CERTIK

Formal Verification Platform for
Smart Contracts and Blockchain Ecosystems

CertiK Verification Report for MXC

Verification Request date: 2018-07-28
Revision: 2018-08-05
Company Website: https://www.mxc.org/

Summary

This is the report for smart contract verification service on MXC smart contract. The goal of the audit is to guarantee that verified smart contracts are robust enough to avoid potentially unexpected loopholes. The source code is passed around from email.

Conclusion: **PASS**

Our formal verification engine concludes that the MXC smart contract meets its specification, with 100% code coverage. CertiK believes this contract is trustworthy and hack-resistant.

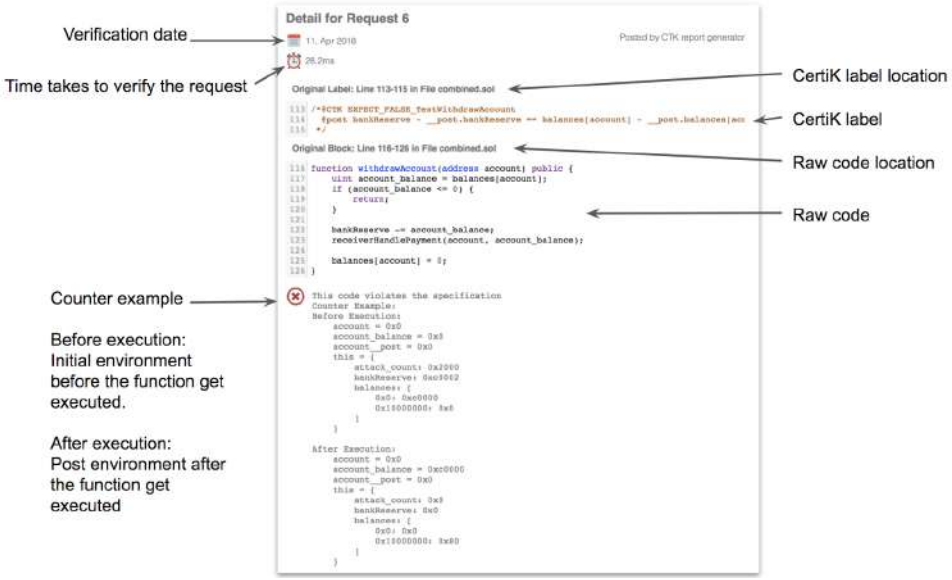
Details

1. Vulnerability

CertiK applies smart labels on the source code with 100% coverage to detect 2 types of errors: Function Correctness, and Integer Overflow. For each failed verification request, CertiK categorizes its severity into 3 brackets: Critical, Medium and Low. Other than issues falling into the Low bracket, CertiK will push back and require the client to update the source code to meet criteria, before proceeding to the next step.

Severity	Result
Critical	Not found
Medium	Not found
Low	Not found

3. How to Read the Verification Report



4. 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 MXC (the Client), or the scope of verification, and terms and conditions provided to the Client 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.

5. Verification Report

CertiK

Certi Request Report

29 out of 29 specs are satisfied.

Detail for Request 0: SafeMath_div

05, Aug 2018

Posted by CTK report generator

2.8ms

Line 45-51 in File MXC.sol

```
45 /*@CTK SafeMath_div
46     @tag spec
47     @post b == 0 -> __reverted == true  // solidity throws on 0.
48     @post __reverted == false -> __return == a / b
49     @post msg == msg__post
50     @post __addr_map == __addr_map__post
51 */
```

Line 52-57 in File MXC.sol

```
52 function div(uint256 a, uint256 b) internal pure returns (uint256) {
53     // assert(b > 0); // Solidity automatically throws when dividing by 0
54     // uint256 c = a / b;
55     // assert(a == b * c + a % b); // There is no case in which this doesn't hold
56     return a / b;
57 }
```

The code meets the specification

Detail for Request 1: vestBalanceOf

05, Aug 2018

Posted by CTK report generator

3.3ms

Line 395-402 in File MXC.sol

```
395 /*@CTK vestBalanceOf
396     @tag assume_completion
397     @post amount == timeLocks[who].amount
398     @post vestedAmount == timeLocks[who].vestedAmount
399     @post start == timeLocks[who].start
400     @post cliff == timeLocks[who].cliff
401     @post vesting == timeLocks[who].vesting
402 */
```

Line 407-417 in File MXC.sol

```
407 function vestBalanceOf(address who)
408     public view
409     returns (uint256 amount, uint256 vestedAmount, uint256 start, uint256 cliff, ui
410 {
411     require(who != address(0));
412     amount = timeLocks[who].amount;
413     vestedAmount = timeLocks[who].vestedAmount;
414     start = timeLocks[who].start;
415     cliff = timeLocks[who].cliff;
416     vesting = timeLocks[who].vesting;
417 }
```

The code meets the specification

Detail for Request 2: vestBalanceOfFailure

05, Aug 2018

Posted by CTK report generator

0.6ms

Line 403-406 in File MXC.sol

```
403 /*@CTK vestBalanceOfFailure
404     @pre __reverted == true
405     @post who == address(0)
406 */
```

Line 407-417 in File MXC.sol

```
407 function vestBalanceOf(address who)
408     public view
409     returns (uint256 amount, uint256 vestedAmount, uint256 start, uint256 cliff, ui
410 {
411     require(who != address(0));
412     amount = timeLocks[who].amount;
413     vestedAmount = timeLocks[who].vestedAmount;
414     start = timeLocks[who].start;
415     cliff = timeLocks[who].cliff;
416     vesting = timeLocks[who].vesting;
417 }
```

The code meets the specification

Detail for Request 3: get_allowance

05, Aug 2018

Posted by CTK report generator

2.6ms

Line 265-269 in File MXC.sol

```
265 /*@CTK get_allowance
266     @post __reverted == false
267     @post __return == allowed[_owner][_spender]
268     @post this == __post
269 */
```

Line 270-279 in File MXC.sol

```
270  function allowance(
271      address _owner,
272      address _spender
273 )
274      public
275      view
276      returns (uint256)
277  {
278      return allowed[_owner][_spender];
279  }
```

The code meets the specification

Detail for Request 4: grantTokenStartNow_succeed

05, Aug 2018

Posted by CTK report generator

10121.1ms

Line 486-497 in File MXC.sol

```
/*@CTK grantTokenStartNow_succeed
    @tag assume_completion
    @post __post.timeLocks[_to].amount == _amount
    @post __post.timeLocks[_to].vestedAmount == 0
    @post __post.timeLocks[_to].vestedMonths == 0
    @post __post.timeLocks[_to].start == now
    @post __post.timeLocks[_to].cliff == __post.timeLocks[_to].start + _cliffMonths
    @post __post.timeLocks[_to].vesting == __post.timeLocks[_to].start + _vestingMo
    @post __post.timeLocks[_to].from == msg.sender
    @post __post.balances[msg.sender] == balances[msg.sender] - _amount
    @post success == true
*/
```

Line 504-520 in File MXC.sol

```
function grantTokenStartNow(
    address _to,
    uint256 _amount,
    uint256 _cliffMonths,
    uint256 _vestingMonths
)
    public
    returns (bool success)
{
    return grantToken(
        _to,
        _amount,
        now,
        now.add(_cliffMonths.mul(MONTH)),
        now.add(_vestingMonths.mul(MONTH))
        );
}
```

The code meets the specification

Detail for Request 5: grantTokenStartNow_fail

05, Aug 2018

Posted by CTK report generator

561ms

Line 498-503 in File MXC.sol

```
/*@CTK grantTokenStartNow_fail
    @pre _to == address(0) || _amount > balances[msg.sender] || _amount == 0 ||
         timeLocks[_to].amount - timeLocks[_to].vestedAmount > 0 ||
         _vestingMonths <= 0 || _cliffMonths < 0
```

```
502     @post __reverted == true
503 */
```

Line 504-520 in File MXC.sol

```
504 function grantTokenStartNow(
505     address _to,
506     uint256 _amount,
507     uint256 _cliffMonths,
508     uint256 _vestingMonths
509 )
510     public
511     returns (bool success)
512 {
513     return grantToken(
514         _to,
515         _amount,
516         now,
517         now.add(_cliffMonths.mul(MONTH)),
518         now.add(_vestingMonths.mul(MONTH))
519         );
520 }
```

The code meets the specification

Detail for Request 6: decreaseApproval0

05, Aug 2018

Posted by CTK report generator

20.3ms

Line 315-319 in File MXC.sol

```
315 /*@CTK decreaseApproval0
316     @pre __return == true
317     @pre allowed[msg.sender][_spender] <= _subtractedValue
318     @post __post.allowed[msg.sender][_spender] == 0
319 */
```

Line 326-341 in File MXC.sol

```
326 function decreaseApproval(
327     address _spender,
328     uint256 _subtractedValue
329 )
330     public
331     returns (bool)
332 {
333     uint256 oldValue = allowed[msg.sender][_spender];
334     if (_subtractedValue > oldValue) {
335         allowed[msg.sender][_spender] = 0;
336     } else {
337         allowed[msg.sender][_spender] = oldValue.sub(_subtractedValue);
338     }
339     emit Approval(msg.sender, _spender, allowed[msg.sender][_spender]);
340     return true;
341 }
```

The code meets the specification

Detail for Request 7: decreaseApproval

05, Aug 2018

Posted by CTK report generator

33.4ms

Line 320-325 in File MXC.sol

```
320 /*@CTK decreaseApproval
321     @pre __return == true
322     @pre allowed[msg.sender][_spender] > _subtractedValue
323     @post __post.allowed[msg.sender][_spender] ==
324         allowed[msg.sender][_spender] - _subtractedValue
325 */
```

Line 326-341 in File MXC.sol

```
326 function decreaseApproval(
327     address _spender,
328     uint256 _subtractedValue
329 )
330     public
331     returns (bool)
332 {
333     uint256 oldValue = allowed[msg.sender][_spender];
334     if (_subtractedValue > oldValue) {
335         allowed[msg.sender][_spender] = 0;
336     } else {
337         allowed[msg.sender][_spender] = oldValue.sub(_subtractedValue);
338     }
339     emit Approval(msg.sender, _spender, allowed[msg.sender][_spender]);
340     return true;
341 }
```

The code meets the specification

Detail for Request 8: totalSupply

05, Aug 2018

Posted by CTK report generator

1.7ms

Line 141-143 in File MXC.sol

```
141 /*@CTK totalSupply
142     @post __return == totalSupply_
143 */
```

Line 144-146 in File MXC.sol

```
144 function totalSupply() public view returns (uint256) {
145     return totalSupply_;
146 }
```

The code meets the specification

Detail for Request 9: transfer_success

05, Aug 2018

Posted by CTK report generator

8.8ms

Line 153-159 in File MXC.sol

```
153 /*@CTK transfer_success
154     @pre _to != address(0)
155     @pre balances[msg.sender] >= _value
156     @pre __reverted == false
157     @post __reverted == false
158     @post __return == true
159 */
```

Line 172-180 in File MXC.sol

```
172 function transfer(address _to, uint256 _value) public returns (bool) {
173     require(_to != address(0));
174     require(_value <= balances[msg.sender]);
175 
176     balances[msg.sender] = balances[msg.sender].sub(_value);
177     balances[_to] = balances[_to].add(_value);
178     emit Transfer(msg.sender, _to, _value);
179     return true;
180 }
```

The code meets the specification

Detail for Request 10: transfer_conditions

05, Aug 2018

Posted by CTK report generator

115.5ms

Line 160-165 in File MXC.sol

```
160 /*@CTK transfer_conditions
```

```
161     @tag assume_completion
162     @pre _to != msg.sender
163     @post __post.balances[_to] == balances[_to] + _value
164     @post __post.balances[msg.sender] == balances[msg.sender] - _value
165 */
```

Line 172-180 in File MXC.sol

```
172 function transfer(address _to, uint256 _value) public returns (bool) {
173     require(_to != address(0));
174     require(_value <= balances[msg.sender]);
175 
176     balances[msg.sender] = balances[msg.sender].sub(_value);
177     balances[_to] = balances[_to].add(_value);
178     emit Transfer(msg.sender, _to, _value);
179     return true;
180 }
```

The code meets the specification

Detail for Request 11: transfer_same_address

05, Aug 2018

Posted by CTK report generator

107.1ms

Line 166-171 in File MXC.sol

```
166 /*@CTK transfer_same_address
167     @tag assume_completion
168     @tag no_overflow
169     @pre _to == msg.sender
170     @post this == __post
171 */
```

Line 172-180 in File MXC.sol

```
172 function transfer(address _to, uint256 _value) public returns (bool) {
173     require(_to != address(0));
174     require(_value <= balances[msg.sender]);
175 
176     balances[msg.sender] = balances[msg.sender].sub(_value);
177     balances[_to] = balances[_to].add(_value);
178     emit Transfer(msg.sender, _to, _value);
179     return true;
180 }
```

The code meets the specification

Detail for Request 12: balanceOf

05, Aug 2018

Posted by CTK report generator

1.9ms

Line 187-189 in File MXC.sol

```
187 /*@CTK balanceOf
188     @post __return == balances[_owner]
189 */
```

Line 190-192 in File MXC.sol

```
190 function balanceOf(address _owner) public view returns (uint256) {
191     return balances[_owner];
192 }
```

The code meets the specification

Detail for Request 13: returnGrantedToken_succeed

05, Aug 2018

Posted by CTK report generator

220.6ms

Line 633-638 in File MXC.sol

```
633 /*@CTK returnGrantedToken_succeed
634     @tag assume_completion
635     @post __post.timeLocks[msg.sender].from == timeLocks[msg.sender].from
636     @post __post.timeLocks[msg.sender].amount == timeLocks[msg.sender].amount - _an
637     @post __post.balances[timeLocks[msg.sender].from] == balances[timeLocks[msg.ser
638 */
```

Line 643-658 in File MXC.sol

```
643 function returnGrantedToken(uint256 _amount)
644     public
645     returns (bool success)
646 {
647     address to = timeLocks[msg.sender].from;
648     require(to != address(0));
649     require(_amount > 0, "Nothing to transfer.");
650     require(timeLocks[msg.sender].amount > 0, "Nothing to return.");
651     require(_amount <= timeLocks[msg.sender].amount.sub(timeLocks[msg.sender].veste
652
653     timeLocks[msg.sender].amount = timeLocks[msg.sender].amount.sub(_amount);
654     balances[to] = balances[to].add(_amount);
655
656     emit GrantedTokenReturned(msg.sender, to, _amount);
657     return true;
658 }
```

The code meets the specification

Detail for Request 14: returnGrantedToken_fail

05, Aug 2018

Posted by CTK report generator

130ms

Line 639-642 in File MXC.sol

```
639 /*@CTK returnGrantedToken_fail
640     @pre timeLocks[msg.sender].from == address(0) || _amount == 0 || timeLocks[msg.
641     @post __reverted == true
642 */
```

Line 643-658 in File MXC.sol

```
643 function returnGrantedToken(uint256 _amount)
644     public
645     returns (bool success)
646 {
647     address to = timeLocks[msg.sender].from;
648     require(to != address(0));
649     require(_amount > 0, "Nothing to transfer.");
650     require(timeLocks[msg.sender].amount > 0, "Nothing to return.");
651     require(_amount <= timeLocks[msg.sender].amount.sub(timeLocks[msg.sender].veste
652 
653     timeLocks[msg.sender].amount = timeLocks[msg.sender].amount.sub(_amount);
654     balances[to] = balances[to].add(_amount);
655 
656     emit GrantedTokenReturned(msg.sender, to, _amount);
657     return true;
658 }
```

The code meets the specification

Detail for Request 15: calcVestedToken_fail

05, Aug 2018

Posted by CTK report generator

13.5ms

Line 529-531 in File MXC.sol

```
529 /*@CTK calcVestedToken_fail
530     @post timeLocks[_to].amount == 0 -> __reverted == true
531 */
```

Line 557-592 in File MXC.sol

```
557  function calcVestableToken(address _to)
558     internal view
559     returns (uint256 amount, uint256 vestedMonths, uint256 curTime)
560 {
561     uint256 vestTotalMonths;
562     uint256 vestedAmount;
563     uint256 vestPart;
564     amount = 0;
565     vestedMonths = 0;
566     curTime = now;
567 
568     require(timeLocks[_to].amount > 0, "Nothing was granted to this address.");
569 
570     if (curTime <= timeLocks[_to].cliff) {
571         return (0, 0, curTime);
572     }
573 
574     vestedMonths = curTime.sub(timeLocks[_to].start) / MONTH;
575     vestedMonths = vestedMonths.sub(timeLocks[_to].vestedMonths);
576 
577     if (curTime >= timeLocks[_to].vesting) {
578         return (timeLocks[_to].amount.sub(timeLocks[_to].vestedAmount), vestedMonth
579     }
580 
581     if (vestedMonths > 0) {
582         vestTotalMonths = timeLocks[_to].vesting.sub(timeLocks[_to].start) / MONTH;
583         vestPart = timeLocks[_to].amount.div(vestTotalMonths);
584         amount = vestedMonths.mul(vestPart);
585         vestedAmount = timeLocks[_to].vestedAmount.add(amount);
586         if (vestedAmount > timeLocks[_to].amount) {
587             amount = timeLocks[_to].amount.sub(timeLocks[_to].vestedAmount);
588         }
589     }
590 
591     return (amount, vestedMonths, curTime);
592 }
```

The code meets the specification

Detail for Request 16: calcVestedToken_not_eligible_for_vesting

05, Aug 2018

Posted by CTK report generator

127.3ms

Line 532-538 in File MXC.sol

```
532 /*@CTK calcVestedToken_not_eligible_for_vesting
533     @tag assume_completion
534     @pre now <= timeLocks[_to].cliff
535     @post amount == 0
536     @post vestedMonths == 0
537     @post curTime == now
538 */
```

Line 557-592 in File MXC.sol

```
557  function calcVestableToken(address _to)
558     internal view
559     returns (uint256 amount, uint256 vestedMonths, uint256 curTime)
```

```
{
    uint256 vestTotalMonths;
    uint256 vestedAmount;
    uint256 vestPart;
    amount = 0;
    vestedMonths = 0;
    curTime = now;

    require(timeLocks[_to].amount > 0, "Nothing was granted to this address.");

    if (curTime <= timeLocks[_to].cliff) {
        return (0, 0, curTime);
    }

    vestedMonths = curTime.sub(timeLocks[_to].start) / MONTH;
    vestedMonths = vestedMonths.sub(timeLocks[_to].vestedMonths);

    if (curTime >= timeLocks[_to].vesting) {
        return (timeLocks[_to].amount.sub(timeLocks[_to].vestedAmount), vestedMonth
    }

    if (vestedMonths > 0) {
        vestTotalMonths = timeLocks[_to].vesting.sub(timeLocks[_to].start) / MONTH;
        vestPart = timeLocks[_to].amount.div(vestTotalMonths);
        amount = vestedMonths.mul(vestPart);
        vestedAmount = timeLocks[_to].vestedAmount.add(amount);
        if (vestedAmount > timeLocks[_to].amount) {
            amount = timeLocks[_to].amount.sub(timeLocks[_to].vestedAmount);
        }
    }

    return (amount, vestedMonths, curTime);
}
```

The code meets the specification

Detail for Request 17: calcVestedToken_vesting_period_end

05, Aug 2018

Posted by CTK report generator

2862.3ms

Line 539-546 in File MXC.sol

```
/*@CTK calcVestedToken_vesting_period_end
    @tag assume_completion
    @pre now > timeLocks[_to].cliff
    @pre now >= timeLocks[_to].vesting
    @post amount == timeLocks[_to].amount - timeLocks[_to].vestedAmount
    @post vestedMonths == (now - timeLocks[_to].start) / MONTH - timeLocks[_to].ves
    @post curTime == now
*/
```

Line 557-592 in File MXC.sol

```
 function calcVestableToken(address _to)
    internal view
    returns (uint256 amount, uint256 vestedMonths, uint256 curTime)
{
    uint256 vestTotalMonths;
```

```
562     uint256 vestedAmount;
563     uint256 vestPart;
564     amount = 0;
565     vestedMonths = 0;
566     curTime = now;
567 
568     require(timeLocks[_to].amount > 0, "Nothing was granted to this address.");
569 
570     if (curTime <= timeLocks[_to].cliff) {
571         return (0, 0, curTime);
572     }
573 
574     vestedMonths = curTime.sub(timeLocks[_to].start) / MONTH;
575     vestedMonths = vestedMonths.sub(timeLocks[_to].vestedMonths);
576 
577     if (curTime >= timeLocks[_to].vesting) {
578         return (timeLocks[_to].amount.sub(timeLocks[_to].vestedAmount), vestedMonth
579     }
580 
581     if (vestedMonths > 0) {
582         vestTotalMonths = timeLocks[_to].vesting.sub(timeLocks[_to].start) / MONTH;
583         vestPart = timeLocks[_to].amount.div(vestTotalMonths);
584         amount = vestedMonths.mul(vestPart);
585         vestedAmount = timeLocks[_to].vestedAmount.add(amount);
586         if (vestedAmount > timeLocks[_to].amount) {
587             amount = timeLocks[_to].amount.sub(timeLocks[_to].vestedAmount);
588         }
589     }
590 
591     return (amount, vestedMonths, curTime);
592 }
```

The code meets the specification

Detail for Request 18: calcVestedToken_in_between_cliff_and_vesting

05, Aug 2018

Posted by CTK report generator

77872.2ms

Line 547-556 in File MXC.sol

```
547 /*@CTK "calcVestedToken_in_between_cliff_and_vesting"
548     @tag assume_completion
549     @pre now > timeLocks[_to].cliff && now < timeLocks[_to].vesting
550     @pre (now - timeLocks[_to].start) / MONTH > timeLocks[_to].vestedMonths
551     @post vestedMonths == (now - timeLocks[_to].start) / MONTH - timeLocks[_to].ves
552     @post (amount == timeLocks[_to].amount / ((timeLocks[_to].vesting - timeLocks[_
553           (amount == timeLocks[_to].amount - timeLocks[_to].vestedAmount)
554     @post amount <= timeLocks[_to].amount - timeLocks[_to].vestedAmount
555     @post curTime == now
556 */
```

Line 557-592 in File MXC.sol

```
557  function calcVestableToken(address _to)
558     internal view
559     returns (uint256 amount, uint256 vestedMonths, uint256 curTime)
560 {
561     uint256 vestTotalMonths;
```

```
562     uint256 vestedAmount;
563     uint256 vestPart;
564     amount = 0;
565     vestedMonths = 0;
566     curTime = now;
567 
568     require(timeLocks[_to].amount > 0, "Nothing was granted to this address.");
569 
570     if (curTime <= timeLocks[_to].cliff) {
571         return (0, 0, curTime);
572     }
573 
574     vestedMonths = curTime.sub(timeLocks[_to].start) / MONTH;
575     vestedMonths = vestedMonths.sub(timeLocks[_to].vestedMonths);
576 
577     if (curTime >= timeLocks[_to].vesting) {
578         return (timeLocks[_to].amount.sub(timeLocks[_to].vestedAmount), vestedMonth
579     }
580 
581     if (vestedMonths > 0) {
582         vestTotalMonths = timeLocks[_to].vesting.sub(timeLocks[_to].start) / MONTH;
583         vestPart = timeLocks[_to].amount.div(vestTotalMonths);
584         amount = vestedMonths.mul(vestPart);
585         vestedAmount = timeLocks[_to].vestedAmount.add(amount);
586         if (vestedAmount > timeLocks[_to].amount) {
587             amount = timeLocks[_to].amount.sub(timeLocks[_to].vestedAmount);
588         }
589     }
590 
591     return (amount, vestedMonths, curTime);
592 }
```

The code meets the specification

Detail for Request 19: SafeMath_mul

05, Aug 2018

Posted by CTK report generator

463.5ms

Line 20-28 in File MXC.sol

```
20 /*@CTK SafeMath_mul
21     @tag spec
22     @post __reverted == __has_assertion_failure
23     @post __has_assertion_failure == __has_overflow
24     @post __reverted == false -> c == a * b
25     @post msg == msg__post
26     @post (a > 0 && (a * b / a != b)) == __has_assertion_failure
27     @post __addr_map == __addr_map__post
28 */
```

Line 29-40 in File MXC.sol

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

```
35      }
36
37      c = a * b;
38      assert(c / a == b);
39      return c;
40  }
```

The code meets the specification

Detail for Request 20: redeemVestableToken_succeeds

05, Aug 2018

Posted by CTK report generator

573.1ms

Line 598-604 in File MXC.sol

```
598 /*@CTK redeemVestableToken_succeeds
599     @tag assume_completion
600     @post _to != address(0)
601     @post timeLocks[_to].amount > 0
602     @post timeLocks[_to].vestedAmount < timeLocks[_to].amount
603     @post __post.balances[_to] > balances[_to]
604 */
```

Line 605-627 in File MXC.sol

```
605 function redeemVestableToken(address _to)
606     public
607     returns (bool success)
608 {
609     require(_to != address(0));
610     require(timeLocks[_to].amount > 0, "Nothing was granted to this address!");
611     require(timeLocks[_to].vestedAmount < timeLocks[_to].amount, "All tokens were v
612
613     (uint256 amount, uint256 vestedMonths, uint256 curTime) = calcVestableToken(_to
614     require(amount > 0, "Nothing to redeem now.");
615
616     TimeLock storage t = timeLocks[_to];
617     balances[_to] = balances[_to].add(amount);
618     t.vestedAmount = t.vestedAmount.add(amount);
619     t.vestedMonths = t.vestedMonths + uint16(vestedMonths);
620     t.cliff = curTime;
621
622     emit VestedTokenRedeemed(_to, amount, vestedMonths);
623     return true;
624 }
```

The code meets the specification

Detail for Request 21: approve

05, Aug 2018

Posted by CTK report generator

2.3ms

Line 249-252 in File MXC.sol

```
249 /*@CTK approve
250     @tag assume_completion
251     @post __post.allowed[msg.sender][_spender] == _value
252 */
```

Line 253-257 in File MXC.sol

```
253 function approve(address _spender, uint256 _value) public returns (bool) {
254     allowed[msg.sender][_spender] = _value;
255     emit Approval(msg.sender, _spender, _value);
256     return true;
257 }
```

The code meets the specification

Detail for Request 22: increaseApproval

05, Aug 2018

Posted by CTK report generator

15.3ms

Line 289-293 in File MXC.sol

```
289 /*@CTK increaseApproval
290     @pre __return == true
291     @post __post.allowed[msg.sender][_spender] ==
292         allowed[msg.sender][_spender] + _addedValue
293 */
```

Line 294-305 in File MXC.sol

```
294 function increaseApproval(
295     address _spender,
296     uint256 _addedValue
297 )
298     public
299     returns (bool)
300 {
301     allowed[msg.sender][_spender] = (
302         allowed[msg.sender][_spender].add(_addedValue));
303     emit Approval(msg.sender, _spender, allowed[msg.sender][_spender]);
304     return true;
305 }
```

The code meets the specification

Detail for Request 23: transferFrom

05, Aug 2018

Posted by CTK report generator

138.2ms

Line 212-219 in File MXC.sol

```
212 /*@CTK transferFrom
213     @tag assume_completion
214     @pre _from != _to
215     @post __return == true
216     @post __post.balances[_to] == balances[_to] + _value
217     @post __post.balances[_from] == balances[_from] - _value
218     @post __has_overflow == false
219 */
```

Line 220-237 in File MXC.sol

```
220 function transferFrom(
221     address _from,
222     address _to,
223     uint256 _value
224 )
225     public
226     returns (bool)
227 {
228     require(_to != address(0));
229     require(_value <= balances[_from]);
230     require(_value <= allowed[_from][msg.sender]);
231 
232     balances[_from] = balances[_from].sub(_value);
233     balances[_to] = balances[_to].add(_value);
234     allowed[_from][msg.sender] = allowed[_from][msg.sender].sub(_value);
235     emit Transfer(_from, _to, _value);
236     return true;
237 }
```

The code meets the specification

Detail for Request 24: grantToken_succeed

05, Aug 2018

Posted by CTK report generator

705.8ms

Line 427-438 in File MXC.sol

```
427 /*@CTK grantToken_succeed
428     @tag assume_completion
429     @post __post.timeLocks[_to].amount == _amount
```

```
    @post __post.timeLocks[_to].vestedAmount == 0
    @post __post.timeLocks[_to].vestedMonths == 0
    @post __post.timeLocks[_to].start == _start
    @post __post.timeLocks[_to].cliff == _cliff
    @post __post.timeLocks[_to].vesting == _vesting
    @post __post.timeLocks[_to].from == msg.sender
    @post success == true
    @post __post.balances[msg.sender] == balances[msg.sender] - _amount
*/
```

Line 445-477 in File MXC.sol

```
function grantToken(
    address _to,
    uint256 _amount,
    uint256 _start,
    uint256 _cliff,
    uint256 _vesting
)
    public
    returns (bool success)
{
    require(_to != address(0));
    require(_amount <= balances[msg.sender], "Not enough balance to grant token.");
    require(_amount > 0, "Nothing to transfer.");
    require((timeLocks[_to].amount.sub(timeLocks[_to].vestedAmount) == 0), "The pre
    require(_cliff >= _start, "_cliff must be >= _start");
    require(_vesting > _start, "_vesting must be bigger than _start");
    require(_vesting > _cliff, "_vesting must be bigger than _cliff");


    balances[msg.sender] = balances[msg.sender].sub(_amount);
    timeLocks[_to] = TimeLock(_amount, 0, 0, _start, _cliff, _vesting, msg.sender);

    emit NewTokenGrant(msg.sender, _to, _amount, _start, _cliff, _vesting);
    return true;
}
```

The code meets the specification

Detail for Request 25: grantToken_fail

05, Aug 2018

Posted by CTK report generator

416.7ms

Line 439-444 in File MXC.sol

```
/*@CTK grantToken_fail
    @pre _to == address(0) || _amount > balances[msg.sender] || _amount == 0 ||
         timeLocks[_to].amount - timeLocks[_to].vestedAmount > 0 ||
         _vesting <= _start || _cliff < _start
    @post __reverted == true
*/
```

Line 445-477 in File MXC.sol

```
function grantToken(
    address _to,
    uint256 _amount,
```

```
448     uint256 _start,
449     uint256 _cliff,
450     uint256 _vesting
451 )
452     public
453     returns (bool success)
454 {
455     require(_to != address(0));
456     require(_amount <= balances[msg.sender], "Not enough balance to grant token.");
457     require(_amount > 0, "Nothing to transfer.");
458     require((timeLocks[_to].amount.sub(timeLocks[_to].vestedAmount) == 0), "The pre
459     require(_cliff >= _start, "_cliff must be >= _start");
460     require(_vesting > _start, "_vesting must be bigger than _start");
461     require(_vesting > _cliff, "_vesting must be bigger than _cliff");
462
463
464     balances[msg.sender] = balances[msg.sender].sub(_amount);
465
466     timeLocks[_to] = TimeLock(_amount, 0, 0, _start, _cliff, _vesting, msg.sender);
467
468     emit NewTokenGrant(msg.sender, _to, _amount, _start, _cliff, _vesting);
469     return true;
470 }
```

The code meets the specification

Detail for Request 26: SafeMath_add

05, Aug 2018

Posted by CTK report generator

7.7ms

Line 79-87 in File MXC.sol

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

Line 88-92 in File MXC.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 27: MXCToken

05, Aug 2018

Posted by CTK report generator

190.6ms

Line 385-388 in File MXC.sol

```
385 /*@CTK MXCToken
386     @post __post.totalSupply_ == 2664965800 * (10 ** uint256(decimals))
387     @post __post.balances[msg.sender] == __post.totalSupply_
388 */
```

Line 389-393 in File MXC.sol

```
389 constructor() public {
390     totalSupply_ = 2664965800 * (10 ** uint256(decimals));
391     balances[msg.sender] = totalSupply_;
392     emit Transfer(address(0), msg.sender, totalSupply_);
393 }
```

The code meets the specification

Detail for Request 28: SafeMath_sub

05, Aug 2018

Posted by CTK report generator

2.7ms

Line 62-70 in File MXC.sol

```
62 /*@CTK SafeMath_sub
63     @tag spec
64     @post __reverted == __has_assertion_failure
65     @post __has_overflow == true -> __has_assertion_failure == true
66     @post __reverted == false -> __return == a - b
67     @post msg == msg__post
68     @post (a < b) == __has_assertion_failure
69     @post __addr_map == __addr_map__post
70 */
```

Line 71-74 in File MXC.sol

```
71 function sub(uint256 a, uint256 b) internal pure returns (uint256) {
72     assert(b <= a);
73     return a - b;
74 }
```

The code meets the specification