

Registration Smart Contract

```
Parameter Registration.registerEntity(address,Registration.EntityType)._entityAddress (Registration.sol#36) is not in mixedCase
Parameter Registration.registerEntity(address,Registration.EntityType)._entityType (Registration.sol#36) is not in mixedCase
Parameter Registration.removeEntity(address)._entityAddress (Registration.sol#46) is not in mixedCase
Parameter Registration.getEntity(address)._entityAddress (Registration.sol#52) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
Registration.sol analyzed (1 contracts with 81 detectors), 4 result(s) found
```

Verifier Smart Contract

Pairing.addition(Pairing.G1Point,Pairing.G1Point) (Verifier.sol#40-53) uses assembly

- INLINE ASM (Verifier.sol#47-51)

Pairing.scalar_mul(Pairing.G1Point,uint256) (Verifier.sol#58-70) uses assembly

- INLINE ASM (Verifier.sol#64-68)

Pairing.pairing(Pairing.G1Point[],Pairing.G2Point[]) (Verifier.sol#75-98) uses assembly

- INLINE ASM (Verifier.sol#91-95)

Reference: <https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage>

Verifier.sol analyzed (2 contracts with 81 detectors), 3 result(s) found

Violation Reporting Smart Contract

```
Reentrancy in ViolationsReporting.grantAccess(address) (ViolationReporting.sol#341-345):
  External calls:
  - onlyVerifyingOracle() (ViolationReporting.sol#341)
    - (entitytype,isRegistered) = Registration.getEntity(msg.sender) (ViolationReporting.sol#297)
  State variables written after the call(s):
  - accessWhitelist[_user] = true (ViolationReporting.sol#342)
Reentrancy in ViolationsReporting.recordViolation(string,int256,int256) (ViolationReporting.sol#314-319):
  External calls:
  - onlyReportingOracle() (ViolationReporting.sol#314)
    - (entitytype,isRegistered) = Registration.getEntity(msg.sender) (ViolationReporting.sol#285)
  State variables written after the call(s):
  - nonReentrant() (ViolationReporting.sol#314)
    - _status = _NOT_ENTERED (ViolationReporting.sol#71)
    - _status = _ENTERED (ViolationReporting.sol#65)
  - reportCount ++ (ViolationReporting.sol#315)
  - violationReports[reportCount] = ViolationReport(msg.sender,block.timestamp,_latitude,_longitude,bytes32(bytes(_evidencehash)),true) (ViolationReporting.sol#316)
Reentrancy in ViolationsReporting.revokeAccess(address) (ViolationReporting.sol#348-351):
  External calls:
  - onlyRegulatoryAuthority() (ViolationReporting.sol#348)
    - (entitytype,isRegistered) = Registration.getEntity(msg.sender) (ViolationReporting.sol#279)
  State variables written after the call(s):
  - accessWhitelist[_user] = false (ViolationReporting.sol#349)
Reentrancy in ViolationsReporting.updateVerifier(address) (ViolationReporting.sol#354-357):
  External calls:
  - onlyRegulatoryAuthority() (ViolationReporting.sol#354)
    - (entitytype,isRegistered) = Registration.getEntity(msg.sender) (ViolationReporting.sol#279)
  State variables written after the call(s):
  - Verifier = IVerifier(_verifier) (ViolationReporting.sol#355)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-2

Reentrancy in ViolationsReporting.grantAccess(address) (ViolationReporting.sol#341-345):
  External calls:
  - onlyVerifyingOracle() (ViolationReporting.sol#341)
    - (entitytype,isRegistered) = Registration.getEntity(msg.sender) (ViolationReporting.sol#297)
  Event emitted after the call(s):
  - Whitelisted(_user,block.timestamp) (ViolationReporting.sol#343)
Reentrancy in ViolationsReporting.recordViolation(string,int256,int256) (ViolationReporting.sol#314-319):
  External calls:
  - onlyReportingOracle() (ViolationReporting.sol#314)
    - (entitytype,isRegistered) = Registration.getEntity(msg.sender) (ViolationReporting.sol#285)
  Event emitted after the call(s):
  - NewViolationReport(reportCount,msg.sender,_latitude,_longitude,block.timestamp,bytes32(bytes(_evidencehash))) (ViolationReporting.sol#318)
Reentrancy in ViolationsReporting.revokeAccess(address) (ViolationReporting.sol#348-351):
  External calls:
  - onlyRegulatoryAuthority() (ViolationReporting.sol#348)
    - (entitytype,isRegistered) = Registration.getEntity(msg.sender) (ViolationReporting.sol#279)
  Event emitted after the call(s):
  - AccessRevoked(msg.sender,_user,block.timestamp) (ViolationReporting.sol#350)

Reentrancy in ViolationsReporting.revokeReportValidity(uint256) (ViolationReporting.sol#329-336):
  External calls:
  - onlyRegulatoryAuthority() (ViolationReporting.sol#329)
    - (entitytype,isRegistered) = Registration.getEntity(msg.sender) (ViolationReporting.sol#279)
  Event emitted after the call(s):
  - ReportValidityRevoked(msg.sender,_violationReportId,block.timestamp) (ViolationReporting.sol#335)
Reentrancy in ViolationsReporting.updateVerifier(address) (ViolationReporting.sol#354-357):
  External calls:
  - onlyRegulatoryAuthority() (ViolationReporting.sol#354)
    - (entitytype,isRegistered) = Registration.getEntity(msg.sender) (ViolationReporting.sol#279)
  Event emitted after the call(s):
  - VerifierUpdated(msg.sender,_verifier,block.timestamp) (ViolationReporting.sol#356)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

ReentrancyGuard._reentrancyGuardEntered() (ViolationReporting.sol#78-80) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
ViolationReporting.sol analyzed (5 contracts with 81 detectors), 10 result(s) found
```