1. Is smart contract really smart?

smart contract is anything but smart. 智能合约就是一段写死的自动执行的合约代码，并不智能，而且写好之后就不能修改。

1. Irrevocability is a double edged sword.

一方面不可篡改性增加了合约的公信力，所有人都只能按照合约中的规则来，没有人能够篡改这个规则；另一方面不可篡改性意味着如果规则有漏洞，想要修补漏洞或者软件升级都是很困难的，硬分叉是非常麻烦的，和传统的中心化系统相比，没有办法及时发布补丁修复，必须征得绝大多数矿工的同意才行。硬分叉的时候又需要说明理由，一旦说明理由又会泄露系统的安全漏洞，有恶意的攻击者会在还没来得及升级前抢先发动攻击。

另外，即使我们已经发现了系统漏洞，已经有人进行恶意攻击了，想要冻结账户终止交易都是很困难的，比如个人的私钥泄露，没有办法进行软分叉，因为没有办法发行一个软件的更新并设置凡是跟这个账户相关的交易都是不合法的，这才能够冻结，但是对于个人账户来说明显是不可能的，只能把账户剩下的钱尽快转到安全的账户。智能合约一旦发布到区块链上，没有办法阻止对它的调用。比如The DAO事件，1/3的钱被黑客盗走了，剩下的2/3的钱也非常危险，没有办法阻止别人调用智能合约，唯一的办法是用黑客的方法把钱转到另一个安全的合约。

1. Nothing is irrevocable.

篡改是很难的，但是遇到特殊情况也是可以篡改的。分叉攻击可以篡改区块链的交易，The DAO攻击之后以太坊团队也是强行修改数据使得交易恢复被攻击之前的账户状态。所以在生活中不可篡改不是绝对的。

1. Is solidity the right programming language?
2. solidity语言设计上有什么问题？为什么会有重入攻击？

solidity的语言特性是反自然的，一般的理解，我给你转账，你是一个被动的接受者，你不可能反过来调用我，但是solidity的语言特性是说我给你转账的操作等于隐性地调用了你的fallback函数，结果你就可以再来调用我。这个和生活常识不同所以安全漏洞容易被忽略。

有人提出应该用函数式的编程语言，函数式语言(例如：ocaml)比较安全不容易出现这种漏洞，而且从长远看要实现的是对智能合约功能的理论上要证明他的正确性。formal verification:理想化能够证明这一段智能合约语言的编写只能实现我们想让他实现的功能，现实中是不存在的。

1. 编写智能合约的语言应该有什么样的表达力？

solidity 语言是图灵完备的，但是会有漏洞，比特币脚本语言比较简单，目前没有发现任何漏洞。能不能找到一个比比特币语言复杂又比solidity简单的语言？不容易出现安全漏洞。

答案是找不到的。但是我们可以预测未来的发展：我们可以向常用的智能合约提供一些模板，也有可能有专门编写智能合约的机构就像律师事务所一样。

去中心化的系统像如区块链都是开源的，也就是透明的，因为必须要让所有的节点都执行同样的内容才能达成共识。开源的一个好处就是增加合约的公信力，接受群众的监督。

有些人认为开源的另外一个好处是安全，因为全世界的人都在看着这些代码，但是已经看到智能合约代码出现漏洞，全世界这么多双眼睛看着开源代码，为什么会出现错误呢？这种现象叫做many eyeball fallacy错误认知的意思，相当于misbelief。但实际上是真正有时间看代码的人少之又少，看的人不是很多，也不一定能看得懂。

1. what does decentralized mean?

区块链的追随者一般都是去中心化理念的追随者。以太坊的硬分叉是以太坊的开发团队说了算的吗？

当然不是，以太坊的团队升级软件之后，也是90%绝大多数的矿工用行动支持了硬分叉，剩下的一小部分虽然没有支持，但是也依然在旧链上继续挖矿，以太坊团队也没有办法强制所有人都升级软件。去中心化并不是不能修改已经制定的规则，而是修改规则要用去中心化的方式进行。硬分叉的成功是因为90%的矿工认为以太坊团队的措施是符合公众利益的。

而这种分叉正好是去中心化系统的体现，因为只有去中心化系统，用户才可以选择分叉，中心化系统只能选择继续或者放弃。存在分叉的现象恰恰是民主的体现，比如系统私自增多以太币供给量，使得以太币贬值，矿工就可以选择分叉继续维护原来的以太币。

1. Decentralized ≠Distributed

一个去中心化的系统一定是分布式的，如果这个系统只运行在一台计算机上，显然不能叫去中心化；但是分布式系统不一定是去中心化的，即使这个系统运行在成千上万的计算机上，如果计算机都是由同一个组织管辖的，那也不是去中心化，比如谷歌的search engine；在一个分布式的平台上可以运行一个中心化的应用，也可以运行一个去中心化的应用。

比特币和以太坊都是交易驱动的状态机，state machine的特点是让系统中几千台机器重复做同一组操作，付出很大的代价来维护状态的一致性，这个并不是分布式系统常用的工作模式，大多数的分布式是让每台机器做不同的事情，然后再把各台机器的工作结果汇总起来，目的是比单机速度快。状态机的目的不是为了比一台计算机的处理速度快，而是为了容错。状态机最早的应用场景：mission critical application.应用程序必须无间断的对外提供服务，哪怕宕机一分钟都会造成很大的损失，所以他才有好几组计算机重复同一组操作，这样即使有一台计算机故障，剩下的计算机也可以对外提供服务。 eg: airtraffic control; stock exchange; space shuttle.这样付出的代价是效率很低，几台机器合在一起比一台机器慢，因为需要同步状态，而且集群里的数目越多速度越慢，所以传统利用状态机的应用，机器的数目都是比较少的，可能就是个位数字。

**智能合约是编写控制逻辑的，只有那些互不信任的实体之间建立共识的操作才需要写在智能合约里。大规模存储和计算不适用这种场景(因为这种场景下默认所有节点之间时相互信任的)，又慢又贵，因为要耗汽油费，相比之下云服务更好。**