在NEO采用dBFT机制实现共识节点之间的“拜占庭容错”，并在NEO白皮书中描述恶意共识节点小于1/3的时候，该共识机制能够保证系统的安全性和可用性。我们经过研究发现，目前NEO的dBFT机制仅能保证诚实的共识节点之间达成共识。但共识节点之间不存在分叉，并不意味着全网不会存在分叉。NEO目前对dBFT共识机制的实现还不满足的拜占庭容错性质。

NEO dBFT共识机制简介

NEO区块链是一个分布式的智能合约平台。NEO实现了一种委托的拜占庭容错共识算法，它借鉴了一些 PoS 的特点(NEO持有人需要对共识节点进行投票)
利用最小的资源来保障网络免受拜占庭故障的影响，同时也弥补了 PoS 的一些问题。dBFT对由n个共识节点组成的共识系统，提供的容错能力，这种容错能力同时包含安全性和可用性，并适用于任何网络环境。

NEO dBFT共识机制可以详细见NEO官方共识机制白皮书：

http://docs.neo.org/zh-cn/basic/consensus/whitepaper.html(中文版)

http://docs.neo.org/en-us/basic/consensus/whitepaper.html(英文版)

拜占庭将军问题和区块链

拜占庭位于如今的土耳其的伊斯坦布尔，是东罗马帝国的首都。由于当时拜占庭罗马帝国国土辽阔，为了防御目的，因此每个军队都分隔很远，将军与将军之间只能靠信差传消息。 在战争的时候，拜占庭军队内所有将军和副官必需达成一致的共识，决定是否有赢的机会才去攻打敌人的阵营。但是，在军队内有可能存有叛徒和敌军的间谍，左右将军们的决定又扰乱整体军队的秩序。在进行共识时，结果并不代表大多数人的意见。这时候，在已知有成员谋反的情况下，其余忠诚的将军在不受叛徒的影响下如何达成一致的协议，拜占庭问题就此形成。PBFT算法，是解决拜占庭将军问题的一个经典算法。

区块链是一种去中心化的分布式账本系统，它可以用于登记和发行数字化资产、产权凭证、积分等，并以点对点的方式进行转账、支付和交易。区块链技术最早是由中本聪在一个密码学的邮件列表中提出的，也就是比特币。

此后，基于区块链技术的各种应用纷纷出现，比如基于区块链的电子现金系统、基于区块链的股权交易系统、基于区块链的智能合约系统等。区块链系统与传统的中心化账本系统相比，具有完全公开、不可篡改、防止多重支付等优点，并且不依赖

于任何的可信第三方。

然而，和任何分布式系统一样，区块链系统会面临网络延迟、传输错误、软件错误、安全漏洞、黑客入侵等问题。此外，去中心化的特点决定了此系统的任何一个参与者都不能被信任，可能会出现恶意节点，以及因各方利益不一致导致的数据分歧等问题。

为了防范这些潜在的错误，区块链系统需要一个高效的共识机制来确保每一个节点都有一个唯一公认的全局账本。传统的针对某些特定问题的容错方法，并不能完全解决分布式系统以及区块链系统的容错问题，人们需要一种能够容忍任何种类错误的容错方案。

NEO区块链共识机制细节

采用了拜占庭容错委托(dBFT)作为共识机制(http://docs.neo.org/en-us/basic/consensus/consensus.html)。全网中的NEO节点分为两类节点：一类为共识记点，负责和其他共识记点之间进行共识通讯，产生新的区块;另外一类为普通节点，不参与共识，但能够验证和接受新的区块。

共识节点由全网用户通过投票产生。NEO节点提出dBFT的背后思想是：PBFT算法能够很好的解决分布式节点的共识问题，但是PBFT共识参与节点数量越大性能就会越低。采用投票选取出相对较小数量的共识节点内部进行PBFT共识生成新区块，然后将该新区块发布到全网中达成全网共识。NEO共识节点之间，产生新区块的正常共识流程如下：

开启共识的节点分为两大类，非记账人和记账人节点，非记账人的不参与共识，记账人参与共识流程

选择议长，Neo议长产生机制是根据当前块高度和记账人数量做MOD运算得到，议长实际上按顺序当选

节点初始化，议长为primary节点，议员为backup节点。

满足出块条件后议长发送PrepareRequest

议员收到请求后，验证通过签名发送PrepareResponse

记账节点接收到PrepareResponse后，节点保存对方的签名信息，检查如果超过三分之二则发送 block

节点接收到block，PersistCompleted事件触发后整体重新初始化，

为了防止恶意的共识节点或议长，保证系统的安全性和可靠性，NEO提出changeview机制进一步增强dBFT的安全性。当节点在经过的时间间隔后仍未达成共识，或接收到包含非法交易的提案后，开始进入视图更换流程：

节点发出视图更换请求

任意节点收到至少个来自不同的相同后，视图更换达成，令并开始共识;

如果在经过的时间间隔后，视图更换仍未达成，则递增并回到第 2 步;
NEO共识节点总体流程如图：

dBFT共识机制问题分析

dBFT的核心思想，是想通过选举出的共识节点通过pBFT协议达成共识，从而产生全网的共识。这看上去是一个很好的思路，但由于dBFT和pBFT中，共识节点为非共识节点提供服务的模型不一样。在pBFT中，非共识节点(客户端节点)需要收到至少f+1个共识节点的相同的执行结果，从而获得服务结果。而dBFT中，非共识节点需要获得一个由至少2f+1个共识节点共同签名的block。dBFT的拜占庭容错，不能想当然的从pBFT中得出。证明一个共识协议的安全性，需要严谨的证明。

在分析网络协议安全性的一个重要的前提是，网络是一个复杂的环境，我们不能保证先发去的包一定先到达。这是网络协议之所以复杂的原因。下面我们给出两个反例，证明dBFT无法提供的拜占庭容错。

攻击案例1：

假设7个节点A1 A2 A3 A4 A5 A6 A7.其中议长A1和A2是恶意节点(小于1/3节点数)，同时A1担任本轮议长。A1生成block1发送PrepareRequest(block1)给A2 A3 A4.同时生成block2发送PrepareRequest(block2)给A5 A6 A7.当大家都收到PrepareRequest消息后。

A2 A3 A4会返回PrepareRespone(block1)消息，A5 A6 A7会返回PrepareRespone(block2)消息。此时，Block1和Block2的签名已经在网络上公开4个。

网络还未达成共识，还需进一步协商，最终生成的共识区块未定。但此时恶意节点A2实际上可以产生block2的签名(他自己手上能有5个block2的签名)，故A2有能力在网络中产生一个分叉。dBFT在该攻击案例下，无法实现拜占庭容错(2个恶意节点

)。

攻击案例2：假设7个节点A1 A2 A3 A4 A5 A6 A7.
其中A2为恶意节点(只有一个恶意节点)。 假设A1当选本轮议长，由于大量交易处理或其他网络原因，A1产生了延迟。并在临界timeout的事件发送一个block1出去。其中 A5 A6 A7 刚刚好在限定时间内收到并完全整个block1验证，回复了签名。A2 A3 A4还没有收到这个block1.此时，按照dBFT协议，timeout时间到，所有人将要求change view。

刚好，change view的包，比Block1提前到达了A2 A3 A4处。大家达成了change view的共识。然后进行下一个view阶段。 如果在此阶段，A2 A3 A4收到了block1.但是view和议长已经改变了，他们并不会接受block1.那么大家会进入下一轮，产生block2. Block2生产过程一切正常，A3 A4 A5 A6
A7几个诚实的节点达成了共识。
但是此时，A2拥有Block1的5个签名。它能够构造Block1. 由此，A2可以在网络中产生了一个分叉。dBFT在该攻击案例下，无法实现拜占庭容错(1个恶意节点)。

NEO当前的对dBTF的实现有所不足的本质原因在于：NEO的dBTF实现试图用共识节点之间的共识代替全网共识;然而这个假设并不严格成立。dBFT仅能保证诚实的共识节点之间产生共识，而诚实的共识节点之间的共识与全网节点之间的共识并没有严格的绑定关系。

在上面的例子1中，恶意的出块节A1 A2点无法影响共识节点之间产生共识Block2.但它能产生区块Block1从而造成网络中普通节点的分叉。按NEO目前设计，一旦分叉产生，分叉节点就无法回到NEO网络了。

问题的影响

在发现问题的当天，我们就将该问题邮件发给NEO创始人Erik Zhang。 Erik也在当天回复说他们已经发现了这个问题，并且在社区内做过相关声明。之前NEO在github项目上发起了一个Pull Request处理该问题，得到了社区成员的积极响应，为该问题建言献策，目前问题已经初步解决，正在测试稳定性，具体连接:https://github.com/neo-project/neo/pull/320.

通过对该PR具体讨论和实现的研究，我们发现实际上该PR主要是针对NEO当前dBTF中对PBFT的实现不够完善进行的修复。社区目前的理解是：由于NEO中dBFT缺乏PBFT的commit阶段，所以可能因为共识节点之间的网络延迟或者宕机问题，造成共识节点之间无法达成共识。

但即使增加commit过程，也无法防止恶意共识节点构造分叉块。NEO团队针对此问题，已经在该PR下提出了新的改进方案，目前NEO社区正在完善之前的PR从而彻底解决该问题。在这次问题的解决过程中我们发现NEO团队对于安全问题处理专业高效，社区响应热情及时。360安全团队会继续与NEO一起测试、分析并完善相关问题，推动NEO与区块链技术向前发展。