Anon Aadhaar PSE audit

By Mridul Garg, Yufei Li, Kyle Charbonnet

March 2024

Overview

Background

Anon Aadhaar is a protocol designed to prove properties based on Aadhaar card (Indian ID) without revealing its content.

Scope

Github repo: https://github.com/anon-aadhaar/anon-aadhaar

Commit Hash: c4ef0180cddda39a92779d432265f7f75190977b

Files: Smart contracts, circuits.

External documentation: Aadhaar Card v1 spec

Fix commit: bbdfd9d3037f5126dd69c672b72c668b763f4ac7

Findings

Critical Severity

1. Incorrect order of inputs from smart contract to circuit

Context: AnonAadhaar.sol#L57-L60, aadhaar-verifier.circom#L46-L47, Verifier.ts#L106

Description: `state` and `pinCode` is declared in circuits in the following order:

```
signal output state;
signal output pinCode;
```

However, smart contract and TS code assign `pinCode` to `state` and `state` to `pinCode`:

```
// revealPincode
revealArray[2],
// revealState
revealArray[3],
```

Recommendation: Flip the order of values to correctly assign `state` and `pinCode`.

Implemented fix: PR 203.

High Severity

1. User can bypass nullifier by changing photo

Context: aadhaar-verifier.circom#L79

Description: User photo provided to Aadhaar card is included in nullifier calculation. Since a photo can be updated by the user, user can bypass nullifier meant to be restrict user to

access something only once by changing the photo. Even if the voting period is less than 2 to 3 weeks, the user who's in the period of changing photo already can create one nullifier just before and after the photo change.

Recommendation: There is no way to avoid this situation. Signed data doesn't include the timestamp at which photo was updated, so there is no way to identify this change. Assuming the user cannot update a photo frequently, the abuse of the system is possible but limited.

2. Incorrect age calculation logic

Context: extractor.circom#L180

Description: If current date is 2024/2/10, birthday is 2023/3/6, the age should be 0. With the current logic, since `10 > 6`, age is calculated as 1.

Recommendation: Update `age` calculation as:

```
monthGt = currentMonth > birthMonth
dayGt = currentMonth == birthMonth and currentDay >= birthDay

age = ageByYear + (monthGt or dayGt)
```

Implemented fix: PR 203.

Medium severity

1. Wrong indexing for `pubkeyHasherInput`

Context: signature.circom#L71

Description: The following loop works well for `k=17` but for other values it doesn't loop over the entire `pubkeyHasherInput` array:

```
for (var i = 0; i < poseidonInputSize; i++) {
    if (i == poseidonInputSize - 1 && poseidonInputSize % 2 == 1) {
        pubkeyHasherInput[i] <== pubKey[i * 2];
    } else {
        pubkeyHasherInput[i] <== pubKey[i * 2] + (1 << n) * pubKey[i * 2 + 1];
    }
}
```

Recommendation: Update it as:

```
-if (i == poseidonInputSize - 1 && poseidonInputSize % 2 == 1) {
+if (i == poseidonInputSize - 1 && k % 2 == 1) {
```

Implemented fix: PR 203.

2. `log2Ceil()` returns `log2(a)+1` for powers of 2

Context: zk-email/functions.circom

Description: `log2Ceil()` is incorrectly implemented and doesn't return the ceiling of the log2 value of the input.

Recommendation: Update as:

```
-var n = a+1;
+var n = a-1;
```

Implemented fix: Commit 99174374f1109a0511b193407037ed2acb556ff4 in zkemail repo.

Gas Optimizations

1. Set variables as `immutable` if they cannot be updated after deployment

Context: AnonAadhaar.sol#L8-L9, AnonAadhaarVote.sol#L8-L9

Description: The highlighted storage variables are only set in `constructor`. Since they are fixed after deployment, they can be made `immutable` to save gas by storing them in the code itself instead of EVM storage.

Recommendation: Make these variables `immutable`.

Implemented fix: PR 203.

2. Cache length

Context: AnonAadhaarVote.sol#L117

Description: `proposals.length` can be cached before the loop to save gas.

Recommendation: Cache length.

Implemented fix: PR 203.

3. Arguments can be kept in calldata

Context: AnonAadhaarVote.sol#L58-L59, AnonAadhaar.sol#L37-L38

Description: There's no need to copy the highlited arguments to memory:

Recommendation: Keep these arguments in calldata.

Implemented fix: PR 203.

4. No need to compare a boolean with `true`

Context: AnonAadhaarVote.sol#L70, AnonAadhaarVote.sol#L81

Description: A boolean variable can be used in an `if` condition directly. `if (b == true)` is equivalent to `if (b)`.

Recommendation: Remove the equality check with `true` and use the boolean expression directly.

Implemented fix: PR 203.

Informational

1. Remove dead code

Context: AnonAadhaar.sol#L19

Description: `verifyPublicKeyHash()` is a private function, and hence can't be called from contracts inheriting `AnonAadhaar`, so the function is dead code.

Recommendation: Delete the function.

Implemented fix: PR 203.

2. Solidity supports common time units

Context: AnonAadhaarVote.sol#L41

Description: Solidity supports `hours` as a time unit. `1 hour` is equal to `60 * 60` seconds.

Recommendation: Use `hours`:

```
-return timestamp > (block.timestamp - 3 * 60 * 60);
+return timestamp > (block.timestamp - 3 hours);
```

Implemented fix: PR 203.

3. Circom supports array assignments

Context: signature.circom#L54-L57

Description: The following loop can be syntactically avoided by using the new Circom syntax:

```
for (var i = 0; i < k; i++) {
    rsa.modulus[i] <== pubKey[i];
    rsa.signature[i] <== signature[i];
}
```

Recommendation: Replace it as:

```
rsa.modulus <== pubKey;
rsa.signature <== signature;
```

Implemented fix: PR 203.

4. `dobDelimiterIndex` isn't used

Context: extractor.circom#L147

Description: `doblimitedIndex` is declared but isn't used in circuits:

```
var dobDelimiterIndex = dobPosition();
```

Recommendation: Delete the line.

Implemented fix: PR 203.

5. `_hash()` can shift by fewer bits

Context: AnonAadhaar.sol#L71

Description: The goal with right shifting hash is to make it fit into Bn254 curve order. The curve order is of 254 bits, so shifting by 3 bits is fine.

```
return uint256(keccak256(abi.encodePacked(message))) >> 8;
```

Recommendation: Shift by 3 bits instead.

Implemented fix: PR 203.