ERC-1329: Inalienable Reputation Token #1329

New issue

Open **dr-orlovsky** opened this issue 25 days ago · 51 comments

dr-orlovsky commented 25 days ago • edited

```
eip: EIP-1329
title: Inalienable Reputation Token
author: Maxim Orlovsky <orlovsky@pandoraboxchain.ai>, Andrey Sobol <sobol@pandoraboxchain.a
discussions-to: this issue
status: WIP
type: Standards Track
category: ERC
created: 2018-08-13
```

Simple Summary

Reputation belonging to an identity is frequently used in designing repeated economic games and protect them from malicious actors. The main property of such reputation should be inalienability. This ERC proposes the standard in creating inalienable reputation tokens.

Abstract

Standard in creating inalienable reputation tokens.

Reputation tokens are emitted and burned by a contract depending on balance holders actions and their consequences (i.e. there is a proof of the Byzantine behaviour of the owner). Reputation balances can be queried, but not directly changed or transferred from outside of the contract.

The token balance is protected by two-tier key design that allows signing transactions that can affect the reputation from one address while keeping the ability to replace the address with a compromised key by using the primary key stored in an airgap/secure cold storage.

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Notifications

Unsubscribe

You're receiving notifications because you were mentioned.

12 participants

https://github.com/ethereum/EIPs/issues/1329

Specification

```
interface ERCReputationToken {
    /// ERC20-like properties providing general information
    /// about token name and symbol
    function name() public view returns (string);
    function symbol() public view returns (string);

    /// ERC777-like granularity
    function granularity() public view returns (uint256);

    /// Reputation may be limited or onlimited by the supply. These functions
    /// provide information whether the supply is limited and, if not, the
    /// `totalLimit()` and `currentSupply()` will be returning the maximum amount
    /// of the tokens that can be produced and current token issuance
    function hasLimit() public view returns (bool);
    function totalLimit() public view returns (uint256);
    function currentSupply() public view returns (uint256);

    /// Function returns the balance of reputation tokens on an account
    function balanceOf(address owner) public view returns (uint256);

    /// Function returns the address that is authorised to sign transactions
    /// to this contract that can affect amount of the reputation on the account
    function authAddress(address owner) public view returns (address auth, uint duration);

    /// Authorizes address to interact with the contract on behalf
    /// of the balance owner for a some duration (amount of blocks)
    function grantAddressAuth(address auth, uint duration) public returns (address prevAddr
    /// Extends authorized duration for the registered authorized address
    function extendAuthDuration(uint forDuration)
    /// Remokes authorisation right from the currently authorised address
    /// to interact with the contract on behalf of account owner
    function revokeAddressAuth() public;

    /// Produced when contract generates some about of reputation and assigns
    /// it to a certain account
    event Issued(address owner, uint amountProduced);
    /// Produced when contract burns some about of reputation on a certain account
    event Burned(address owner, uint amountBurned);

    /// Events for operations with authorised accounts
    event AuthGranted(address owner, address auth, uint duration);
    event AuthRevoked(address owner, address auth);
    event AuthExpired(address owner, address auth);
}
```

https://github.com/ethereum/EIPs/issues/1329

```solidity
pragma solidity ^0.4.24;

import "./IReputation.sol";
import "openzeppelin-solidity/contracts/math/SafeMath.sol";


contract Reputation is IReputation {

    using SafeMath for uint256;

    string internal constant _name = "ERC1329";
    string internal constant _symbol = "REP";
    uint256 internal constant _granularity = 1;

    // key : owner , value : balance
    mapping(address => uint256) internal _balances;
    // key : auth  , value : duration (blocks)
    mapping(address => uint256) internal _authorized_duration;
    // key : owner , value : auth
    mapping(address => address) internal _authorized_addresses;
    // key : auth  , value : owner
    mapping(address => address) internal _owner_addresses;
    // key : owner , value : banned
    mapping(address => bool) internal _banneds;

    uint256 internal _totalLimit;
    uint256 internal _currentSupply;


    // ------------------------------------------------------------------------
    // Constructor
    // ------------------------------------------------------------------------
    constructor() public {
        // no tokens minted on deploy
        _currentSupply = 0;
        _totalLimit = 2**256 - 1;
    }


    // ------------------------------------------------------------------------
    // Functions
    // ------------------------------------------------------------------------
```

https://github.com/pandoraboxchain/token-1329-hackathon/blob/master/contracts/reputation/Reputation.sol

```solidity
pragma solidity ^0.4.24;

import "openzeppelin-solidity/contracts/ownership/Ownable.sol";
import "./Reputation.sol";


contract ReputationIssuable is Reputation, Ownable {

    function issueByAuth(address auth, uint256 value) public onlyOwner() {
        require(_owner_addresses[auth] != address(0));

        address owner = _owner_addresses[auth];

        require(!_banneds[owner]);

        uint256 duration = _authorized_duration[auth];
        if (duration >= block.number) {
            _balances[owner] = _balances[owner].add(value);
            _currentSupply = _currentSupply.add(value);
            emit Issued(owner, value);
        } else {
            emit AuthExpired(owner, auth);
        }
    }

    function  burnedByAuth(address auth, uint256 value) public onlyOwner() {
        require(_owner_addresses[auth] != address(0));

        address owner = _owner_addresses[auth];
        uint256 duration = _authorized_duration[auth];
        if (duration >= block.number) {
            uint256 balance = _balances[owner];
            if (value < balance) {
                _balances[owner] = _balances[owner].sub(value);
                _currentSupply = _currentSupply.sub(value);
                emit Burned(owner, value);
            } else {
                uint256 burned = _balances[owner];
                delete _balances[owner];
                _currentSupply = _currentSupply.sub(burned);
                delete _authorized_addresses[owner];
                delete _authorized_duration[auth];
```

https://github.com/pandoraboxchain/token-1329-hackathon/blob/master/contracts/reputation/ReputationIssuable.sol

Transaction 0x5b6e4365892fd2d78279386603c4e42246c15603ea8bc071af3f5165576cbe56

Home / Transactions / Tx Info

Overview

Transaction Information

Tools & Utilities

[This is a Rinkeby **Testnet** Transaction Only]

TxHash:	0x5b6e4365892fd2d78279386603c4e42246c15603ea8bc071af3f5165576cbe56
TxReceipt Status:	Success
Block Height:	2956870 (117 block confirmations)
TimeStamp:	29 mins ago (Sep-09-2018 07:56:37 AM +UTC)
From:	0x640874795fb57178a06bb4f6dff93fbc15183404
To:	[Contract 0x75d2a768a00cf07abd2f0994e0318db9ed8dc48a Created]
Value:	0 Ether ($0.00)
Gas Limit:	6700000
Gas Used By Txn:	2157291
Gas Price:	0.0000001 Ether (100 Gwei)
Actual Tx Cost/Fee:	0.2157291 Ether ($0.000000)
Nonce & {Position}:	1 | {0}

Input Data:

0x608060405260006006819055507fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff600581905550336007
60006101000a81548173ffffffffffffffffffffffffffffffffffffffff021916908373ffffffffffffffffffffffffffffffffffffffff
160217905550611e248061008360003960 00f3006080604052600436106100e6576000357c01000000000000000000000000000000000000
00000000000000000000900463ffffffff16806306fdde03146100eb57806344489ad11461017b578063556f0dc7146101aa57806370a082

View Input As

https://rinkeby.etherscan.io/tx/0x5b6e4365892fd2d78279386603c4e42246c15603ea8bc071af3f5165576cbe56

Tests:

https://github.com/pandoraboxchain/token-1329-hackathon/blob/master/test/reputation.js

https://github.com/pandoraboxchain/token-1329-hackathon/blob/master/test/reputation.issuable.js

Code test coverage

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
reputation/	94.92	67.86	100	95.59	
IReputation.sol	100	100	100	100	
Reputation.sol	96.88	75	100	97.3	105
ReputationIssuable.sol	91.3	58.33	100	92.59	22,48
ReputationProxy.sol	100	100	100	100	
All files	**94.92**	**67.86**	**100**	**95.59**	

Thank You

Repo link:

https://github.com/pandoraboxchain/token-1329-hackathon/