



## Project 3: 用 circom 实现 poseidon2 哈希 算法的电路

学院: 网络空间安全学院  
专业: 密码科学与技术  
姓名: 李双平  
学号: 202200180026

2025 年 8 月 15 日

## 目录

<b>1</b>	<b>实验步骤</b>	<b>2</b>
<b>2</b>	<b>代码解释</b>	<b>2</b>
2.1	poseidon2.circom . . . . .	2
2.2	compute_witness.js . . . . .	3
<b>3</b>	<b>实验结果</b>	<b>3</b>

## 1 实验步骤

本实验旨在使用 Circom 实现 Poseidon2 哈希算法的零知识证明电路，并利用 snarkjs 工具生成与验证 Groth16 协议的证明。实验步骤如下：

### 1. 编写电路文件与辅助脚本

编写 poseidon2.circom 实现 Poseidon2 哈希算法逻辑；编写 compute\_witness.js 作为生成 Witness 文件的辅助脚本。

### 2. 编译 Circom 电路

使用以下命令将 Circom 电路编译为 R1CS 和 WebAssembly 文件：

```
circom poseidon2.circom --r1cs --wasm --sym -o zkout/
```

其中：

- `--r1cs`：生成约束系统文件（.r1cs）。
- `--wasm`：生成 WebAssembly 文件和相关 JS 文件。
- `--sym`：生成调试符号文件。
- `-o`：指定输出目录。

### 3. 准备输入文件

创建 input.json，其中包含：

- 私有输入（原像）。
- 公共输出（哈希值）。

### 4. 生成 Witness

使用编译生成的 poseidon2.wasm 和 compute\_witness.js 脚本生成 Witness 文件：

```
node compute_witness.js zkout/poseidon2.wasm input.json zkout/witness.wtns
```

### 5. 生成证明（Groth16 协议）

使用 snarkjs 工具，执行可信设置、生成证明密钥、导出验证密钥，并生成证明。

### 6. 验证证明

验证生成的证明是否有效，若输出 OK，表示证明有效，零知识证明过程完成。

## 2 代码解释

### 2.1 poseidon2.circom

该文件实现了一个参数为  $(n, t, d) = (256, 3, 5)$  的 Poseidon2 哈希电路，主要模块如下：

- **Exp5Box**：实现 S-box 功能，将输入提升到五次幂，作为非线性变换核心。

- Poseidon2Step :

1. 加入轮常数。
2. 按照轮类型（全 S-box 或部分 S-box）对状态向量进行幂运算。
3. 与 MDS 矩阵相乘以实现混合。

- Poseidon2Main :

1. 初始化状态，将私有输入与常数元素组合。
2. 执行指定轮数的 Poseidon2Step。
3. 最终输出第一个状态元素作为哈希结果。

## 2.2 compute\_witness.js

该脚本用于生成 `witness.wtns` 文件，其流程如下：

1. 读取命令行参数，包括 `wasm` 文件路径、输入 JSON 文件路径和输出 `wtns` 文件路径。
2. 读取输入数据和 `wasm` 文件二进制内容。
3. 调用 `witness_calculator.js` 提供的 `calculateWTNSBin` 方法计算 `witness`。
4. 将生成的二进制 `witness` 写入指定文件路径。

其核心作用是在电路编译完成后，将具体输入数据代入电路计算出所有中间信号和输出，生成可供证明生成使用的 `.wtns` 文件。

## 3 实验结果

实验结果如下：




图 1: 生成证明

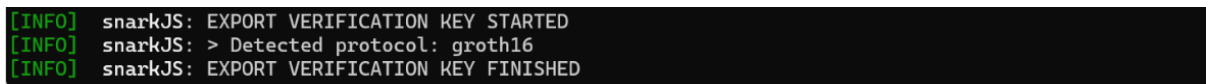


图 2: 验证证明