CertiK Verification Report For Ados

Ad-OS

Request Date: 2018-02-01
Revision Date: 2019-02-07

Disclaimer

This Report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Verification Services Agreement between CertiK and Ados(the "Company"), or the scope of services/verification, and terms and conditions provided to the Company in connection with the verification (collectively, the "Agreement"). This Report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This Report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes without CertiK's prior written consent.

Summary

This is the report for smart contract verification service requestd by Ados. The goal of the audition is to guarantee that verified smart contracts are robust enough to avoid potentially unexpected loopholes.
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.

Type of Issues

CertiK smart label engine applied 100% coveraged formal verification labels on the source code, and scanned the code by static analysis and formal verification engine to detect the follow type of issues.

Title	Description	Issues	SWC ID
Integer Overflow and Underflow	An overflow/underflow happens when an arithmetic operation reaches the maximum or minimum size of a type.	0	SWC-101
Function incorrectness	Function implementation does not meet the specification, leading to intentional or unintentional vulnerabilities.	0	
Buffer Overflow	An attacker is able to write to arbitrary storage locations of a contract if array of out bound happens	0	SWC-124
Reentrancy	A malicious contract can call back into the calling contract before the first invocation of the function is finished.	0	SWC-107
Transaction Order Dependence	A race condition vulnerability occurs when code depends on the order of the transactions submitted to it.	0	SWC-114
Timestamp Dependence	Timestamp can be influenced by minors to some degree.	4	SWC-116

Insecure Compiler Version	Using an fixed outdated compiler version or floating pragma can be problematic, if there are publicly disclosed bugs and issues that affect the current compiler version used.	4	SWC-102 SWC-103
Insecure Randomness	Block attributes are insecure to generate random numbers, as they can be influenced by minors to some degree.	0	SWC-120
"tx.origin" for authorization	tx.origin should not be used for authorization. Use msg.sender instead.	0	SWC-115
Delegatecall to Untrusted Callee	Calling into untrusted contracts is very dangerous, the target and arguments provided must be sanitized.	0	SWC-112
State Variable Default Visibility	Labeling the visibility explicitly makes it easier to catch incorrect assumptions about who can access the variable.	0	SWC-108
Function Default Visibility	Functions are public by default. A malicious user is able to make unauthorized or unintended state changes if a developer forgot to set the visibility.	0	SWC-100
Uninitialized variables	Uninitialized local storage variables can point to other unexpected storage variables in the contract.	0	SWC-109
Assertion Failure	The assert() function is meant to assert invariants. Properly functioning code should never reach a failing assert statement.	0	SWC-110
Deprecated Solidity Features	Several functions and operators in Solidity are deprecated and should not be used as best practice.	0	SWC-111
Unused variables	Unused variables reduce code quality	0	

Vulnerability Details

Critical

No issue found.

Medium

No issue found.

Low

No issue found. A minor nit is that Migration can inherit from Ownerable token. It seems to be using a modifier (`restricted`) that is the same as `onlyOwner`.

For every issues found, CertiK categorizes them into 3 buckets based on its risk level:

- Critical: The code implementation does not match the specification, or it could result in loss of funds for contract owner or users.

- Medium: The code implementation does not match the specification at certain condition, or it could affect the security standard by lost of access control.
- Low: The code implementation is not a best practice, or use a suboptimal design pattern, which may lead to security vulnerability, but no concern found yet.

Source Code with CertiK Labels

File VestingVault.sol

```
pragma solidity ^0.4.23;

import "./tokens/ERC20.sol";
import "./utils/SafeMath.sol";
import "./VestingManager.sol";

contract VestingVault {
    using SafeMath for uint256;

    event Released(uint256 amount);

    // beneficiary of tokens after they are released
    ERC20 public token;
    address public beneficiary;

    // VestingManager public vestingManager;

    uint256[] public tokenAmounts;
    uint256[] public unlockTimes;

    uint256 public released;

    /**
     * @dev Creates a vesting contract that vests its balance of any ERC20 token to the
     * _beneficiary, gradually in a linear fashion until _start + _duration. By then all
     * of the balance will have vested.
     * @param _token address of ERC20 token which this contract controls
     * @param _beneficiary address of the beneficiary to whom vested tokens are transferred
     */
    /*@CTK VestingVault
      @tag assume_completion
      @post _token != address(0)
      @post _beneficiary != address(0)
      @post _tokenAmounts.length > 0 && _unlockTimes.length > 0
      @post _tokenAmounts.length == _unlockTimes.length
      @post __post.beneficiary == _beneficiary
      @post __post.tokenAmounts == _tokenAmounts
      @post __post.unlockTimes == _unlockTimes
     */
    constructor(
        address _token,
        address _beneficiary,
        uint256[] _tokenAmounts,
        uint256[] _unlockTimes
    )
    public
    {
        require(_token != address(0));
        require(_beneficiary != address(0));
        require(_tokenAmounts.length > 0 && _unlockTimes.length > 0);
        require(_tokenAmounts.length == _unlockTimes.length);
```

```
        // vestingManager = VestingManager(msg.sender);

        token = ERC20(_token);
        beneficiary = _beneficiary;
        tokenAmounts = _tokenAmounts;
        unlockTimes = _unlockTimes;
    }

    /**
     * @notice Transfers vested tokens to beneficiary.
     */
    function release() public {
        require(tokenAmounts.length > 0);
        uint256 initialUnlockTime = vestingManager.getInitialUnlockTime();
        require(initialUnlockTime > 0);

        uint256 totalAmount;
        for (uint256 i = 0; i < tokenAmounts.length; i++) {
            if (block.timestamp >= initialUnlockTime.add(unlockTimes[i])) {
                totalAmount = totalAmount.add(tokenAmounts[i]);
                tokenAmounts[i] = 0;
            }
        }
        require(totalAmount > 0);

        token.transfer(beneficiary, totalAmount);
        released = released.add(totalAmount);
        emit Released(totalAmount);
    }

    /**
     * @dev Calculates the amount that has already vested but hasn't been released yet
       .
     */
    function releasableAmount() public view returns (uint256) {
        uint256 totalAmount;
        uint256 initialUnlockTime = vestingManager.getInitialUnlockTime();

        if (initialUnlockTime == 0) {
            totalAmount = 0;
        } else {
            for (uint256 i = 0; i < tokenAmounts.length; i++) {
                if (block.timestamp >= initialUnlockTime.add(unlockTimes[i])) {
                    totalAmount = totalAmount.add(tokenAmounts[i]);
                }
            }
        }

        return totalAmount;
    }

    /**
     * @dev Calculates the amount that has already vested.
     */
    function vestedAmount() public view returns (uint256) {
        return releasableAmount().add(released);
    }
```

```
}
```

File VestingManager.sol

```
pragma solidity ^0.4.23;

import "./utils/Ownable.sol";
import "./tokens/ERC20.sol";
import "./utils/SafeMath.sol";
import "./VestingVault.sol";

contract VestingManager is Ownable {
    using SafeMath for uint256;

    // ex) 1547019753
    uint256 initialUnlockTime;

    bool public initialUnlockTimeSet = false;

    event CreateVault(address vault, address beneficiary, uint256 tokenAmount);
    ERC20 public token;

    /**
     * @param _token address of ERC20 token which this contract controls
     */
    constructor(
        address _token
    )
    public
    {
        require(_token != address(0));

        token = ERC20(_token);
    }

    modifier isSetPossible() {
        require(!initialUnlockTimeSet);
        _;
    }

    function createVault(
        address _beneficiary,
        uint256[] _tokenAmounts,
        uint256[] _unlockTimes
    )
    public onlyOwner returns (address vaultAddress) {
        require(_beneficiary != address(0));
        require(_tokenAmounts.length > 0 && _unlockTimes.length > 0);
        require(_tokenAmounts.length == _unlockTimes.length);
        require(_tokenAmounts.length <= 100);
        // Prevent block gas limit exceed
        // gas(length == 1) : 420,000gas
        // 41,000 gas per 1 length+ => set maximum length : 100

        uint256 totalAmount;
        for (uint256 i = 0; i < _tokenAmounts.length; i++) {
            totalAmount = totalAmount.add(_tokenAmounts[i]);
        }

```

```
        require(totalAmount > 0);
        require(token.balanceOf(this) >= totalAmount);


        address vestingVault = address(new VestingVault(token, _beneficiary,
            _tokenAmounts, _unlockTimes));
        token.transfer(vestingVault, totalAmount);

        emit CreateVault(vestingVault, _beneficiary, totalAmount);

        vaultAddress = vestingVault;
    }

    function withdraw() public onlyOwner {
        token.transfer(owner, token.balanceOf(this));
    }

    /*@CTK setInitialUnlockTime
      @tag assume_completion
      @post owner == msg.sender
      @post !initialUnlockTimeSet
      @post __post.initialUnlockTime == _initialUnlockTime
      @post __post.initialUnlockTimeSet
     */
    function setInitialUnlockTime(uint256 _initialUnlockTime) public onlyOwner
        isSetPossible {
        initialUnlockTime = _initialUnlockTime;
        initialUnlockTimeSet = true;
    }

    /*@CTK getInitialUnlockTime
      @post __return == initialUnlockTime
     */
    function getInitialUnlockTime() public returns(uint256) {
        return initialUnlockTime;
    }
}
```

File MainToken.sol

```
pragma solidity ^0.4.23;


import "./Consts.sol";
import "./tokens/FreezableToken.sol";
import "./tokens/TransferableToken.sol";
import "./tokens/PausableToken.sol";
import "./tokens/MintableToken.sol";
import "./tokens/BurnableToken.sol";


/**
 * @title MainToken
 */
contract MainToken is Consts, FreezableToken, TransferableToken, PausableToken,
    MintableToken, BurnableToken {
    string public constant name = TOKEN_NAME; // solium-disable-line uppercase
    string public constant symbol = TOKEN_SYMBOL; // solium-disable-line uppercase
    uint8 public constant decimals = TOKEN_DECIMALS; // solium-disable-line uppercase
```

```

    uint256 public constant INITIAL_SUPPLY = TOKEN_AMOUNT * (10 ** uint256(decimals));

    /*@CTK MainToken
      @post __post.totalSupply_ == __post.balances[msg.sender]
     */
    constructor() public {
        totalSupply_ = INITIAL_SUPPLY;
        balances[msg.sender] = INITIAL_SUPPLY;
        emit Transfer(0x0, msg.sender, INITIAL_SUPPLY);
    }
}
```

File Migrations.sol

```
pragma solidity ^0.4.23;


contract Migrations {
    address public owner;
    uint public last_completed_migration; // solium-disable-line mixedcase

    constructor() public {
        owner = msg.sender;
    }

    modifier restricted() {
        require(msg.sender == owner);
        _;
    }

    /*@CTK setCompleted
      @tag assume_completion
      @post owner == msg.sender
      @post __post.last_completed_migration == completed
     */
    function setCompleted(uint completed) public restricted {
        last_completed_migration = completed;
    }

    function upgrade(address new_address) public restricted { // solium-disable-line
        mixedcase
        Migrations upgraded = Migrations(new_address);
        upgraded.setCompleted(last_completed_migration);
    }
}
```

File utils/Ownable.sol

```
pragma solidity ^0.4.23;


/**
 * @title Ownable
 * @dev The Ownable contract has an owner address, and provides basic authorization
     control
 * functions, this simplifies the implementation of "user permissions".
 */
contract Ownable {
    address public owner;
```

```


    event OwnershipRenounced(address indexed previousOwner);
    event OwnershipTransferred(address indexed previousOwner, address indexed newOwner
        );


    /**
     * @dev The Ownable constructor sets the original `owner` of the contract to the
         sender
     * account.
     */
    /*@CTK Ownable
      @post __post.owner == msg.sender
     */
    constructor() public {
        owner = msg.sender;
    }

    /**
     * @dev Throws if called by any account other than the owner.
     */
    modifier onlyOwner() {
        require(msg.sender == owner);
        _;
    }

    /**
     * @dev Allows the current owner to transfer control of the contract to a newOwner
         .
     * @param newOwner The address to transfer ownership to.
     */
    /*@CTK transferOwnership
      @tag assume_completion
      @post newOwner != address(0)
      @post owner == msg.sender
      @post __post.owner == newOwner
     */
    function transferOwnership(address newOwner) public onlyOwner {
        require(newOwner != address(0));
        emit OwnershipTransferred(owner, newOwner);
        owner = newOwner;
    }

    /**
     * @dev Allows the current owner to relinquish control of the contract.
     */
    /*@CTK renounceOwnership
      @tag assume_completion
      @post owner == msg.sender
      @post __post.owner == address(0)
     */
    function renounceOwnership() public onlyOwner {
        emit OwnershipRenounced(owner);
        owner = address(0);
    }
}
```

File utils/SafeMath.sol

```
pragma solidity ^0.4.23;


/**
 * @title SafeMath
 * @dev Math operations with safety checks that throw on error
 */
library SafeMath {
    /**
      * @dev Multiplies two numbers, throws on overflow.
      */
    /*@CTK "SafeMath mul"
      @post (a > 0) && (((a * b) / a) != b) -> __reverted
      @post __reverted -> (a > 0) && (((a * b) / a) != b)
      @post !__reverted -> c == a * b
      @post !__reverted == !__has_overflow
    */
    function mul(uint256 a, uint256 b) internal pure returns (uint256 c) {
        if (a == 0) {
            return 0;
        }
        c = a * b;
        assert(c / a == b);
        return c;
    }

    /**
    * @dev Integer division of two numbers, truncating the quotient.
    */
    /*@CTK "SafeMath div"
      @post b != 0 -> !__reverted
      @post !__reverted -> __return == a / b
      @post !__reverted -> !__has_overflow
    */
    function div(uint256 a, uint256 b) internal pure returns (uint256) {
        // assert(b > 0); // Solidity automatically throws when dividing by 0
        // uint256 c = a / b;
        // assert(a == b * c + a % b); // There is no case in which this doesn't hold
        return a / b;
    }

    /**
    * @dev Subtracts two numbers, throws on overflow (i.e. if subtrahend is greater
        than minuend).
    */
    /*@CTK "SafeMath sub"
      @post (a < b) == __reverted
      @post !__reverted -> __return == a - b
      @post !__reverted -> !__has_overflow
    */
    function sub(uint256 a, uint256 b) internal pure returns (uint256) {
        assert(b <= a);
        return a - b;
    }

    /**
    * @dev Adds two numbers, throws on overflow.
```

```
    */
    /*@CTK "SafeMath add"
      @post (a + b < a || a + b < b) == __reverted
      @post !__reverted -> c == a + b
      @post !__reverted -> !__has_overflow
    */
    function add(uint256 a, uint256 b) internal pure returns (uint256 c) {
        c = a + b;
        assert(c >= a);
        return c;
    }
}
```

File utils/Pausable.sol

```
pragma solidity ^0.4.23;


import "./Ownable.sol";

/**
 * @title Pausable
 * @dev Base contract which allows children to implement an emergency stop mechanism.
 */
contract Pausable is Ownable {
    event Pause();
    event Unpause();

    bool public paused = false;


    /**
     * @dev Modifier to make a function callable only when the contract is not paused.
     */
    modifier whenNotPaused() {
        require(!paused);
        _;
    }

    /**
     * @dev Modifier to makeWhitelist a function callable only when the contract is
       paused.
     */
    modifier whenPaused() {
        require(paused);
        _;
    }

    /**
     * @dev called by the owner to pause, triggers stopped state
     */
    /*@CTK pause
      @tag assume_completion
      @post owner == msg.sender
      @post paused == false
      @post __post.paused == true
     */
    function pause() onlyOwner whenNotPaused public {
        paused = true;
```

```
        emit Pause();
    }

    /**
     * @dev called by the owner to unpause, returns to normal state
     */
    /*@CTK unpause
      @tag assume_completion
      @post owner == msg.sender
      @post paused == true
      @post __post.paused == false
     */
    function unpause() onlyOwner whenPaused public {
        paused = false;
        emit Unpause();
    }
}
```

File tokens/TransferableToken.sol

```
pragma solidity ^0.4.23;

import "../utils/Ownable.sol";
import "./StandardToken.sol";


/**
 * @title TransferableToken
 */
contract TransferableToken is StandardToken, Ownable {
    bool public isLock;

    mapping (address => bool) public transferableAddresses;

    /*@CTK TransferableToken
      @post __post.isLock == true
      @post __post.transferableAddresses[msg.sender] == true
     */
    constructor() public {
        isLock = true;
        transferableAddresses[msg.sender] = true;
    }

    event Unlock();
    event TransferableAddressAdded(address indexed addr);
    event TransferableAddressRemoved(address indexed addr);

    /*@CTK unlock
      @tag assume_completion
      @post owner == msg.sender
      @post __post.isLock == false
     */
    function unlock() public onlyOwner {
        isLock = false;
        emit Unlock();
    }

    /*@CTK isTransferable
      @post __return == !isLock || transferableAddresses[addr]
```

```
     */
    function isTransferable(address addr) public view returns(bool) {
        return !isLock || transferableAddresses[addr];
    }

    function addTransferableAddresses(address[] addrs) public onlyOwner returns(bool
        success) {
        for (uint256 i = 0; i < addrs.length; i++) {
            if (addTransferableAddress(addrs[i])) {
                success = true;
            }
        }
    }

    function addTransferableAddress(address addr) public onlyOwner returns(bool
        success) {
        if (!transferableAddresses[addr]) {
            transferableAddresses[addr] = true;
            emit TransferableAddressAdded(addr);
            success = true;
        }
    }

    function removeTransferableAddresses(address[] addrs) public onlyOwner returns(
        bool success) {
        for (uint256 i = 0; i < addrs.length; i++) {
            if (removeTransferableAddress(addrs[i])) {
                success = true;
            }
        }
    }

    /*@CTK removeTransferableAddress_success
      @pre transferableAddresses[addr] == true
      @tag assume_completion
      @post owner == msg.sender
      @post __post.transferableAddresses[addr] == false
      @post success
     */
    function removeTransferableAddress(address addr) public onlyOwner returns(bool
        success) {
        if (transferableAddresses[addr]) {
            transferableAddresses[addr] = false;
            emit TransferableAddressRemoved(addr);
            success = true;
        }
    }

    /*@CTK transferFromTransferable
      @tag assume_completion
      @pre _to != _from
      @post !isLock || transferableAddresses[_from]
      @post __post.balances[_to] == balances[_to] + _value
      @post __post.balances[_from] == balances[_from] - _value
    */
    function transferFrom(address _from, address _to, uint256 _value) public returns (
        bool) {
        require(isTransferable(_from));
```

```
        return super.transferFrom(_from, _to, _value);
    }

    /*@CTK transferFromTransferable
      @tag assume_completion
      @pre _to != msg.sender
      @post !isLock || transferableAddresses[msg.sender]
      @post __post.balances[_to] == balances[_to] + _value
      @post __post.balances[msg.sender] == balances[msg.sender] - _value
    */
    function transfer(address _to, uint256 _value) public returns (bool) {
        require(isTransferable(msg.sender));
        return super.transfer(_to, _value);
    }
}
```

File tokens/FreezableToken.sol

```
pragma solidity ^0.4.23;

import "../utils/Ownable.sol";
import "./StandardToken.sol";


/**
 * @title FreezableToken
 * @dev Freeze transfer of the specific addresses, if the address is hacked
 */
contract FreezableToken is StandardToken, Ownable {
    mapping (address => bool) public freezeAddresses;

    event FreezableAddressAdded(address indexed addr);
    event FreezableAddressRemoved(address indexed addr);

    function addFreezableAddresses(address[] addrs) public onlyOwner returns(bool
        success) {
        for (uint256 i = 0; i < addrs.length; i++) {
            if (addFreezableAddress(addrs[i])) {
                success = true;
            }
        }
    }

    /*@CTK addFreezableAddress
      @tag assume_completion
      @pre !freezeAddresses[addr]
      @post owner == msg.sender
      @post __post.freezeAddresses[addr] == true
      @post success == true
     */
    function addFreezableAddress(address addr) public onlyOwner returns(bool success)
        {
        if (!freezeAddresses[addr]) {
            freezeAddresses[addr] = true;
            emit FreezableAddressAdded(addr);
            success = true;
        }
    }

```

```
    function removeFreezableAddresses(address[] addrs) public onlyOwner returns(bool
        success) {
        for (uint256 i = 0; i < addrs.length; i++) {
            if (removeFreezableAddress(addrs[i])) {
                success = true;
            }
        }
    }

    /*@CTK removeFreezableAddress
      @tag assume_completion
      @pre freezeAddresses[addr]
      @post owner == msg.sender
      @post __post.freezeAddresses[addr] == false
      @post success == true
     */
    function removeFreezableAddress(address addr) public onlyOwner returns(bool
        success) {
        if (freezeAddresses[addr]) {
            freezeAddresses[addr] = false;
            emit FreezableAddressRemoved(addr);
            success = true;
        }
    }

    /*@CTK transferFromFreezable
      @tag assume_completion
      @pre _from != _to
      @post !freezeAddresses[_from]
      @post !freezeAddresses[_to]
      @post __post.balances[_from] == balances[_from] - _value
      @post __post.balances[_to] == balances[_to] + _value
     */
    function transferFrom(address _from, address _to, uint256 _value) public returns (
        bool) {
        require(!freezeAddresses[_from]);
        require(!freezeAddresses[_to]);
        return super.transferFrom(_from, _to, _value);
    }

    /*@CTK transferFreezable
      @tag assume_completion
      @pre msg.sender != _to
      @post !freezeAddresses[msg.sender]
      @post !freezeAddresses[_to]
      @post __post.balances[msg.sender] == balances[msg.sender] - _value
      @post __post.balances[_to] == balances[_to] + _value
     */
    function transfer(address _to, uint256 _value) public returns (bool) {
        require(!freezeAddresses[msg.sender]);
        require(!freezeAddresses[_to]);
        return super.transfer(_to, _value);
    }
}
```

File tokens/StandardToken.sol

```
pragma solidity ^0.4.23;

```

```
import "./BasicToken.sol";
import "./ERC20.sol";

/**
 * @title Standard ERC20 tokens
 *
 * @dev Implementation of the basic standard tokens.
 * @dev https://github.com/ethereum/EIPs/issues/20
 * @dev Based on code by FirstBlood: https://github.com/Firstbloodio/token/blob/master
     /smart_contract/FirstBloodToken.sol
 */
contract StandardToken is ERC20, BasicToken {
    mapping (address => mapping (address => uint256)) internal allowed;


    /**
     * @dev Transfer tokens from one address to another
     * @param _from address The address which you want to send tokens from
     * @param _to address The address which you want to transfer to
     * @param _value uint256 the amount of tokens to be transferred
     */
    /*@CTK "transferFrom"
      @tag assume_completion
      @pre _from != _to
      @pre balances[_from] >= _value
      @pre allowed[_from][msg.sender] >= _value
      @post __post.balances[_to] == balances[_to] + _value
      @post __post.balances[_from] == balances[_from] - _value
      @post __post.allowed[_from][msg.sender] == allowed[_from][msg.sender] - _value
    */
    function transferFrom(address _from, address _to, uint256 _value) public returns (
        bool) {
        require(_to != address(0));
        require(_value <= balances[_from]);
        require(_value <= allowed[_from][msg.sender]);

        balances[_from] = balances[_from].sub(_value);
        balances[_to] = balances[_to].add(_value);
        allowed[_from][msg.sender] = allowed[_from][msg.sender].sub(_value);
        emit Transfer(_from, _to, _value);
        return true;
    }

    /**
     * @dev Approve the passed address to spend the specified amount of tokens on
         behalf of msg.sender.
     *
     * Beware that changing an allowance with this method brings the risk that someone
          may use both the old
     * and the new allowance by unfortunate transaction ordering. One possible
         solution to mitigate this
     * race condition is to first reduce the spender's allowance to 0 and set the
         desired value afterwards:
     * https://github.com/ethereum/EIPs/issues/20#issuecomment-263524729
     * @param _spender The address which will spend the funds.
     * @param _value The amount of tokens to be spent.
     */
    /*@CTK "approve"
```

```
     @tag assume_completion
     @post __post.allowed[msg.sender][_spender] == _value
   */
   function approve(address _spender, uint256 _value) public returns (bool) {
       allowed[msg.sender][_spender] = _value;
       emit Approval(msg.sender, _spender, _value);
       return true;
   }

   /**
    * @dev Function to check the amount of tokens that an owner allowed to a spender.
    * @param _owner address The address which owns the funds.
    * @param _spender address The address which will spend the funds.
    * @return A uint256 specifying the amount of tokens still available for the spender.
    */
   /*@CTK allowance
     @post __return == allowed[_owner][_spender]
   */
   function allowance(address _owner, address _spender) public view returns (uint256) {
       return allowed[_owner][_spender];
   }

   /**
    * @dev Increase the amount of tokens that an owner allowed to a spender.
    *
    * approve should be called when allowed[_spender] == 0. To increment
    * allowed value is better to use this function to avoid 2 calls (and wait until
    * the first transaction is mined)
    * From MonolithDAO Token.sol
    * @param _spender The address which will spend the funds.
    * @param _addedValue The amount of tokens to increase the allowance by.
    */
   /*@CTK increaseApproval
     @tag assume_completion
     @post __post.allowed[msg.sender][_spender] == allowed[msg.sender][_spender] + _addedValue
   */
   function increaseApproval(address _spender, uint _addedValue) public returns (bool) {
       allowed[msg.sender][_spender] = allowed[msg.sender][_spender].add(_addedValue);
       emit Approval(msg.sender, _spender, allowed[msg.sender][_spender]);
       return true;
   }

   /**
    * @dev Decrease the amount of tokens that an owner allowed to a spender.
    *
    * approve should be called when allowed[_spender] == 0. To decrement
    * allowed value is better to use this function to avoid 2 calls (and wait until
    * the first transaction is mined)
    * From MonolithDAO Token.sol
    * @param _spender The address which will spend the funds.
    * @param _subtractedValue The amount of tokens to decrease the allowance by.
    */
   /*@CTK decreaseApproval0
     @tag assume_completion
```

```
     @pre allowed[msg.sender][_spender] <= _subtractedValue
     @post __post.allowed[msg.sender][_spender] == 0
   */
   /*@CTK decreaseApproval
     @tag assume_completion
     @pre allowed[msg.sender][_spender] > _subtractedValue
     @post __post.allowed[msg.sender][_spender] == allowed[msg.sender][_spender] -
         _subtractedValue
   */
   function decreaseApproval(address _spender, uint _subtractedValue) public returns
       (bool) {
       uint oldValue = allowed[msg.sender][_spender];
       if (_subtractedValue > oldValue) {
           allowed[msg.sender][_spender] = 0;
       } else {
           allowed[msg.sender][_spender] = oldValue.sub(_subtractedValue);
       }
       emit Approval(msg.sender, _spender, allowed[msg.sender][_spender]);
       return true;
   }
}
```

File tokens/MintableToken.sol

```
pragma solidity ^0.4.23;

import "../utils/Ownable.sol";
import "./StandardToken.sol";


/**
 * @title Mintable token
 * @dev Simple ERC20 Token example, with mintable token creation
 * @dev Issue: * https://github.com/OpenZeppelin/openzeppelin-solidity/issues/120
 * Based on code by TokenMarketNet: https://github.com/TokenMarketNet/ico/blob/master/
     contracts/MintableToken.sol
 */
contract MintableToken is StandardToken, Ownable {
   event Mint(address indexed to, uint256 amount);
   event MintFinished();

   bool public mintingFinished = false;


   modifier canMint() {
       require(!mintingFinished);
       _;
   }

   /**
    * @dev Function to mint tokens
    * @param _to The address that will receive the minted tokens.
    * @param _amount The amount of tokens to mint.
    * @return A boolean that indicates if the operation was successful.
    */
   /*@CTK mint
     @tag assume_completion
     @post owner == msg.sender
     @post !mintingFinished
```

```
     @post __post.totalSupply_ == totalSupply_ + _amount
     @post __post.balances[_to] == balances[_to] + _amount
    */
   function mint(address _to, uint256 _amount) onlyOwner canMint public returns (bool
       ) {
       totalSupply_ = totalSupply_.add(_amount);
       balances[_to] = balances[_to].add(_amount);
       emit Mint(_to, _amount);
       emit Transfer(address(0), _to, _amount);
       return true;
   }

   /**
    * @dev Function to stop minting new tokens.
    * @return True if the operation was successful.
    */
   /*@CTK finishMinting
     @tag assume_completion
     @post owner == msg.sender
     @post !mintingFinished
     @post __post.mintingFinished == true
    */
   function finishMinting() onlyOwner canMint public returns (bool) {
       mintingFinished = true;
       emit MintFinished();
       return true;
   }
}
```

File tokens/BasicToken.sol

```
pragma solidity ^0.4.23;


import "./ERC20Basic.sol";
import "../utils/SafeMath.sol";


/**
 * @title Basic tokens
 * @dev Basic version of StandardToken, with no allowances.
 */
contract BasicToken is ERC20Basic {
   using SafeMath for uint256;

   mapping(address => uint256) balances;

   uint256 totalSupply_;

   /**
   * @dev total number of tokens in existence
   */
   /*@CTK totalSupply
     @post __return == totalSupply_
    */
   function totalSupply() public view returns (uint256) {
       return totalSupply_;
   }

```

```
    /**
    * @dev transfer tokens for a specified address
    * @param _to The address to transfer to.
    * @param _value The amount to be transferred.
    */
    /*@CTK transfer
      @tag assume_completion
      @pre _to != msg.sender
      @post __post.balances[msg.sender] == balances[msg.sender] - _value
      @post __post.balances[_to] == balances[_to] + _value
    */
    function transfer(address _to, uint256 _value) public returns (bool) {
        require(_to != address(0));
        require(_value <= balances[msg.sender]);

        balances[msg.sender] = balances[msg.sender].sub(_value);
        balances[_to] = balances[_to].add(_value);
        emit Transfer(msg.sender, _to, _value);
        return true;
    }

    /**
    * @dev Gets the balance of the specified address.
    * @param _owner The address to query the the balance of.
    * @return An uint256 representing the amount owned by the passed address.
    */
    /*@CTK balanceOf
      @post __return == balances[_owner]
    */
    function balanceOf(address _owner) public view returns (uint256) {
        return balances[_owner];
    }

}
```

File tokens/PausableToken.sol

```
pragma solidity ^0.4.23;

import "./StandardToken.sol";
import "../utils/Pausable.sol";


/**
 * @title Pausable tokens
 * @dev StandardToken modified with pausable transfers.
 **/
contract PausableToken is StandardToken, Pausable {
    /*@CTK transferPausable
      @tag assume_completion
      @post paused == false
     */
    function transfer(address _to, uint256 _value) public whenNotPaused returns (bool)
        {
        return super.transfer(_to, _value);
    }

    /*@CTK transferFromPausable
      @tag assume_completion
```

```
     @post paused == false
    */
   function transferFrom(address _from, address _to, uint256 _value) public whenNotPaused returns (bool) {
       return super.transferFrom(_from, _to, _value);
   }

   /*@CTK approvePausable
     @tag assume_completion
     @post paused == false
    */
   function approve(address _spender, uint256 _value) public whenNotPaused returns (bool) {
       return super.approve(_spender, _value);
   }

   /*@CTK increaseApprovalPausable
     @tag assume_completion
     @post paused == false
    */
   function increaseApproval(address _spender, uint _addedValue) public whenNotPaused returns (bool success) {
       return super.increaseApproval(_spender, _addedValue);
   }

   /*@CTK decreaseApprovalPausable
     @tag assume_completion
     @post paused == false
    */
   function decreaseApproval(address _spender, uint _subtractedValue) public whenNotPaused returns (bool success) {
       return super.decreaseApproval(_spender, _subtractedValue);
   }
}
```

File tokens/BurnableToken.sol

```
pragma solidity ^0.4.23;

import "./BasicToken.sol";
import "../utils/Pausable.sol";


/**
 * @title Burnable Token
 * @dev Token that can be irreversibly burned (destroyed).
 */
contract BurnableToken is BasicToken, Pausable {

   event Burn(address indexed burner, uint256 value);

   /**
    * @dev Burns a specific amount of tokens.
    * @param _value The amount of tokens to be burned.
    */

   /*@CTK burn
     @tag assume_completion
     @post paused == false
```

```
     @post _value <= balances[msg.sender]
     @post __post.balances[msg.sender] == balances[msg.sender] - _value
     @post __post.totalSupply_ == totalSupply_ - _value
    */
   function burn(uint256 _value) whenNotPaused public {
       _burn(msg.sender, _value);
   }

   /*@CTK _burn
     @tag assume_completion
     @post _value <= balances[_who]
     @post __post.balances[_who] == balances[_who] - _value
     @post __post.totalSupply_ == totalSupply_ - _value
    */
   function _burn(address _who, uint256 _value) internal {
       require(_value <= balances[_who]);
       // no need to require value <= totalSupply, since that would imply the
       // sender's balance is greater than the totalSupply, which *should* be an
           assertion failure

       balances[_who] = balances[_who].sub(_value);
       totalSupply_ = totalSupply_.sub(_value);
       emit Burn(_who, _value);
       emit Transfer(_who, address(0), _value);
   }
}
```

How to read

Detail for Request 1

transferFrom to same address

Verification date	20, Oct 2018
Verification timespan	395.38 ms

CERTIK *label location*	Line 30-34 in File howtoread.sol

CERTIK *label*

```
30  /*@CTK FAIL "transferFrom to same address"
31      @tag assume_completion
32      @pre from == to
33      @post __post.allowed[from][msg.sender] ==
34  */
```

Raw code location	Line 35-41 in File howtoread.sol

Raw code

```
35  function transferFrom(address from, address to
        ) {
36      balances[from] = balances[from].sub(tokens
37      allowed[from][msg.sender] = allowed[from][
38      balances[to] = balances[to].add(tokens);
39      emit Transfer(from, to, tokens);
40      return true;
41  }
```

Counterexample — This code violates the specification

Initial environment

```
1  Counter Example:
2  Before Execution:
3      Input = {
4          from = 0x0
5          to = 0x0
6          tokens = 0x6c
7      }
8      This = 0
53         balance: 0x0
54     }
55 }
56
```

Post environment

```
57 After Execution:
58     Input = {
59         from = 0x0
60         to = 0x0
61         tokens = 0x6c
```

Static Analysis Request

INSECURE_COMPILER_VERSION

Line 1 in File VestingVault.sol

```
1 pragma solidity ^0.4.23;
```

Only these compiler versions are safe to compile your code: 0.4.25

TIMESTAMP_DEPENDENCY

Line 71 in File VestingVault.sol

```
71          if (block.timestamp >= initialUnlockTime.add(unlockTimes[i])) {
```

"block.timestamp" can be influenced by minors to some degree

TIMESTAMP_DEPENDENCY

Line 94 in File VestingVault.sol

```
94              if (block.timestamp >= initialUnlockTime.add(unlockTimes[i])) {
```

"block.timestamp" can be influenced by minors to some degree

INSECURE_COMPILER_VERSION

Line 1 in File VestingManager.sol

```
1 pragma solidity ^0.4.23;
```

Only these compiler versions are safe to compile your code: 0.4.25

INSECURE_COMPILER_VERSION

Line 1 in File MainToken.sol

```
1 pragma solidity ^0.4.23;
```

Only these compiler versions are safe to compile your code: 0.4.25

INSECURE_COMPILER_VERSION

Line 1 in File Migrations.sol

```
1 pragma solidity ^0.4.23;
```

Only these compiler versions are safe to compile your code: 0.4.25

INSECURE_COMPILER_VERSION

Line 1 in File Ownable.sol

```
1 pragma solidity ^0.4.23;
```

Only these compiler versions are safe to compile your code: 0.4.25

INSECURE_COMPILER_VERSION

Line 1 in File SafeMath.sol

```
1 pragma solidity ^0.4.23;
```

Only these compiler versions are safe to compile your code: 0.4.25

INSECURE_COMPILER_VERSION

Line 1 in File Pausable.sol

```
1 pragma solidity ^0.4.23;
```

Only these compiler versions are safe to compile your code: 0.4.25

INSECURE_COMPILER_VERSION

Line 1 in File TransferableToken.sol

```
1 pragma solidity ^0.4.23;
```

Only these compiler versions are safe to compile your code: 0.4.25

INSECURE_COMPILER_VERSION

Line 1 in File FreezableToken.sol

```
1 pragma solidity ^0.4.23;
```

Only these compiler versions are safe to compile your code: 0.4.25

INSECURE_COMPILER_VERSION

Line 1 in File StandardToken.sol

```
1 pragma solidity ^0.4.23;
```

Only these compiler versions are safe to compile your code: 0.4.25

INSECURE_COMPILER_VERSION

Line 1 in File MintableToken.sol

```
1 pragma solidity ^0.4.23;
```

Only these compiler versions are safe to compile your code: 0.4.25

INSECURE_COMPILER_VERSION

Line 1 in File BasicToken.sol

```
1 pragma solidity ^0.4.23;
```

Only these compiler versions are safe to compile your code: 0.4.25

INSECURE_COMPILER_VERSION

Line 1 in File PausableToken.sol

```
1 pragma solidity ^0.4.23;
```

Only these compiler versions are safe to compile your code: 0.4.25

INSECURE_COMPILER_VERSION

Line 1 in File BurnableToken.sol

```
1 pragma solidity ^0.4.23;
```

Only these compiler versions are safe to compile your code: 0.4.25

Formal Verification Request 1

VestingVault

07, Feb 2019
79.37 ms

Line 30-39 in File VestingVault.sol

```
30  /*@CTK VestingVault
31    @tag assume_completion
32    @post _token != address(0)
33    @post _beneficiary != address(0)
34    @post _tokenAmounts.length > 0 && _unlockTimes.length > 0
35    @post _tokenAmounts.length == _unlockTimes.length
36    @post __post.beneficiary == _beneficiary
37    @post __post.tokenAmounts == _tokenAmounts
38    @post __post.unlockTimes == _unlockTimes
39   */
```

Line 40-59 in File VestingVault.sol

```
40  constructor(
41      address _token,
42      address _beneficiary,
43      uint256[] _tokenAmounts,
44      uint256[] _unlockTimes
45  )
46  public
47  {
48      require(_token != address(0));
49      require(_beneficiary != address(0));
50      require(_tokenAmounts.length > 0 && _unlockTimes.length > 0);
51      require(_tokenAmounts.length == _unlockTimes.length);
52
53      // vestingManager = VestingManager(msg.sender);
54
55      token = ERC20(_token);
56      beneficiary = _beneficiary;
57      tokenAmounts = _tokenAmounts;
58      unlockTimes = _unlockTimes;
59  }
```

The code meets the specification

Formal Verification Request 2

setInitialUnlockTime

07, Feb 2019
29.74 ms

Line 72-78 in File VestingManager.sol

```
72  /*@CTK setInitialUnlockTime
73    @tag assume_completion
74    @post owner == msg.sender
```

```
75      @post !initialUnlockTimeSet
76      @post __post.initialUnlockTime == _initialUnlockTime
77      @post __post.initialUnlockTimeSet
78     */
```

Line 79-82 in File VestingManager.sol

```
79    function setInitialUnlockTime(uint256 _initialUnlockTime) public onlyOwner
          isSetPossible {
80        initialUnlockTime = _initialUnlockTime;
81        initialUnlockTimeSet = true;
82    }
```

The code meets the specification

Formal Verification Request 3

getInitialUnlockTime

07, Feb 2019
6.51 ms

Line 84-86 in File VestingManager.sol

```
84    /*@CTK getInitialUnlockTime
85      @post __return == initialUnlockTime
86     */
```

Line 87-89 in File VestingManager.sol

```
87    function getInitialUnlockTime() public returns(uint256) {
88        return initialUnlockTime;
89    }
```

The code meets the specification

Formal Verification Request 4

MainToken

07, Feb 2019
21.36 ms

Line 22-24 in File MainToken.sol

```
22    /*@CTK MainToken
23      @post __post.totalSupply_ == __post.balances[msg.sender]
24     */
```

Line 25-29 in File MainToken.sol

```
25    constructor() public {
26        totalSupply_ = INITIAL_SUPPLY;
27        balances[msg.sender] = INITIAL_SUPPLY;
28        emit Transfer(0x0, msg.sender, INITIAL_SUPPLY);
29    }
```

The code meets the specification

Formal Verification Request 5

setCompleted

07, Feb 2019
20.18 ms

Line 17-21 in File Migrations.sol

```
17  /*@CTK setCompleted
18    @tag assume_completion
19    @post owner == msg.sender
20    @post __post.last_completed_migration == completed
21   */
```

Line 22-24 in File Migrations.sol

```
22  function setCompleted(uint completed) public restricted {
23      last_completed_migration = completed;
24  }
```

The code meets the specification

Formal Verification Request 6

Ownable

07, Feb 2019
6.57 ms

Line 21-23 in File Ownable.sol

```
21  /*@CTK Ownable
22    @post __post.owner == msg.sender
23   */
```

Line 24-26 in File Ownable.sol

```
24  constructor() public {
25      owner = msg.sender;
26  }
```

The code meets the specification

Formal Verification Request 7

transferOwnership

07, Feb 2019
26.17 ms

Line 40-45 in File Ownable.sol

```
40    /*@CTK transferOwnership
41      @tag assume_completion
42      @post newOwner != address(0)
43      @post owner == msg.sender
44      @post __post.owner == newOwner
45     */
```

Line 46-50 in File Ownable.sol

```
46    function transferOwnership(address newOwner) public onlyOwner {
47        require(newOwner != address(0));
48        emit OwnershipTransferred(owner, newOwner);
49        owner = newOwner;
50    }
```

✅ The code meets the specification

Formal Verification Request 8

renounceOwnership

📅 07, Feb 2019
⏱ 17.19 ms

Line 55-59 in File Ownable.sol

```
55    /*@CTK renounceOwnership
56      @tag assume_completion
57      @post owner == msg.sender
58      @post __post.owner == address(0)
59     */
```

Line 60-63 in File Ownable.sol

```
60    function renounceOwnership() public onlyOwner {
61        emit OwnershipRenounced(owner);
62        owner = address(0);
63    }
```

✅ The code meets the specification

Formal Verification Request 9

SafeMath mul

📅 07, Feb 2019
⏱ 475.65 ms

Line 12-17 in File SafeMath.sol

```
12    /*@CTK "SafeMath mul"
13      @post (a > 0) && (((a * b) / a) != b) -> __reverted
14      @post __reverted -> (a > 0) && (((a * b) / a) != b)
15      @post !__reverted -> c == a * b
16      @post !__reverted == !__has_overflow
17    */
```

Line 18-25 in File SafeMath.sol

```
18  function mul(uint256 a, uint256 b) internal pure returns (uint256 c) {
19      if (a == 0) {
20          return 0;
21      }
22      c = a * b;
23      assert(c / a == b);
24      return c;
25  }
```

The code meets the specification

Formal Verification Request 10

SafeMath div

07, Feb 2019
7.53 ms

Line 30-34 in File SafeMath.sol

```
30  /*@CTK "SafeMath div"
31    @post b != 0 -> !__reverted
32    @post !__reverted -> __return == a / b
33    @post !__reverted -> !__has_overflow
34  */
```

Line 35-40 in File SafeMath.sol

```
35  function div(uint256 a, uint256 b) internal pure returns (uint256) {
36      // assert(b > 0); // Solidity automatically throws when dividing by 0
37      // uint256 c = a / b;
38      // assert(a == b * c + a % b); // There is no case in which this doesn't hold
39      return a / b;
40  }
```

The code meets the specification

Formal Verification Request 11

SafeMath sub

07, Feb 2019
14.41 ms

Line 45-49 in File SafeMath.sol

```
45  /*@CTK "SafeMath sub"
46    @post (a < b) == __reverted
47    @post !__reverted -> __return == a - b
48    @post !__reverted -> !__has_overflow
49  */
```

Line 50-53 in File SafeMath.sol

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

The code meets the specification

Formal Verification Request 12

SafeMath add

07, Feb 2019
18.07 ms

Line 58-62 in File SafeMath.sol

```
/*@CTK "SafeMath add"
  @post (a + b < a || a + b < b) == __reverted
  @post !__reverted -> c == a + b
  @post !__reverted -> !__has_overflow
*/
```

Line 63-67 in File SafeMath.sol

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

The code meets the specification

Formal Verification Request 13

pause

07, Feb 2019
30.63 ms

Line 36-41 in File Pausable.sol

```
/*@CTK pause
  @tag assume_completion
  @post owner == msg.sender
  @post paused == false
  @post __post.paused == true
 */
```

Line 42-45 in File Pausable.sol

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

The code meets the specification

Formal Verification Request 14

unpause

07, Feb 2019
29.32 ms

Line 50-55 in File Pausable.sol

```
50  /*@CTK unpause
51    @tag assume_completion
52    @post owner == msg.sender
53    @post paused == true
54    @post __post.paused == false
55   */
```

Line 56-59 in File Pausable.sol

```
56  function unpause() onlyOwner whenPaused public {
57      paused = false;
58      emit Unpause();
59  }
```

The code meets the specification

Formal Verification Request 15

TransferableToken

07, Feb 2019
10.72 ms

Line 15-18 in File TransferableToken.sol

```
15  /*@CTK TransferableToken
16    @post __post.isLock == true
17    @post __post.transferableAddresses[msg.sender] == true
18   */
```

Line 19-22 in File TransferableToken.sol

```
19  constructor() public {
20      isLock = true;
21      transferableAddresses[msg.sender] = true;
22  }
```

The code meets the specification

Formal Verification Request 16

unlock

07, Feb 2019
16.02 ms

Line 28-32 in File TransferableToken.sol

```
28    /*@CTK unlock
29      @tag assume_completion
30      @post owner == msg.sender
31      @post __post.isLock == false
32     */
```

Line 33-36 in File TransferableToken.sol

```
33    function unlock() public onlyOwner {
34        isLock = false;
35        emit Unlock();
36    }
```

✅ The code meets the specification

Formal Verification Request 17

isTransferable

📅 07, Feb 2019
⏱ 7.8 ms

Line 38-40 in File TransferableToken.sol

```
38    /*@CTK isTransferable
39      @post __return == !isLock || transferableAddresses[addr]
40     */
```

Line 41-43 in File TransferableToken.sol

```
41    function isTransferable(address addr) public view returns(bool) {
42        return !isLock || transferableAddresses[addr];
43    }
```

✅ The code meets the specification

Formal Verification Request 18

removeTransferableAddress_success

📅 07, Feb 2019
⏱ 23.1 ms

Line 69-75 in File TransferableToken.sol

```
69    /*@CTK removeTransferableAddress_success
70      @pre transferableAddresses[addr] == true
71      @tag assume_completion
72      @post owner == msg.sender
73      @post __post.transferableAddresses[addr] == false
74      @post success
75     */
```

Line 76-82 in File TransferableToken.sol

```
function removeTransferableAddress(address addr) public onlyOwner returns(bool
    success) {
    if (transferableAddresses[addr]) {
        transferableAddresses[addr] = false;
        emit TransferableAddressRemoved(addr);
        success = true;
    }
}
```

✅ The code meets the specification

Formal Verification Request 19

transferFromTransferable

📅 07, Feb 2019
⏱ 502.02 ms

Line 84-90 in File TransferableToken.sol

```
/*@CTK transferFromTransferable
  @tag assume_completion
  @pre _to != _from
  @post !isLock || transferableAddresses[_from]
  @post __post.balances[_to] == balances[_to] + _value
  @post __post.balances[_from] == balances[_from] - _value
*/
```

Line 91-94 in File TransferableToken.sol

```
function transferFrom(address _from, address _to, uint256 _value) public returns (
    bool) {
    require(isTransferable(_from));
    return super.transferFrom(_from, _to, _value);
}
```

✅ The code meets the specification

Formal Verification Request 20

transferFromTransferable

📅 07, Feb 2019
⏱ 313.48 ms

Line 96-102 in File TransferableToken.sol

```
/*@CTK transferFromTransferable
  @tag assume_completion
  @pre _to != msg.sender
  @post !isLock || transferableAddresses[msg.sender]
  @post __post.balances[_to] == balances[_to] + _value
  @post __post.balances[msg.sender] == balances[msg.sender] - _value
*/
```

Line 103-106 in File TransferableToken.sol

```
103 function transfer(address _to, uint256 _value) public returns (bool) {
104     require(isTransferable(msg.sender));
105     return super.transfer(_to, _value);
106 }
```

The code meets the specification

Formal Verification Request 21

addFreezableAddress

07, Feb 2019
21.8 ms

Line 25-31 in File FreezableToken.sol

```
25 /*@CTK addFreezableAddress
26   @tag assume_completion
27   @pre !freezeAddresses[addr]
28   @post owner == msg.sender
29   @post __post.freezeAddresses[addr] == true
30   @post success == true
31  */
```

Line 32-38 in File FreezableToken.sol

```
32 function addFreezableAddress(address addr) public onlyOwner returns(bool success)
       {
33     if (!freezeAddresses[addr]) {
34         freezeAddresses[addr] = true;
35         emit FreezableAddressAdded(addr);
36         success = true;
37     }
38 }
```

The code meets the specification

Formal Verification Request 22

removeFreezableAddress

07, Feb 2019
22.69 ms

Line 48-54 in File FreezableToken.sol

```
48 /*@CTK removeFreezableAddress
49   @tag assume_completion
50   @pre freezeAddresses[addr]
51   @post owner == msg.sender
52   @post __post.freezeAddresses[addr] == false
53   @post success == true
54  */
```

Line 55-61 in File FreezableToken.sol

```
55    function removeFreezableAddress(address addr) public onlyOwner returns(bool
          success) {
56        if (freezeAddresses[addr]) {
57            freezeAddresses[addr] = false;
58            emit FreezableAddressRemoved(addr);
59            success = true;
60        }
61    }
```

The code meets the specification

Formal Verification Request 23

transferFromFreezable

07, Feb 2019
494.65 ms

Line 63-70 in File FreezableToken.sol

```
63    /*@CTK transferFromFreezable
64      @tag assume_completion
65      @pre _from != _to
66      @post !freezeAddresses[_from]
67      @post !freezeAddresses[_to]
68      @post __post.balances[_from] == balances[_from] - _value
69      @post __post.balances[_to] == balances[_to] + _value
70     */
```

Line 71-75 in File FreezableToken.sol

```
71    function transferFrom(address _from, address _to, uint256 _value) public returns (
          bool) {
72        require(!freezeAddresses[_from]);
73        require(!freezeAddresses[_to]);
74        return super.transferFrom(_from, _to, _value);
75    }
```

The code meets the specification

Formal Verification Request 24

transferFreezable

07, Feb 2019
304.29 ms

Line 77-84 in File FreezableToken.sol

```
77    /*@CTK transferFreezable
78      @tag assume_completion
79      @pre msg.sender != _to
80      @post !freezeAddresses[msg.sender]
```

```
81      @post !freezeAddresses[_to]
82      @post __post.balances[msg.sender] == balances[msg.sender] - _value
83      @post __post.balances[_to] == balances[_to] + _value
84     */
```

Line 85-89 in File FreezableToken.sol

```
85    function transfer(address _to, uint256 _value) public returns (bool) {
86        require(!freezeAddresses[msg.sender]);
87        require(!freezeAddresses[_to]);
88        return super.transfer(_to, _value);
89    }
```

The code meets the specification

Formal Verification Request 25

transferFrom

07, Feb 2019
420.42 ms

Line 23-31 in File StandardToken.sol

```
23    /*@CTK "transferFrom"
24      @tag assume_completion
25      @pre _from != _to
26      @pre balances[_from] >= _value
27      @pre allowed[_from][msg.sender] >= _value
28      @post __post.balances[_to] == balances[_to] + _value
29      @post __post.balances[_from] == balances[_from] - _value
30      @post __post.allowed[_from][msg.sender] == allowed[_from][msg.sender] - _value
31    */
```

Line 32-42 in File StandardToken.sol

```
32    function transferFrom(address _from, address _to, uint256 _value) public returns (
          bool) {
33        require(_to != address(0));
34        require(_value <= balances[_from]);
35        require(_value <= allowed[_from][msg.sender]);
36
37        balances[_from] = balances[_from].sub(_value);
38        balances[_to] = balances[_to].add(_value);
39        allowed[_from][msg.sender] = allowed[_from][msg.sender].sub(_value);
40        emit Transfer(_from, _to, _value);
41        return true;
42    }
```

The code meets the specification

Formal Verification Request 26

approve

07, Feb 2019
11.48 ms

Line 54-57 in File StandardToken.sol

```
54  /*@CTK "approve"
55    @tag assume_completion
56    @post __post.allowed[msg.sender][_spender] == _value
57  */
```

Line 58-62 in File StandardToken.sol

```
58  function approve(address _spender, uint256 _value) public returns (bool) {
59      allowed[msg.sender][_spender] = _value;
60      emit Approval(msg.sender, _spender, _value);
61      return true;
62  }
```

✅ The code meets the specification

Formal Verification Request 27

allowance

📅 07, Feb 2019
⏱ 5.9 ms

Line 70-72 in File StandardToken.sol

```
70  /*@CTK allowance
71    @post __return == allowed[_owner][_spender]
72  */
```

Line 73-75 in File StandardToken.sol

```
73  function allowance(address _owner, address _spender) public view returns (uint256)
        {
74      return allowed[_owner][_spender];
75  }
```

✅ The code meets the specification

Formal Verification Request 28

increaseApproval

📅 07, Feb 2019
⏱ 37.0 ms

Line 87-90 in File StandardToken.sol

```
87  /*@CTK increaseApproval
88    @tag assume_completion
89    @post __post.allowed[msg.sender][_spender] == allowed[msg.sender][_spender] +
          _addedValue
90  */
```

Line 91-95 in File StandardToken.sol

```
function increaseApproval(address _spender, uint _addedValue) public returns (bool) {
    allowed[msg.sender][_spender] = allowed[msg.sender][_spender].add(_addedValue);
    emit Approval(msg.sender, _spender, allowed[msg.sender][_spender]);
    return true;
}
```

The code meets the specification

Formal Verification Request 29

decreaseApproval0

07, Feb 2019
67.31 ms

Line 107-111 in File StandardToken.sol

```
/*@CTK decreaseApproval0
  @tag assume_completion
  @pre allowed[msg.sender][_spender] <= _subtractedValue
  @post __post.allowed[msg.sender][_spender] == 0
*/
```

Line 117-126 in File StandardToken.sol

```
function decreaseApproval(address _spender, uint _subtractedValue) public returns (bool) {
    uint oldValue = allowed[msg.sender][_spender];
    if (_subtractedValue > oldValue) {
        allowed[msg.sender][_spender] = 0;
    } else {
        allowed[msg.sender][_spender] = oldValue.sub(_subtractedValue);
    }
    emit Approval(msg.sender, _spender, allowed[msg.sender][_spender]);
    return true;
}
```

The code meets the specification

Formal Verification Request 30

decreaseApproval

07, Feb 2019
21.07 ms

Line 112-116 in File StandardToken.sol

```
/*@CTK decreaseApproval
  @tag assume_completion
  @pre allowed[msg.sender][_spender] > _subtractedValue
  @post __post.allowed[msg.sender][_spender] == allowed[msg.sender][_spender] - _subtractedValue
*/
```

Line 117-126 in File StandardToken.sol

```
117 function decreaseApproval(address _spender, uint _subtractedValue) public returns
        (bool) {
118     uint oldValue = allowed[msg.sender][_spender];
119     if (_subtractedValue > oldValue) {
120         allowed[msg.sender][_spender] = 0;
121     } else {
122         allowed[msg.sender][_spender] = oldValue.sub(_subtractedValue);
123     }
124     emit Approval(msg.sender, _spender, allowed[msg.sender][_spender]);
125     return true;
126 }
```

✅ The code meets the specification

Formal Verification Request 31

mint

📅 07, Feb 2019
⏱ 225.68 ms

Line 31-37 in File MintableToken.sol

```
31 /*@CTK mint
32   @tag assume_completion
33   @post owner == msg.sender
34   @post !mintingFinished
35   @post __post.totalSupply_ == totalSupply_ + _amount
36   @post __post.balances[_to] == balances[_to] + _amount
37  */
```

Line 38-44 in File MintableToken.sol

```
38 function mint(address _to, uint256 _amount) onlyOwner canMint public returns (bool
       ) {
39     totalSupply_ = totalSupply_.add(_amount);
40     balances[_to] = balances[_to].add(_amount);
41     emit Mint(_to, _amount);
42     emit Transfer(address(0), _to, _amount);
43     return true;
44 }
```

✅ The code meets the specification

Formal Verification Request 32

finishMinting

📅 07, Feb 2019
⏱ 29.77 ms

Line 50-55 in File MintableToken.sol

```
50    /*@CTK finishMinting
51      @tag assume_completion
52      @post owner == msg.sender
53      @post !mintingFinished
54      @post __post.mintingFinished == true
55     */
```

Line 56-60 in File MintableToken.sol

```
56    function finishMinting() onlyOwner canMint public returns (bool) {
57        mintingFinished = true;
58        emit MintFinished();
59        return true;
60    }
```

✅ The code meets the specification

Formal Verification Request 33

totalSupply

📅 07, Feb 2019
⏱ 6.81 ms

Line 22-24 in File BasicToken.sol

```
22    /*@CTK totalSupply
23      @post __return == totalSupply_
24     */
```

Line 25-27 in File BasicToken.sol

```
25    function totalSupply() public view returns (uint256) {
26        return totalSupply_;
27    }
```

✅ The code meets the specification

Formal Verification Request 34

transfer

📅 07, Feb 2019
⏱ 241.11 ms

Line 34-39 in File BasicToken.sol

```
34    /*@CTK transfer
35      @tag assume_completion
36      @pre _to != msg.sender
37      @post __post.balances[msg.sender] == balances[msg.sender] - _value
38      @post __post.balances[_to] == balances[_to] + _value
39    */
```

Line 40-48 in File BasicToken.sol

```
    function transfer(address _to, uint256 _value) public returns (bool) {
        require(_to != address(0));
        require(_value <= balances[msg.sender]);

        balances[msg.sender] = balances[msg.sender].sub(_value);
        balances[_to] = balances[_to].add(_value);
        emit Transfer(msg.sender, _to, _value);
        return true;
    }
```

The code meets the specification

Formal Verification Request 35

balanceOf

07, Feb 2019
6.38 ms

Line 55-57 in File BasicToken.sol

```
55  /*@CTK balanceOf
56    @post __return == balances[_owner]
57  */
```

Line 58-60 in File BasicToken.sol

```
58  function balanceOf(address _owner) public view returns (uint256) {
59      return balances[_owner];
60  }
```

The code meets the specification

Formal Verification Request 36

transferPausable

07, Feb 2019
209.09 ms

Line 12-15 in File PausableToken.sol

```
12  /*@CTK transferPausable
13    @tag assume_completion
14    @post paused == false
15   */
```

Line 16-18 in File PausableToken.sol

```
16  function transfer(address _to, uint256 _value) public whenNotPaused returns (bool)
        {
17      return super.transfer(_to, _value);
18  }
```

The code meets the specification

Formal Verification Request 37

transferFromPausable

07, Feb 2019
302.13 ms

Line 20-23 in File PausableToken.sol

```
20  /*@CTK transferFromPausable
21    @tag assume_completion
22    @post paused == false
23   */
```

Line 24-26 in File PausableToken.sol

```
24  function transferFrom(address _from, address _to, uint256 _value) public
        whenNotPaused returns (bool) {
25      return super.transferFrom(_from, _to, _value);
26  }
```

The code meets the specification

Formal Verification Request 38

approvePausable

07, Feb 2019
48.85 ms

Line 28-31 in File PausableToken.sol

```
28  /*@CTK approvePausable
29    @tag assume_completion
30    @post paused == false
31   */
```

Line 32-34 in File PausableToken.sol

```
32  function approve(address _spender, uint256 _value) public whenNotPaused returns (
        bool) {
33      return super.approve(_spender, _value);
34  }
```

The code meets the specification

Formal Verification Request 39

increaseApprovalPausable

07, Feb 2019
101.71 ms

Line 36-39 in File PausableToken.sol

```
36    /*@CTK increaseApprovalPausable
37      @tag assume_completion
38      @post paused == false
39     */
```

Line 40-42 in File PausableToken.sol

```
40    function increaseApproval(address _spender, uint _addedValue) public whenNotPaused
          returns (bool success) {
41        return super.increaseApproval(_spender, _addedValue);
42    }
```

✅ The code meets the specification

Formal Verification Request 40

decreaseApprovalPausable

📅 07, Feb 2019
⏱ 124.78 ms

Line 44-47 in File PausableToken.sol

```
44    /*@CTK decreaseApprovalPausable
45      @tag assume_completion
46      @post paused == false
47     */
```

Line 48-50 in File PausableToken.sol

```
48    function decreaseApproval(address _spender, uint _subtractedValue) public
          whenNotPaused returns (bool success) {
49        return super.decreaseApproval(_spender, _subtractedValue);
50    }
```

✅ The code meets the specification

Formal Verification Request 41

burn

📅 07, Feb 2019
⏱ 359.74 ms

Line 20-26 in File BurnableToken.sol

```
20    /*@CTK burn
21      @tag assume_completion
22      @post paused == false
23      @post _value <= balances[msg.sender]
24      @post __post.balances[msg.sender] == balances[msg.sender] - _value
25      @post __post.totalSupply_ == totalSupply_ - _value
26     */
```

Line 27-29 in File BurnableToken.sol

```
    function burn(uint256 _value) whenNotPaused public {
        _burn(msg.sender, _value);
    }
```

The code meets the specification

Formal Verification Request 42

_burn

07, Feb 2019
37.56 ms

Line 31-36 in File BurnableToken.sol

```
31    /*@CTK _burn
32      @tag assume_completion
33      @post _value <= balances[_who]
34      @post __post.balances[_who] == balances[_who] - _value
35      @post __post.totalSupply_ == totalSupply_ - _value
36     */
```

Line 37-46 in File BurnableToken.sol

```
37    function _burn(address _who, uint256 _value) internal {
38        require(_value <= balances[_who]);
39        // no need to require value <= totalSupply, since that would imply the
40        // sender's balance is greater than the totalSupply, which *should* be an
            assertion failure
41
42        balances[_who] = balances[_who].sub(_value);
43        totalSupply_ = totalSupply_.sub(_value);
44        emit Burn(_who, _value);
45        emit Transfer(_who, address(0), _value);
46    }
```

The code meets the specification