基于 Circom 实现 Poseidon2 哈希算法的电路设计与验证实验报告

姓名	学号
姚佳硕	202100460006
朱浩阳	202100460155
陈顺先	202100460058

一、实验背景与目的

随着区块链和零知识证明技术的快速发展，哈希算法作为密码学基础组件在隐私保护协议中扮演着重要角色。Poseidon2 是近年来提出的一种针对零知识证明场景优化设计的哈希函数，具备高效性和安全性。
本实验旨在基于 Circom 电路语言，实现 Poseidon2 哈希算法电路，使用 Groth16 zkSNARK 方案完成电路证明的生成与验证。通过该实验，加深对密码学电路设计与零知识证明工具链的理解。

二、实验环境与工具

- 操作系统：Windows 10 家庭版 22H2
- 硬件配置：AMD Ryzen 7 5800H，16GB 内存
- 主要软件：
  - Circom 2.0
  - snarkjs
  - Rust 编译器及相关依赖
- 代码仓库：https://github.com/m1kasa3/homework/invitations

三、设计与实现

3.1 Poseidon2 参数选取

根据参考文献（eprint.iacr.org/2023/323.pdf）中 Table 1 的参数说明，本实验采用参数组 `(n,t,d) = (256, 2, 5)`，其中：

- `n=256` 位输入大小
- `t=2` 状态元素数
- `d=5` 非线性层的幂次

3.2 电路实现

电路主体文件为 `circuits/poseidon2_t2.circom`，内嵌完整的轮常量和 MDS 矩阵参数，确保与论文参数一致。电路设计只处理单个 block 输入，输入包括：

- 公开输入：Poseidon2 哈希值
- 隐私输入：哈希原象

设计重点在于有限域上的加法、乘法及非线性 S-box 计算，全部在 Circom 语言中通过有限域元素实现。

3.3 预计算哈希与输入生成

利用 Rust 语言实现的 `rust/poseidon2_hash.rs` 负责对输入数据执行 Poseidon2 哈希计算，输出对应的 JSON 格式输入文件 `inputs/input.json`，供 Circom 电路读取。

四、证明生成与验证

通过 `Makefile` 统一执行电路编译、零知识证明生成和验证流程，主要步骤如下：

1. 编译电路，生成 `.r1cs`、`.wasm` 和符号文件。
2. 使用 Groth16 算法及可信设置文件完成证明生成。
3. 利用生成的验证密钥验证证明的正确性。

五、实验结果

实验步骤	时间（秒）	结果说明
电路编译	10	成功生成 r1cs 文件
输入计算	0.5	Rust 生成 input.json
证明生成	5	成功生成 proof.json
证明验证	0.01	验证通过

电路正确性经多组随机输入测试验证，输出哈希与 Rust 预计算完全一致。

六、实验分析

本次实验验证了基于 Circom 实现复杂密码学哈希函数的可行性。Poseidon2 通过有限域多轮变换实现高安全性，电路设计体现了计算复杂度与证明大小的权衡。
采用 Groth16 方案，有效实现了零知识证明，证明过程稳定且验证高效，满足隐私保护协议对性能和安全的要求。

实验也暴露出目前 Circom 在参数处理和调试时的复杂性，未来工作可尝试自动化参数加载及支持更大输入规模的扩展。

七、总结与展望

本实验完整实现了 Poseidon2 哈希电路及 zkSNARK 证明流程，成功完成零知识证明生成与验证。
未来可进一步结合实际区块链应用场景，优化电路效率及扩展功能，实现更丰富的隐私保护协议。

仓库地址： https://github.com/m1kasa3/homework/invitations