doi: 10.14132/j.cnki.1673-5439.2019.04.009

智能合约中的安全与隐私保护技术

王化群,张 帆,李 甜,高梦婕,杜心雨

(南京邮电大学 计算机学院,江苏 南京 210023)

摘要: 区块链是一种全新的分布式基础架构与计算范式,利用有序的链式数据结构存储数据,利用共识算法更新数据,利用密码学技术保障数据安全。区块链 2.0 的最大特性就是引入了智能合约,可以基于其架构开发各种用途的区块链应用。智能合约是一种计算机协议,能够以信息化方式传播、验证或执行合同,这些交易在没有可信第三方情况下执行、可追踪且不可逆转。但目前智能合约存在各种各样的安全和隐私保护问题,为用户带来严重的经济损失和困扰。文中分析了智能合约安全挑战与隐私威胁,整理了智能合约中安全与隐私保护关键技术,最后给出了智能合约未来的研究方向。

关键词: 区块链; 智能合约; 数据安全; 隐私保护

中图分类号: TP309.2 **文献标志码:** A **文章编号:** 1673-5439(2019)04-0063-09

Security and privacy-protection technologies in smart contract

WANG Huaqun, ZHANG Fan, LI Tian, GAO Mengjie, DU Xinyu

(School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

Abstract: The blockchain is a novel distributed infrastructure and computation paradigm. The ordered chain structure is used to store data and a consensus algorithm is used to modify data and cryptography to ensure the data security. As an important character, blockchain 2.0 introduces the smart contract. All kinds of applications can be developed by using smart contract. The smart contract is one type of computer protocol. It can disseminate, verify and execute contract through information technology. These irreversible transactions can be executed and traced without the trusted third party. The security challenge and privacy threats in the smart contract are analyzed. Then, the key security and privacy-protection technologies in smart contract are arranged. Finally, we give the future research directions of smart contract are pointed out.

Keywords: blockchain; smart contract; data security; privacy protection

第一代区块链技术是由中本聪提出的点对点电子现金比特币系统,采用 UTXO[1](Unspent Transaction Output)模型。尽管 UTXO 交易模型具有无状态性、可并发处理、隐私保护性强等优点,但是其非图灵完备性无法实现复杂的系统。2013 年,出现了采用账户模型部署智能合约的以太坊,扩展了区块链的应用范围和灵活性,被称为第二代区块链技术(区块链 2.0)。智能合约的概念是由密码学家 Nick Szabo 在 1994 年根据自动售货机的灵感引入:“一个智能合约是一套以数字形式定义的承诺,包括合约

收稿日期: 2019-07-02; 修回日期: 2019-07-06 本刊网址: http://nyzr.njupt.edu.cn
基金项目: 国家自然科学基金(61872192)、江苏省自然科学基金(BK20181394)、江苏省青蓝工程和南京邮电大学“1311”人才计划资助项目
作者简介: 王化群,男,博士,教授,博士生导师,whq@njupt.edu.cn
引用本文: 王化群,张帆,李甜,等. 智能合约中的安全与隐私保护技术[J]. 南京邮电大学学报(自然科学版),2019,39(4):63-71.

参与方可以在上面执行这些承诺的协议”[2]。然而智能合约一直没有找到适合的应用场景,直到区块链技术的出现。以太坊创始人 Vitalik Buterin 在以太坊白皮书中这样描述智能合约“智能合约不应被视为应履行或遵守的义务,它们更像是 EVM(Ethereum Virtual Machine)中的机器人,当收到外部条件(消息或交易)时就自动执行特定的代码并修改相关地址的余额或其它信息”[3]。以太坊中智能合约的部署流程如图 1 所示。基于智能合约的概念,产生了其他区块链平台,如小蚁(NEO)、EOS(Enterprise Operation System)和超级账本 Hyperledger 等。前两个属于公有链,后一个属于联盟链。在 Hyperledger Fabric 中,智能合约被称为链码(Chaincode)。

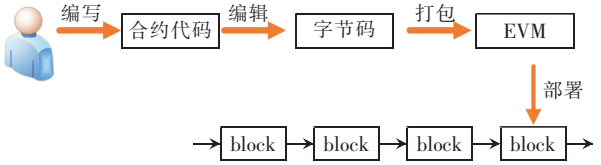

图 1 以太坊中智能合约的部署流程

智能合约由于其图灵完备性可以实现复杂的区块链应用,在物联网[4]、能源[5]、医疗健康[6]以及供应链[7]等领域都得到了广泛的应用。尽管人们对智能合约的发展保持乐观,但频频爆发的智能合约安全漏洞事件和隐私问题引发了人们的担忧。2016 年 6 月,DAO 安全漏洞导致5 000万美元的经济损失;2017 年 7 月,由智能合约引发的 Parity 多签名钱包安全漏洞,导致超过 1.82 亿美元的经济损失;2018 年 4 月,美链的代币 BEC 由于一行代码的安全漏洞导致 9 亿美元市值几乎归零;2019 年 5 月,币安遭遇黑客攻击导致7 000多比特币被盗。由此可以看出,智能合约传递了价值,但是一个安全漏洞可能会引发巨额的损失。与此同时,区块链面临身份信息泄漏、交易信息泄漏等各种风险,这些隐私问题严重阻碍了区块链技术生态圈的发展。比如银行,由于交易记录可能对所有参与者可见,如果不考虑安全性与隐私性,银行将不愿意在智能合约平台上进行交易。因此,智能合约中的安全与隐私保护是一个值得研究的课题。

目前,国内外研究工作者致力于开发能同时满足安全性、隐私保护性和可扩展性的区块链智能合约平台。在金融数字货币领域,尽管 Monero[8]、零币[9]和 Mimblewimble[10]等密码货币均在隐私保护方面做出了贡献,但都放弃了可编程性。2016 年,Kosba 等[11]首次提出了一种智能合约隐私保护系统 Hawk,既保证了合约安全又保证了链上隐私。2017 年,麻省理工学院的学者们开发了一项加密合约 Enigma 协议,应用在 Catalyst(一种加密货币交易投资平台)上。Oasis Labs 融资4 500万美元构建基于区块链的云计算平台,研究的 Ekiden 协议[12]旨在提高智能合约的可扩展性、安全性和隐私性,试图突破公有链的“不可能三角”问题。在基于账户模型的智能合约平台(如以太坊)上,Bünz 等[13]引入了 Zether 协议提供机密性和匿名性。2017 年,在账户模型下利用同态加密和零知识证明,中国科学院与武汉大学密码学专家们合作实现了保护余额与转账资产的分布式智能合约系统[14]。2019 年 3 月,斯坦福大学的 SERO(Super Zero)团队首创了可以运行智能合约的零知识证明加密系统,实现隐私保护,比现有的 ZCash 系统有 20 倍以上的性能提高,提高了实用性[15]。从大量的研究中可以看出,智能合约中的安全性和隐私保护技术已经成为区块链的研究热点。

1 智能合约中的安全挑战与隐私威胁

智能合约是用代码编写,存在于合约地址中,并由交易命令调用的。与传统的编程不同,智能合约比较灵活,任何矿工都可以在区块链上根据交易需求创建智能合约。基于区块链的不可篡改性,一旦智能合约被部署到区块链上,不可用传统的方法修改。另外,由于一个完整智能合约的编写涉及到隐私问题、法律问题、安全问题和机制设计等多个方面[16],因此设计一个没有安全漏洞、公平可信,符合规范和交易流程的智能合约不是一件容易的事情。

以太坊中的智能合约是目前为止使用频率最多的,且较为成熟的,我们就以以太坊中的智能合约为例,分析智能合约中常见的安全问题。

1.1 智能合约的安全挑战

在对以太坊中的智能合约的攻击分析中,根据引入级别,可将漏洞分为 3 类:Solidity、EVM 字节码和区块链[17]。

除了文献[17]描述的 12 种漏洞,智能合约中常见的漏洞还有短地址漏洞、拒绝服务漏洞和整数溢出漏洞[18],如表 1 所示。这 15 种漏洞又可以根据漏洞形成的原因大体分成调用时漏洞和固有属性漏洞。

表 1　智能合约中的 15 种安全漏洞

漏洞分类	漏洞
Solidity	调用未知状态
	无 gas 调用
	调用顺序异常
	类型不匹配
	可重入攻击
	秘密字段
	整数溢出
	拒绝服务
EVM	不可修改的漏洞
	代币传输丢失
	堆栈限制漏洞
	短地址漏洞
区块链	不可预测状态
	生成伪随机数
	时间戳依赖

1.1.1　调用时漏洞

智能合约在调用时会涉及到类型匹配、gas 限制、堆栈限制以及调用逻辑等问题,恶意攻击者能够在调用时利用代码或者逻辑上的漏洞,对合约进行攻击。

由于在以太坊上编译智能合约使用的 solidity 语言是强类型语言,不会进行类型检测,因此,在调用类型不匹配时不会抛出异常,所以,开发者不会察觉。调用未知状态和类型不匹配,这些漏洞都是由于调用类型不匹配造成的。

在以太坊中的智能合约中,gas 用于测量执行某些操作所需的计算量,越复杂的运算导致的 gas 消耗量越高。无 gas 调用计算是 gas 不足够执行任何回调函数,甚至不能执行空回调函数。利用 gas 限制漏洞,恶意攻击者能够通过限制合约调用对智能合约进行攻击。

拒绝服务漏洞,是合约因某种原因在一段时间内或者永远无法处理用户的请求所造成的。造成拒绝服务的主要原因有在编写中循环语句、递归函数、外部合约调用等处理不当以及对资源压制等。这些因素会导致达到 gas 上限、抛出异常和无法访问等问题,从而让合约无法按照正常的交易逻辑执行,或者让合约本身逻辑无法执行。

调用智能合约时调用逻辑也是决定安全调用的关键因素之一。在可重入攻击[19]中,恶意攻击者就是利用了调用逻辑的漏洞,在被调用合约上一次调用未结束之前,再次调用智能合约,如此一来智能合约会响应多次。在调用顺序异常漏洞中,智能合约拥有不同的行为,交易能否安全进行,也取决于合约之间能否按照需求相互调用。

1.1.2　固有属性漏洞

恶意攻击者会根据区块链的某种特性、智能合约的自身属性、EVM 固有的属性限制以及编译语言的特点,对运行的智能合约进行攻击。

在区块链中,新的区块的时间戳要比上一个区块的时间戳大,而且智能合约的执行是依赖于当前区块的时间戳,合约执行结果会随着时间戳的不同而改变,因此,恶意攻击者可以提前尝试不同的时间戳,控制智能合约的运行结果。

不可修改漏洞根源在于区块链的不可篡改性,智能合约一旦部署在区块链上,就没有直接的方法修补其中的漏洞,若重新部署合约,则之前合约中的数据将会丢失。

在智能合约的设计中,合约调用是有次数限制的,即堆栈限制。当一个合约调用另一个合约时,关联调用堆栈就会增加,当恶意攻击者对堆栈调用进行深度攻击时,会造成堆栈溢出,即堆栈限制漏洞。

在智能合约里还有一个很重要的问题——获取随机数。在智能合约里都不支持随机数的生成,因为如果合约里可以生成真正随机数,那么该合约执行结果就是完全随机的,其他节点无法直接验证该执行结果是否合法。所以,智能合约里的随机数生成器会使用已有的区块哈希生成伪随机数。而有的恶意攻击者利用智能合约的这一性质,控制伪随机数的生成,从而控制智能合约的运行结果。

对于短地址攻击,在智能合约交易中,没有对地址长度的验证过程,由于 EVM 会对交易中长度不足的地址根据 ABI (Application Binary Interface) 规范对其进行末尾补零,如果恶意攻击者故意将 2 个字节的交易额减少了两位,EVM 会补全为 4 个字节,那么恶意攻击者就可以获得原交易额的 256 倍。

对于整数溢出漏洞,因为 Solidity 语言支持从 uint8 到 uint256,对于 uint256 的变量,它的取值范围是 0 到 $2^{256}-1$。如果某个 uint256 变量的值已经为 $2^{256}-1$,那么这个变量再加 1 就会发生整数上溢,同时该变量的值变为 0,比如当数值 0 × FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF 加 1 的之和,该数值会整数上溢成 0 × 0000000000000000000000000000000000000000。当 uint256 变量超过最小值时会造成整数下溢出,比如对于某个 uint256 变量的值已经为 0,即 0 × 0000000000000000000000000000000000000000,当变量减 1 时,该变量会变成 0 ×

FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF,恶意攻击者可以用这种漏洞花费比他拥有的更多的代币。智能合约对数字非常敏感,所有整数溢出都非常危险,可能会导致合约失效、无限发币等风险。

虽然智能合约中的交易双方是匿名的,但是恶意攻击者可以根据公开的交易信息推断出交易双方等隐私信息,这就是秘密字段漏洞的原理。

1.2 智能合约中的隐私威胁

智能合约运行在区块链上,没有任何第三方进行信用背书,因此智能合约不同于传统合约的隐私保护。智能合约中的隐私威胁主要是攻击者对身份隐私和交易隐私的攻击,目前主要隐私威胁包括匿名性隐私威胁、访问控制隐私威胁和链上信息隐私威胁等。

匿名性隐私威胁:区块链上的信息是以分布式账本的形式存储在链上,任何一个节点都可以从链上获取完整的信息。虽然在区块链上的交易具备一定的匿名性,但随着技术的发展,匿名性已经不能完全保护个人身份隐私。对于链上的交易都是使用匿名地址,如果这些地址直接或间接地与链下生活发生了联系,就会失去交易的匿名性,从而泄露其个人隐私。另外,不同的地址之间如果有着稳定的关联交易,攻击者可以通过分析交易规律,甚至能够推测出用户的身份信息和位置信息。文献[20]在分析区块链上的匿名交易时发现,攻击者可以通过窃听并分析交易者的公共信息,查找具有近似金额的所有可能交易,就可以获取相应的交易地址。文献[21]通过分析交易图谱,推导出一些用户行为特征数据。

访问控制隐私威胁:访问控制技术是对用户权限进行管理,只有合法的用户使用合法的行为才有访问系统相应资源的权限。设置合适的访问控制权限,可以防止恶意节点窃取交易隐私数据。出于隐私安全的考虑,智能合约对整个结构中的数据访问进行控制[22]。如果恶意攻击者对访问权限进行攻击,会造成越权,从而造成数据的流失甚至财产的丢失。在 ERC223 智能合约中,由于访问控制权限被攻击,导致了越权发币。具体漏洞攻击如下:在 ERC223 智能合约中,恶意攻击者利用自定义回调函数回调 setOwner 方法,将合约的 owner 变成自己控制的地址,从而获得高级权限。

链上信息隐私威胁:传统的信息保护措施是通过对信息加密,防止攻击者获得有用的信息。但是在链上加密交易信息的过程中,一方面要保证不能让非交易者看到交易信息,另一方面需要验证交易的正确性,对交易内容不能完全加密。这两者本身存在矛盾,也是隐私保护技术上的挑战。

1.3 经典漏洞攻击事件分析

The DAO 攻击:The DAO 是基于区块链技术的全球最大众筹项目,于 2016 年 4 月 30 日开始,28 天融资筹得 1.5 亿美元。2016 年 6 月 18 日,黑客利用可重入漏洞对 The DAO 项目进行攻击,导致项目损失了价值5 000万美元的以太币。

在合约中,代码的逻辑是:如果用户不同意其他用户的投票,可以选择分裂出去(调用 splitDAO 函数),也就是用户反悔之后可以通过调用 splitDAO 函数退钱。

从 splitDAO 代码[23]中我们可以看到,用户提出分裂之后,合约先计算退还金额,然后退还金额(调用 withdrawRewardFor 函数),最后设置账户余额归零。这里,先退还金额(第 12 行),再设置账户余额为零(第 15 行),为攻击 the DAO 埋下了伏笔。

withdrawRewardFor 函数的源码如下:

```
(1) function withdrawRewardFor(address _account) noEther internal returns(bool _success) {
(2)     if((balanceOf(_account) * rewardAccount.accumulatedInput()) / totalSupply < paidOut[account])
(3)         throw;
(4)     uint reward = (balanceOf(_account) * rewardAccount.accumulatedInput()) / totalSupply - paidOut[account];
(5)     if(!rewardAccount.payOut(_account,reward)) //XXXXX vulnerable
(6)         throw;
(7)     paidOut[account] += reward;
(8)     return true;
(9) }
```

在第 6 行代码中,我们可以看到,在 withdrawRewardFor 中调用了 payOut 函数执行转账。payOut 函数如下:

```
(1) function payOut(address _recipient,uint _amount) returns(bool) {
(2)     if(msg.sender != owner || msg.value > 0 || (payOwnerOnly && _recipient != owner))
(3)         throw;
(4)     if(_recipient.call.value(_amount)()) { //XXXXX vulnerable
(5)         PayOut(_recipient,_amount);
(6)         return true;
(7)     } else {
(8)         return false;
(9)     }
(10) }
```

在 payOut 函数中,调用了 call 函数,而在 solidity 中返回结果为 true 的 call 函数的调用不一定会正

确调用,当在合约调用没有匹配到函数或者没有任何数据时,会自动调用 fallback 函数。当调用 call.value 时,会把所有的 gas 发送到合约地址上并执行 fallback 函数,fallback 函数有足够的 gas 执行任何操作。恶意攻击者正是利用了这一点,对 The DAO 进行了可重入攻击。

恶意攻击者提交 splitDAO 申请退还代币,The DAO 调用 splitDAO 函数,恶意攻击者在 withdrawRewardFor 操作和设置账户余额归零操作之间再次提交 splitDAO,由于上一次账户余额还未置零,所以第二次代币也会退还到恶意攻击者的账户中。如此重复申请 splitDAO,导致 The DAO 损失惨重。

目前,数量最多的分布式应用类型是符合 ERC-20 标准的代币(token) 合约。虽然大量的代币合约声称自己符合 ERC-20 标准,但是这些代码合约的程序实现是否真的与标准一致却无人知晓。Chen 等[24] 研究了 ERC-20 代币合约不一致行为,提出了检测不一致行为的新方法,这种方法对比了从 3 个渠道获悉的代币行为: 记录代币拥有者和余额的内部数据结构的修改,标准接口以及标准事件。如果从这 3 个渠道获悉的行为不匹配,则发现了不一致行为。作者还发现了导致不一致行为的 11 个原因,例如: 缺陷代币合约、标准方法缺失以及缺失标准事件等。另外,Chen 等[25] 利用图分析技术对区块链中的用户特征进行分类,并用交叉图技术研究了以太坊安全问题。在智能合约安全问题方面的研究成果还有: 文献 [26] 提出了一种基于区块链的公开、高效的证书认证机制; 文献 [27] 利用以太坊智能合约设计了一种分布式的支持隐私保护的搜索方案,数据所有者能够可靠地接收正确的搜索结果; 文献 [28] 建立了随机网络模型对区块链系统的安全性进行测试和验证,研究了系统的稳态和瞬态性能特征等。

2　智能合约中安全与隐私保护关键技术

上文提出智能合约漏洞分为调用漏洞和固有属性漏洞。调用漏洞主要是由代码漏洞和逻辑漏洞造成的,对于这种安全漏洞,文献 [29 – 35] 提出的自动化安全审计可以在智能合约编写时进行有效地避免,固有属性漏洞也有对应的解决方案。

2.1　调用漏洞的自动化安全审计

针对调用漏洞问题,理想化的方法是将海量智能合约部署到以太坊网络中,在未使用前进行安全审计,以便发现错误,找出漏洞并规避风险。审计可以分为人工审计和自动化审计,显然,人工审计不切实际,耗时耗力且效率低下,不符合当前海量智能合约日益增长的现状。自动化审计可以分为 3 类: 特征代码的匹配、基于形式化验证的自动化审计以及基于符号执行和符号抽象的自动化审计[29] 。

2.1.1　特征代码的匹配

特征代码匹配是对恶意代码进行分析,提取特征,抽象形成语义匹配模块,然后对要检测的源码进行检测。特征代码匹配的优点是速度快、响应迅速,缺点则是使用范围有限、漏报率高。

2.1.2　基于形式化验证的自动化审计

智能合约的形式化验证是指在智能合约的生命周期内用数学手段以及智能化分析工具对合约进行建模、推导与证明,以验证智能合约是否满足一致性、无二异性、可观察性、可验证性和接入控制等关键特征[30] 。2016 年,Hirai 首次提出基于 Isabelle 高阶逻辑交互式定理证明器,判断逻辑代码是否有漏洞,之后考虑到 Lem 语言转化效率低,于是改进方案,采用 F*　framework 和 K framework,将 EVM 转化为一个形式化模型。Bhargavan 等[31] 提出了一种采用形式化验证、针对以太坊 Solidity 合约功能正确性的验证框架,该框架先将 Solidity 语言和 EVM 字节码转换为 F* 语言,然后验证代码的各种属性,确保排除逻辑漏洞和安全漏洞。但是,这种方法的自动化程度较低,需要人工进行二次检验。

2.1.3　基于符号执行和符号抽象的自动化审计

这种方法先对智能合约进行分析,然后通过编译源码形成 EVM Opcode,再将 Opcode 输入到自动化分析引擎,自动化分析引擎将一些共有特性转化为控制流图,最后利用符号执行和符号抽象两种方法进行分析。基于符号执行和符号抽象的自动化审计比较典型的有 Oyente[32] (符号执行验证) 和 Securify[33] (符号抽象分析) 系统,可以显著降低特征代码匹配的误报率和漏报率,但缺点是分析方法繁琐耗时,并且合约的功能正确性无法验证,可检测的安全漏洞有限,而且可能导致错误的警报。Zhou 等[34] 提出了一种智能合约的安全保障解决方案,方案不仅能够生成跨文件智能合约调用关系的拓扑图,还扩展了时间戳风险、Tx. origin 风险、Zero Division 风险,并且可以通过符号执行和语法分析,检测和定位逻辑风险。类似的智能合约验证工具还有 ZEUS[35] 、Mythril、Solgraph 等。目前这些验证工具大多停留在试验阶段,还未在真实系统中接受检验

以证明其可靠性。因此开发完备的、规范的、可靠的形式化验证框架极具现实意义,形式化验证将成为未来智能合约的重要发展方向。

2.2 固有属性漏洞解决方案

固有属性漏洞的一般性解决措施如表 2 所示。例如,短地址攻击是由于以太坊 EVM 没有严格校验地址的位数,且具有自动补全位数的特性,所以为了消除这种漏洞,只能在交易提交的时候检查用户输入地址数是否正确;在发送交易之前,检查函数参数位数是否正确;并且对 transfer 函数校验 len(msg.data) == 68。再有,为了维护新旧合约,采取对智能合约升级的方法,从合约代码层将业务逻辑和数据分离,把合约分为两类:逻辑合约以及数据合约,实现智能合约中逻辑可插拔、数据可迁移。数据合约为逻辑合约提供数据接口,同时逻辑合约为数据合约提供数据处理。智能合约升级后,通过引导用户使用新的逻辑合约,并更新数据合约的权限允许新的逻辑合约调用。具体的改造要和实际需求相结合,针对逻辑合约与数据合约的操作关系进行改造。

表 2 固有属性漏洞解决措施

固有属性漏洞	解决措施
时间戳依赖生成伪随机数	采用第三方服务获取
不可修改漏洞	升级智能合约
堆栈溢出	在调用外部函数时确保已经完成内部操作
短地址攻击 整数溢出	对输入参数进行校验
秘密字段漏洞	采用密码学算法来保护

2.3 隐私保护关键技术

针对区块链数据全部上链并且公开透明的隐私保护问题,目前已经使用的隐私保护技术有混币器、环签名、同态加密、零知识证明以及安全多方计算[36]等。混币器可以将一组币输入与其他币进行混合输出,除了混币器自身,任何人都不知道输入币与输出币的对应关系。所以,一组币经过混币器后价值不变,但是匿名性增加,可以一定程度上避免追踪。达世币(Dash)就采用了混币器技术。门罗币[8](Monero)相较于比特币,具有隐私性更高、不可追踪、匿名性的特点,原因在于门罗币使用了环签名和隐匿地址的技术。环签名是一种特殊的群签名,没有中心管理者,环成员利用自己的私钥和其它成员的公钥进行签名,验证者只知道签名者来自这个环却不知道真实的签名者,环签名满足无条件匿名性和不可伪造性。门罗币使用环签名这一性质实现隐藏交易发送方地址信息的功能,使外部攻击者无法建立地址之间的关联性。同时隐匿地址确保了交易的匿名性,RingCT 环形保密交易技术(RCT)隐藏了交易金额。同态加密可应用在区块链上,实现密文计算。区块链上应用最广泛的零知识证明是简洁化的非交互式零知识证明(zk-SNARKs)[37]。零知识证明允许证明方向验证方在不暴露除声明有效性之外的任何信息前提下证明声明是真的,非交互式指的是证明仅包含从证明方发送给验证方的单个消息,即证明方与验证方没有双向通信。零币[9](Zcash)就利用了 zk-SNARKs 技术实现交易信息的完全隐藏,包括交易账号和交易金额。

以上是区块链的隐私保护技术,而对于智能合约的隐私保护,更多的是在区块链隐私保护的基础上进行改进,将多种密码技术结合。保护匿名性大多通过密码学技术,如零知识证明等实现。Hyperledger Fabric 在访问控制方面做出了优化,细化了智能合约节点的权限粒度,不同的用户对智能合约具有部署、查询、执行等权限;加密链上信息可通过可信硬件,如 SGX(SoftwareGuard Extensions)中的计算节点执行智能合约的计算,确保原始数据及计算过程的保密性和安全性。

2.3.1 零知识证明

从增强匿名性出发,Origo 等[38]提出了一个隐私保护应用平台 Origo,通过将自己设计的隐私协议加入零知识证明框架,为以太坊智能合约添加匿名保护,搭建一个支持 Dapp 数据隐私保护和私密交易的平台。Origo 协议通过初始化、提交、执行和结束 4 个步骤将用户的信息、交易金额和合约执行细节进行保密,但是其他用户仍然可以验证这笔合约是否被正确执行。此外,系统会先要求冷冻参与方和执行方的代币,以防参与方发生恶意行为和执行方在合约结束后公布参与方信息泄漏隐私,如若发生了恶意行为,冷冻的代币将被没收。Kosba 等[11]提出了一个用于构建保护用户隐私的智能合约框架 Hawk 框架。在 Hawk 中,专业的匿名程序员编写 Hawk 程序,无需实施任何加密。Hawk 智能合约分为私有合约和公共合约,私有合约存储用户私人数据和财务流动信息,通过向区块链发送加密信息,并依靠零知识证明确保合约正确执行,实现资金保护。以上操作实现了链上隐私并确保私有数据隐藏于公共的视野之外。此外,Hawk 还保证合约参与方的公平性,任何恶意终止协议的参与方都会受到财务

处罚。

2.3.2　可信执行环境

以上解决方案使用范围有限,仅限于一些简单的智能合约场景,当涉及到安全多方计算等复杂密码学问题时,可信执行环境 TEE(Trusted Execution Environment) 可提供通用的高性能解决方案。例如,英特尔公司与区块链初创公司 Enigma 合作开发了一种增强智能合约隐私性的区块链协议。该协议使用安全多方计算,实现匿名智能合约,允许节点使用加密的智能合约碎片进行计算,而无需解密。其中,加密计算在 TEE 中进行并与区块链其他部分隔离开。Zhang 等[39] 提出了一种可信数据输入系统 TC(Town Crier) ,TC 将以太坊的智能合约前端和基于 SGX 的可信硬件后端结合起来实现: (1) 在没有可信赖的服务运营商的情况下,向智能合约提供经过身份验证的数据; (2) 支持私有和自定义数据请求,用 TC 公钥加密请求和安全使用离链数据源,而区块链中的其他用户无法查看请求内容,从而保证智能合约的用户隐私。可信执行环境为智能合约数据提供了区块链所不具有的机密性,而通过链下可信计算环境,TEE 也解决了智能合约和区块链无法应对的复杂计算场景问题。

2.3.3　Zether 协议

Zether[13] 是一种分散式、保密的针对以太坊智能合约平台的隐私协议,由 Bünz 等提出,能够在多种账户模型上实现匿名支付。Zether 智能合约不仅能够保密用户账户的余额,还能够隐藏智能合约交易金额。Zether 拥有自己的代币 ZTH,这些代币可以在用户交易中流通。为了让 Zether 变得更有效,研究者还提供了新的隐私算法——新的零知识证明机制,使得 Bulletproofs[40] 与 Sigma 协议更具互操作性。此外,Bünz 展示了 Zether 可构建的 4 种应用[13] ,Zether 易于实现的特点使其更具现实意义。近日,Quorum 团队对 Zether 协议进行了巧妙扩展,称为匿名 Zether[13] ,除了交易本身的细节之外,还允许混淆交易中各方的身份,隐藏参与者的身份。Zether 的功能与 Quorum 堆栈中的访问控制特性相结合,可以提供更加健壮的安全服务。Zether 技术还有很多不足,比如成本高昂、以太坊的 gas 机制可能会导致隐私泄露等,还需要进一步研究。

3　未来研究方向

本文阐述了当前有关智能合约的安全机制以及隐私保护机制,但是大部分机制还处于研究讨论阶段,仍有很多的不足。所以未来可以在以下 3 个方向开展研究。

(1) 对智能合约本身进行形式化验证。智能合约一旦上链就不可修改,而本身可能出现的代码漏洞问题对其安全有着很大的隐患。对智能合约进行形式化验证能有效避免常见的安全漏洞,但目前诸如 Oyente[32] 等一些验证工具只能验证一部分漏洞,又或是仍处于试验阶段,所以仍需要研究更加完善的形式化验证框架[16] 。

(2) 结合可信硬件对智能合约用户及数据进行安全性保证和隐私保护。大部分研究方案仅采用了传统密码学技术。例如,用同态加密算法对数据进行加密,实现密文计算; 用环签名实现交易方无条件匿名。这些密码学算法目前来讲相对安全,但随着数学、密码学和计算技术的发展,尤其是人工智能和量子计算的兴起,这些算法面临着被破解的可能性。再加上在现实世界里,利用密码学技术完成安全隐私的代价较大,很多实际应用并不会采用安全系数高的方案,因此与可信硬件结合往往是更加实际的解决方案[41] 。

(3) 构造更加友好的智能合约编译器。当前智能合约只能支持简单的逻辑处理,并且操作起来比较复杂。对于非密码学专家来说,人为犯错的可能性就更大,若是能构造更加友好的智能合约编译器,对智能合约的安全也有很大的保障。

4　结束语

区块链 2.0 提供了图灵完备的智能合约,使得区块链技术变成了一个分布式应用的底层环境。目前,以太坊系统上已经部署了 60 万个智能合约,超过5 000万个以太坊账户。随着智能合约迅速发展,其安全和隐私保护问题遇到了挑战,对该问题深入研究能够促进智能合约进一步发展。通过深入分析与比较,智能合约安全和隐私保护在国内外已取得较多成果,但仍有很多问题尚待探讨,需要不断提高智能合约安全和隐私保护技术。本文重点介绍了智能合约中各类已知安全威胁和隐私保护风险,并分析了这些问题产生的原因和特点,给出了智能合约中安全与隐私保护的关键技术。基于最新的研究成果,展望了智能合约安全方面未来的研究方向,以期能够为智能合约安全和隐私保护的未来研究做出一些有益的探索。

参考文献:

[1] NAKAMOTO S. Bitcoin: a peer-to-peer electronic cash sys-

tem [EB/OL]. [2019-06-03]. https://bitcoin.org/bitcoin.pdf.

[2] SZABO N. Smart contracts: formalizing and securing relationships on public networks [EB/OL]. [2019-06-03]. http://firstmonday.org/ojs/index.php/fm/article/view/548/469.

[3] BUTERIN V. A next-generation smart contract and decentralized application platform [EB/OL]. [2019-06-03]. https://cryptorating.eu/whitepapers/Ethereum/Ethereum_white_paper.pdf.

[4] CHRISTIDIS K, DEVETSIKIOTIS M. Blockchains and smart contracts for the Internet of Things [J]. IEEE Access, 2016, 4: 2292 - 2303.

[5] POP C, CIOARA T, ANTAL M, et al. Blockchain based decentralized management of demand response programs in smart energy grids [J]. Sensors, 2018, 18(1): 162.

[6] GRIGGS K, OSSIPOVA O, KOHLIOS C, et al. Healthcare blockchain system using smart contracts for secure automated remote patient monitoring [J]. Journal of Medical Systems, 2018, 42(7): 130.

[7] KIM H, LASKOWSKI M. Towards an ontology-driven blockchain design for supply chain provenance [J]. Intelligent Systems in Accounting, Finance and Management, 2018, 25(1): 18 - 27.

[8] SABERHAGEN N. Cryptonote V2.0 [EB/OL]. [2019-06-03]. http://cryptonote.org/whitepaper.pdf.

[9] MIERS I, GARMAN C, GREEN M, et al. ZeroCoin: anonymous distributed E-cash from bitcoin [C] // IEEE Symposium on Security & Privacy. 2013: 397 - 411.

[10] FUCHSBAUER G, MICHELE O, YANNICK S. Aggregate cash systems: a cryptographic investigation of mimblewimble [C] // EUROCRYPT. 2019: 657 - 689.

[11] KOSBA A, MILLER A, SHI E, et al. Hawk: the blockchain model of cryptography and privacy-preserving smart contracts [C] // IEEE Symposium on Security and Privacy. 2016: 839 - 858.

[12] CHENG R, ZHANG F, KOS J, et al. Ekiden: a platform for confidentiality-preserving, trustworthy, and performant smart contracts [EB/OL]. [2019-06-03]. https://arxiv.org/abs/1804.05141v4.

[13] BÜNZ B, AGRAWAL S, ZAMANI M, et al. Zether: towards privacy in a smart contract world [EB/OL]. [2019-06-03]. https://eprint.iacr.org/2019/191.pdf.

[14] MA S, DENG Y, HE D, et al. An efficient NIZK scheme for privacy-preserving transactions over account-model blockchain [EB/OL]. [2019-06-03]. https://eprint.iacr.org/2017/1239.

[15] Super Zero(SERO). Technical white paper [EB/OL]. [2018-09-10]. http://sero-media.s3-website-ap-southeast-1.amazonaws.com/Sero_ENG_V1.06.pdf.

[16] 欧阳丽炜,王帅,袁勇,等. 智能合约:架构及进展 [J]. 自动化学报,2019,45(3):445 - 457.
OUYANG Liwei, WANG Shuai, YUAN Yong, et al. Smart contracts: architecture and research progresses [J]. Acta Automatica Sinica, 2019, 45(3): 445 - 457. (in Chinese)

[17] ATZEI N, BARTOLETTI M, CIMOLI T. A survey of attacks on ethereum smart contracts (SOK) [C] // International Conference on Principles of Security and Trust. 2017: 164 - 186.

[18] 邱欣欣,马兆丰,徐明昆. 以太坊智能合约安全漏洞分析及对策 [J]. 信息安全与通信保密,2019(2):10.
QIU Xinxin, MA Zhaofeng, XU Mingkun. Ethereum smart contract security vulnerability scenario analysis [J]. Information Security and Communications Privacy, 2019(2): 10. (in Chinese)

[19] LIU C, LIU H, CAO Z, et al. ReGuard: finding reentrancy bugs in smart contracts [C] // ICSE. 2018: 65 - 68.

[20] FLEDER M, KESTER M, PILLAI S. Bitcoin transaction graph analysis [EB/OL]. [2019-06-03]. https://arxiv.org/pdf/1502.01657.pdf.

[21] RON D, SHAMIR A. Quantitative analysis of the full bitcoin transaction graph [C] // International Conference on Financial Cryptography and Data Security. 2013: 6 - 24.

[22] AZARIA A, EKBLAW A, VIEIRA T, et al. Medrec: using blockchain for medical data access and permission management [C] // International Conference on Open and Big Data. 2016: 25 - 30.

[23] Figo. 智能合约安全事故回顾 [EB/OL]. [2019-06-03]. http://blockgeek.com/t/topic/1929.
Figo. Review of security incident in smart contract [EB/OL]. [2019-06-03]. http://blockgeek.com/t/topic/1929. (in Chinese)

[24] CHEN T, ZHANG Y F, LI Z H, et al. TokenScope: automatically discovering inconsistent cryptocurrency tokens [C] // ACM Conference on Computer and Communications Security. 2019.

[25] CHEN T, ZHU Y X, LI Z H, et al. Understanding ethereum via graph analysis [C] // INFOCOM. 2018: 1484 - 1492.

[26] CHEN J, YAO S, YUAN Q, et al. CertChain: public and efficient certificate audit based on blockchain for TLS connections [C] // INFOCOM. 2018: 2060 - 2068.

[27] HU S, CAI C, WANG Q, et al. Searching an encrypted cloud meets blockchain: a decentralized, reliable and fair realization [C] // INFOCOM. 2018: 792 - 800.

[28] PAPADIS N, BORST S, WALID A, et al. Stochastic models and wide-area network measurements for blockchain design and analysis [C] // INFOCOM. 2018: 2546 - 2554.

[29] 杨文玉. 智能合约自动化安全检测技术浅析 [EB/OL]. [2019-06-03]. https://bbs.csdn.net/topics/392445825.
YANG Wenyu. Analysis on the intelligent detection technology of smart contracts [EB/OL]. [2019-06-03]. https:

//bbs. csdn. net/topics/392445825. (in Chinese)

[30] 胡凯,白晓敏,高灵超,等. 智能合约的形式化验证方法[J]. 信息安全研究,2016,2(12):1080 - 1089.
HU Kai,BAI Xiaomin,GAO Lingchao,et al. Formal verification method for smart contracts[J]. Journal of Information Security Research,2016,2(12):1080 - 1089. (in Chinese)

[31] BHARGAVAN K,DELIGNAT-LAVAUD A,FOURNET C,et al. Formal verification of smart contracts[C]//PLAS. 2016:91 - 96.

[32] LUU L,CHU D,OLICKEL H,et al. Making smart contracts smarter[C]//ACM SIGSAC Conference on Computer Communications Security. 2016:254 - 269.

[33] PETAR T,ANDREI D,DANA D,et al. Securify: practical security analysis of smart contracts[C]//ACM SIGSAC Conference on Computer Communications Security. 2018:67 - 82.

[34] ZHOU E,HUA S,PI B,et al. Security assurance for smart contract[C]//International Conference on New Technologies,Mobility and Security. 2018:1 - 5.

[35] KALRA S,GOEL S,DHAWAN M,et al. ZEUS: analyzing safety of smart contracts[C]//Annual Network and Distributed System Security Symposium. 2018:1 - 15.

[36] ANDRYCHOWICZ M,DZIEMBOWSKI S,MALINOWSKI D,et al. Secure multiparty computations on bitcoin[C]//IEEE Symposium on Security and Privacy. 2014:443 - 458.

[37] SUN S,AU M,LIU J,et al. RingCT 2.0: a compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency[C]//European Symposium on Research in Computer Security. 2017:456 - 474.

[38] ORIGO F. Privacy preserving platform for decentralized applications[EB/OL]. [2019-06-03]. https://origo.network/whitepaper/.

[39] ZHANG F,CECCHETTI E,CROMAN K,et al. Town crier: an authenticated data feed for smart contracts[C]//ACM SIGSAC Conference on Computer Communications Security. 2016:270 - 282.

[40] BÜNZ B,BOOTLE J,BONEH D,et al. Bulletproofs: efficient range proofs for confidential transactions[C]//IEEE Symposium on Security and Privacy. 2018:315 - 334.

[41] DUAN H,ZHENG Y,DU Y,et al. Aggregating crowd wisdom via blockchain: a private,correct,and robust realization[C]//IEEE PerCom. 2019:43 - 52.