0.48 ms