CERTIK

**F**ormal **V**erification **P**latform for
**S**mart **C**ontracts
and **B**lockchain **E**cosystems

# CERTIK VERIFICATION REPORT FOR IOGTOKEN

Request Date: September 17th, 2018
Report Revision: September 18th, 2018
Company Name: IOG

# Disclaimer

This report is subject to the terms and conditions (including without limitation, description of the services, confidentiality, disclaimer and limitation of liability) set forth in the Verification Services Agreement between CertiK and IOG, or the scope of verification, and terms and conditions provided to IOG in connection with this verification. No third party shall be entitled to rely on this report or have any legal or equitable right, benefit or remedy of any nature whatsoever, under or by reason of this report. CertiK assumes no liability to any third party because of reliance on this report.

# Summary

This is the report for smart contract verification service on iogtoken from IOG. The goal of the audition is to guarantee that verified smart contracts are robust enough to avoid potentially unexpected loopholes.

# Methodology

Certik applied 100% coveraged smart labels on the source code to detect 4 types of issues:

- Function Correctness
- Integer Overflow
- Assertion Failure
- Buffer Overflow

For each verification request issue, CertiK categorizes its result into 3 buckets, based on its risk level:

| Risk Level | Reason | Action Needed |
|---|---|---|
| Critical | The code implementation does not match the specification, or it could result in s loss of funds for contract owner or users. | Fix required. |
| Medium | The code implementation does not match the specification at certain condition, or it could affect the security standard by mis-operations of owner or admin. | Fix highly recommended. |
| Low | The code implementation is not a best practice, or use a suboptimal design pattern, but no major security concerns on that. | Fix are just suggestions. |

# Scope

The source code to verify is confirmed with client, and appended to the appendix at the end of the report.

# Files

- Migrations.sol
- IOGToken.sol

# MD5

The result of this report is only a reflection of the source code that was determined in this scope, and of the source code at the audit time.

- `4317a95838205f33a1a5d69ca8ff22d4`
- `40386e5d86eb7936337f72da86d4b05d`

# Conclusion: PASS

CertiK formal verification engine concludes that the IOGś smart contract meets **97.9%** of the specification out of 100% code coverage. There are **0** critical issues, **0** medium risk issues, and **1** low risk issues found. There are **1** of suggestions provided from CertiK.

The correctness of distribution function depends on input lockedPeriodList, where improper may cause overflow. Overall, CertiK believes this contract is trustworthy and hack-resistant.

# Detail for Request 0

SafeMath_mul

17, Sep 2018
337.549 ms

Line 13-21 in File IOGToken.sol

```
13  /*@CTK SafeMath_mul
14    @tag spec
15    @post __reverted == __has_assertion_failure
16    @post __has_assertion_failure == __has_overflow
17    @post __reverted == false -> c == a * b
18    @post msg == msg__post
19    @post (a > 0 && (a * b / a != b)) == __has_assertion_failure
20    @post __addr_map == __addr_map__post
21  */
```

Line 23-34 in File IOGToken.sol

```
23  function mul(uint256 a, uint256 b) internal pure returns (uint256 c) {
24    // Gas optimization: this is cheaper than asserting 'a' not being zero, but the
25    // benefit is lost if 'b' is also tested.
26    // See: https://github.com/OpenZeppelin/openzeppelin-solidity/pull/522
27    if (a == 0) {
28      return 0;
29    }
30
31    c = a * b;
32    assert(c / a == b);
33    return c;
34  }
```

The code meets the specification

# Detail for Request 1

Buffer overflow / array index out of bound would never happen.

17, Sep 2018
0.443 ms

Line 22 in File IOGToken.sol

```
22  //@CTK NO_BUF_OVERFLOW
```

Line 23-34 in File IOGToken.sol

```
23  function mul(uint256 a, uint256 b) internal pure returns (uint256 c) {
24    // Gas optimization: this is cheaper than asserting 'a' not being zero, but the
25    // benefit is lost if 'b' is also tested.
26    // See: https://github.com/OpenZeppelin/openzeppelin-solidity/pull/522
27    if (a == 0) {
28      return 0;
29    }
```

```
30
31      c = a * b;
32      assert(c / a == b);
33      return c;
34    }
```

The code meets the specification

# Detail for Request 2

**If method completes, integer overflow would not happen.**

17, Sep 2018
0.27799999999999997 ms

Line 39 in File IOGToken.sol

```
39    //@CTK NO_OVERFLOW
```

Line 50-55 in File IOGToken.sol

```
50    function div(uint256 a, uint256 b) internal pure returns (uint256) {
51      // assert(b > 0); // Solidity automatically throws when dividing by 0
52      // uint256 c = a / b;
53      // assert(a == b * c + a % b); // There is no case in which this doesn't hold
54      return a / b;
55    }
```

The code meets the specification

# Detail for Request 3

**SafeMath_div**

17, Sep 2018
0.29300000000000004 ms

Line 40-49 in File IOGToken.sol

```
40    /*@CTK SafeMath_div
41      @tag spec
42      @post __reverted == __has_assertion_failure
43      @post __has_overflow == true -> __has_assertion_failure == true
44      @post !__reverted -> __return == a / b
45      @post msg == msg__post
46      @post (b == 0) == __has_assertion_failure
47      @post __addr_map == __addr_map__post
48      @post !__has_buf_overflow
49    */
```

Line 50-55 in File IOGToken.sol

```
50  function div(uint256 a, uint256 b) internal pure returns (uint256) {
51    // assert(b > 0); // Solidity automatically throws when dividing by 0
52    // uint256 c = a / b;
53    // assert(a == b * c + a % b); // There is no case in which this doesn't hold
54    return a / b;
55  }
```

The code meets the specification

# Detail for Request 4

SafeMath_sub

17, Sep 2018
0.966 ms

Line 60-68 in File IOGToken.sol

```
60  /*@CTK SafeMath_sub
61    @tag spec
62    @post __reverted == __has_assertion_failure
63    @post __has_overflow -> __has_assertion_failure
64    @post __reverted == false -> __return == a - b
65    @post msg == msg__post
66    @post (b > a) == __has_assertion_failure
67    @post __addr_map == __addr_map__post
68  */
```

Line 69-72 in File IOGToken.sol

```
69  function sub(uint256 a, uint256 b) internal pure returns (uint256) {
70    assert(b <= a);
71    return a - b;
72  }
```

The code meets the specification

# Detail for Request 5

If method completes, integer overflow would not happen.

17, Sep 2018
1.7799999999999998 ms

Line 77 in File IOGToken.sol

```
77  //@CTK NO_OVERFLOW
```

Line 88-92 in File IOGToken.sol

```
88  function add(uint256 a, uint256 b) internal pure returns (uint256 c) {
89    c = a + b;
90    assert(c >= a);
91    return c;
92  }
```

The code meets the specification

# Detail for Request 6

SafeMath_add

17, Sep 2018
5.6049999999999995 ms

Line 78-87 in File IOGToken.sol

```
78  /*@CTK SafeMath_add
79    @tag spec
80    @post __reverted == __has_assertion_failure
81    @post __has_assertion_failure == __has_overflow
82    @post __reverted == false -> c == a + b
83    @post msg == msg__post
84    @post ((a + b < a) || (a + b < b)) == __has_assertion_failure
85    @post __addr_map == __addr_map__post
86    @post !__has_buf_overflow
87  */
```

Line 88-92 in File IOGToken.sol

```
88  function add(uint256 a, uint256 b) internal pure returns (uint256 c) {
89    c = a + b;
90    assert(c >= a);
91    return c;
92  }
```

The code meets the specification

# Detail for Request 7

**If method completes, integer overflow would not happen.**

17, Sep 2018
0.34400000000000003 ms

Line 134 in File IOGToken.sol

```
134  //@CTK NO_OVERFLOW
```

Line 137-140 in File IOGToken.sol

```
137  function renounceOwnership() public onlyOwner {
138    emit OwnershipRenounced(owner);
139    owner = address(0);
140  }
```

The code meets the specification

# Detail for Request 8

**Method will not encounter an assertion failure.**

17, Sep 2018
0.32 ms

Line 135 in File IOGToken.sol

```
135  //@CTK NO_ASF
```

Line 137-140 in File IOGToken.sol

```
137  function renounceOwnership() public onlyOwner {
138    emit OwnershipRenounced(owner);
139    owner = address(0);
140  }
```

The code meets the specification

# Detail for Request 9

**Buffer overflow / array index out of bound would never happen.**

17, Sep 2018
0.318 ms

Line 136 in File IOGToken.sol

```
136  //@CTK NO_BUF_OVERFLOW
```

Line 137-140 in File IOGToken.sol

```
137  function renounceOwnership() public onlyOwner {
138    emit OwnershipRenounced(owner);
139    owner = address(0);
140  }
```

The code meets the specification

# Detail for Request 10

**transferOwnership**

17, Sep 2018
2.6029999999999998 ms

Line 146-150 in File IOGToken.sol

```
146  /*@CTK transferOwnership
147    @post __reverted == false -> (msg.sender == owner -> __post.owner == _newOwner)
148    @post (owner != msg.sender) -> (__reverted == true)
149    @post (_newOwner == address(0)) -> (__reverted == true)
150  */
```

Line 154-156 in File IOGToken.sol

```
154  function transferOwnership(address _newOwner) public onlyOwner {
155    _transferOwnership(_newOwner);
156  }
```

The code meets the specification

# Detail for Request 11

**If method completes, integer overflow would not happen.**

17, Sep 2018
0.536 ms

Line 151 in File IOGToken.sol

```
151  //@CTK NO_OVERFLOW
```

Line 154-156 in File IOGToken.sol

```
154  function transferOwnership(address _newOwner) public onlyOwner {
155    _transferOwnership(_newOwner);
156  }
```

The code meets the specification

# Detail for Request 12

**Method will not encounter an assertion failure.**

17, Sep 2018
0.593 ms

Line 152 in File IOGToken.sol

```
152  //@CTK NO_ASF
```

Line 154-156 in File IOGToken.sol

```
154  function transferOwnership(address _newOwner) public onlyOwner {
155    _transferOwnership(_newOwner);
156  }
```

The code meets the specification

# Detail for Request 13

**Buffer overflow / array index out of bound would never happen.**

17, Sep 2018
0.5339999999999999 ms

Line 153 in File IOGToken.sol

```
153  //@CTK NO_BUF_OVERFLOW
```

Line 154-156 in File IOGToken.sol

```
154  function transferOwnership(address _newOwner) public onlyOwner {
155    _transferOwnership(_newOwner);
156  }
```

The code meets the specification

# Detail for Request 14

**If method completes, integer overflow would not happen.**

17, Sep 2018
0.32699999999999996 ms

Line 162 in File IOGToken.sol

```
162  //@CTK NO_OVERFLOW
```

Line 165-169 in File IOGToken.sol

```
165  function _transferOwnership(address _newOwner) internal {
166    require(_newOwner != address(0));
167    emit OwnershipTransferred(owner, _newOwner);
168    owner = _newOwner;
169  }
```

The code meets the specification

# Detail for Request 15

**Method will not encounter an assertion failure.**

17, Sep 2018
0.321 ms

Line 163 in File IOGToken.sol

```
163  //@CTK NO_ASF
```

Line 165-169 in File IOGToken.sol

```
165  function _transferOwnership(address _newOwner) internal {
166    require(_newOwner != address(0));
167    emit OwnershipTransferred(owner, _newOwner);
168    owner = _newOwner;
169  }
```

The code meets the specification

# Detail for Request 16

**Buffer overflow / array index out of bound would never happen.**

17, Sep 2018
0.318 ms

Line 164 in File IOGToken.sol

```
164  //@CTK NO_BUF_OVERFLOW
```

Line 165-169 in File IOGToken.sol

```
165  function _transferOwnership(address _newOwner) internal {
166    require(_newOwner != address(0));
167    emit OwnershipTransferred(owner, _newOwner);
168    owner = _newOwner;
169  }
```

The code meets the specification

# Detail for Request 17

**If method completes, integer overflow would not happen.**

17, Sep 2018
0.455 ms

Line 203 in File IOGToken.sol

```
203  //@CTK NO_OVERFLOW
```

Line 206-209 in File IOGToken.sol

```
206  function pause() onlyOwner whenNotPaused public {
207    paused = true;
208    emit Pause();
209  }
```

The code meets the specification

# Detail for Request 18

**Method will not encounter an assertion failure.**

17, Sep 2018
0.429 ms

Line 204 in File IOGToken.sol

```
204  //@CTK NO_ASF
```

Line 206-209 in File IOGToken.sol

```
206  function pause() onlyOwner whenNotPaused public {
207    paused = true;
208    emit Pause();
209  }
```

The code meets the specification

# Detail for Request 19

**Buffer overflow / array index out of bound would never happen.**

17, Sep 2018
0.425 ms

Line 205 in File IOGToken.sol

```
205  //@CTK NO_BUF_OVERFLOW
```

Line 206-209 in File IOGToken.sol

```
206  function pause() onlyOwner whenNotPaused public {
207    paused = true;
208    emit Pause();
209  }
```

The code meets the specification

# Detail for Request 20

**If method completes, integer overflow would not happen.**

17, Sep 2018
0.5569999999999999 ms

Line 214 in File IOGToken.sol

```
214  //@CTK NO_OVERFLOW
```

Line 217-220 in File IOGToken.sol

```
217  function unpause() onlyOwner whenPaused public {
218    paused = false;
219    emit Unpause();
220  }
```

The code meets the specification

# Detail for Request 21

**Method will not encounter an assertion failure.**

17, Sep 2018
0.425 ms

Line 215 in File IOGToken.sol

```
215  //@CTK NO_ASF
```

Line 217-220 in File IOGToken.sol

```
217  function unpause() onlyOwner whenPaused public {
218    paused = false;
219    emit Unpause();
220  }
```

The code meets the specification

# Detail for Request 22

**Buffer overflow / array index out of bound would never happen.**

17, Sep 2018
0.419 ms

Line 216 in File IOGToken.sol

```
216  //@CTK NO_BUF_OVERFLOW
```

Line 217-220 in File IOGToken.sol

```
217  function unpause() onlyOwner whenPaused public {
218    paused = false;
219    emit Unpause();
220  }
```

The code meets the specification

# Detail for Request 23

**If method completes, integer overflow would not happen.**

17, Sep 2018
0.35100000000000003 ms

Line 271 in File IOGToken.sol

```
271  //@CTK NO_OVERFLOW
```

Line 274-276 in File IOGToken.sol

```
274  function totalSupply() public view returns (uint256) {
275    return totalSupply_;
276  }
```

The code meets the specification

# Detail for Request 24

**Method will not encounter an assertion failure.**

17, Sep 2018
0.285 ms

Line 272 in File IOGToken.sol

```
272  //@CTK NO_ASF
```

Line 274-276 in File IOGToken.sol

```
274  function totalSupply() public view returns (uint256) {
275    return totalSupply_;
276  }
```

The code meets the specification

# Detail for Request 25

**Buffer overflow / array index out of bound would never happen.**

17, Sep 2018
0.23399999999999999 ms

Line 273 in File IOGToken.sol

```
273  //@CTK NO_BUF_OVERFLOW
```

Line 274-276 in File IOGToken.sol

```
274  function totalSupply() public view returns (uint256) {
275    return totalSupply_;
276  }
```

The code meets the specification

# Detail for Request 26

**transfer_failure_addressEqualZero**

17, Sep 2018
0.8 ms

Line 283-287 in File IOGToken.sol

```
283  /*@CTK transfer_failure_addressEqualZero
284    @pre _to == address(0)
285    @pre balances[msg.sender] >= _value
286    @post __reverted == true
287  */
```

Line 305-313 in File IOGToken.sol

```
305  function transfer(address _to, uint256 _value) public returns (bool) {
306    require(_to != address(0));
307    require(_value <= balances[msg.sender]);
308
309    balances[msg.sender] = balances[msg.sender].sub(_value);
310    balances[_to] = balances[_to].add(_value);
311    emit Transfer(msg.sender, _to, _value);
312    return true;
313  }
```

The code meets the specification

# Detail for Request 27

transfer_failure_balanceSmallerValue

17, Sep 2018
3.565 ms

Line 288-292 in File IOGToken.sol

```
288  /*@CTK transfer_failure_balanceSmallerValue
289    @pre _to != address(0)
290    @pre balances[msg.sender] < _value
291    @post __reverted == true
292  */
```

Line 305-313 in File IOGToken.sol

```
305  function transfer(address _to, uint256 _value) public returns (bool) {
306    require(_to != address(0));
307    require(_value <= balances[msg.sender]);
308
309    balances[msg.sender] = balances[msg.sender].sub(_value);
310    balances[_to] = balances[_to].add(_value);
311    emit Transfer(msg.sender, _to, _value);
312    return true;
313  }
```

The code meets the specification

# Detail for Request 28

transfer_conditions

17, Sep 2018
82.425 ms

Line 293-298 in File IOGToken.sol

```
293  /*@CTK transfer_conditions
294    @tag assume_completion
295    @pre _to != msg.sender
296    @post __post.balances[_to] == balances[_to] + _value
297    @post __post.balances[msg.sender] == balances[msg.sender] - _value
298  */
```

Line 305-313 in File IOGToken.sol

```
305  function transfer(address _to, uint256 _value) public returns (bool) {
306    require(_to != address(0));
307    require(_value <= balances[msg.sender]);
308
309    balances[msg.sender] = balances[msg.sender].sub(_value);
310    balances[_to] = balances[_to].add(_value);
311    emit Transfer(msg.sender, _to, _value);
312    return true;
313  }
```

The code meets the specification

# Detail for Request 29

transfer_same_address

17, Sep 2018

10.581 ms

Line 299-304 in File IOGToken.sol

```
299  /*@CTK transfer_same_address
300    @tag assume_completion
301    @tag no_overflow
302    @pre _to == msg.sender
303    @post this == __post
304  */
```

Line 305-313 in File IOGToken.sol

```
305  function transfer(address _to, uint256 _value) public returns (bool) {
306    require(_to != address(0));
307    require(_value <= balances[msg.sender]);
308
309    balances[msg.sender] = balances[msg.sender].sub(_value);
310    balances[_to] = balances[_to].add(_value);
311    emit Transfer(msg.sender, _to, _value);
312    return true;
313  }
```

The code meets the specification

# Detail for Request 30

balanceOf

17, Sep 2018
0.306 ms

Line 320-323 in File IOGToken.sol

```
320  /*@CTK balanceOf
321    @post __reverted == false
322    @post __return == balances[_owner]
323  */
```

Line 324-326 in File IOGToken.sol

```
324  function balanceOf(address _owner) public view returns (uint256) {
325    return balances[_owner];
326  }
```

The code meets the specification

# Detail for Request 31

transferFrom

17, Sep 2018
139.942 ms

Line 349-356 in File IOGToken.sol

```
349  /*@CTK transferFrom
350    @tag assume_completion
351    @pre _from != _to
352    @post __return == true
353    @post __post.balances[_to] == balances[_to] + _value
354    @post __post.balances[_from] == balances[_from] - _value
355    @post __has_overflow == false
356  */
```

Line 361-378 in File IOGToken.sol

```
361  function transferFrom(
362    address _from,
363    address _to,
364    uint256 _value
365  )
366    public
367    returns (bool)
368  {
369    require(_to != address(0));
370    require(_value <= balances[_from]);
371    require(_value <= allowed[_from][msg.sender]);
372
373    balances[_from] = balances[_from].sub(_value);
374    balances[_to] = balances[_to].add(_value);
375    allowed[_from][msg.sender] = allowed[_from][msg.sender].sub(_value);
```

```
376     emit Transfer(_from, _to, _value);
377     return true;
378   }
```

The code meets the specification

# Detail for Request 32

transferFrom_sameOwner

17, Sep 2018
91.472 ms

Line 357-360 in File IOGToken.sol

```
357   /*@CTK FAIL "transferFrom_sameOwner"
358     @pre _from == _to
359     @post __post.allowed[_from][msg.sender] == allowed[_from][msg.sender]
360   */
```

Line 361-378 in File IOGToken.sol

```
361   function transferFrom(
362     address _from,
363     address _to,
364     uint256 _value
365   )
366     public
367     returns (bool)
368   {
369     require(_to != address(0));
370     require(_value <= balances[_from]);
371     require(_value <= allowed[_from][msg.sender]);
372 
373     balances[_from] = balances[_from].sub(_value);
374     balances[_to] = balances[_to].add(_value);
375     allowed[_from][msg.sender] = allowed[_from][msg.sender].sub(_value);
376     emit Transfer(_from, _to, _value);
377     return true;
378   }
```

This code violates the specification

```
1  Counter Example:
2  Before Execution:
3     Input = {
4         _from = 0x2
5         _to = 0x2
6         _value = 0x31
7     }
8     This = 0
9     Internal = {
10        __has_assertion_failure = False
11        __has_buf_overflow = False
12        __has_overflow = False
13        __has_returned = False
14        __reverted = False
```

```
        msg = {
            gas: 0x0
            sender: 0x0
            value: 0x0
        }
    }
    Other = {
        __return = False
    }
    Address_Map = {
        address_wrapper @ 0x0: {
            StandardToken: {
                allowed: {
                    0x2: {
                        0x80: 0x0
                        0x0: 0x80
                        0x8: 0x0
                        else: 0x31
                    }
                    else: {
                        0x4f: 0x4f
                        else: 0x4f
                    }
                }
                balances: {
                    0x80: 0x20
                    0x0: 0x0
                    0x8: 0x8
                    0x2: 0x40
                    else: 0x31
                }
                totalSupply_: 0x0
            }
            balance: 0x0
        }
    }

After Execution:
    Input = {
        _from = 0x2
        _to = 0x2
        _value = 0x31
    }
    This = 0
    Internal = {
        __has_assertion_failure = False
        __has_buf_overflow = True
        __has_overflow = False
        __has_returned = True
        __reverted = False
        msg = {
            gas: 0x0
            sender: 0x0
            value: 0x0
        }
    }
    Other = {
        __return = True
```

```
    }
    Address_Map = {
        address_wrapper @ 0x0: {
            StandardToken: {
                allowed: {
                    0x2: {
                        0x80: 0x0
                        0x0: 0x4f
                        0x8: 0x0
                        else: 0x31
                    }
                    else: {
                        0x4f: 0x4f
                        else: 0x4f
                    }
                }
                balances: {
                    0x80: 0x20
                    0x0: 0x0
                    0x8: 0x8
                    0x2: 0x40
                    else: 0x31
                }
                totalSupply_: 0x0
            }
            balance: 0x0
        }
    }
```

# Detail for Request 33

approve_success

17, Sep 2018
0.306 ms

Line 389-392 in File IOGToken.sol

```
/*@CTK approve_success
  @post _value == 0 -> __reverted == false
  @post allowed[msg.sender][_spender] == 0 -> __reverted == false
*/
```

Line 397-401 in File IOGToken.sol

```
function approve(address _spender, uint256 _value) public returns (bool) {
  allowed[msg.sender][_spender] = _value;
  emit Approval(msg.sender, _spender, _value);
  return true;
}
```

The code meets the specification

# Detail for Request 34

approve

17, Sep 2018
1.331 ms

Line 393-396 in File IOGToken.sol

```
393  /*@CTK approve
394    @tag assume_completion
395    @post __post.allowed[msg.sender][_spender] == _value
396  */
```

Line 397-401 in File IOGToken.sol

```
397  function approve(address _spender, uint256 _value) public returns (bool) {
398    allowed[msg.sender][_spender] = _value;
399    emit Approval(msg.sender, _spender, _value);
400    return true;
401  }
```

The code meets the specification

# Detail for Request 35

get_allowance

17, Sep 2018
0.277 ms

Line 409-413 in File IOGToken.sol

```
409   /*@CTK get_allowance
410    @post __reverted == false
411    @post __return == allowed[_owner][_spender]
412    @post this == __post
413  */
```

Line 414-423 in File IOGToken.sol

```
414  function allowance(
415    address _owner,
416    address _spender
417   )
418    public
419    view
420    returns (uint256)
421  {
422    return allowed[_owner][_spender];
423  }
```

The code meets the specification

# Detail for Request 36

increaseApproval

17, Sep 2018
2.7920000000000003 ms

Line 434-440 in File IOGToken.sol

```
434  /*@CTK increaseApproval
435    @tag assume_completion
436    @post __post.allowed[msg.sender][_spender] ==
437          allowed[msg.sender][_spender] + _addedValue
438    @post __post.allowed[msg.sender][_spender] ==
439          allowed[msg.sender][_spender] + _addedValue -> __return == true
440  */
```

Line 441-452 in File IOGToken.sol

```
441  function increaseApproval(
442    address _spender,
443    uint256 _addedValue
444  )
445    public
446    returns (bool)
447  {
448    allowed[msg.sender][_spender] = (
449      allowed[msg.sender][_spender].add(_addedValue));
450    emit Approval(msg.sender, _spender, allowed[msg.sender][_spender]);
451    return true;
452  }
```

The code meets the specification

# Detail for Request 37

decreaseApproval0

17, Sep 2018
32.695 ms

Line 463-467 in File IOGToken.sol

```
463  /*@CTK decreaseApproval0
464    @pre __return == true
465    @pre allowed[msg.sender][_spender] <= _subtractedValue
466    @post __post.allowed[msg.sender][_spender] == 0
467  */
```

Line 474-489 in File IOGToken.sol

```
474  function decreaseApproval(
475    address _spender,
476    uint256 _subtractedValue
477  )
478    public
```

```
479     returns (bool)
480   {
481     uint256 oldValue = allowed[msg.sender][_spender];
482     if (_subtractedValue > oldValue) {
483       allowed[msg.sender][_spender] = 0;
484     } else {
485       allowed[msg.sender][_spender] = oldValue.sub(_subtractedValue);
486     }
487     emit Approval(msg.sender, _spender, allowed[msg.sender][_spender]);
488     return true;
489   }
```

The code meets the specification

# Detail for Request 38

decreaseApproval

17, Sep 2018
9.518 ms

Line 468-473 in File IOGToken.sol

```
468   /*@CTK decreaseApproval
469     @pre __return == true
470     @pre allowed[msg.sender][_spender] > _subtractedValue
471     @post __post.allowed[msg.sender][_spender] ==
472           allowed[msg.sender][_spender] - _subtractedValue
473   */
```

Line 474-489 in File IOGToken.sol

```
474   function decreaseApproval(
475     address _spender,
476     uint256 _subtractedValue
477   )
478     public
479     returns (bool)
480   {
481     uint256 oldValue = allowed[msg.sender][_spender];
482     if (_subtractedValue > oldValue) {
483       allowed[msg.sender][_spender] = 0;
484     } else {
485       allowed[msg.sender][_spender] = oldValue.sub(_subtractedValue);
486     }
487     emit Approval(msg.sender, _spender, allowed[msg.sender][_spender]);
488     return true;
489   }
```

The code meets the specification

# Detail for Request 39

canTransferIfLocked_address_0

17, Sep 2018
2.048 ms

Line 550-553 in File IOGToken.sol

```
550     /*@CTK canTransferIfLocked_address_0
551       @pre 0 == addressLocks[_sender]
552       @post __return == true
553     */
```

Line 558-563 in File IOGToken.sol

```
558     function canTransferIfLocked(address _sender) internal view returns(bool) {
559         if (0 == addressLocks[_sender])
560             return true;
561 
562         return (now >= addressLocks[_sender]);
563     }
```

The code meets the specification

# Detail for Request 40

canTransferIfLocked_address_not_0

17, Sep 2018
1.554 ms

Line 554-557 in File IOGToken.sol

```
554     /*@CTK canTransferIfLocked_address_not_0
555       @pre 0 != addressLocks[_sender]
556       @post __return == (now >= addressLocks[_sender])
557     */
```

Line 558-563 in File IOGToken.sol

```
558     function canTransferIfLocked(address _sender) internal view returns(bool) {
559         if (0 == addressLocks[_sender])
560             return true;
561 
562         return (now >= addressLocks[_sender]);
563     }
```

The code meets the specification

# Detail for Request 41

timeLock_not_owner

17, Sep 2018
11.048 ms

Line 565-568 in File IOGToken.sol

```
565     /*@CTK timeLock_not_owner
566       @pre owner != msg.sender
567       @post __reverted == true
568     */
```

Line 569-573 in File IOGToken.sol

```
569     function timeLock(address _to, uint256 _value, uint256 releaseDate) onlyOwner
            public {
570         addressLocks[_to] = releaseDate;
571         transfer(_to, _value);
572         emit AddressLocked(_to, _value);
573     }
```

The code meets the specification

# Detail for Request 42

transfer_paused

17, Sep 2018
1.879 ms

Line 575-578 in File IOGToken.sol

```
575     /*@CTK transfer_paused
576       @pre paused == true
577       @post __reverted == true
578     */
```

Line 583-585 in File IOGToken.sol

```
583     function transfer(address _to, uint256 _value) canTransfer(msg.sender)
            whenNotPaused public returns (bool success) {
584         return super.transfer(_to, _value);
585     }
```

The code meets the specification

# Detail for Request 43

transfer_cant_transfer

17, Sep 2018
7.9799999999999995 ms

Line 579-582 in File IOGToken.sol

```
579     /*@CTK transfer_cant_transfer
580       @pre msg.sender == address(0)
581       @post __reverted == true
582     */
```

Line 583-585 in File IOGToken.sol

```
583     function transfer(address _to, uint256 _value) canTransfer(msg.sender)
            whenNotPaused public returns (bool success) {
584         return super.transfer(_to, _value);
585     }
```

The code meets the specification

# Detail for Request 44

transfer_from_paused

17, Sep 2018
2.12 ms

Line 587-590 in File IOGToken.sol

```
587     /*@CTK transfer_from_paused
588       @pre paused == true
589       @post __reverted == true
590     */
```

Line 595-597 in File IOGToken.sol

```
595     function transferFrom(address _from, address _to, uint256 _value) canTransfer(
            _from) whenNotPaused public returns (bool success) {
596         return super.transferFrom(_from, _to, _value);
597     }
```

The code meets the specification

# Detail for Request 45

transfer_from_cant_transfer

17, Sep 2018
8.867999999999999 ms

Line 591-594 in File IOGToken.sol

```
591     /*@CTK transfer_from_cant_transfer
592       @pre _from == address(0)
593       @post __reverted == true
594     */
```

Line 595-597 in File IOGToken.sol

```
595     function transferFrom(address _from, address _to, uint256 _value) canTransfer(
            _from) whenNotPaused public returns (bool success) {
596         return super.transferFrom(_from, _to, _value);
597     }
```

The code meets the specification

# Detail for Request 46

approve_paused

17, Sep 2018
1.191 ms

Line 599-602 in File IOGToken.sol

```
599     /*@CTK approve_paused
600       @pre paused == true
601       @post __reverted == true
602     */
```

Line 603-605 in File IOGToken.sol

```
603     function approve(address _spender, uint256 _value) whenNotPaused public returns (
            bool) {
604         return super.approve(_spender, _value);
605     }
```

The code meets the specification

# Detail for Request 47

increaseApproval_paused

17, Sep 2018
1.425 ms

Line 607-610 in File IOGToken.sol

```
607     /*@CTK increaseApproval_paused
608       @pre paused == true
609       @post __reverted == true
610     */
```

Line 611-613 in File IOGToken.sol

```
611     function increaseApproval(address _spender, uint _addedValue) whenNotPaused public
             returns (bool success) {
612         return super.increaseApproval(_spender, _addedValue);
613     }
```

The code meets the specification

# Detail for Request 48

decreaseApproval_paused

17, Sep 2018
1.6440000000000001 ms

Line 615-618 in File IOGToken.sol

```
615     /*@CTK decreaseApproval_paused
616       @pre paused == true
617       @post __reverted == true
618     */
```

Line 619-621 in File IOGToken.sol

```
619     function decreaseApproval(address _spender, uint _subtractedValue) whenNotPaused
            public returns (bool success) {
620         return super.decreaseApproval(_spender, _subtractedValue);
621     }
```

The code meets the specification