Understanding Ethereum via Graph Analysis

以太坊不仅提供了Ether这种加密货币，而且在以太坊虚拟机中提供了一个去中心化的平台来执行智能合约。以太币的价格高涨，许多智能合约也被应用到以太坊中，但是关于其用户、智能合约以及两者之间关系的特性知之甚少。

1. **Introduction**

   本论文进行了对以太坊的第一个系统的研究：利用图分析来特征化以太坊上三种主要的活动：money transfer, smart contract creation, smart contract invocation.（三种活动都是由交易触发）

   构建了三种图：Money Flow Graph，Smart Contract Creation Graph，smart Contract Innovation Graph。

   Transaction是包含有用信息的签名数据包。比如 transaction里面的value表示交易金额总量。 一个transaction可以是外部的（由External owned account 发送，通常是用户提供数据的钱包应用），也可以是内部的（由于外部交易，执行智能合约）。内部的transaction不储存在区块链中，为了得到所有内部交易，我们设计了一种新方法来replay所有的外部交易，record内部交易。

2. **Background**

   通过智能合约，developers可以创建不同的应用，普通users可以转ehter币给他人或者使用那些应用。

   以太坊的基本单位是account。

   account种类：external owned accounts, smart contracts。区别：智能合约有可执行的code，前者没有。

   以太坊的basic fields：recipient（交易收方地址），signature（交易发方的地址），value（交易总额），data（智能合约或者调用智能合约的信息的bytecode）

3. **Methodology**

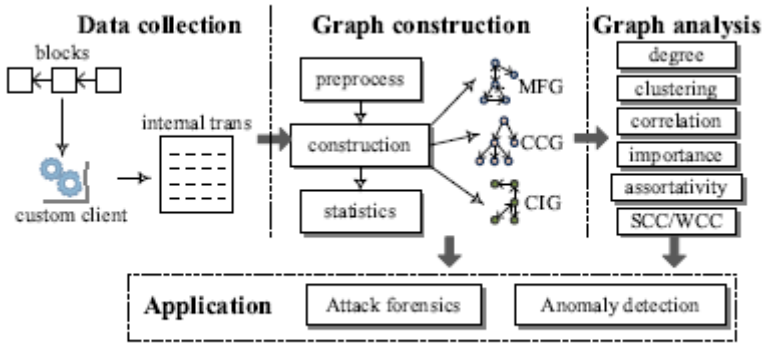

Fig. 1. An overview of our approach.

Methodology分为三个阶段：数据采集，图重建，图分析

4. **数据采集**

   以太坊为以太虚拟机定义了130中操作，智能合约的bytecode可以认为是一序列的这种操作。

   五种操作可以导致内部交易：CREATE(创建合约), CALL（调用合约）, CALL-CODE, DELEGATECALL, SELFDESTRUCT（从区块链remove掉智能合约，将剩下的ether装给一个指定目标）

5. **图重建**

preprocess：将四种类型的交易排除：1）从一个EOA到另一个EOA，交易金额为零；2）自毁智能合约，智能合约没有剩下ether；3）EOA间失败的交易；4）智能合约创建失败所导致的失败交易

没有合约调用的失败交易排除在外，因为合约会部分执行。

实验表明，大部分的accounts都不会涉及超过五个交易，即大部分的accounts不会频繁的转移money；大部分（99%）的EOA不会创建合约，创建合约的是developers；60%的智能合约既不转移money，也没有被调用。所以相当的资源被浪费了。

A. MFG重建：$MFG = (V, E, w)$，V是nodes的集合，E是边的集合，w是边到边的权重的映射函数。$V = V_n \cup V_{sc}$, V_n是EOA的集合，V_{sc}是智能合约的集合，E是节点对的集合，$E = \{(v_i, v_j)|v_i, v_j \in V\}$.

B. CCG重建：$CCG = (V, E)$, $E = \{(v_i, v_j)|v_i \in V, v_j \in V_{sc}\}$, 边$(v_i, v_j)$表示account v_i创建了智能合约v_j.

C. CIG重建：$CIG = (V, E)$，$E = \{(v_i, v_j)|v_i \in V, v_j \in V_{sc}\}$，边$(v_i, v_j)$表示account v_i调用了智能合约v_j.

几点**insights**：

1. 用户更喜欢在以太坊转移money，而不是使用智能合约。可能的原因之一是许多以太坊用户有使用比特币等其他加密货币的经验，智能合约对他们来说是比较新。
2. 智能合约没有被广泛应用。可能的原因之一是智能合约支持的应用很少，其中大部分还是用于金融应用的。
3. 不是所有用户都频繁使用以太坊。可能的原因之一是许多用户仅仅是尝试ethereum，在以太坊中deploy了toy contracts（不值钱的合约？）
4. 少数的developers创建了许多智能合约，许多EOA创建的合约是相同的。
5. 金融应用，比如外汇市场，支配了以太坊，因为他们是money transfer，contract creation和contract invocation中的最重要的节点。

6. **图分析**

   SCC（strong connected component, 一个有向图的最大的节点集合，其中每对节点都有一条有向的路径）、WCC（weak connected componet，一个有向图的最大节点集合，其中每对节点都有一条无向的路径）

   推断节点的Identity是比较困难的。因为以太坊允许用户将**name tags** 和**account**结合，所以从这些name tags我们可以得到identity。如果可以得到一个智能合约的**源代码**，我们也可以从他得到identity。更进一步的，论文提出了一种新的方法：利用**WCC**。WCC的根是一个eoa，其他节点是这个root直接或间接创建的智能合约。因此方法就是检查WCC中的所有节点来查询他们的identity。

   MFG分析：degree小的节点占大部分。大degree的account更喜欢和小degree的交易。可能原因之一是exchange market（大degree节点）倾向于服务许多个人用户（小degree节点），相比于其他exchange market。一个节点的出度很大，它的入度往往也很大；入度很大，出度往往也很大。SCC的数量远多于WCC。

   CCG分析：account的入度为0（EOA）或1（智能合约）。CCG中没有环，大度节点倾向于连接小度节点（assortativity coefficient为负）。智能合约很少创造智能合约。

   CIG分析：

7. **基于cross-graph analysis的应用**：两个重要的安全问题：**attack forensics**（攻击取证），**anomaly detection**（异常检测）

   attack forensics（攻击取证）：给定一个恶意的智能合约，attack forensics查找所有被攻击者控制的account。为了达到这个目标，将CCG和CIG关联来得到所有的攻击者创建的智能合约以及所有调用这种智能合约的account。

   anomaly detection（异常检测）：不正常的合约创建（消耗大量的资源，因为创建了大量的合约，每个以太坊客户端都要维持区块链的一个copy）

异常检测算法：

Algorithm 1 Detection of abnormal contract creation

Inputs: x, the detected account
MFG, money flow graph
CCG/CIG, contract creation/invocation graphs
T1,T2,T3, thresholds
Outputs: True/False, x is abnormal/benign

```
1   sc_set = created_sc(CCG, x);
2   if size(sc_set) < T1 return False;
3   for each node y in sc_set
4       caller_set = inedge(CIG, y);
5       for each edge z in caller_set
6           num += z.weight;
7       sender_set = inedge(MFG, y);
8       for each edge s in sender_set
9           value += s.weight;
10  if num > T2 × size(sc_set) || value > T3 × size(sc_set)
11      return False;
12  else return True;
```

8. 相关工作

1. 以太坊和比特币的比较

比特币不支持复杂的程序比如智能合约。

TABLE VII
COMPARISON OF ETHEREUM WITH BITCOIN

Blockchain	Account	Balance	Change	Multi-inputs	Multi-outputs	Many addresses
Ethereum	✓	✓	×	×	×	×
Bitcoin	×	×	✓	✓	✓	✓

2. 比特币的图分析