

智能合约的安全

欧链科技—区块链高端技术沙龙(一)



扫一扫关注欧链小秘书

添加时请备注“技术社区”

8月26日，欧链科技区块链高端技术沙龙有幸请到清华大学特别研究员、博导，张超博士做《智能合约的安全》主题讲座。张超博士首次演示虚拟机最新漏洞，推演ETH矿场攻击可能性，并探讨如何在智能合约中实现代码、协议的安全。该沙龙采取闭门形式，将陆续邀请国内外区块链知名技术专家做主题报告，并诚邀知名区块链行业专家和企业负责人参与讨论，会后将发布高质量的技术报告。欧链科技立志打造区块链第一技术品牌！

欧链科技：非常感谢大家在周末这样一个大好的时光，来参加我们“区块链高端技术沙龙”，我们欧链科技主要瞄准的是预言机服务，主要目的把链外的数据搬到链上去，实现不同区块链之间的数据通道。我们举办这个高端技术沙龙想解决的是产业界和学业界之间的问题，也是一种通道，我们希望把这个技术沙龙打造成国内第一区块链技术品牌，我们非常有信心达到这个目标，因为今天到会的都是各个公司企业技术的直接负责人，邀请的都是在区块链领域里面国内外非常知名的专家，我们希望在沙龙里面有更多的智慧的碰撞，甚至可以孵化出更优秀的项目。

张超博士：谢谢大家，非常高兴在这里跟大家探讨一下区块链相关以及安全的问题。我是北大毕业的，在加州大学做三年博士后，去年年底是去到清华做研究员。

About Me



- | | | | |
|---------------------|---------------------|--------------------|-----------------------|
| ▪ Peking University | ▪ Peking University | ▪ UC Berkeley | ▪ Tsinghua University |
| ▪ Mathematics | ▪ Computer Science | ▪ Computer Science | ▪ Cybersecurity |
| ▪ Undergraduate | ▪ Ph.D | ▪ Postdoc | ▪ Associate Prof. |
| ▪ Prof. Wei Zou | ▪ Prof. Dawn Song | | |

我的研究是关于网络空间安全的，网络空间安全现在非常热，去年全国各地都在建网络空间安全的学院，非常热。其中有很多话题，包括密码学大家都知道，还有其他的信息安全、网络安全，我其实主要做信息安全。首先研究的是什么，第一个是漏洞，有了漏洞才能做攻击，然后是我们就需要做防御。举一个例子，今年的 WannaCry 病毒背后的故事，它其实是去年每一年 8 月份的时候有一个黑客组织叫影子经纪人，他把 NSA 国家安全局给黑了，把他们的东西偷出来了，偷出来的是什么？网络军火库，包含了什么东西？有很多漏洞，以及怎么利用这些漏洞发起攻击。他们偷出来之后，往外拍卖，100 万比特币，当时的价格差不多是在几亿美元的级别，大概去年 8 月份。价格太高了，其实后来没人拍了，他们拍的时候希望这个东西是由政府来买，政府出得起这个价格的，但最后没人买了。他们就在今年 2017 年分了几批，4 月份大规模公开了，放到网上，谁都可以下载，包括我都下了，搞安全研究的应该都下了。影子经纪人然后用这个东西包装了一下，加了一些勒索功能，加了传播的功能，变成 WannaCry，5 月份就爆发了。

简单说一下我做过的事情。我们的学科特点就是它在实际中关联非常紧密，它不光是纯粹做研究的。实际上国际上有很多的对抗比赛，举一个例子，一个是 Pwn2Own，这是世界上最权威的，它针对的最新的系统，最新的浏览器等等，现

场攻击。今年前三名是 360、腾讯，还有我们实验室毕业的学生创业的公司长亭，他是第一次参加，拿第三名很好。美国前几年都参加了，后面没有参加，是因为这个比赛要求技术公开，但这个东西在美国是限制出口的，它在 2013 年就立法了，所以美国团队是不能参加这个比赛的。中国最近几年在这个赛场非常火。

第二个是 DEFCON CTF，这个历史更长，今年 25000 人参加。其中有 CTF 的竞赛，大陆在 2013 年首次打入决赛，就是清华的蓝莲花站队，最近几年拿了第五名，去年拿了第二名，这是目前国内最好的奖项。而且去年比较特殊是人机大战，不光是人比赛。近几年最火的是 PPP 站队来自卡内基梅隆的。2015 年韩国拿了第一，是因为他们出了一个天才，那个人非常年轻，当时只有 20 岁，那个小孩子是来自韩国一个叫做“优中的最优”的一个计划，从高中就开始培养这些黑客，一直培养到大学，其中优秀的人可以直接免服兵役，对他们很有影响力。

中国队去年拿了第二名，是最好的成绩。今年腾讯组织了四支队伍连队，国内的四支队伍组成了一个连队，拿了第三名。还有一个队伍，说一下是 HITCON，是台湾的队伍，台湾的队伍规模也是很强的，基本每年都在第二名、第四名、第五名这样的一个水平，比清华蓝莲花这边综合成绩还要好一些。

国内其实也有一些赛事。我们实验室在 2015 年的时候，是拿了第一名和第三名，拿了很多奖金。2016 年的话，我们拿了第一名，2016 年 10 月份这一次是做了 ps4 越狱，跟苹果手机越狱差不多，难度非常大。

刚才介绍的这些，在实际的黑客圈里边，在做安全的圈里边，这是一个竞赛，相互之间的攻防赛事。这些赛事基本都是靠人在做，这里边有一些问题，就是都不好做。从攻击性来看的话，通常先是找漏洞，找漏洞很难，根据目标程序不一样、目标环境不一样，它可能需要简单的是几年，复杂的要几年才能找到一个漏

洞。找到漏洞之后，找攻击更难。防守也不乐观，是被动的事情。有些发生的真实的攻击，这种攻击被防守方检测到，平均下来一年的时间就过去了。防守方知道之后，修补打补丁，有了补丁之后，你得部署上去才行。十年来的统计数据很糟，用户通常要很久很久才能更新上去。刚才说到打补丁，可能要两个月，实际上这不算极端的，安卓用户比较多，安卓系统现在最新的是 8.0 的，国内基本是 6 点几，还有一个手机可能 4 点几，都是三四年前，在我们的研究来看可以直接拿来用，来更新的，我们的用户太多了。所以最终到用户的手上好几年，防守非常弱。

大家看到的问题，人在做的事情攻击防御都不好做，怎么让机器来做？包括人工智能现在非常热的。去年我带队在拉斯维加斯参加 DARPA 挑战赛，成绩是资格赛防御是 1，决赛攻击是 2。这边说一下 DARPA 挑战赛，DARPA 是美国的机构，它非常有意思，它这个挑战赛，在此之前，它还有几个，一个是无人车，第二个是机器人，这次挑战的就是人工智能攻防。无人车大战 2007 年搞的，无人车基本上是产业化了。机器人这边的话，大概在 DARPA2011 年左右发行，机器人现在这边也有公司，波斯顿动力公司，这个公司是 DARPA 资助的，他们每年现在会有他们的产品，做得效果非常好，内容、机器各种稳定性都比其他做得好多了。



DARPA 的挑战赛，它的目标是说把一些研究最前沿的东西，一定程度上没有落地的东西，他能推动起来，让工业界同时做这个事情，把它产业化。网络攻防的事情做到往前推一步。伯克利花了 5500 万美元做这个事情，我当时是项目的负责人，做方案设计和开发。

前面简单介绍一下我的一些背景，所以可以看到基本上我是在针对传统的这些软件的系统来做的，最近我自己也做了一些关于区块链的东西，下面跟大家简单分享探讨一下。

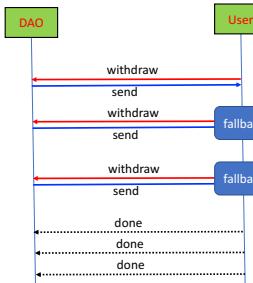
先从 DAO 说起，这是简单的时间线，DAO 的时间非常短，但是它涉及到 360 万的 ETH。然后它背后的问题是什么呢，这个不是 DAO Hack 自己来，核心的是他给请求，他把钱发送给了用户，送完之后，他才把这个记账的信息更新一下。这个直接看没什么大问题，但是区块链比较特殊的一个特性，可以看一下它有什么问题。实际上是用户发出的请求，要取钱，他可以把钱发给用户，正常是发完了就回来了，但是实际上是提供了一个 callback，就是收到钱之后他知道用户在干嘛，返回之后才到这个合约，把它下一步再做，这是正常的程序。所以攻击者可以做的事情是先不回去，我接下来发起另一个取钱的请求，这边接着再发起。为什么会发？是因为刚才发的前边的信息，它并没有更新，它还在，所以接着发。这样他就一直做，理论上把这个钱全给转走了，这就是它的漏洞。

The DAO Hack

- Sample code:

```
function withdrawBalance() {
    uint amountToWithdraw = userBalances[msg.sender];
    if (msg.sender.call.value(amountToWithdraw)() == false) {
        throw;
    }
    userBalances[msg.sender] = 0;
}
```

The DAO Hack: reentry



执行出来之后，大家都知道分权了，分成 ETH 和 ETC。ETC 里面漏洞给修掉了，但是攻击者钱还在里面，这个钱已经被转出来了，陆陆续续转出来 30 万，历史都可以查到。怎么修补的，其实比较简单，把这个记账这句话放到前面来就行了，如果发送没成功，把这个还回去，这是最保险的做法，记账放到前面去。实际上后来就是还有其他的问题，不光是刚才说的一个漏洞，还有其他的问题。

The DAO Fix

- payee pull payments, rather than payer push payments

```
/*
 * PullPayment
 * Base contract supporting async send for pull payments.
 * Inherit from this contract and use asyncSend instead of send.
*/
contract PullPayment {
    mapping(address => uint) public payments;

    // store sent amount as credit to be pulled, called by payer
    function asyncSend(address dest, uint amount) internal {
        payments[dest] += amount;
    }

    // withdraw accumulated balance, called by payee
    function withdrawPayments() external {
        uint payment = payments[msg.sender];
        payments[msg.sender] = 0;
        if (!msg.sender.send(payment)) {
            payments[msg.sender] = payment;
        }
    }
}
```

Other issues in The DAO

- Roles
 - token holders
 - curators
- Proposals
 - invest, minimum 14-day voting period
 - at least 20%~53.3% token holders must have voted
 - 20% + transfer_rate/3
 - split, 7-day voting period
- Lifetime
 - 27-day creation period
 - propose, whitelist, vote
 - split

在看这个东西之前，先简单回顾一下 DAO 的概念，两种角色，一种是用户，一种是做仲裁的。比如总共 11 个裁判，但是有一个人不参与，就 10 个裁判。10 个裁判要做出规定，参与人可以提提案，怎么投资，或者说我想把这些钱取出来。提案需要裁判来决定，这就需要投票，假设你想转移 100% 的资金，意味着你需要 53.3% 的这个人的同意才行，如果你是操作投资，只有 20% 的人同意，操作之后才可以进行投资。如果用户 split，会分出一个新的子的链，有 27 天的购买创始的时期，在这 27 天别人可以跟你买。这是简单的一个回顾。

那么 DAO 还有什么问题呢？第一个问题，DAO 虽然看起来大家可以公平投票，大家 50%、50%，但是实际上它这个下面投反对票的人可以不投，直接退出，可以不投，分裂出去。为什么这么做？因为它有限制，如果投票，这个东西冻结在那，你的钱冻结在那边，你不能其他事，只有投票结束之后才能恢复。这样如果我不想投它，不参与，就退出，等这个投票结束再回来。所以他想做到这个就退出了，可以投赞成票，也可以投退出票。这是它的一个设计上的弱点。

第二个是被打埋伏了，就是围歼了的意思。什么意思？假设有一个庄家，他掌握了很大的资源，他提了一个对其他人不公平的意见。这种情况下就是刚才说的，理性的讲我不赞成，我退出了，剩下的人不退出，但是他可能去投了一个反对票，其实这部分很多人都投了反对票，也没用，因为发起这个合作的人他带的资源很多，他可以带着这个投票快结束的时候，投赞成票这样就会投过，这样相当于他通过这种形式，把大家一点点资源搜集起来，来实现他的目标。

Other issues in The DAO (1)

- The Affirmative Bias, and the Disincentive to Vote No
 - DAO blocks token holders from splitting from the DAO or from selling their TDT once they have voted on a proposal, until the voting period ends.
 - Token holders who would vote No, could just split and leave DAO

Other issues in The DAO (2)

- Ambush Attack
 - The attacker (a large investor) proposes a Bad proposal
 - Most rational actors would leave The DAO
 - Remaining actors may vote NO, even with a high rate
 - The attacker votes YES at the last minute of the voting period

第三个是尾随跟踪。这个攻击是什么意思呢？如果你一个人想把自己的钱退出来，需要用分裂来做。什么意思？也是创建新的链，它跟分裂是一样的。它有 27 天创始期，这个期间内，别人是可以跟进来的，可以买进来，进入它的子链。所以攻击者也可以跟进，而且攻击者还跟进了许多，占 70%。这个人再想把钱取出来的时候，他想把钱转出来，转到外面去，发起一个提案。但是这个新的 DAO，

有这么多人赞成才行，但是用户自己占的小于 30%，所以最后你想退钱的这个人，你自己来说票权不够，它占了 70% 的投票，它只要不投票，就没法通过，你的钱就无法从这个子链出来。但是这个时候呢，就是说我这个用户虽然不能从这个子链取钱，但是它可以接着再分，再提起这个要求，但是黑客还可以再跟进，这样不断的跟着你进子链。它的攻击项目就是你要退出去，钱从这里边取出去，它的攻击效果就是勒索，给他点好处他可能就这么干。

第四个是割韭菜，这个可能就是赌博了。攻击者不一定从 DAO 挣钱，它可以从外面挣钱，它可以从外面买入卖出的操作来挣钱。所以他只要让市场上不去买或者不去卖，他就能操控这个价格去落地。比如前面说到的，这个不让你退钱出去，外边想参与的人这个就不愿意加入，不愿意买了。还有一个就是说我可以提一个很自私的提案，我很多这样的投资，这个提案相当于垃圾邮件，大家不胜其烦，大家也不愿意参与，所以大家打压这个价格，都从外面的市场去做。

Other issues in The DAO (3)

- Stalking Attack
 - the attacker buys tokens when child-DAO is created by victim
 - controls at least 70% tokens
 - does not vote on the victim's refund proposal
 - the victim controls x tokens
 - when $x < 30\%$, $(20\% + x/3) > x$
 - he could never get enough votes to approve the refund
 - but the victim can split the child-DAO again, and
 - the attacker could stalk again...
- Attacker could thus
 - ransom, blackmail

Other issues in The DAO (4)

- Token Raid
 - The attacker (a large investor) drives TDTs lower in value
 - if he could encourage users to sell TDTs on exchanges, or
 - if he could discourage users to buy TDTs on exchanges, by
 - Amplify attack threat: make public worry about attacks, e.g., stalking
 - Disturb the voting: propose self-serving proposals
 - The attacker could then benefit by
 - shorts or put option
 - buy TDT back at low price

第五个也是 DAO 的问题。DAO 设计的时候考虑这个问题，它最主要的就是防御这种问题。如果一个账号占了 53% 的投票权，他可以提一个只对自己有利的方案，他可以把所有的钱转给自己，因为他已经超过了 53% 的投票权，所以投票一定是通过的。所以在 DAO 的收购的时候，有裁判，裁判可以用来检查这种提案，确保它不会有这种比较过分的要求。但是让一个人来看这个提案，只要把它

做得很复杂，这其实很难的所以裁判看不出来这个到底有什么问题。

第六个，这个是说先提一个好的方案，这样先赞成它，而且投票期设的比较长一点，在投票期之前进行另外一个投票，那个投票跟这个反着来，但是很多人已经投了之前的，他的钱冻结了，不能再参与这个投票的，这个方案就控制成功了，它就可以通过了。这是 DAO 的问题。

Other issues in The DAO (5)

- Majority Takeover Attack
 - a large voting bloc, of size 53% or more, votes to award 100% of the funds to a proposal that benefits solely that bloc.
 - The curators are supposed to audit this type of proposals and forbid them from voting
 - but, the proposal could be made complicated and hard to validate

Other issues in The DAO (6)

- Concurrent Proposal Trap
 - The attacker first proposes a **GOOD** proposal with long period
 - to attract token holders to vote YES
 - The attacker waits for a while, and propose an **opposite** proposal
 - token holders are trapped in the previous voting
 - the attacker could easily affect the result of this proposal

下面接着讲整个区块链的其他的问题。在讲之前，先给大家介绍一个概念，叫 Intel SGX。这是一个环境，现在包括互联网，系统的硬件的一些东西，很多时候不可信，你都不知道什么地方会被替换过和更改过，都不可信。所以大家找一个可信的执行环境，这是可行性计算一帮人做的这个事情，这是最新的研究成果，SGX 包含了这个特性，都包含了。它是怎么实现的这个特性，它加了一些特殊指令，帮我们做了测算，他提供了一个环境，他可以把比较敏感的代码，比方你处理你的私钥的比较敏感的代码，把这个代码放到保护环境内执行。它最后安全环节是可以加上 CPU，假设这两个安全就行了。所以只要你 CPU 是安全的，内存、系统全部不安全没问题，他可以给你提供保障。

如果代码在里面是很好的，它很安全。但是大家很关心的是我的代码是不是真的在里面跑呢？这个事情可以证明，Intel 提供了一个证明代码在里面跑的，向它发送请求，证明一下代码确实在跑。具体的问题不展开了，时间关系，总的就

是说，它可以确保代码的安全执行，所以这样可以比较确信它没有什么大的问题，这是 Intel 背书的。代码一旦在里面跑的时候，它的代码的机密性、完整性，处理数据，外面的人看不到，大家知道这两个就行。我会从效果、安全和隐私这三个方面来介绍它在 Blockchain 中的应用。

大家知道效率很重要，我们来关心一下效率的问题，分两个来讨论，第一个是 POW，另外还有一些方案提出来。第二个是关于以太坊的 Gas。

这个就是刚才说的 POW，它可以跟 SGX 结合。这个要解决什么问题呢？你看现在有多少人买 ASIC 矿机来挖矿，这样是对普通人不公平的。有了 SGX 可以限制，你不能用矿机，你只能用 CPU。做法其实就是让它验证过来所有的节点，我要验证你上面跑的确实是 SGX 芯片，不是别的芯片，传统的做法是 PoW 还在，它的算法加一个签名，可以把它发出去。其他的节点可以验证这个方案的东西，它是 SGX 的东西，这个可以确信，节点是在 CPU 上，不是在矿机上。

这个方案的本质还是用传统的 PoW，所以它会占领大量的资源，做一些无用的事，这个不环保，还浪费电，所以有人想替换它。一个是叫 Proof of time，现在无非就是每个人你找出这个解，可能花的时间不一样，最后验证的是谁先找出来的。这个方案 Proof of time，我把这个时间放在那，我这个机器休息了一段时间，就是随机的。谁最先醒过来，谁就有话语权。所以它的核心就是说我要保障这个睡眠时间是随机的，这个如果没法保证，就是作弊了，我说我自己随机的，但是没有人证明，没有办法证明，但是如果把这个代码放到网上去，它通过验证可以知道，它通过协议是不是我的代码，这个代码是随机休眠的，我就可以确信，最后这是 SGX 执行的任务，基本上确信。它的好处就是说休眠起来不会浪费你的计算能力和能源。

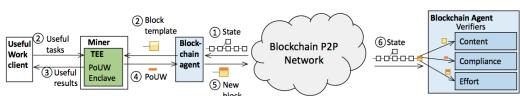
这个方案把所有的权力都交给 CPU 了，这样就有了非常强大的动力去攻击 CPU，一旦攻击破就没法保障。但是攻破 CPU 非常困难，但是这个事情害怕万一，如果出现了也可以搞定，就是一个巨大的安全事件。

这个方案比刚才的方案更好一点，它依赖芯片，不会浪费资源，但是黑客就直接去攻击 CPU 了。把上面这个方案改一改，现在变成验证拥有证明，这个有另外一个方案出现了 **Proof of Ownership**，叫做证明我有这个 CPU 的对象，做法就是投票，每一轮的时候要投票。其中收到票数最多的就是出块者，投票的时候每个 CPU 只能投一票，这样验证了你其实投多数票，你可以验证，这两个票是一样的，投的就是 EPID 这个特性，这个特性可以验证，所以这样每个验证只能投一票，票数最多的就是出块者。另一个方面，时间、工作量量都非常好，为什么说时间比较好，因为它不用休眠，它直接投票就好了，但是问题是它需要对每个 CPU 投票的进行验证东西。

还有今年最新的一个工作 **Proof of Useful work**，也是替换这个 PoW，现在 PoW 它做大量的计算，这个计算没有什么用，这个想法就是把有用的计算放进去，告诉你有用的计算算了多少。当然不是所有的计算都参与进来，它需要的就是进入特定的编码编译一下，跟这个相互认证，进行采集它的工作量信息。这个方案同样需要放在 SGX 里面去，这样才能保证它不会作弊，然后它会算出它的实际工作量，实际工作量算出来之后，它还除以随机数，因为机器有云服务器，不除随机数不公平，这样就是放到区块链里面去。这是今年最新的一个方案。

	ASIC resistant	Energy efficient	Time efficient	Scalable
Bitcoin	no	no	no	yes
SGX proof of work	yes	no	no	yes
Proof of time	yes	yes	no	yes
Proof of ownership	yes	yes	yes	no

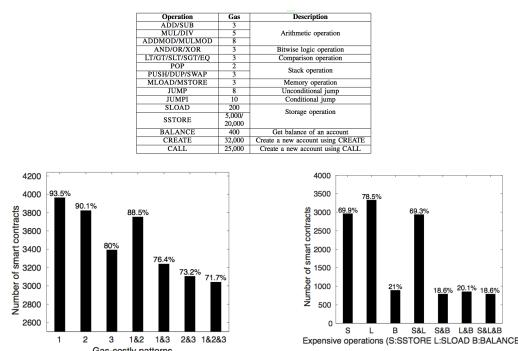
Proof of Useful Work



useful_work() / random(), to be fair.

刚才介绍了 PoW 的优化，还有一个是我们做过的关于 Gas，智能合约 Gas 的优化。我们做了一个分析，每条命令需要的 Gas 是不一样的，有的要 1 个 Gas，有的是几万个 Gas。我们做优化，可能是比较高，消耗的需要操作的比较多，尤其是循环，最后的能量更大。我们最后就是做了一个检测工具，能检测出一些优化的点。这个图显示的是 Gas-costly patterns 额，绝大部分的合约都没有优化。第一种是 93.5%，第二个是 90%，第三种是 80%。三种都有是 70% 多，所以现在绝大部分的都不优化，都浪费了 Gas。这是各个操作，SSTORE 是 2 万，这是操作的数量。出现一个循环的数量，大概 70% 的智能合约的硬件它叫做 SSTORE，而且是在一个循环里面。我们正在做的工作就是说自动去做一个优化。

We could find non-optimized code in smart contracts, and fix them.



刚才讲的是区块链的效率，然后再看一下安全。现在看一下智能合约。智能合约，它是操作系统，跟应用融合的东西。这个合约的操作需要有一个交易形成，或者是其他的智能合约给它发的数据。智能合约是运行在操作系统上的，它依赖

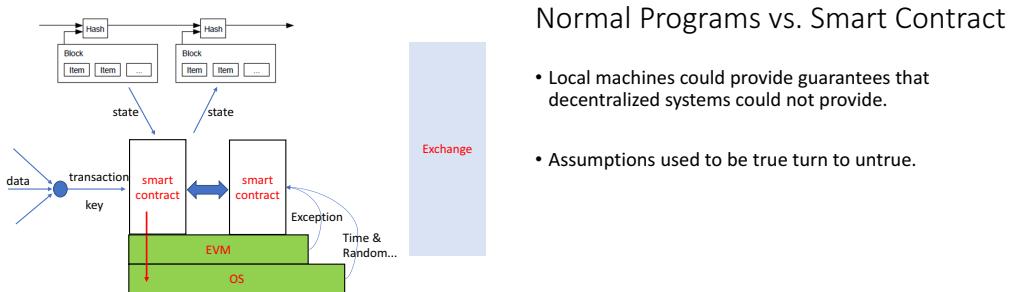
操作系统，有时间的信息等等。

我们需要从外部输入数据进入智能合约，那就需要比较可信的数据录入方式，不是随便的数据就可以输入进来。这有一个参考方案，借用了 **SGX**。我们从一些加密的官方的网站去获取最新的真实的信息，这个数据拿到之后，经过一段时间的包装，它负责发给智能合约的代理，再发消息，发给智能合约，这是它的总体的解决方案。这里面有时间和信任问题，这个代理因为它进行过检查，它是比较可信的。这一块执行起来不需要共识，但是问题是 **SGX** 只具备一些计算能力，像交互的网络的能力没有，所以网络的东西需要有特殊支持，需要设计方案的人员做处理。这个方案要求的网络数据可信，通过 **https** 传输，如果这个数据不可信，我们可以有多个数据源，采用投票机制，投票比较多的数据是我们的最终数据。这是简单的一个解决的方案，当然也会有其他的方案。

第二，就是 **Key**。**Key** 其实本身是怎么保护，钱包、服务，还有一个是资源管道。另外一个问题，在区块链也出现过，分权的时候，**Key** 也有问题，比如说以太分权，还有 **Bitcoin** 分权也有问题。因为它可以重放，因为它两边的情况不一样，所以用户有交易，可以把这个交易去放一下。如果在区块链上付钱给攻击者，另外一个链上受害者的钱直接到这个上面解决的方案是用户把钱转走，在主链上做，在另外的子链上去做，有自己的链上去。其中主链上高一点，别的链交易费低一点。

其中有一个核心的问题，它本质上可以看做是一个程序，大家看传统程序有什么特点？它之间的输入输出是 **state**，可能有变化，很多环境下不会有这个情况，它的状态不会说被别人改了，这是它的一个特性。你在 **windows** 跑的时候，它还是给你提供一定的保障，不会说乱，你的数据不会变，它是你自己操作的，

不会在别的链上出现。但是在区块链上有很多这样的假设，大家认为其实开发人员认为是正常的东西，在区块链上是正常的，但是他的假设不对，我们可以举几个例子。



第一个是交易顺序的问题，假设这边是发奖金，这个合约可以设计成我可以撤销这个奖金了。有意思的是，如果有一个人他提交一个解决方案，准备领悬赏的奖金，但是这个时候发布赏金的人发现到有这么一个交易，他自己也发起一个交易，说我这个悬赏取消了，不会再发了。但是注意，之前那个交易已经传播出去了，就是这个解决方案他已经知道了，但是如果后面这个交易先处理将意味着这个悬赏取消了，但是你的答案悬赏这个人已经知道了。这是第一个问题。

第二个问题是关于时间的，智能合约需要时间的信息，这个信息由底层提供。但是这个时间，因为它是底层的东西，如果操作系统不可信，我可以改，这样影响时间，把这个改变在最终对它有利的方向。还有一个是随机数，这个底层就可以提供一个对它有利的随机数。

还有，所有的程序执行都一样，都会有异常出现，如果你没有把异常处理好，这样就会出问题。主要的问题就是给别人发钱、转钱的时候，如果没做检查，它有问题。在一定的情况下，这个 `send` 会触发异常，比如这个函数序列太长了的话，把栈撑爆了，最后这个 `send` 的函数就没法执行了。但是你如果外面没有检

查异常，它并没有检查 `send` 出现，它会接着往下走，这个状态就不对了，这里面就存在着攻击。这个例子显示的就是说在以太网选一个国王，谁给钱多谁就称为国王，其中可以拍卖给下一个人。他给了现在的国王就是报一个价。攻击者报了一个价，然后这个钱并没有送回去，这就是一个例子。

这是个统计数据，现在存在着刚才说的几个问题，有 5000 多个智能合约没有处理 `send` 异常，有问题。**3000** 多个是有交易顺序的问题，他们验证交易顺序是有问题的。还有一个是智能合约，它可能会影响下面的，一个恶意的智能合约能影响别的智能合约，还是这个智能合约会影响下一个。

Mishandling Exceptions

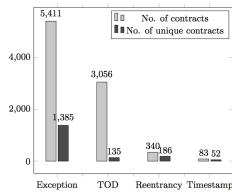
- **Unchecked send():**
- too-long call stack will fail, then no send operation is done

```

1 contract KingOfTheEtherThrone {
2     struct Monarch {
3         // address of the king.
4         address ethAddr;
5         uint256 valuePaid;
6         // how much he pays to previous king
7         uint claimPrice;
8         uint coronationTimestamp;
9     }
10    Monarch public currentMonarch;
11    // claim the throne
12    function claimThrone(string name) {
13        ...
14        if (currentMonarch.ethAddr != wizardAddress)
15            currentMonarch.ethAddr.send(compensation);
16        ...
17        // assign the new king
18        currentMonarch = Monarch(
19            msg.sender, name,
20            valuePaid, Block.timestamp);
21    }
}

```

Buggy Contracts



还有一个是智能合约，智能合约自己是不是都是好的？因为智能合约是谁都可以上传的，上传恶意的智能合约会有什么问题？这是我们团队目前在做的事情，它可以攻击最底层的主机。这个能干什么呢？最简单的就是耗时，或者说我把资源占用很多。我给大家看一下我们的视频。这是我们实验室的同学做的，这是最新版本的 VM 虚拟机，这是虚拟机内部打开 firefox 浏览器，这是当时最新的 firefox 浏览器，然后让它去访问一个网页，这个网页是我们事先搭建好的一个网页。OK，这边刚才已经访问完了，这边是虚拟机的管理器，大家看一看。这还是在虚拟机内部，它很快就要打穿虚拟机了。好的，这已经在宿主机了，已经弹出计算器，实际上是任意代码都可以执行，我们只是调其了计算器。【欧链科技注：隐去攻

【击细节】

这里面每一层都有很多的防护机制，防护考虑，突破一层很困难，刚才演示的是突破了三层，浏览器，然后 windows 系统，最后 VM 虚拟机，突破三层。这是实际的现实的威胁。回到刚才说的区块链，它同样是有这种问题存在的，刚才说它的架构，虚拟机、操作系统，一样的架构，原理上我们如果找到类似的漏洞，可以做这个事情，这样想要控制你的 Gas 不是问题，因为我知道代码，所以它可以做很多事情。最有效的是去消耗其他节点的资源，让我们自己来完成挖矿。

刚才讲到很多问题，上面面临很多的威胁，怎么保护他们。其实这个和传统安全是差不多，也面临同样的问题，现实的环境中同样是数不清的漏洞，虽然是数不进的漏洞，但是想挖出一个新的还是很难，尤其现在有大量的工具在，各种挖。区块链的安全相对来说研究得比较早，第一种方式是说区块链的程序需要加固，在你发布之前是需要经过安全检查，经过安全加固之后才能真的去部署，要不然一旦出现像前面提到的漏洞的时候，你再去替换，太难了。你即使想替换，就是先部署，后面再替换两种方式，要么就是说你有一个比较全力的组织，他可以决定什么来替换，它的好处就是可以比较快，一旦出现这个事情，这个组织决定要换下去，但是它就是违背了区块链的去中心化的东西。所以如果一个组织是恶意的，想做这个是比较麻烦的事情。还有一个是盗亦有道，大家投票也行，但是问题是这个事情要做，他就非常慢，大家达成共识之后，可能攻击者就没影子了，非常慢，这些其实都不是特别好的办法。

我们做的方案是什么，第一个方面，前面我们介绍相关工作的时候也说了，核心的想法就是我们对这个程序，或者智能合约，我们对它安全检查，对它进行加固。经过这样处理之后，它可能面临的安全风险也小一些。

还有一块区块链的安全问题，就是交易所。交易所其实它自己的问题也挺多的，交易所现在很多的安全情况就是原来传统网站的问题，举几个例子，这是交易所被黑客攻击了，门头沟。所以它为什么会发生这个事，是因为它没有传统世界安全的概念，这本来就是传统的安全。而且这个事情很多大学出过，大家听过各种各样的被脱库，就是网易、迅雷、雅虎等等，它的用户数据主要是用户名、密码，还有其他的输入用户信息全部都被拿走了，这个可能绝大部分的大公司都出过这种事，只是有的被曝出来，有的没被曝出来。最有名的就是雅虎，最后曝出来的是雅虎的用户信息全被偷了。这样来说，我们仔细分析一下是什么原因，它可能是差不多的，就是它的服务器，信息安全、网络安全没做好，被人攻击了，传统的网络安全。

最后讲一下隐私，我在做的一个东西，就是对现在比特币上面和以太坊用户的身份和识别，我想跟踪一下，到底哪些地址是属于同一个人、同一个组织，这个是交易所，还是黑市的，还是地下经济犯罪的这些人，就是从各种蛛丝马迹可以追踪，我现在是在做这样一个事情。当然这个可能有一些地方的朋友关心某一个人有多少比特币，把这个信息识别出来，推断某一个人拥有比特币这样的信息，然后追踪他的金钱流动的情况，这是一个问题，这是我们正在做的事情。

OK，今天我就讲这些内容，谢谢大家。

欧链科技：非常感谢张超博士给我们做这样一个演讲，相信大家对很多细节比较感兴趣，接下来可以进入自由讨论的环节。我先问一下现场的嘉宾有没有谁有什么问题，想要提什么问题？我来引一个头吧，刚才讲到的穿透打透虚拟机的实例，我相信在场的矿场主应该会比较关心这个问题。其实刚才张博士讲了，在

实际的场景当中，我们有没有这样一种可能，让别人的机器慢下来，因为我们之前老想着怎么提高自己的效率，这种 Hash 算法怎么提高。但是好像张超博士给我们提了一个新的思路，我们自己快不了，就可以希望别人慢下来，所以这个问题可以多探讨一下。张超博士，你前面举的例子是 VM 虚拟机。比如像以太坊，我们用的 EVM，像接下来的 EOS 用的 JS 引擎。比如像 JS 这种引擎，我们现在有没有一些已知的对它的研究，发现它有没有什么漏洞在？

张超博士：这个一直是有的，里面问题还挺多的，而且 JS 表达的能力很强，JS 它基本是图灵完备的语言，它能够做的事情特别多，它里面的安全问题，现在有一些研究，它没有像传统的软件，大家研究了特别经典的问题，没有那么深入的研究，因为它是动态的语言，它的特性太多了，太多了的时候，我们没办法对它进行一个综述，就是我们在测试的时候，我们找它的漏洞的时候，就不能像传统的漏洞有很好的建模，或者一个模型去表达它。大家现在没有花太多成果把它很好的做出来，但是大家有一些尝试，就是不是特别完备的方案，就是暴力地去测，用别的方法去测，去发现一些问题。这个每年的话，关于 JS 都会爆出几十个漏洞，他们通常拿这个漏洞，去突破浏览器的防御，打到系统里面去。这边如果是区块链的话，EVM 也接近 JS，它们其实从原理上讲存在同样的问题，而且还有一个问题，下面的阶段性，如果他们是传统的引擎，现有的方案已经可以很好的对他们进行挖掘，这是我们正在做的事情。我现在没有看到相关工作，没有看到具体的例子，但是这个事情肯定可以做，而且从理论上讲，一个程序是没法保证它没有 bug，而且它的 bug 还很多，我们利用一些已有的方法和工具来做这个事情。

【欧链科技：隐去提问者姓名】

提问：那是不是可以这样假设，现在大家认为智能合约都是为大家服务的，提供了某种的服务方式，是不是将来智能合约有可能是病毒，它有可能造出一个炸弹，直接穿透 EVM，有没有一些手段，传统的有防火墙软件，可不可以设想一些哪些防御手段是我们能利用到的？

张超博士：这个方法还要从现实来看。最早大家用的 windows 操作系统比较多，windows 应用的发布，没有正式渠道，从网上就能下载。但是实际上后面的 windows 也有 Store，他们提供一个 windows 相应的集中的渠道。另外一方面，在手机端他们也一样，他们只在市场下，不能自己装自己的程序。google 也是，有很多的地方可以下。

回到智能合约，目前智能合约的发布没有限制，跟 windows 当年最早的情况一样。没有限制的话，有可能存在什么问题，这个体系其实有问题，大家随便发布自己的东西肯定有问题。解决方案是什么？跟之前还是比较像，现实中我们发布的软件，苹果的软件，或者 google 的软件，我们要让它到市场上才行，这个时候苹果和 google 不会给你放到市场上去，他们要做一个事情，他们先检查。早期的时候是人工检查，后来自动化。他检查到恶意行为就不让你放进去。虽然很简单，但是它确实对这个安全有很大的提升，但是它并不能完全的解决问题。现在市场上还有相当部分的情况，包括苹果，很大一部分，应该说基本可以 20%、30% 的，很多，都是这个情况，这个做不了。对应的区块链也是一样，我们可以建立一个市场，所有的区块链由一个商店发布，这个商店提供服务，做一个安全检查，明显的是恶意行为的，不要让它发布上去，这是起码第一个要做的事情，因为如果这个门槛不卡住，基本所有的东西都会发布。但是这个不够，不够怎么办？那就需要第三方服务来做，第三方提供这个服务。第三方的服务可能就更全

面一些，不光是帮你检测一些漏洞，就像我们刚才说的智能合约，去解决里面的漏洞，这是第三方情况，可以来提供这个测试，包括众包，跟现在的传统的区别，用各种方式里对你的原因做一个分析，最后会出一份报告，包括里面的漏洞，还包括提供一些解决方案，由第三方来做。这样之后会有一个相对安全的环境，但是还是没有办法 100% 的，因为这个确实是满足不了的事情，但是至少经过这两个，一个是市场做逐步的筛选，还有第三方服务来做，这样才会有比较好的安全环境。

提问：我有一个问题，我们区块链是一个底层设计，实际上回到区块链有其他的服务存在，比如我们有交易所，我们有做自己的钱包。这里面很重要的一点，它的安全性很大是依赖于随机数，用户去生成账户也好，但是我们可能很多情况下，无论是我们的软件环境、硬件环境，以及其他其他的环境，得没有办法去形成一个很好的随机数，能不能给我们讲一下随机数有多重要，我们知道它很重要，但是不知道重要性有多大，如果随机数的质量不好，我们后面怎么去对它进行一个防御？

张超博士：随机数如果有一个非常好的随机源，问题比较好解决。随机数其实刚才 PPT 里面也讲了，上面有一个智能合约，它需要这个随机数，用随机数来做这个东西。如果判断依赖随机数就会有问题，至少现在的随机数的来源是不可信的。刚才前面讲的 SGX 的时候，SGX 也是这个事情，也需要随机数，比如他替换 POW 的时候，就是随机本人，这一块硬件上也在做，包括现在叫 FPU 的一个硬件，它的原理是说制造这个东西，它不能被复制，它肯定有差异输出，有这个硬件在做这个事情。总的来说随机数很多地方算法可能需要，只要把随机数放上去，做一个值才能确保安全。

提问：我提一个简单的问题，你们刚才提的太深了，之前你们提过重放，就是有很多的分叉，BTC，我们分出来一个叫BCC。事实上现在如果在BCC网络里面，你们发了一个转账的请求，大家知道别的网没有这样的东西，其实它是完全可以在这个BTC原网重放。大家知道这个消息，如果说有一个工具在BTC和BCC网络去找，感觉应该能找到。就是说相当于在BCC网络里面转了一笔钱，它完全可以把这个放到BTC里面，再到A和B，再发这笔钱。其实攻击和防御是配对的，为了这个是不是需要注册一个新的账户，转移到新的账户上去，来避免这种情况。所以在座的如果有这样，还是要多一点点。

嘉宾：BCC有解决这个问题，它和BTC是双向防重放的。用了不同的签名标签。

提问：这是第一个问题。第二个问题是这样的，其实您分析DAO的这个攻击原理，也讲了DAO有很多问题，我看您提了大概有差不多四到五个漏洞。其实会不会有这样一个问题，就是DAO它的想法起来很纯粹的，它是想通过大家投票的原理，我们来达成一个在参加DAO这个项目里面这么多人形成一个相对公正的投资决策。其实是否能够得出一个结论，这件事是很难的事，就是想通过一个数学和投票，包括通过这种比例权重的控制，来压制人的恶意，就是从这个规则里面去获益的可能性是很难的。

张超博士：它这个非常难，首先这个事情很复杂，更复杂的事情，它只是一个小的环境，交易所，它可以在DAO上不挣钱，在外面挣钱。所以它做的决策，投票不一定从它自身的意义出发。

提问：首先非常感谢这个演讲。两个小问题，一个是隐私保护，我们提供了很多应用都是做企业级的。能不能简单介绍一下隐私保护这个状况，我们正在考

虑这个事，就是你能不能介绍一下，国内都有哪些团队，科技部，什么清华、北大都在怎么搞这个事情，这是第一个问题。第二个问题，国内有一些钱包，钱包里面到底有多安全，有几十个亿美金在里面，但是每个人都要验证过，这是一个国计民生的大事，所以你们现在做这个攻防，有没有考虑做这个东西，或者有没有合作做这个事情？一个小问题，谢谢。

张超博士：数据隐私保护这个问题，其实我们没怎么做，但具体应用中漏洞挺多的，隐私是和实用是在一块的，隐私也是相对的，可能在某种角度上都一样。但是实际上区块链的数据隐私保护这块，我了解其实并不多。我知道国内南大有个团队在做，复旦那边在做，其他的我没有做，这第一个问题我没法回答。

第二个问题，钱包的安全真的是值得去做，这个里面可能是有很多传统的安全问题，我建议需要找一下安全服务公司，来帮忙做一个什么支持，这个东西非常危险，像我自己现在做一个大型的服务器都非常慎重，对外开放端口非常慎重。我们搞安全的人特别害怕被别人搞，防不住，现在搞安全的都防不住。所以建议最好是找专业测试的人做一下。

提问：现在钱包已经是一个非常庞大的体系在，有这个因素，我觉得我们业界要有这个共识，甚至于有个机制在里面。现在非常害怕，我挣的钱，我不敢用，不敢放哪个钱包。

张超博士：确实影响非常大，而且做黑产的人比做安全防护的人更有驱动力。我们做安全防护的人，你们不请我们做测试，我们不敢去测的，他们没有这个顾忌。所以大家一定要去找专业做安全的公司，像前面提到的长亭，其实也有相关的服务，就是大家感兴趣的化可以找他们。

提问：这个事情可以做出商业服务来，这个事情非常关键，包括找第三方验

证这个事情，现在找国外的，远远不够。

张超博士：这个安全的问题，对我们来说做防守的相对来说开放一点，前几年我们找微软合作，微软很烦的，他觉得在给他们找麻烦。还有很多的公司，你找他的漏洞，他就报警。现在让大家意识到，我们不是坏人，我是真的帮你找出问题，提升你的安全性。所以这两年，有的公司悬赏，你找到这个我来给你悬赏，可以奖励你，你把它修掉。去外面，大家很多人没有这个安全意识，我看了一下，查了一下，以太坊有这个基金，做安全防护的有这种奖金，包括一些服务行业。但是其他的很多各种钱包里面的还没看到，说有这个钱什么的，没有。所以导致白帽子的人没法参与，只有真正黑帽子的人在干这个事。

提问：我问一下，第一安全问题的出现是一个行业发展到一定阶段的必然结果，因为早期这个行业不挣钱，现在出现这个问题，正是说明这个行业到了这样一个时期，但是它是在应用，具体的讲安全，实际上面大家更多的还是讲做应用，在这个时期把安全做出去，是这样一个点。再回到刚才钱包问题，这个行业缺少资深同学，缺少十年以上的大 V，现在的区块链程序员这个价是浮动的，它比起传统的那些在 BAT 这些地方真正做过大型的系统，真正做解决这些问题的人是不够的，我觉得这是人才的问题。然后再一个，比如说钱包，交易所也是这样的，现在一个交易所包括很多 BUG，这根本不是安全问题，也不是区块链的问题，只是一个非常普通的网站的问题。这个行业这种人都没有的，再像安全的领域，这是更高级的，像特别牛的人没有进入到这个行业里边来，这个行业的安全问题很难解决。

欧链科技：我们国内交易所都还是做得不错的，至少今天在场的这两个交易所技术就都很完备嘛。

张超博士：对，大家说到安全问题，就是产品建设的问题，确实是这样，安全问题哪都有，但是只有当一个市场足够大，或者市场影响力足够大，尤其是直接剩下钱的，因为传统的需求更高，这个漏洞不看，钱就没了，这是本质区别，所以区块链更早把安全信息考虑进去。

但是非常好的人才，某种程度上不会有大大提高，我们都见过，特别资深的成员也没有用，它涉及到一个大的分布式系统的时候，你要把这个协议设计清楚，很难，很多东西都没法建模，很多的东西想不到，这个是非常困难的事情。所以这个说话直接靠一两个架构师，或者比较有经验的人，他是解决不了的，你一定是说有很多人帮你做这个专业的安全服务的公司。你刚才说的第三个问题，就是需求他满足不了，这个解决方案，确实可能没有找准人才，可能这就是比较好的，把预算解决掉。

提问：我了解之前还做过形式化验证，去验证智能合约，这方面什么情况吗？

张超博士：形式化证明理论上很好，听着很美好，但是它有几大难题，一个是说你要验证建模，你到底验证什么，你不是所有的问题都验证，你非常准确的刻划属性，你要把它刻划出来非常难，因为它要涉及到一些交互，所涉及的东西非常复杂。你有了这个问题之后，只要程序规模一大，它就进行不了，就要终止，因为太复杂了，解决不了。现在能做大规模的，这是两个核心的问题，很难解决，某些特殊的厂家你可以用，确实满足了这个属性。我相信这个可能会遇到一些常见的问题，这个不是他自己做的事。

提问：我问一个比较外行的问题，因为我在这个区块链是比较外行的。传统信息安全我发现是这样，从技术上来讲，补一个漏洞，这个漏洞还会出现，反复的往下进行，永远在这个漏洞上。所以我们认为在技术安全之外，管理安全是一

块。但是在区块链，去中心化的平台，引入一个中心化的一个商店也好，或者上面跑的应用，智能合约，或者他们说的加的第三方的评测的这种，对你的钱包也好，其他的东西也好，进行一个评测，第三方认证。这些东西都加进来更好，还是纯粹的智能化，纯粹从技术角度解决问题？在咱们国内的圈里有没有这种模式或者一些东西？

欧链科技：这个问题我来替张博士回答。从安全的角度来讲，它是需要一个中心化的，还是去中心化的方式，还要看后边的时间去操作。比特币是一个完全去中心化的网络，结果它还是爆出了中心化的分叉方案来，完全去中心化是理想下面，但它要和现实生活中结合起来，多多少少还是有中心化的，因为大家参与得越多，那么这个网络才更有价值，人们想去参与，还是有这种传统的思维在里边，需要有一个新的主题在里面，我是这样的一个解释，所以我觉得安全问题还得往后看，还得去往后看它的一些解决方案。

欧链科技：看大家刚才激烈的讨论，我来总结了。大家说到安全，具体的实例大家特别感兴趣，特别是前面的 ETH 炸弹，这是矿场主关心的，还有钱包的安全，这是在坐的几位大佬关心的，还有包括数据隐私保护，做应用的企业都在关心这个吗，张超博士都给大家做了很专业的解读。大家可以持续关注我们的这个区块链高端技术沙龙，后面还会有更有意思、更加技术的主题演进。

刚才有位老总提到人才培养的问题，实际现在我们和北大的网络空间安全研究院正在合作，做区块链人才的定向培养，在座知名企有人才需求的时候，可以考虑一下到北大去找。

【完】

OracleChain 团队
2017 年 8 月 28 日