
山东大学
SHANDONG UNIVERSITY

# 实验 3

## 用 circom 实现 poseidon2 哈希算法的电路

学院 网络空间安全学院

# 目录

# 1 实验目的

本实验在 Ubuntu 20.04 系统中使用 Circom 实现 Poseidon2 哈希电路并生成 Groth16 证明。

Table 1: Some instantiations for POSEIDON2$^{\pi}$, where $n = \lceil \log_2(p) \rceil$.

| $(n, t, d)$ | $R_F$ | $R_P$ |
|---|---|---|
| $(31, 16, 5)$ | 8 | 14 |
| $(31, 24, 5)$ | 8 | 22 |
| $(64, 8, 7)$ | 8 | 22 |
| $(64, 12, 7)$ | 8 | 22 |
| $(256, 2, 5)$ | 8 | 56 |
| $(256, 3, 5)$ | 8 | 56 |

图 1: table1

# 2 实验内容

## 2.1 实验步骤

### 2.1.1 环境配置

依据 circom 说明文档 https://docs.circom.io/，首先进行系统更新；

```
sudo apt update && sudo apt upgrade -y
```

sudo kill -9 手动关闭冲突进程，并修复包管理器状态

```
#安装基础依赖
sudo apt install -y build-essential git curl npm nodejs
```

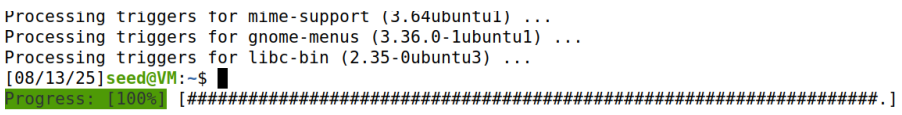


图 2: 安装基础依赖

```
#安装 Node.js 16
curl -fsSL https://deb.nodesource.com/setup_16.x | sudo -E bash -
sudo apt install -y nodejs
```

```
Removing libnode64:amd64 (10.19.0~dfsg-3ubuntu1.6) ...
(Reading database ... 208843 files and directories currently installed.)
Preparing to unpack .../nodejs_16.20.2-1nodesource1_amd64.deb ...
Unpacking nodejs (16.20.2-1nodesource1) over (10.19.0~dfsg-3ubuntu1.6) ...
Setting up nodejs (16.20.2-1nodesource1) ...
Processing triggers for libc-bin (2.35-0ubuntu3) ...
Processing triggers for man-db (2.9.1-1) ...
[08/13/25]seed@VM:~$
```

图 3: 安装 Node.js 16

```
#安装 circom 和 snarkjs
npm install -g circom snarkjs
```

```
Run `npm audit` for details.
npm notice
npm notice New major version of npm available! 8.19.4 -> 11.5.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.5.2
npm notice Run npm install -g npm@11.5.2 to update!
npm notice
[08/13/25]seed@VM:~$
```

图 4: 安装 circom 和 snarkjs

```
#安装 circomlib
git clone https://github.com/iden3/circomlib.git
cd circomlib && npm install
```

```
[08/13/25]seed@VM:~$ git clone https://github.com/iden3/circomlib.git
Cloning into 'circomlib'...
remote: Enumerating objects: 4769, done.
remote: Counting objects: 100% (1153/1153), done.
remote: Compressing objects: 100% (221/221), done.
remote: Total 4769 (delta 1011), reused 932 (delta 932), pack-reused 3616 (from 2)
Receiving objects: 100% (4769/4769), 9.25 MiB | 3.83 MiB/s, done.
Resolving deltas: 100% (2992/2992), done.
[08/13/25]seed@VM:~$ cd circomlib && npm install
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: 'mocha@11.1.0',
npm WARN EBADENGINE   required: { node: '^18.18.0 || ^20.9.0 || >=21.1.0' },
npm WARN EBADENGINE   current: { node: 'v16.20.2', npm: '8.19.4' }
npm WARN EBADENGINE }

added 243 packages, and audited 244 packages in 8s

81 packages are looking for funding
  run `npm fund` for details

2 low severity vulnerabilities

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
[08/13/25]seed@VM:~/circomlib$
```

图 5: 安装 circomlib

接着创建项目结构，

```
mkdir poseidon-circuit && cd poseidon-circuit
```

```
mkdir -p circuits scripts artifacts
#创建电路文件（核心实现）
touch circuits/poseidon2.circom
#创建输入模板文件
touch circuits/input.json
```

初期结构目录如下，

```
[08/13/25]seed@VM:~/.../poseidon-circuit$ tree -a
.
├── artifacts
├── circuits
└── scripts

4 directories, 0 files
```

图 6: 结构

因为 circom 的等级太低，经过一系列方法的尝试，最终选择了使用构建好的镜像的方法，并在过程中通过手动安装新版 Rust 来升级 rust，最终成功得到最新版本的 circom；

```
   Compiling constraint_generation v2.1.7 (/home/seed/poseidon-circuit/circom/co
nstraint_generation)
   Compiling type_analysis v2.1.7 (/home/seed/poseidon-circuit/circom/type_analy
sis)
warning: `parser` (lib) generated 1 warning (run `cargo fix --lib -p parser` to
apply 1 suggestion)
   Compiling exitcode v1.1.2
   Compiling circom v2.1.7 (/home/seed/poseidon-circuit/circom/circom)
    Finished `release` profile [optimized] target(s) in 4m 03s
  Installing /home/seed/.cargo/bin/circom
   Installed package `circom v2.1.7 (/home/seed/poseidon-circuit/circom/circom)`
 (executable `circom`)
[08/13/25]seed@VM:~/.../circom$ circom --version
circom compiler 2.1.7
[08/13/25]seed@VM:~/.../circom$
```

图 7: 升级 circom 版本成功

### 2.1.2 poseidon2.circom

Poseidon2 的核心是海绵结构和 Hades 置换；关于约束，S-box、线性变换约束分别为：$x_{out} = x_{in}^5$、$M \times state_{in} = state_{out}$，poseidon2.circom 内容如下：

```
pragma circom 2.1.7;
include "/home/seed/poseidon-circuit/node_modules/circomlib/circuits/poseidon.circom"

template Poseidon2() {
    signal input in[2];
```

```
    signal output out;

    component hasher = Poseidon(2);
    hasher.inputs[0] <== in[0];
    hasher.inputs[1] <== in[1];

    out <== hasher.out;
}

component main = Poseidon2();
```

### 2.1.3 compile.sh

分为编译电路、下载 PTAU、Groth16 密钥生成、导出验证密钥四个步骤:

```
#!/bin/bash
set -eo pipefail
mkdir -p artifacts

# 1. 编译电路
echo "编译电路..."
circom circuits/poseidon2.circom \
    --r1cs --wasm --sym \
    -o artifacts

# 2. 下载PTAU文件
if [ ! -f artifacts/pot12_final.ptau ]; then
    echo "下载PTAU文件..."
    for i in {1..5}; do
        if wget -q https://hermez.s3-eu-west-1.amazonaws.com/powersOfTau28_hez_final_
            break
        elif [ $i -eq 5 ]; then
            echo "PTAU文件下载失败"
            exit 1
        else
            sleep 5
        fi
    done
fi
```

```
# 3. 生成Groth16密钥
snarkjs groth16 setup \
    artifacts/poseidon2.r1cs \
    artifacts/pot12_final.ptau \
    artifacts/poseidon2.zkey

# 4. 导出验证密钥
snarkjs zkey export verificationkey \
    artifacts/poseidon2.zkey \
    artifacts/verification_key.json
```

## 2.2 实验结果

最终目录结构如下:

```
[08/13/25]seed@VM:~/poseidon-circuit$ tree -L 1
.
├── artifacts
├── build
├── circom
├── circuits
├── node_modules
├── package.json
├── package-lock.json
├── poseidon2.sym
└── scripts

7 directories, 3 files
```

图 8: 最终目录结构

```
[08/13/25]seed@VM:~/poseidon-circuit$ ./scripts/compile.sh
编译电路...
template instances: 72
non-linear constraints: 240
linear constraints: 0
public inputs: 0
private inputs: 2
public outputs: 1
wires: 243
labels: 771
Written successfully: artifacts/poseidon2.r1cs
Written successfully: artifacts/poseidon2.sym
Written successfully: artifacts/poseidon2_js/poseidon2.wasm
Everything went okay
```

图 9: 实验结果