Oracle Server Study

lixiaoyu 2017.2.28

学习路径

1. Vitalik Buterin, SchellingCoin: A Minimal-Trust Universal Data Feed, 2014
2. Vitalik Buterin, Ethereum and Oracles，2014
3. 第三方服务网站Oraclize.it

- **Oracles定义**

> `An oracle, in the blockchain sense, is a third party which is sending to your on-chain smart contract some data your smart contract code cannot fetch by itself.`

看上去等同于Data Feed。高大上的说法是: A reliable bridge between Ethereum smart contracts and the Internet.(TLSNotary proof Oraclize could.)

SchellingCoin

SchellingCoin出现的背景是解决以太坊智能合约与金融衍生品的应用遇到的问题。“对冲”，一种金融衍生品，是外贸交易中常用的金融工具，可以有效地减少汇率波动引起的损失。虚拟货币的价格波动非常大，用户使用虚拟货币进行交易也会面临价格波动引起损失的风险。使用“对冲”合约可以避免这样的损失，但区块链完整节点本身并不能读取链下的“价格”（如：ETH/USD）。因此，提出各种方法来解决链下数据上链的方法，SchellingCoin就是其中的一种去中心化的方法。

Scheling Point 谢林点

“人们在没有沟通的情况下的选择倾向，做出这一选择可能因为它看起来自然、特别、或者与选择者有关。”

举例：假设你和另一个囚犯被关在单独的房间里,和保安给你两个相同的纸和几个数字。如果你选择相同数字的,那么你们将被释放;否则,将在狱中度过余生。这些数字是：

14237 59049 76241 81259 90215 100000 132156 157604

由于数字“100000”在几个数字中最特殊，绝大多数人都会选择它。根据人类的这一趋利避害的天性设计了SchellingCoin大致原理如下：

1. 每偶数个区块，所有用户都能用各自的地址提交ETH/USD的价格的Hash值。
2. 在下一个区块，用户能提交上次提交的ETH/USD的价格的实际值。
3. 当实际值N+用户地址做hash，即H(N+ADDR)前后一致，且交易签名方是系统合法的，数字签名也正确时，我们就称之为“正确的提交值”。
4. 将正确的提交值进行排序。
5. 每个提交正确且提交值处于所有值25%~75%用户，将获得SchellingCoin奖励。
6. *假设使用PoW或PoS，可以避免女巫攻击。*

问题和限制

1. 共谋攻击，类似51%攻击。
2. micro-cheating，对问题的理解有歧义或不够具体。解决方法：一、清晰描述问题。二、改善系统，减少出错几率。

总结：

Vitalik提出了POC设计，将SchellingCoin作为激励，让更多用户透过智能合约参与投票，选取符合中间值（25%~75%）的参与者进行奖励。除了引入汇率，SchellingCoin还可以被用来作为分布式的AWS。

思考：

Vitalik在SchellingCoin一文中指出：

> `In fact Bitcoin’s mining algorithm basically is a SchellingCoin on the order of transactions.`

意味着比特币挖矿算法（记录账本的规则，并非共识演算法）就是一种交易排序的SchellingCoin。进一步思考，把挖矿算法换成EVM就是以太坊，把挖矿算法换成Oracles就是更有意思的组合。

Ethereum and Oracles

Oracle的优点：

1. 不需要所有全节点的计算，仅需要少量的oracle计算。
2. 理论上不需要比现有Bitcoin或Ripple更复杂的平台。
3. 合约隐私性更强，虽然出口交易任然可见，但内部的运算过程可以不可见。这一计划可以通过安全多方计算协议进行扩充，这样合约甚至能够包含隐私信息。
4. 合约可以引入外部信息。在N个节点中达成HTTP请求共识远比在整个区块链网络中要容易的多。

Oracle方式的效率并不总是更高：

- Oracle方法的效率并非总是比全网共识的效率更高，有些应用甚至没有通过Oracle系统的实现。
  - 场景假设：执行一份特殊合约，这份合约的功能是实现对两个账户地址可以对一个合约地址的交易进行操作，其中一个账户的转账金额受限，另一个账户不受限。
  - 使用全网共识（Ethereum智能合约）：需要消耗1次椭圆签名检查、读2次数据、写3次数据。用Serpent语言编写，每次交易消耗350bytes的存储空间和160bytes的交易传输带宽。
  - 使用Oracle方式：假设是5取3签名，需要验证三次交易，由于Oracle多重签名消耗350-400bytes存储空间。
  - 上述例子很好地描述了以太坊使用状态机制和非图灵完备，同时由于记录的撤销条件由合约自身控制带来的神奇效果。

Oracle和区块链如何协同？

1. 预测市场：SchellingCoin和TruthCoin
2. 可验证的计算Oracle：通过Hash算法和惩罚机制，让Oracle说实话。
3. 签名分批处理：三台Oracle在多签交易时，每次交易消耗195bytes的数据和三次昂贵的签名验证。聪明地使用以太坊“oracle contract”，让Oracle依次签名，从而减少签名验证成本。Cost of Verify(N sig)>Cost of [Verify(a)+V(b)+……+V(n)],when N>2.
4. **基于区块链的审计：**

   基于Oracle的计算远不只“Ethereum multisig oracle” 的想法。极端情况下，我们让Oracle如果只做一件事——就是把交易的顺序，这个比特币的任务仍然留给区块链来决定。如果我们放弃这种需求，通过让一台Oracle维护它自己提出的中心化地记录状态和交易的数据库，就有可能来实现更高的效能，允许像微交易、高频交易这样的应用。但是，这显然存在一个信任的问题；万一oracle双花了怎么办？

   以太坊的合约可以解决这个问题。所有事情都默认运行在oracle上，如果oracle选择签发两个不同的、交易结果相矛盾的余额状态，那么这两个签名能被导入Ethereum，如果他们的合约可以拿走oracle的保证金，合约还会检验两个签名的有效性。为处理其他攻击行为的更复杂的计划也可能实现。

   （个人理解：Oracle可以通过以太坊的合约再建立一层交易和状态，由以太坊来避免Oracle出现双花。）
5. 可验证的安全多方计算：使用Oracle维护私有数据，可以建立一种协议每24小时让Oracle通过多方随机数生成新密钥。任何在审计上的违规都会损失保证金。

总结：Ethereum为代表的blockchain与M-of-N Oracles应可以各自为政。基础协议（Ethereum or Oracles）只是用户的仆人而非主人。

后续还关注

- Oraclize
- Smart Oracles
- Augur