1. 算法概述

Poseidon2 是一种基于置换的密码学哈希函数，专为零知识证明系统优化设计。它采用海绵结构（Sponge Construction），具有以下核心特性：

算术友好性：在有限域上高效运算

低约束复杂度：适合 ZK-SNARK 电路实现

安全性：提供 256 位安全级别

2. 数学基础

2.1 有限域运算

在素数域 \mathbb{F}\_p 上操作，其中 p 为质数（如 BN254 曲线的标量域）：

p=21888242871839275222246405745257275088548364400416034343698204186575808495617

2.2 海绵结构

Poseidon2 使用海绵函数处理输入：

\begin{aligned}

\text{状态} & : \mathbf{s} \in \mathbb{F}\_p^t \\

\text{吸收阶段} & : \mathbf{s} \leftarrow f(\mathbf{s} \oplus \mathbf{m}\_i) \\

\text{挤出阶段} & : \mathbf{z}\_j = \text{truncate}(f(\mathbf{s}))

\end{aligned}

其中：

t = 状态大小 (本实现 t=3)

\mathbf{m}\_i = 输入消息块

f = Poseidon 置换函数

3. 核心置换函数

3.1 状态初始化

对于输入 \mathbf{m} = [m\_1, m\_2]：

\mathbf{s}^{(0)} =

\begin{bmatrix}

0 \\ // \text{容量字段}

m\_1 \\ // \text{数据字段}

m\_2 // \text{数据字段}

\end{bmatrix}

3.2 轮函数结构

Poseidon2 置换由三种操作组成：

1. AddRoundConstants (ARC)：

\mathbf{s} \leftarrow \mathbf{s} + \mathbf{rc}^{(i)}

2. SubWords (S-box)：

完全轮：对所有元素应用 x^\alpha

部分轮：仅对第一个元素应用 x^\alpha

s\_j \leftarrow

\begin{cases}

s\_j^\alpha & \text{完全轮} \\

s\_0^\alpha & \text{部分轮 } (j=0) \\

s\_j & \text{部分轮 } (j \neq 0)

\end{cases}

其中 \alpha = 5 (本实现参数)

3. MixLayer (MDS)：

\mathbf{s} \leftarrow M \cdot \mathbf{s}

其中 M 是最大距离可分矩阵

3.3 轮次结构

使用论文推荐的轮次参数：

R\_F = 8 \quad (\text{完全轮}), \quad R\_P = 5 \quad (\text{部分轮})

轮次顺序：

1. R\_F/2 = 4 个完全轮

2. R\_P = 5 个部分轮

3. R\_F/2 = 4 个完全轮

4. 参数规范

4.1 本实现参数

参数 值 描述

n 256 安全级别 (位)

t 3 状态大小

d 5 S-box 指数

R\_F 8 完全轮数

R\_P 5 部分轮数

c 1 容量字段数

r 2 数据字段数

4.2 常数生成

轮常数 \mathbf{rc}^{(i)} 和 MDS 矩阵 M 使用以下方法生成：

1. 基于 \pi 的伪随机数生成器

2. 满足最大距离可分属性

3. 在 circomlib 中预计算优化

5. 电路实现

5.1 约束系统

Poseidon2 置换的 R1CS 约束：

template Poseidon(nInputs) {

signal input inputs[nInputs];

signal output out;

// 状态初始化

component sbox[3];

component mix[3];

// 轮函数实现

for (var r = 0; r < totalRounds; r++) {

// AddRoundConstants

for (var i = 0; i < t; i++) {

state[i] <== state[i] + roundConstants[r][i];

}

// S-box层

if (r < R\_F/2 || r >= R\_F/2 + R\_P) {

// 完全轮

for (var i = 0; i < t; i++) {

sbox[i] = Quint(i);

sbox[i].in <== state[i];

state[i] <== sbox[i].out;

}

} else {

// 部分轮

sbox[0] = Quint();

sbox[0].in <== state[0];

state[0] <== sbox[0].out;

}

// MDS混合层

for (var i = 0; i < t; i++) {

mix[i] = MDSRow(i);

for (var j = 0; j < t; j++) {

mix[i].in[j] <== state[j];

}

nextState[i] <== mix[i].out;

}

state = nextState;

}

out <== state[0];

}

5.2 复杂度分析

操作 约束数 描述

S-box (x⁵) 3 每个域元素

MDS 乘法 t² 矩阵乘法

总计 ~3,000 完整置换

6. Groth16 证明系统

6.1 算术电路

将 Poseidon2 表示为二次算术程序：

C(\mathbf{x}, \mathbf{w}) = 0

其中：

\mathbf{x} = [out] (公开输入)

\mathbf{w} = [in\_0, in\_1] (隐私输入)

6.2 Groth16 证明生成

\pi = (\mathbf{A}, \mathbf{B}, \mathbf{C})

满足：

\mathbf{A} = \alpha + \sum a\_i\mathbf{u}\_i + r\delta \\

\mathbf{B} = \beta + \sum b\_i\mathbf{v}\_i + s\delta \\

\mathbf{C} = \frac{\sum a\_i(\beta\mathbf{u}\_i + \alpha\mathbf{v}\_i + \mathbf{w}\_i) + h(\tau)\delta + s\mathbf{A} + r\mathbf{B} - rs\delta}{\gamma}

6.3 验证方程

[\mathbf{A}]\_1 \cdot [\mathbf{B}]\_2 = [\alpha\beta]\_1 \cdot [\gamma]\_2 + [\mathbf{C}]\_1 \cdot [\delta]\_2 + \sum x\_i[\mathbf{u}\_i\beta + \mathbf{v}\_i\alpha + \mathbf{w}\_i]\_1