yAudit Spartan-ecdsa Review

Review Resources:

- Spartan-ecdsa

Auditors:

- 0xnagu
- Antonio Viggiano
- Bahurum
- Chen Wen Kang
- garfam
- Igor Line
- lwltea
- nullity
- Oba
- parsley
- Rajesh
- Vincent Owen
- whoismatthewmc

Table of Contents

Review Summary

Spartan-ecdsa

Spartan-ecdsa is a library for proving and verifying ECDSA (secp256k1) signatures in zero-knowledge. Group membership proving time is 10x faster in Spartan-ecdsa compared to efficient-zk-ecdsa, the previous implemenation by Personae Labs. It is developed using the Spartan proof system which does not require trusted setup. However, Spartan uses `secp256k1` curve intead of `curve25519-dalek` in Spartan.

The Spartan-ecdsa circuits, commit 3386b30d9b, were reviewed by 13 auditors between June 19, 2023 and July 5, 2023.

Scope

The scope of the review consisted of the following circuits at commit 3386b30d9b:

- eff_ecdsa.circom
- tree.circom
- add.circom
- double.circom
- mul.circom
- poseidon.circom
- pubkey_membership.circom

After the findings were presented to the Spartan-ecdsa team, fixes were made and included in several PRs.

This review is for identifying potential vulnerabilities in the code. The reviewers did not investigate security practices or operational security and assumed that privileged accounts could be trusted. The reviewers did not evaluate the security of the code relative to a standard or specification. The review may not have identified all potential attack vectors or areas of vulnerability.

yAudit and the auditors make no warranties regarding the security of the code and do not warrant that the code is free from defects. yAudit and the auditors do not represent nor imply to third parties that the code has been audited nor that the code is free from defects. By deploying or using the code, Spartan-ecdsa and users of the circuits agree to use the code at their own risk.

Code Evaluation Matrix

Category	Mark	Description
Access Control	N/A	Spartan-ecdsa is a permissionless protocol, and as such no access control is required
Mathematics	Good	Sage scripts were created to assess the security of some parameters used in the algorithms
Complexity	High	Complexity is reduced compared to previous implementations due to doing right-field arithmetic on secq and eliminating SNARK-unfriendly range checks and big integer math. This led to an overall reduction of R1CS constraints from 1.5M to ~5k.
Libraries	Average	Well-known libraries such as circomlib are used, but Poseidon was custom-implemented with Spartan-ecdsa's own constants since the finite field that Spartan uses isn't supported
Decentralization	Good	Spartan-ecdsa is a permissionless protocol
Cryptography	Good	Spartan-ecdsa operates on the `secp256k1` curve which provides a security level of `128 bits`. It makes use of the Poseidon hash function known for its zk-friendlinesss, simplicity, and resistance against various cryptanalytic attacks. However, it's essential to note that cryptographic algorithms and functions are always subject to ongoing analysis, and new attacks or weaknesses may be discovered in the future.
Code stability	Average	The code was reviewed at a specific commit. The code did not change during the review. However, due to its focus on efficiency, it is likely to change with the addition of features or updates, or to achieve further performance gains.
Documentation	Low	Spartan-ecdsa documentation comprises blog posts from Personae Labs, the Github README documentation, and reference materials from Filecoin and Neptune. It is recommended to aggregate the resources necessary of the protocol under a single repository
Monitoring	N/A	The protocol is intended to be integrated by a dApps who will be responsible for any monitoring needed
Testing and verification	Low	The protocol contains only a few tests for the circuits. During audit, the circom-mutator testing tool was developed for finding potential blind spots in the test coverage of circom projects. The `circom-mutator` tool found that several edge cases were not tested by the project. It is recommended to add more tests to increase test coverage

Findings Explanation

Findings are broken down into sections by their respective Impact:

- Critical, High, Medium, Low Impact
  - These are findings that range from attacks that may cause loss of funds, proof malleability, or cause any unintended consequences/actions that are outside the scope of the requirements
- Informational
  - Findings including Recommendations and best practices

Critical Findings

None.

High Findings

1. High - Input signal s is not constrained in eff_ecdsa.circom

It is possible to submit `s = 0`, `Ux = pubX`, `Uy = pubY` or `s = 0`, `Ux = pubX`, `Uy = -pubY` and get back `(pubX, pubY)`, though this is not a valid signature.

Technical Details

Given check $s \cdot T + U == pubKey$,

$$s * T + U == pubKey$$

$$s = 0, \forall T \in secp256k1$$

$$s * T + U = 0 * T + U = O + U = U == pubKey$$

$$or$$

$$T = 0, \forall s \in secp256k1$$

$$s * T + U = s * 0 + U = O + U = U == pubKey$$

where `U = (pubX, pubY)`. -U would work as well, where `-U = (pubX, -pubY)`. Here is a POC to explain the same.

Impact

High. The missing constraints can be used to generate fake proof.

Recommendation

Add the constraints to the circuit and/or documentation

Developer Response

Acknowledged

Reported by Antonio Viggiano, Igor Line, Oba

2. High - Knowledge of any member signature allow to generate proof of membership

Knowledge of any valid signature by an account stored in the merkle tree allows generating membership proof

Technical Details

There is no check on message supplied by the user. Anyone can submit valid past signatures with arbitrary message hash

Impact

High. The missing constraints can be used to generate fake proof.

Recommendation

Add the constraints to the circuit and/or documentation

Developer Response

Acknowledged

Reported by Antonio Viggiano, Igor Line, Oba

3. High - Under constrained circuits compromising the soundness of the system

In the file mul.circom, the signals `slo` & `shi` are assigned but not constrained.

Technical Details

```
    signal slo <-- s & (2  (128) - 1);
    signal shi <-- s >> 128;
```

Impact

High. Underconstraining allows malicious provers to generate fake proofs.

Developer Response

> *"Adding the line `slo + shi * 2  128 === s;` would fix this, but it turns out that actually, that calculation of `k = (s + tQ) % q` doesn't have to be constrained at all (so the entire template K is unnecessary). Regardless, your discovery made me realize K is unnecessary, which results in solid constraint count reduction!"*

Reported by nullity

4. High - X, Y pair may be an invalid point on the curve

Circuits do not check whether the point (x, y) is on the curve E.

Technical Details

The pair (x, y) forms a group G of order N under $E(\mathbb{F}_p)/\mathcal{P}$ where E represents an elliptic curve, $x, y < P, \mathbb{F}_p$ denotes a finite field, and $\mathcal{P}$ represents the prime order of the base point. There is no check validating that $(x, y) \in G$.

Impact

User may provide a public key (which is just a point (x, y)) that is not a valid point on the curve. This may leak the private key if the point is chosen from small order N' of another curve C'

Recommendation

Validate the given point (x, y) outside of the circuit.

Developer Response

Acknowledged

Reported by [Rajesh]

Medium Findings

None.

Low Findings

1. Low - Unchecked edge case in complete addition

`Secp256k1AddComplete()` returns an incorrect value when `yP + yQ = 1`.

Technical Details

`zeroizeA.out` should be 0 when `P` and `Q` are different points, but when `xP != xQ` and `yP + yQ = 1` it would be 1.

In this case the output point would be the point at infinity instead of the actual sum.

Impact

Low. secp256k1 arithmetics is incorrect in some edge cases.

Recommendation

Document the proof that when $yP + yQ = 1$, the points P and Q either do not exist on the curve or are highly improbable to occur.

If this can't be done, then add a `isYEqual` component as done for `X` and use `AND()` instead of `IsEqual()`

```
    component zeroizeA = AND();
    zeroizeA.in[0] <== isXEqual.out;
    zeroizeA.in[1] <== isYEqual.out;
```

There should be similar informational warnings to the client implementations for many edge cases like zero point, points at infinity, additions/multiplications with p & $-p$

Developer Response

Acknowledged

Reported by Bahurum, 0xnagu

Informational Findings

1. Informational - Over-allocation of circom components

In mul.circom:Secp256k1Mul, the value `accIncomplete` and `PComplete` are over-allocated.

Technical Details

In mul.circom:Secp256k1Mul, the value `accIncomplete` and `PComplete` are over-allocated.

```
    component accIncomplete[bits];
    // ...
    component PComplete[bits-3];
```

Impact

Optimization.

Recommendation

Reduce the allocation of these component arrays to `accIncomplete[bits-p3]` and `PIncomplete[3]`.

Developer Response

Acknowledged

Reported by Antonio Viggiano, Igor Line, Oba, nullity, parsley

2. Informational - Check if the input scalar is within the valid range

Technical Details

Add assertions and constraints to check for invalid inputs and edge cases

Impact

Informational.

Recommendation

Add a constraint to ensure that the input scalar is within the valid range of the secp256k1 elliptic curve. You can do this by adding an assertion to check if the scalar is less than the curve's order.

```
// Add this line after the signal input scalar declaration
assert(scalar <
0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEBAAEDCE6AF48A03BBFD25E8CD0364141);
```

Developer Response

Acknowledged

Reported by Oxnagu

3. Informational - Unused value `bits`

Technical Details

In `eff_ecdsa.circom`, the value `bits` is assigned but never read.

Impact

Informational.

Recommendation

Remove the unused value.

Developer Response

Acknowledged

Reported by Antonio Viggiano, Igor Line, Oba, garfam, parsley, Bahurum, lwltea

4. Informational - No constraints on input signals

Technical Details

There are no constraints on input signals in any of the circuits (presumably to reduce the number of constraints to a bare minimum). This could potentially cause issues for third party developers integrating Spartan-ECDSA.

Impact

Informational.

Recommendation

In order to keep the number of constraints to a minimum, simply document the absence of input signal constraints clearly and suggest that they be validated in the application code.

Developer Response

Acknowledged

Reported by whoismatthewmc

5. Informational - Missing & Extra Imports in `eff_ecdsa.circom`

Technical Details

The `add.circom` import is missing in `eff_ecdsa.circom`. The `bitify.circom` is imported in `eff_ecdsa.circom` but not used.

Impact

Informational. This is not an issue as `add.circom` is imported in `mul.circom` which is in turn imported in `eff_ecdsa.circom`.

Recommendation

But recommendation is to explicitly import like `include "./secp256k1/add.circom";` & remove `bitify.circom` import.

Developer Response

Acknowledged

Reported by lwltea, Vincent Owen

6. Informational - Constraints for add.cicom for values to be non-zero

In signal assignments containing division, the divisor needs to be constrained to be non-zero.

Technical Details

```
   |
31 |     lambda <-- dy / dx;
   |                     ^^ The divisor `dx` must be constrained to be non-zero.
```

Impact

Informational.

Recommendation

Do an additional check for non-zero values.

Developer Response

Acknowledged

Reported by Chen Wen Kang, Vincent Owen

7. Informational - More tests for the circuits

Additional tests are always good to have in order to cover more unexpected cases.

Technical Details

`eff_ecdsa.test.ts` and `eff_ecdsa_to_addr.test.ts` only have 1 positive tests.

Impact

Informational.

Recommendation

Adding more tests for the circuits.

Developer Response

Acknowledged

Reported by Chen Wen Kang, Vincent Owen

Final remarks

- The Spartan-ecdsa circuits assume that the underlying hash function (Poseidon) is:
    - Collision-resistant
    - Resistant to differential, algebraic, and interpolation attacks
    - Behaves as a random oracle
- The Merkle tree used for membership proof is assumed to be secure against second-preimage attacks.
- Social engineering attacks are still a valid way to break the system. ECDSA has several nonce based attacks. It is very important that the client side confirguration doesn't leak any nonce data or any app metadata that can reduce the security of guessing nonce for the ECDSA.
- We recommend clarifying the proper usage of each template, where assertions about the valuation of its inputs (pre-conditions) should be satisfied when calling the template.
- We recommend writing a checklist to be ensured on the client side. This can help dApp developers avoid common mistakes such as missing validation of inputs which can lead to soundness bugs.
- Overall, the code demonstrates good implementation of mathematical operations and basic functionality. However, it could benefit from more documentation and tests.

Automated program analysis tools

Over the course of the audit, in addition to a manual review of the code, we applied different automated program analysis tools and evaluated their output.

1. circomspect
2. Picus
3. Ecne
4. circom-mutator

A few things to note on results from Picus and Ecne:

- Ecne can output false positive. It relies on static analysis but [does not](#) call an SMT solver to concretely find the potential attack vector.
- Picus does not output false positive. Its produces one of the following outputs: `safe` : Picus did not find underconstrained bugs `unsafe` : Picus found an underconstrained bug. It may also output the attack vector. `unknown` : Picus cannot get a result within the given time limit. Manual review is necessary. The time limit for the solver can also be increased.
- Note that two solvers that use different theories are available for Picus: z3 and cvc5. Picus and some of its libraries are still in development and could cointain bugs. In case of conflicting results, `safe` for one and `unsafe` for the other, the only way to know which one is correct would be to manually verify the counter example produced by the solver that reported `unsafe`.

Results

1. Circomspect

[circomspect : A static analyzer and linter for the Circom zero-knowledge DSL](#)

```
circomspect: analyzing template 'SBox'
circomspect: analyzing template 'Secp256k1Mul'
circomspect: analyzing template 'AddRoundConst'
circomspect: analyzing template 'EfficientECDSA'
warning: The variable `bits` is assigned a value, but this value is never read.
   ┌─ packages/circuits/eff_ecdsa_membership/eff_ecdsa.circom:14:5
   │
14 │     var bits = 256;
   │     ^^^^^^^^^^^^^^ The value assigned to `bits` here is never read.
   │
   = For more details, see
https://github.com/trailofbits/circomspect/blob/main/doc/analysis_passes.md#unused-variable-or-parameter.

circomspect: analyzing template 'MatrixMul'
circomspect: analyzing template 'K'
warning: Using the signal assignment operator `<--` does not constrain the
assigned signal.
    ┌─ packages/circuits/eff_ecdsa_membership/secp256k1/mul.circom:123:5
    │
123 │     signal slo <-- s & (2 ** (128) - 1);
```

```
    │     ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ The assigned signal `slo` is not constrained here.
    ·
129 │     inBits.in <== slo + tQlo;
    │     ------------------------ The signal `slo` is constrained here.
    ·
144 │     signal alo <== slo + tQlo - (carry * 2 ** 128);
    │     ---------------------------------------------- The signal `slo` is constrained here.
    ·
170 │     theta.in[1] <== slo + tQlo;
    │     -------------------------- The signal `slo` is constrained here.
    ·
177 │     signal klo <== (slo + tQlo + borrow.out * (2 ** 128)) - isQuotientOne.out * qlo;
    │     ------------------------------------------------------------------------------- The signal `slo` is constrained here.
    │
    = For more details, see https://github.com/trailofbits/circomspect/blob/main/doc/analysis_passes.md#signal-assignment.

warning: Using the signal assignment operator `<--` does not constrain the assigned signal.
    ┌─ packages/circuits/eff_ecdsa_membership/secp256k1/mul.circom:124:5
    │
124 │     signal shi <-- s >> 128;
    │     ^^^^^^^^^^^^^^^^^^^^^^^ The assigned signal `shi` is not constrained here.
    ·
142 │     signal ahi <== shi + tQhi + carry;
    │     --------------------------------- The signal `shi` is constrained here.
    ·
178 │     signal khi <== (shi + tQhi - borrow.out * 1)  - isQuotientOne.out * qhi;
    │     ----------------------------------------------------------------------- The signal `shi` is constrained here.
    │
    = For more details, see https://github.com/trailofbits/circomspect/blob/main/doc/analysis_passes.md#signal-assignment.

warning: Using `Num2Bits` to convert field elements to bits may lead to aliasing
```

```
issues.
    ┌─ packages/circuits/eff_ecdsa_membership/secp256k1/mul.circom:180:25
    │
180 │     component kloBits = Num2Bits(256);
    │                         ^^^^^^^^^^^^^ Circomlib template `Num2Bits`
instantiated here.
    │
    = Consider using `Num2Bits_strict` if the input size may be >= than the prime
size.
    = For more details, see
https://github.com/trailofbits/circomspect/blob/main/doc/analysis_passes.md#non-
strict-binary-conversion.

warning: Using `Num2Bits` to convert field elements to bits may lead to aliasing
issues.
    ┌─ packages/circuits/eff_ecdsa_membership/secp256k1/mul.circom:183:25
    │
183 │     component khiBits = Num2Bits(256);
    │                         ^^^^^^^^^^^^^ Circomlib template `Num2Bits`
instantiated here.
    │
    = Consider using `Num2Bits_strict` if the input size may be >= than the prime
size.
    = For more details, see
https://github.com/trailofbits/circomspect/blob/main/doc/analysis_passes.md#non-
strict-binary-conversion.

circomspect: analyzing template 'Poseidon'
circomspect: analyzing template 'Secp256k1Double'
warning: Using the signal assignment operator `<--` does not constrain the
assigned signal.
   ┌─ packages/circuits/eff_ecdsa_membership/secp256k1/double.circom:22:5
   │
22 │     lambda <-- (3 * xPSquared) / (2 * yP);
   │     ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ The assigned signal `lambda` is
not constrained here.
23 │     lambda * 2 * yP === 3 * xPSquared;
   │     --------------------------------- The signal `lambda` is constrained
here.
24 │
25 │     outX <== lambda * lambda - (2 * xP);
   │     ----------------------------------- The signal `lambda` is constrained
here.
```

```
26 │     outY <== lambda * (xP - outX) - yP;
   │     ---------------------------------- The signal `lambda` is constrained
here.
   │
   = For more details, see
https://github.com/trailofbits/circomspect/blob/main/doc/analysis_passes.md#signa
l-assignment.

warning: In signal assignments containing division, the divisor needs to be
constrained to be non-zero
   ┌─ packages/circuits/eff_ecdsa_membership/secp256k1/double.circom:22:35
   │
22 │     lambda <-- (3 * xPSquared) / (2 * yP);
   │                                   ^^^^^^ The divisor `(2 * yP)` must be
constrained to be non-zero.
   │
   = For more details, see
https://github.com/trailofbits/circomspect/blob/main/doc/analysis_passes.md#uncon
strained-division.

circomspect: analyzing template 'FullRound'
circomspect: analyzing template 'PartialRound'
circomspect: analyzing template 'PubKeyMembership'
circomspect: analyzing template 'MerkleTreeInclusionProof'
circomspect: 7 issues found.
```

2. Picus

[Picus : Automated verification of uniqueness property for ZKP circuits]

Command:
`racket ./test-v3-uniqueness.rkt --r1cs <file>.r1cs --timeout 3000 --smt --solver {z3, cvc5}`

Circuits were compiled with the following command to compare results with Ecne:
`circom <file>.circom --r1cs --O0 --sym`

- tree.circom

```
solver z3
# strong uniqueness: safe.
# weak uniqueness: safe.

solver cvc5
# strong uniqueness: safe.
# weak uniqueness: safe.
```

- add.circom

```
Add incomplete

solver z3
# strong uniqueness: safe.
# weak uniqueness: safe.

solver cvc5
# strong uniqueness: unknown.
# weak uniqueness: unknown.
Add complete

solver z3
# strong uniqueness: safe.
# weak uniqueness: safe.

solver cvc5
# strong uniqueness: unknown.
# weak uniqueness: unknown.
```

- double.circom

```
solver z3
# strong uniqueness: safe.
# weak uniqueness: safe.

solver cvc5
# strong uniqueness: unsafe.
# weak uniqueness: unsafe.
```

- poseidon.circom

```
solver z3
# strong uniqueness: safe.
# weak uniqueness: safe.

solver cvc5
# strong uniqueness: safe.
# weak uniqueness: safe.
```

- pubkey_membership.circom, mul.circom, eff_ecdsa.circom: intractable to run (too many constraints)

3. Ecne

[Ecne: An engine for verifying the soundness of R1CS constraints]()

- eff_ecdsa.circom

```
time to prep inputs 729 milliseconds
setup solver 385 milliseconds
Solved for 300 variables out of 6285 total variables
Solved for 0 target variables out of 2 total target variables
------ Bad Constraints ------
...
R1CS function eff_ecdsa has potentially unsound constraints
```

- tree.circom

```
time to prep inputs 128 milliseconds
setup solver 166 milliseconds
Solved for 1479 variables out of 1479 total variables
Solved for 1 target variables out of 1 total target variables
------ Bad Constraints ------
...
R1CS function tree has sound constraints (No trusted functions needed!)
```

- add.circom

```
time to prep inputs 91 milliseconds
setup solver 171 milliseconds
Solved for 6 variables out of 9 total variables
Solved for 0 target variables out of 2 total target variables
------ Bad Constraints ------
...
R1CS function addIncomplete has potentially unsound constraints
time to prep inputs 92 milliseconds
setup solver 178 milliseconds
Solved for 34 variables out of 48 total variables
Solved for 0 target variables out of 2 total target variables
------ Bad Constraints ------
...
R1CS function addcomplete has potentially unsound constraints
```

- double.circom

```
time to prep inputs 56 milliseconds
setup solver 115 milliseconds
Solved for 3 variables out of 6 total variables
Solved for 0 target variables out of 2 total target variables
------ Bad Constraints ------
...
R1CS function double has potentially unsound constraints
```

- mul.circom

```
time to prep inputs 478 milliseconds
setup solver 248 milliseconds
Solved for 289 variables out of 6231 total variables
Solved for 0 target variables out of 2 total target variables
------ Bad Constraints ------
...
R1CS function mul has potentially unsound constraints
time to prep inputs 144 milliseconds
setup solver 212 milliseconds
Solved for 10 variables out of 1336 total variables
Solved for 0 target variables out of 256 total target variables
------ Bad Constraints ------
```

```
...
R1CS function k has potentially unsound constraints
```

- poseidon.circom

```
time to prep inputs 208 milliseconds
setup solver 262 milliseconds
Solved for 1479 variables out of 1479 total variables
Solved for 1 target variables out of 1 total target variables
------ Bad Constraints ------
...
R1CS function poseidon has sound constraints (No trusted functions needed!)
```

- pubkey_membership.circom

```
time to prep inputs 2626 milliseconds
setup solver 651 milliseconds
Solved for 640 variables out of 37572 total variables
Solved for 0 target variables out of 0 total target variables
------ Bad Constraints ------
...
R1CS function pubkeymembership has sound constraints (No trusted functions needed!)
```

4. Circom-Mutator

The circom-mutator was developed during the review. It intended to help find blind spots in the test coverage of circom projects.

`circom-mutator` works by injecting bugs into existing code in order to generate ”mutants”. The output is then compared with that of existing tests. Mutanted circuits should ideally make tests fail. In many cases, an injected bug will not necessarily mean that the circuit is vulnerable, but rather that the test coverage can be improved, or that the circuit accepts these edge cases but proper validation should be performed elsewhere (such as in the application layer).

The tool can be used either as a CLI (`npx circom-mutator <file>`) or by adding it to the project and calling the `testMutations` helper function on the jest test files.

```
FAIL  tests/eff_ecdsa.test.ts
  ● Console

    console.log
      (AssignedButNotConstrained)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      26c26
      <     sMultT.scalar <== s;
      ---
      >     sMultT.scalar <-- s;

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

    console.log
      (Secp256k1Add)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      31c31
      <     component pubKey = Secp256k1AddComplete();
      ---
      >     component pubKey = Secp256k1AddIncomplete();

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

  ● [mutation] eff_ecdsa › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).toThrow()

    Received function did not throw

      63 |     });
      64 |
    > 65 |     expect(() => circuit.checkConstraints(w)).toThrow();
         |                                               ^
      66 |   }
      67 | });
      68 |

      at tests/eff_ecdsa.test.ts:65:47
      at fulfilled (tests/eff_ecdsa.test.ts:5:58)

  ● [mutation] eff_ecdsa › [mutation] Replace Secp256k1AddComplete by
Secp256k1AddIncomplete. (Secp256k1Add)

    expect(received).toThrow()

    Received function did not throw

      63 |     });
      64 |
    > 65 |     expect(() => circuit.checkConstraints(w)).toThrow();
         |                                               ^
      66 |   }
      67 | });
      68 |

      at tests/eff_ecdsa.test.ts:65:47
      at fulfilled (tests/eff_ecdsa.test.ts:5:58)

 FAIL  tests/poseidon.test.ts (7.047 s)
  ● Console

    console.log
      (AssignedButNotConstrained)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      9c9
      <     signal inDouble <== in * in;
      ---
      >     signal inDouble <-- in * in;

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

    console.log
      (AssignedButNotConstrained)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19
```

```
console.log
  9,10c9,10
  <     signal inDouble <-- in * in;
  <     signal inQuadruple <== inDouble * inDouble;
  ---
  >     signal inDouble <== in * in;
  >     signal inQuadruple <-- inDouble * inDouble;

  at ../../node_modules/circom-mutator/src/tester.ts:52:19

console.log
  (AssignedButNotConstrained)

  at ../../node_modules/circom-mutator/src/tester.ts:51:19

console.log
  10c10
  <     signal inQuadruple <-- inDouble * inDouble;
  ---
  >     signal inQuadruple <== inDouble * inDouble;
  13c13
  <     out <== inQuadruple * in;
  ---
  >     out <-- inQuadruple * in;

  at ../../node_modules/circom-mutator/src/tester.ts:52:19

console.log
  (AssignedButNotConstrained)

  at ../../node_modules/circom-mutator/src/tester.ts:51:19

console.log
  13c13
  <     out <-- inQuadruple * in;
  ---
  >     out <== inQuadruple * in;
  27c27
  <         out[i] <== tmp;
  ---
  >         out[i] <-- tmp;
```

```
      at ../../node_modules/circom-mutator/src/tester.ts:52:19

    console.log
      (AssignedButNotConstrained)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      27c27
      <         out[i] <-- tmp;
      ---
      >         out[i] <== tmp;
      38c38
      <         out[i] <== state[i] + round_keys[pos + i];
      ---
      >         out[i] <-- state[i] + round_keys[pos + i];

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

    console.log
      (AssignedButNotConstrained)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      38c38
      <         out[i] <-- state[i] + round_keys[pos + i];
      ---
      >         out[i] <== state[i] + round_keys[pos + i];
      48c48
      <         constAdded.state[i] <== state[i];
      ---
      >         constAdded.state[i] <-- state[i];

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

    console.log
      (AssignedButNotConstrained)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      48c48
```

```
  <         constAdded.state[i] <-- state[i];
  ---
  >         constAdded.state[i] <== state[i];
  55c55
  <         sBoxes[i].in <== constAdded.out[i];
  ---
  >         sBoxes[i].in <-- constAdded.out[i];

  at ../../node_modules/circom-mutator/src/tester.ts:52:19

console.log
  (AssignedButNotConstrained)

  at ../../node_modules/circom-mutator/src/tester.ts:51:19

console.log
  55c55
  <         sBoxes[i].in <-- constAdded.out[i];
  ---
  >         sBoxes[i].in <== constAdded.out[i];
  60c60
  <         matrixMul.state[i] <== sBoxes[i].out;
  ---
  >         matrixMul.state[i] <-- sBoxes[i].out;

  at ../../node_modules/circom-mutator/src/tester.ts:52:19

console.log
  (AssignedButNotConstrained)

  at ../../node_modules/circom-mutator/src/tester.ts:51:19

console.log
  60c60
  <         matrixMul.state[i] <-- sBoxes[i].out;
  ---
  >         matrixMul.state[i] <== sBoxes[i].out;
  64c64
  <         out[i] <== matrixMul.out[i];
  ---
  >         out[i] <-- matrixMul.out[i];

  at ../../node_modules/circom-mutator/src/tester.ts:52:19
```

```
console.log
  (AssignedButNotConstrained)

  at ../../node_modules/circom-mutator/src/tester.ts:51:19

console.log
  64c64
  <         out[i] <-- matrixMul.out[i];
  ---
  >         out[i] <== matrixMul.out[i];
  75c75
  <         constAdded.state[i] <== state[i];
  ---
  >         constAdded.state[i] <-- state[i];

  at ../../node_modules/circom-mutator/src/tester.ts:52:19

console.log
  (AssignedButNotConstrained)

  at ../../node_modules/circom-mutator/src/tester.ts:51:19

console.log
  75c75
  <         constAdded.state[i] <-- state[i];
  ---
  >         constAdded.state[i] <== state[i];
  79c79
  <     sBox.in <== constAdded.out[0];
  ---
  >     sBox.in <-- constAdded.out[0];

  at ../../node_modules/circom-mutator/src/tester.ts:52:19

console.log
  (AssignedButNotConstrained)

  at ../../node_modules/circom-mutator/src/tester.ts:51:19

console.log
  79c79
  <     sBox.in <-- constAdded.out[0];
```

```
      ---
      >     sBox.in <== constAdded.out[0];
      84c84
      <             matrixMul.state[i] <== sBox.out;
      ---
      >             matrixMul.state[i] <-- sBox.out;

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

    console.log
      (AssignedButNotConstrained)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      84c84
      <             matrixMul.state[i] <-- sBox.out;
      ---
      >             matrixMul.state[i] <== sBox.out;
      86c86
      <             matrixMul.state[i] <== constAdded.out[i];
      ---
      >             matrixMul.state[i] <-- constAdded.out[i];

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

    console.log
      (AssignedButNotConstrained)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      86c86
      <             matrixMul.state[i] <-- constAdded.out[i];
      ---
      >             matrixMul.state[i] <== constAdded.out[i];
      91c91
      <         out[i] <== matrixMul.out[i];
      ---
      >         out[i] <-- matrixMul.out[i];

      at ../../node_modules/circom-mutator/src/tester.ts:52:19
    console.log
      (AssignedButNotConstrained)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      91c91
      <         out[i] <-- matrixMul.out[i];
      ---
      >         out[i] <== matrixMul.out[i];
      108c108
      <     initState[1] <== inputs[0];
      ---
      >     initState[1] <-- inputs[0];

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

    console.log
      (AssignedButNotConstrained)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      108,109c108,109
      <     initState[1] <-- inputs[0];
      <     initState[2] <== inputs[1];
      ---
      >     initState[1] <== inputs[0];
      >     initState[2] <-- inputs[1];

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

    console.log
      (AssignedButNotConstrained)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      109c109
      <     initState[2] <-- inputs[1];
      ---
      >     initState[2] <== inputs[1];
      116c116
      <                 fRoundsFirst[j].state[i] <== initState[i];
      ---
      >                 fRoundsFirst[j].state[i] <-- initState[i];

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

    console.log
      (AssignedButNotConstrained)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      116c116
      <                 fRoundsFirst[j].state[i] <-- initState[i];
      ---
      >                 fRoundsFirst[j].state[i] <== initState[i];
      120c120
      <                 fRoundsFirst[j].state[i] <== fRoundsFirst[j - 1].out[i];
      ---
      >                 fRoundsFirst[j].state[i] <-- fRoundsFirst[j - 1].out[i];

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

    console.log
      (AssignedButNotConstrained)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      120c120
      <                 fRoundsFirst[j].state[i] <-- fRoundsFirst[j - 1].out[i];
      ---
      >                 fRoundsFirst[j].state[i] <== fRoundsFirst[j - 1].out[i];
      132c132
      <                 pRounds[j].state[i] <== fRoundsFirst[numFullRoundsHalf -
1].out[i];
      ---
      >                 pRounds[j].state[i] <-- fRoundsFirst[numFullRoundsHalf -
1].out[i];

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

    console.log
```

```
      (AssignedButNotConstrained)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      132c132
      <                 pRounds[j].state[i] <-- fRoundsFirst[numFullRoundsHalf -
1].out[i];
      ---
      >                 pRounds[j].state[i] <== fRoundsFirst[numFullRoundsHalf -
1].out[i];
      136c136
      <                 pRounds[j].state[i] <== pRounds[j - 1].out[i];
      ---
      >                 pRounds[j].state[i] <-- pRounds[j - 1].out[i];

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

    console.log
      (AssignedButNotConstrained)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      136c136
      <                 pRounds[j].state[i] <-- pRounds[j - 1].out[i];
      ---
      >                 pRounds[j].state[i] <== pRounds[j - 1].out[i];
      147c147
      <                 fRoundsLast[j].state[i] <== pRounds[numPartialRounds -
1].out[i];
      ---
      >                 fRoundsLast[j].state[i] <-- pRounds[numPartialRounds -
1].out[i];

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

    console.log
      (AssignedButNotConstrained)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
```

```
      147c147
      <                 fRoundsLast[j].state[i] <-- pRounds[numPartialRounds -
1].out[i];
      ---
      >                 fRoundsLast[j].state[i] <== pRounds[numPartialRounds -
1].out[i];
      151c151
      <                 fRoundsLast[j].state[i] <== fRoundsLast[j - 1].out[i];
      ---
      >                 fRoundsLast[j].state[i] <-- fRoundsLast[j - 1].out[i];

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

    console.log
      (AssignedButNotConstrained)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      151c151
      <                 fRoundsLast[j].state[i] <-- fRoundsLast[j - 1].out[i];
      ---
      >                 fRoundsLast[j].state[i] <== fRoundsLast[j - 1].out[i];
      157c157
      <     out <== fRoundsLast[numFullRoundsHalf-1].out[1];
      ---
      >     out <-- fRoundsLast[numFullRoundsHalf-1].out[1];

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

  ● [mutation] poseidon › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      53 |     });
      54 |
    > 55 |     expect(() => circuit.checkConstraints(w)).rejects.toThrow();
         |     ^
      56 |   }
      57 | });
      58 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/poseidon.test.ts:55:5
      at fulfilled (tests/poseidon.test.ts:5:58)

  ● [mutation] poseidon › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      53 |     });
      54 |
    > 55 |     expect(() => circuit.checkConstraints(w)).rejects.toThrow();
         |     ^
      56 |   }
      57 | });
      58 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/poseidon.test.ts:55:5
      at fulfilled (tests/poseidon.test.ts:5:58)

  ● [mutation] poseidon › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      53 |     });
      54 |
    > 55 |     expect(() => circuit.checkConstraints(w)).rejects.toThrow();
         |     ^
      56 |   }
      57 | });
      58 |
      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/poseidon.test.ts:55:5
      at fulfilled (tests/poseidon.test.ts:5:58)

  ● [mutation] poseidon › [mutation] Replace constraints by assignments (AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      53 |     });
      54 |
    > 55 |     expect(() => circuit.checkConstraints(w)).rejects.toThrow();
         |     ^
      56 |   }
      57 | });
      58 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/poseidon.test.ts:55:5
      at fulfilled (tests/poseidon.test.ts:5:58)

  ● [mutation] poseidon › [mutation] Replace constraints by assignments (AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      53 |     });
      54 |
    > 55 |     expect(() => circuit.checkConstraints(w)).rejects.toThrow();
         |     ^
      56 |   }
      57 | });
      58 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/poseidon.test.ts:55:5
      at fulfilled (tests/poseidon.test.ts:5:58)
  ● [mutation] poseidon › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      53 |     });
      54 |
    > 55 |     expect(() => circuit.checkConstraints(w)).rejects.toThrow();
         |     ^
      56 |   }
      57 | });
      58 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/poseidon.test.ts:55:5
      at fulfilled (tests/poseidon.test.ts:5:58)

  ● [mutation] poseidon › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      53 |     });
      54 |
    > 55 |     expect(() => circuit.checkConstraints(w)).rejects.toThrow();
         |     ^
      56 |   }
      57 | });
      58 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/poseidon.test.ts:55:5
      at fulfilled (tests/poseidon.test.ts:5:58)

  ● [mutation] poseidon › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)
```

```
    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      53 |     });
      54 |
    > 55 |     expect(() => circuit.checkConstraints(w)).rejects.toThrow();
         |     ^
      56 |   }
      57 | });
      58 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/poseidon.test.ts:55:5
      at fulfilled (tests/poseidon.test.ts:5:58)

  ● [mutation] poseidon › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      53 |     });
      54 |
    > 55 |     expect(() => circuit.checkConstraints(w)).rejects.toThrow();
         |     ^
      56 |   }
      57 | });
      58 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/poseidon.test.ts:55:5
      at fulfilled (tests/poseidon.test.ts:5:58)

  ● [mutation] poseidon › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()
    Received promise resolved instead of rejected
    Resolved to value: undefined

      53 |     });
      54 |
    > 55 |     expect(() => circuit.checkConstraints(w)).rejects.toThrow();
         |     ^
      56 |   }
      57 | });
      58 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/poseidon.test.ts:55:5
      at fulfilled (tests/poseidon.test.ts:5:58)

  ● [mutation] poseidon › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      53 |     });
      54 |
    > 55 |     expect(() => circuit.checkConstraints(w)).rejects.toThrow();
         |     ^
      56 |   }
      57 | });
      58 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/poseidon.test.ts:55:5
      at fulfilled (tests/poseidon.test.ts:5:58)

  ● [mutation] poseidon › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined
```

```
      53 |     });
      54 |
    > 55 |     expect(() => circuit.checkConstraints(w)).rejects.toThrow();
         |     ^
      56 |   }
      57 | });
      58 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/poseidon.test.ts:55:5
      at fulfilled (tests/poseidon.test.ts:5:58)

  ● [mutation] poseidon › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      53 |     });
      54 |
    > 55 |     expect(() => circuit.checkConstraints(w)).rejects.toThrow();
         |     ^
      56 |   }
      57 | });
      58 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/poseidon.test.ts:55:5
      at fulfilled (tests/poseidon.test.ts:5:58)

  ● [mutation] poseidon › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      53 |     });
      54 |
    > 55 |     expect(() => circuit.checkConstraints(w)).rejects.toThrow();
```

```
         |     ^
      56 |   }
      57 | });
      58 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/poseidon.test.ts:55:5
      at fulfilled (tests/poseidon.test.ts:5:58)

  ● [mutation] poseidon › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      53 |     });
      54 |
    > 55 |     expect(() => circuit.checkConstraints(w)).rejects.toThrow();
         |     ^
      56 |   }
      57 | });
      58 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/poseidon.test.ts:55:5
      at fulfilled (tests/poseidon.test.ts:5:58)

  ● [mutation] poseidon › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      53 |     });
      54 |
    > 55 |     expect(() => circuit.checkConstraints(w)).rejects.toThrow();
         |     ^
      56 |   }
      57 | });
      58 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/poseidon.test.ts:55:5
      at fulfilled (tests/poseidon.test.ts:5:58)

  ● [mutation] poseidon › [mutation] Replace constraints by assignments (AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      53 |     });
      54 |
    > 55 |     expect(() => circuit.checkConstraints(w)).rejects.toThrow();
         |     ^
      56 |   }
      57 | });
      58 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/poseidon.test.ts:55:5
      at fulfilled (tests/poseidon.test.ts:5:58)

  ● [mutation] poseidon › [mutation] Replace constraints by assignments (AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      53 |     });
      54 |
    > 55 |     expect(() => circuit.checkConstraints(w)).rejects.toThrow();
         |     ^
      56 |   }
      57 | });
      58 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/poseidon.test.ts:55:5
      at fulfilled (tests/poseidon.test.ts:5:58)

  ● [mutation] poseidon › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      53 |     });
      54 |
    > 55 |     expect(() => circuit.checkConstraints(w)).rejects.toThrow();
         |     ^
      56 |   }
      57 | });
      58 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/poseidon.test.ts:55:5
      at fulfilled (tests/poseidon.test.ts:5:58)

  ● [mutation] poseidon › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      53 |     });
      54 |
    > 55 |     expect(() => circuit.checkConstraints(w)).rejects.toThrow();
         |     ^
      56 |   }
      57 | });
      58 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/poseidon.test.ts:55:5
      at fulfilled (tests/poseidon.test.ts:5:58)
  ● [mutation] poseidon › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      53 |     });
      54 |
    > 55 |     expect(() => circuit.checkConstraints(w)).rejects.toThrow();
         |     ^
      56 |   }
      57 | });
      58 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/poseidon.test.ts:55:5
      at fulfilled (tests/poseidon.test.ts:5:58)

  ● [mutation] poseidon › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      53 |     });
      54 |
    > 55 |     expect(() => circuit.checkConstraints(w)).rejects.toThrow();
         |     ^
      56 |   }
      57 | });
      58 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/poseidon.test.ts:55:5
      at fulfilled (tests/poseidon.test.ts:5:58)

  ● [mutation] poseidon › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)
```

```
   expect(received).rejects.toThrow()

   Received promise resolved instead of rejected
   Resolved to value: undefined

     53 |     });
     54 |
   > 55 |     expect(() => circuit.checkConstraints(w)).rejects.toThrow();
        |     ^
     56 |   }
     57 | });
     58 |

     at expect (../../node_modules/expect/build/index.js:105:15)
     at tests/poseidon.test.ts:55:5
     at fulfilled (tests/poseidon.test.ts:5:58)

FAIL  tests/pubkey_membership.test.ts (18.541 s)
 ● Console

   console.log
     (AssignedButNotConstrained)

     at ../../node_modules/circom-mutator/src/tester.ts:51:19

   console.log
     28c28
     <     ecdsa.Tx <== Tx;
     ---
     >     ecdsa.Tx <-- Tx;

     at ../../node_modules/circom-mutator/src/tester.ts:52:19

   console.log
     (AssignedButNotConstrained)

     at ../../node_modules/circom-mutator/src/tester.ts:51:19

   console.log
     28,29c28,29
     <     ecdsa.Tx <-- Tx;
     <     ecdsa.Ty <== Ty;
     ---
```

```
      >     ecdsa.Tx <== Tx;
      >     ecdsa.Ty <-- Ty;

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

    console.log
      (AssignedButNotConstrained)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      29,30c29,30
      <     ecdsa.Ty <-- Ty;
      <     ecdsa.Ux <== Ux;
      ---
      >     ecdsa.Ty <== Ty;
      >     ecdsa.Ux <-- Ux;

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

    console.log
      (AssignedButNotConstrained)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      30,31c30,31
      <     ecdsa.Ux <-- Ux;
      <     ecdsa.Uy <== Uy;
      ---
      >     ecdsa.Ux <== Ux;
      >     ecdsa.Uy <-- Uy;

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

    console.log
      (AssignedButNotConstrained)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      31,32c31,32
      <     ecdsa.Uy <-- Uy;
      <     ecdsa.s <== s;
      ---
      >     ecdsa.Uy <== Uy;
      >     ecdsa.s <-- s;

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

    console.log
      (AssignedButNotConstrained)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      32c32
      <     ecdsa.s <-- s;
      ---
      >     ecdsa.s <== s;
      35c35
      <     pubKeyHash.inputs[0] <== ecdsa.pubKeyX;
      ---
      >     pubKeyHash.inputs[0] <-- ecdsa.pubKeyX;

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

    console.log
      (AssignedButNotConstrained)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      35,36c35,36
      <     pubKeyHash.inputs[0] <-- ecdsa.pubKeyX;
      <     pubKeyHash.inputs[1] <== ecdsa.pubKeyY;
      ---
      >     pubKeyHash.inputs[0] <== ecdsa.pubKeyX;
      >     pubKeyHash.inputs[1] <-- ecdsa.pubKeyY;

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

    console.log
      (AssignedButNotConstrained)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19
```

```
console.log
  36c36
  <     pubKeyHash.inputs[1] <-- ecdsa.pubKeyY;
  ---
  >     pubKeyHash.inputs[1] <== ecdsa.pubKeyY;
  39c39
  <     merkleProof.leaf <== pubKeyHash.out;
  ---
  >     merkleProof.leaf <-- pubKeyHash.out;

  at ../../node_modules/circom-mutator/src/tester.ts:52:19

console.log
  (AssignedButNotConstrained)

  at ../../node_modules/circom-mutator/src/tester.ts:51:19

console.log
  39c39
  <     merkleProof.leaf <-- pubKeyHash.out;
  ---
  >     merkleProof.leaf <== pubKeyHash.out;
  42c42
  <         merkleProof.pathIndices[i] <== pathIndices[i];
  ---
  >         merkleProof.pathIndices[i] <-- pathIndices[i];

  at ../../node_modules/circom-mutator/src/tester.ts:52:19

console.log
  (AssignedButNotConstrained)

  at ../../node_modules/circom-mutator/src/tester.ts:51:19

console.log
  42,43c42,43
  <         merkleProof.pathIndices[i] <-- pathIndices[i];
  <         merkleProof.siblings[i] <== siblings[i];
  ---
  >         merkleProof.pathIndices[i] <== pathIndices[i];
  >         merkleProof.siblings[i] <-- siblings[i];
```

```
      at ../../node_modules/circom-mutator/src/tester.ts:52:19

    console.log
      (MissingOutputCheckConstraint)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      45c45
      <     root === merkleProof.root;
      ---
      >

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

  ● [mutation] pubkey_membership › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      135 |     const w = await circuit.calculateWitness(input, true);
      136 |
    > 137 |     expect(async () => await
circuit.checkConstraints(w)).rejects.toThrow();
          |     ^
      138 |   }
      139 | });
      140 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/pubkey_membership.test.ts:137:5
      at fulfilled (tests/pubkey_membership.test.ts:5:58)

  ● [mutation] pubkey_membership › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined
```

```
      135 |     const w = await circuit.calculateWitness(input, true);
      136 |
    > 137 |     expect(async () => await
circuit.checkConstraints(w)).rejects.toThrow();
          |     ^
      138 |   }
      139 | });
      140 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/pubkey_membership.test.ts:137:5
      at fulfilled (tests/pubkey_membership.test.ts:5:58)

  ● [mutation] pubkey_membership › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      135 |     const w = await circuit.calculateWitness(input, true);
      136 |
    > 137 |     expect(async () => await
circuit.checkConstraints(w)).rejects.toThrow();
          |     ^
      138 |   }
      139 | });
      140 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/pubkey_membership.test.ts:137:5
      at fulfilled (tests/pubkey_membership.test.ts:5:58)

  ● [mutation] pubkey_membership › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined
```

```
      135 |     const w = await circuit.calculateWitness(input, true);
      136 |
    > 137 |     expect(async () => await
circuit.checkConstraints(w)).rejects.toThrow();
          |     ^
      138 |   }
      139 | });
      140 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/pubkey_membership.test.ts:137:5
      at fulfilled (tests/pubkey_membership.test.ts:5:58)

  ● [mutation] pubkey_membership › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      135 |     const w = await circuit.calculateWitness(input, true);
      136 |
    > 137 |     expect(async () => await
circuit.checkConstraints(w)).rejects.toThrow();
          |     ^
      138 |   }
      139 | });
      140 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/pubkey_membership.test.ts:137:5
      at fulfilled (tests/pubkey_membership.test.ts:5:58)

  ● [mutation] pubkey_membership › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      135 |     const w = await circuit.calculateWitness(input, true);
      136 |
    > 137 |     expect(async () => await
circuit.checkConstraints(w)).rejects.toThrow();
          |     ^
      138 |   }
      139 | });
      140 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/pubkey_membership.test.ts:137:5
      at fulfilled (tests/pubkey_membership.test.ts:5:58)

  ● [mutation] pubkey_membership › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      135 |     const w = await circuit.calculateWitness(input, true);
      136 |
    > 137 |     expect(async () => await
circuit.checkConstraints(w)).rejects.toThrow();
          |     ^
      138 |   }
      139 | });
      140 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/pubkey_membership.test.ts:137:5
      at fulfilled (tests/pubkey_membership.test.ts:5:58)

  ● [mutation] pubkey_membership › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      135 |     const w = await circuit.calculateWitness(input, true);
      136 |
    > 137 |     expect(async () => await
circuit.checkConstraints(w)).rejects.toThrow();
          |     ^
      138 |   }
      139 | });
      140 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/pubkey_membership.test.ts:137:5
      at fulfilled (tests/pubkey_membership.test.ts:5:58)

  ● [mutation] pubkey_membership › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      135 |     const w = await circuit.calculateWitness(input, true);
      136 |
    > 137 |     expect(async () => await
circuit.checkConstraints(w)).rejects.toThrow();
          |     ^
      138 |   }
      139 | });
      140 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/pubkey_membership.test.ts:137:5
      at fulfilled (tests/pubkey_membership.test.ts:5:58)

  ● [mutation] pubkey_membership › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      135 |     const w = await circuit.calculateWitness(input, true);
      136 |
    > 137 |     expect(async () => await
circuit.checkConstraints(w)).rejects.toThrow();
          |     ^
      138 |   }
      139 | });
      140 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/pubkey_membership.test.ts:137:5
      at fulfilled (tests/pubkey_membership.test.ts:5:58)

  ● [mutation] pubkey_membership › [mutation] Arithmetic Over/Under Flows
(Num2Bits)

    expect(received).rejects.toThrow()

    Received promise resolved instead of rejected
    Resolved to value: undefined

      135 |     const w = await circuit.calculateWitness(input, true);
      136 |
    > 137 |     expect(async () => await
circuit.checkConstraints(w)).rejects.toThrow();
          |     ^
      138 |   }
      139 | });
      140 |

      at expect (../../node_modules/expect/build/index.js:105:15)
      at tests/pubkey_membership.test.ts:137:5
      at fulfilled (tests/pubkey_membership.test.ts:5:58)

 FAIL  tests/eff_ecdsa_to_addr.test.ts (42.852 s)
  ● Console

    console.log
      (AssignedButNotConstrained)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      22c22
      <     effEcdsa.s <== s;
      ---
      >     effEcdsa.s <-- s;

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

    console.log
      (Num2Bits)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      28,29d27
      <     component pubKeyXBits = Num2Bits(256);
      <     pubKeyXBits.in <== effEcdsa.pubKeyX;
      30a29,30
      >
      >
      38c38
      <         pubToAddr.pubkeyBits[i + 256] <== pubKeyXBits.out[i];
      ---
      >

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

  ● [mutation] eff_ecdsa_to_addr › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).toThrow()

    Received function did not throw

      70 |     });
      71 |
    > 72 |     expect(() => circuit.checkConstraints(w)).toThrow();
         |                                               ^
      73 |   }
      74 | });
      75 |

      at tests/eff_ecdsa_to_addr.test.ts:72:47
      at fulfilled (tests/eff_ecdsa_to_addr.test.ts:5:58)

  ● [mutation] eff_ecdsa_to_addr › [mutation] Arithmetic Over/Under Flows
(Num2Bits)
```

```
    assert.strictEqual(received, expected)

    Expected value to strictly be equal to:
      undefined
    Received:
      null

    Message:
      circom compiler error
    Error: Command failed: circom --wasm --sym --r1cs --output
/var/folders/pb/g_b_19n15hn4gkl0cvjjlzfm0000gn/T/circom_-64331-7x80QurfBYCb --
prime secq256k1 packages/circuits/tests/circuits/eff_ecdsa_to_addr_test.circom
    error[T3001]: Exception caused by invalid access
       ┌─ "packages/circuits/eff_ecdsa_membership/eff_ecdsa_to_addr.circom":41:14
       │
    41 │     addr <== pubToAddr.address;
       │              ^^^^^^^^^^^^^^^^^ found here
       │
       = call trace:
         ->EfficientECDSAToAddr

    previous errors were found



    Difference:

      Comparing two different types of values. Expected undefined but received
null.

      at compile (../../node_modules/circom_tester/wasm/tester.js:91:2)
      at wasm_tester (../../node_modules/circom_tester/wasm/tester.js:45:2)

 FAIL  tests/addr_membership.test.ts (63.911 s)
  ● Console

    console.log
      (AssignedButNotConstrained)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      29c29
      <     effEcdsa.Tx <== Tx;
      ---
      >     effEcdsa.Tx <-- Tx;

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

    console.log
      (MissingOutputCheckConstraint)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      56c56
      <     root === merkleProof.root;
      ---
      >

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

    console.log
      (Num2Bits)

      at ../../node_modules/circom-mutator/src/tester.ts:51:19

    console.log
      35,36d34
      <     component pubKeyXBits = Num2Bits(256);
      <     pubKeyXBits.in <== effEcdsa.pubKeyX;
      37a36,37
      >
      >
      45c45
      <         pubToAddr.pubkeyBits[i + 256] <== pubKeyXBits.out[i];
      ---
      >

      at ../../node_modules/circom-mutator/src/tester.ts:52:19

  ● [mutation] eff_ecdsa_to_addr › [mutation] Replace constraints by assignments
(AssignedButNotConstrained)

    expect(received).toThrow()
    Received function did not throw

      134 |     const w = await circuit.calculateWitness(input, true);
      135 |
    > 136 |     expect(() => circuit.checkConstraints(w)).toThrow();
          |                                               ^
      137 |   }
      138 | });
      139 |

      at tests/addr_membership.test.ts:136:47
      at fulfilled (tests/addr_membership.test.ts:5:58)

  ● [mutation] eff_ecdsa_to_addr › [mutation] Remove constraints on circuit
outputs (MissingOutputCheckConstraint)

    expect(received).toThrow()

    Received function did not throw

      134 |     const w = await circuit.calculateWitness(input, true);
      135 |
    > 136 |     expect(() => circuit.checkConstraints(w)).toThrow();
          |                                               ^
      137 |   }
      138 | });
      139 |

      at tests/addr_membership.test.ts:136:47
      at fulfilled (tests/addr_membership.test.ts:5:58)

  ● [mutation] eff_ecdsa_to_addr › [mutation] Arithmetic Over/Under Flows
(Num2Bits)

    assert.strictEqual(received, expected)

    Expected value to strictly be equal to:
      undefined
    Received:
      null

    Message:
      circom compiler error
    Error: Command failed: circom --wasm --sym --r1cs --output
/var/folders/pb/g_b_19n15hn4gkl0cvjjlzfm0000gn/T/circom_-64330-yAAED1vVOo1S --
prime secq256k1 packages/circuits/tests/circuits/addr_membership_test.circom
    error[T3001]: Exception caused by invalid access
       ┌─ "packages/circuits/eff_ecdsa_membership/addr_membership.circom":49:26
       │
    49 │     merkleProof.leaf <== pubToAddr.address;
       │                          ^^^^^^^^^^^^^^^^^ found here
       │
       = call trace:
         ->AddrMembership

    previous errors were found


    Difference:

      Comparing two different types of values. Expected undefined but received
null.

      at compile (../../node_modules/circom_tester/wasm/tester.js:91:2)
      at wasm_tester (../../node_modules/circom_tester/wasm/tester.js:45:2)
```