TokenDice - 基于BCH实现的1对1数字竞猜应用

场景:
玩家A和玩家B各自选择一个数字并加密, 交换密文创建脚本, 脚本的具体内容就是游戏规则, 然后向脚本中打钱,
等这笔交易上链, 玩家公开游戏开始时的数字, 接着创建交易去花脚本中的钱, 能够签名成功的即为竞猜赢家

TokenDice 由三部分组成:
1. 底层是基于BitcoinUnlimited-1.3.0.1源码开发的, 对此我们做了两点改动
  1) 开放opcode: OP_DATASIGVERIFY
  2) 添加两个rpc: buildscript, signtx
2. relay服务器, libevent和libcurl实现, 用于玩家交换消息, 并将数据存储到ipfs
3. qt实现的客户端

实现细节:
首先, 我们约定玩家A选择的数字记为 Na, 地址记为 Aa, 公钥为Pa, 地址 Aa 对应的私钥对数字 Na 加密后的密文记为 Sa,
同理, 玩家B对应有 Nb, Ab, Pb, Sb
1. 加密数字, 对应的rpc是 signmessage, 即 Sa = singmessage (Aa, Na)
2. 玩家交换地址和密文依靠relay服务器
3. 创建脚本, 对应的rpc是 buildscript, script = buildscript (Aa, Sa, Ab, Sb)

script = OP_DUP << Sb << OP_EQUALVERIFY << Pb< OP_DATASIGVERIFY
<< OP_ROT << OP_ROT
<< OP_DUP << Sa< OP_EQUALVERIFY << Pa<< OP_DATASIGVERIFY
<< OP_ADD << 0x02 << OP_MOD;
<< OP_IF << OP_DUP << OP_HASH160 << Pb << OP_EQUALVERIFY << OP_CHECKSIG
<< OP_ELSE << OP_DUP << OP_HASH160 << Pa << OP_EQUALVERIFY << OP_CHECKSIG << OP_ENDIF

第1行验证Sb是否由Pb对应的私钥签名
第2行交换栈顶三个元素的次序, 让栈顶的第三个元素交换到栈顶第一个 (x3 x2 x1 -- x2 x1 x3)
第3行验证Sa是否由Pa对应的私钥签名
第4行取栈顶俩元素相加, 然后对2取模
第5行取模后结果为1, 验证玩家B私钥签名, 成功则玩家B获胜
第6行取模后结果为0, 验证玩家A私钥签名, 成功则玩家A获胜

4. 玩家下注, 对应的rpc是 createrawtransction
暂记这笔交易为 FundTx, FundTx的输入有两笔, 分别为玩家A和玩家B的未花费交易, 输出为buildscript产生的脚本地址,
如果需要找零, 则还有一笔找零的输出

5. 双方对 FundTx 进行签名, 对应的rpc是 signrawtransction, 签名时可以验证交易是否正确创建
注意, 到这一步玩家只知道对方的地址和密文, 下注的钱也还未广播上链, 仍然可以终止游戏, 双方都没有损失

6. 广播 FundTx, 对应的rpc是 sendrawtransction, 然后交易由矿工打包上链
7. 游戏双方公开竞猜的数字 Na, Nb
8. 创建 SpendTx, 对应的rpc是 createrawtransction, 交易的输入为 FundTx, 输出为玩家的地址

9. 签名 SpendTx, 对应的rpc是 signtx, 锁定脚本 script 是第3步创建的脚本, 以玩家A为例, 解锁脚本如下:
sigScript(A) = sign(A) << Pa << Na << Sa << Nb << Sb << HASH160(script)
根据脚本的逻辑, 最多只能有一位玩家可以签名成功, 成功的玩家广播 SpendTx即可赢得对方的注码

优点:
游戏的规则只依赖脚本, 完全去中心化的实现
对源码改动非常小, 只需要开放OP_DATASIGVERIFY

缺点:
目前只能够两个玩家游戏, 可玩性较差
如果玩家恶意不公开明文会导致游戏进行不下去

待完善:
完善游戏逻辑使得玩家即使恶意不公开明文也能继续游戏, 如脚本中添加超时机制
拓展成多人游戏
拓展脚本的功能, 实现更丰富的智能合约工具