doi :10.3969/j.issn.1671-1122.2018.11.002

区块链中的智能合约

马春光[1]，安婧[1]，毕伟[2]，袁琪[3]
（1. 哈尔滨工程大学计算机科学与技术学院，黑龙江哈尔滨 150000；2. 中思博安科技（北京）有限公司，北京 100088；
3. 齐齐哈尔大学通信与电子工程学院，黑龙江齐齐哈尔 161006）

摘　要：智能合约是部署在区块链上的可执行代码，可不依赖中心机构自动化地代表各签署方执行合约。因其具有强制执行性、防篡改性和可验证性等特点，可以应用到很多场景中。过去几年中，已经出现很多可以部署智能合约的区块链平台，其中一些已经在实际中实施和使用。文章首先对智能合约的定义和性质进行描述；然后分析各个区块链平台中的智能合约，并对应用最广泛的比特币、以太坊和超级账本等区块链系统中的智能合约进行重点阐述；最后指出智能合约存在的问题和解决方法。

关键字：区块链；智能合约；比特币；以太坊；超级账本

中图分类号：TP309　**文献标识码：**A　**文章编号：**1671-1122（2018）11-0008-10

中文引用格式：马春光，安婧，毕伟，等. 区块链中的智能合约 [J]. 信息网络安全，2018，18（11）：8-17.

英文引用格式：MA Chunguang, AN Jing, BI Wei, et al. Smart Contract in Blockchain[J]. Netinfo Security, 2018, 18(11): 8-17.

Smart Contract in Blockchain

MA Chunguang[1], AN Jing[1], BI Wei[2], YUAN Qi[3]
(1. *School of Computer Science and Technology*, *Harbin Engineering University*, *Harbin Heilongjiang* 150000, *China*; 2. *Zsbatech Corporation*, *Beijing* 100088, *China*; 3. *College of Communication and Electronic Engineering*, *Qiqihar University*, *Qiqihar Heilongjiang* 161006, *China*)

Abstract: Smart contracts are executable code that is deployed on the blockchain and can be executed on behalf of the signatories automatically, independent of the central authority. Due to its enforceability, tamper resistance and verifiability, it can be applied to many scenarios. In the past few years, there have been many blockchain platforms where smart contracts can be deployed, some of which have actually been implemented and used. This paper first describes the definition and nature of smart contract, then analyzes the smart contract in each blockchain platform, and the intelligence in the most widely used blockchain systems of Bitcoin, Ethereum and Hyperledger. The contract was highlighted. Finally, the problems and solutions in the smart contract are pointed out.

Key words: blockchain; smart contract; Bitcoin; Ethereum; Hyperledger

收稿日期：2018-7-5
基金项目：国家自然科学基金 [61472097]
作者简介：马春光（1974—），男，黑龙江，教授，博士，主要研究方向为密码学、数据安全与隐私、云计算安全、区块链等；安婧（1994—），女，黑龙江，硕士研究生，主要研究方向为区块链；毕伟（1980—），男，黑龙江，博士，主要研究方向为机器学习与数据分析、区块链、密码学、可信协同计算等；袁琪（1973—），女，黑龙江，副教授，博士，主要研究方向为信息安全、区块链、博弈论。
通信作者：马春光 machunguang@hrbeu.edu.cn

0 引言

区块链是随着比特币等数字货币的日益普及而逐渐兴起的一种技术。近年来，区块链已经逐渐成为独立于比特币的一个平台架构，其核心理念是将区块链作为一个可编程的分布式信用基础设施，支持智能合约应用，与过去比特币区块链作为一个虚拟货币支撑平台区别开来。

智能合约是部署在区块链上的可执行代码，可不依赖中心机构自动化地代表各签署方执行合约。在金融区块链中，智能合约可以被认为是一种系统，一旦预先定义的规则得到满足，它就向所有或部分相关方发布数字资产 [1]。更广义地讲，智能合约是用编程语言编码的一组规则，一旦满足这些规则的事件发生，就会触发智能合约中事先预设好的一系列操作，而不需要可信第三方参与。这一性质使得智能合约有着广泛的应用。目前已有不同的区块链平台可以用来开发智能合约。如图 1 所示，2012 年 1 月至 2018 年 3 月，GitHub.com 上与区块链和智能合约有关的项目数分别达到 19204 个和 5304 个。与此同时，一些信息通信技术公司和国家政府已经开始关注区块链和智能合约 [2]，大部分国家政府对推动区块链技术的发展也持积极态度。

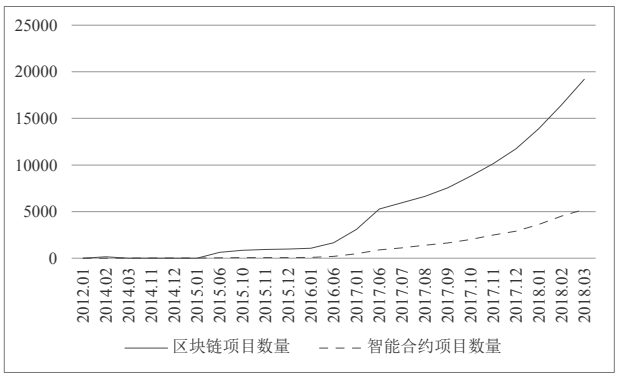

图 1 GitHub.com 上与区块链和智能合约有关的项目数量

1 区块链概述

区块链起源于化名为中本聪（Satoshi Nakamoto）的学者在 2008 年发表的奠基性论文“比特币：一种点对点电子现金系统”[3]。狭义来讲，区块链是一种按照时间顺序将数据区块以顺序相连的方式组合成一种链式数据结构，并以密码学方式保证不可篡改和不可伪造的分布式账本。广义来讲，区块链是利用块链式数据结构来验证与存储数据，利用分布式节点共识算法来生成和更新数据，利用密码学方式保证数据传输和访问的安全，利用由自动化脚本代码组成的智能合约来编程和操作数据的一种全新的分布式基础架构与计算范式 [4]，具有去中心化、时序数据、集体维护、准匿名性、安全可信、可编程性等特点。

区块链根据应用场景和设计体系不同分为公有链（Public Blockchain）、联盟链（Consortium Blockchain）和私有链（Private Blockchain）[5]。公有链的各个节点可以自由加入和退出网络，并参加链上数据的读写，运行时以扁平的拓扑结构互联互通，网络中不存在任何中心化的服务端节点。联盟链的各个节点通常有与之对应的实体机构组织，通过授权后才能加入与退出网络。各机构组织组成利益相关的联盟，共同维护区块链的健康运转。私有链各个节点的写入权限收归内部控制，而读取权限可视需求选择性地对外开放。私有链仍然具备区块链多节点运行的通用结构，适用于特定机构的内部数据管理与审计。

SWAN[6] 按照应用范围和发展阶段将区块链应用分为 1.0、2.0 和 3.0 三个阶段。区块链 1.0 支撑虚拟货币应用，一般用于支付，其典型特征有：链状数据块结构、全网共享账本、非对称加密、源代码开源。数字货币的典型应用是比特币。比特币是第一个解决双重支付问题的去中心化虚拟货币。但是由于比特币的编程能力有限，无法支持复杂的交易，因此比特币不支持在其上创建复杂的分布式应用程序（Decentralized Application，DAPP）。区块链 2.0 支撑智能合约应用，并考虑使用区块链将其他类型的资产转移到货币之外，典型特征有：智能合约、分布式应用、虚拟机。其核心理念是把区块链作为一个可编程的分布式信用基础设施，支持智能合约应用。当前区块链应用处于 1.0

和 2.0 阶段。区块链 2.0 技术架构 [4] 如图 2 所示。

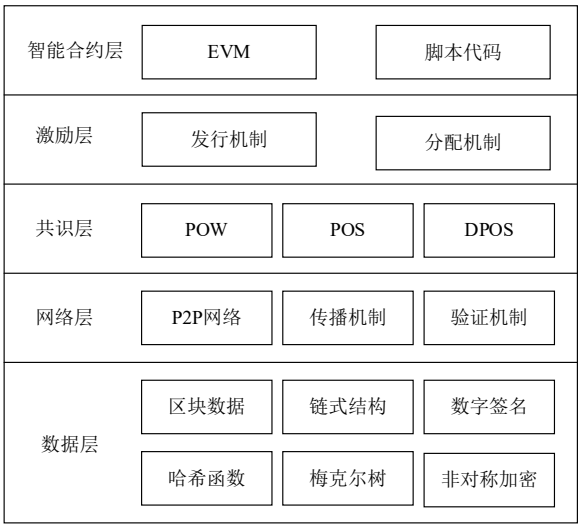

图 2 区块链 2.0 技术架构

SWAN[6] 将超越货币、金融范围内的区块链应用归为区块链 3.0。未来随着人工智能、物联网和量子技术的发展，区块链会有越来越广泛的应用场景。

2 智能合约概述

2.1 智能合约的定义和性质

1994 年，SZABO[7] 首次提出了智能合约的概念：智能合约就是执行合约条款的可计算交易协议。同时 SZABO[7] 还给出了智能合约应具有的性质：可见性、强制执行性、可验证性、隐私性。1997 年，SZABO[8] 将智能合约定义为一套以数字形式定义的承诺（Promises），包括合约参与方可以在上面执行这些承诺的协议。承诺包括用于执行业务逻辑的合约条款和基于规则的操作，这些承诺定义了合约的本质和目的。数字形式意味着合约由代码组成，其输出可以预测并可以自动执行。协议是参与方必须遵守的一系列规则。

2008 年比特币出现之后，人们意识到比特币的底层技术区块链可以为智能合约提供可信的执行环境，使得智能合约技术重新受到了关注并飞速发展。智能合约是区块链的核心构成要素，是由事件驱动的、具有状态的、运行在可复制的共享区块链数据账本上的计算机程序，能够实现主动或被动的数据处理功能，具有接受、存储和发送价值，以及控制和管理各类链上智能资产等功能 [9]。2016 年 10 月工信部发布的《中国区块链技术和应用发展白皮书》[4] 将智能合约视为一段部署在区块链上可自动运行的程序，涵盖范围包括编程语言、编译器、虚拟机、时间、状态机、容错机制等。

STARK[10] 将智能合约所有定义分为两类：智能合约代码（Smart Contract Code）和智能法律合约（Smart Legal Contract）。智能合约代码指在区块链中存储、验证和执行的代码。由于这些代码运行在区块链上，因此也具有区块链的一些特性，如不可篡改性和去中心化。该程序本身也可以控制区块链资产，即可以存储和传输数字货币。智能法律合约更像是智能合约代码的一种特例，是使用区块链技术补充或替代现有法律合同的一种方式，也可以说是智能合约代码和传统的法律语言的结合。本文主要关注智能合约代码。

在区块链系统上运行的智能合约如图 3 所示。智能合约运行后自动产生智能合约账户，智能合约账户包括账户余额、存储等内容，存储在区块链中 [11]。区块链中各个节点在虚拟机或者 Docker 容器中执行合约代码（也可称作调用智能合约），就执行结果达成共识，并相应地更新区块链上智能合约的状态。智能合约可以基于其收到的交易读 / 写用户私人存储，将钱存入其账户余额，可以发送 / 接收消息或来自用户 / 其他智能合约的数字资产，甚至创建新的智能合约。

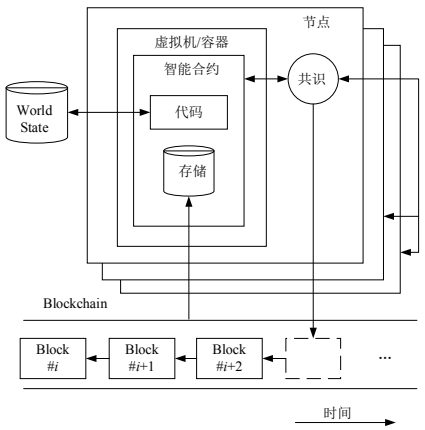

图 3 区块链系统上运行的智能合约

2.2 智能合约的应用

由于智能合约拥有较低的运行成本，较低的人为干预风险，并且能准确执行等特性，现已被应用到很多领域，如交易与公平交换、身份管理、物联网、医疗记录隐私、众筹等。

1）交易与公平交换。BOGNER[12] 等人在以太坊区块链上开发了一个智能合约应用程序，允许不受信任的参与者共享日常物品（如租赁设备）。PARK[13] 等人建议实施基于区块链的能源交易平台，以实现供应商之间高效的电能交易。

2）身份管理。AL-BASSAM[14] 等人在以太坊区块链上建立了一个名为 SCPKI 的系统，以克服公钥基础设施的局限性（如缺乏透明度）。该系统允许实体以透明的方式管理他们的身份，不需要可信第三方参与。YASIN[15] 等人提出一个系统框架来汇总在线身份和信用信息，为个人在线行为评分提供方法。

3）物联网。HUH[16] 等人使用以太坊智能合约来定义和管理特定条件下一些设备的行为。HUCKLE[17] 等人讨论了如何利用物联网和区块链创建安全的共享经济分布式应用程序。ZHANG[18] 等人基于区块链和智能合约实现了智能财产和物联网支付交易。

4）医疗记录隐私。XIA[19] 等人提出了 MeDShare 系统，该系统采用智能合约和访问控制机制来有效地跟踪数据的行为，解决了医疗大数据保管人在无信任环境中共享医疗数据的问题。BENCHOUFI[20] 等人探讨了将区块链和智能合约应用到临床试验的核心功能与潜在影响。CUNNINGHAM[21] 等人使用以太坊智能合约设计了一个系统记录患者的电子健康记录。

5）众筹。JACYNYCZ[22] 等人提出了 Betfunding，基于区块链的分散式众筹平台解决了目前众筹平台公信力不足、无法监管等问题。黄洁华 [23] 等人将众筹项目的规则制定成智能合约代码，将代码与项目状态全部记录在众筹区块链（Crowdfunding Private Blockchain，CPBC）上，并由区块链执行交易代码，保证了众筹项目的真实性、可靠性与强制性，使得项目的执行具有可信性，提升了众筹平台的公信力，还可减少众筹业务在第三方审核上的人工花费与计算成本。

3 典型区块链平台中的智能合约

比特币区块链架构支持数字货币的实现，虽然具有一定的灵活性，但用来支撑除了数字货币之外的应用场景还显得十分局限。区块链 2.0 领域泛指比特币 2.0、比特币 2.0 协议、智能合约、智能财产、分布式应用程序、去中心化的自治组织（Decentralized Autonomous Organizations，DAOs）和去中心化的自治公司（Decentralized Autonomous Corporations，DACs）[5]。智能合约可以开发并部署在不同的区块链平台上，不同的平台为开发智能合约提供了独特的功能。表 1 列出了典型的区块链平台的准入机制、数据模型、共识算法、智能合约执行环境和智能合约语言。

表 1 区块链平台对比

	准入机制	数据模型	共识算法	智能合约执行环境	智能合约语言
Bitcoin	公有链	基于交易	PoW	脚本引擎	基于栈的脚本
Ethereum	公有链	基于账户	Pow, PoS	EVM	Solidity, Serpent, LLL
CounterParty	公有链	基于交易	--	脚本引擎	Solidity, Serpent, LLL
Quorum	公有链	基于账户	Raft	EVM	GoLang
Monax	公有链	基于账户	Tendermint	EVM	Solidity
Fabric v0.6.0	联盟链	基于键值	PBFT	Docker 容器	Go, Java
Fabric v1.0.0	联盟链	基于键值	用户定制的	Docker 容器	Go, Java
Sawtooth	联盟链	基于键值	PoET	本地执行	Python
Tbaas	联盟链	基于账户	用户定制的	Docker 容器	JavaScript
BigchainDB	联盟链	基于交易	Quorum Voting	本地执行	Python, Crypto-Conditions
Corda	联盟链	基于交易	Raft	JVM	Kotlin, Java
BCS	联盟链	基于账户	用户定制的	Docker 容器	GoLang
Parity	私有链	基于账户	PoA	EVM	Solidity, Serpent, LLL
Multichain	私有链	基于交易	PoW	本地执行	C++
HydraChain	私有链	基于账户	PBFT	EVM	Solidity, Serpent, LLL

DERMODY[24] 等人在 2013 年创建了 Counterparty，Counterparty 是比特币链上的寄生应用，没有自己的区块链，而是将其数据嵌入到比特币交易中，用户可以在其上编写智能合约。不同于其他平台，Counterparty 的智能合约并没有形成共识，每一个网络参与者只是以相同的方式执行各自合约（如发送交

易）。由于每个网络节点具有相同的智能合约代码及相同的协议代码，每个智能合约的调用、执行、输出都是相同的（因为其执行所有代码在本质上具有确定性）。2013 年，BUTERIN[1] 和 WOOD[25] 创建了以太坊，首次将智能合约应用到区块链上，并提供了图灵完备的编程语言编写智能合约在以太坊虚拟机（Ethereum Virtual Machine，EVM）中执行。Quorum[26]、Monax[27]、Dfinity[28]、HydraChain[29]、Parity[30] 等众多区块链平台都是基于以太坊构建和扩展的。

2015 年 12 月，Linux 基金会启动了名为超级账本（Hyperledger）的开源项目，旨在发展跨行业的商业区块链平台 [31]。该基金会目前拥有 IBM、Intel、R3、百度、CISCO 等 200 多个成员。超级账本包括 Fabric[32,33]、Sawtooth[34]、Burrow[35] 和 Iroha[36] 等多个区块链项目，其中最受关注的是 Fabric。目前 Fabric 有 v0.6.0 和 v1.0.0 两个版本，v0.6.0 版本使用的共识算法是实用拜占庭容错算法（PBFT），而在 v1.0.0 版本中用户可以根据需要自行选择共识算法，包括 PBFT、Raft、PoW、PoS 等算法。Sawtooth 提供了一个构建和运行分布式账本的高度模块化平台，使用了时间消逝证明（Proof of ElapedTime，PoET）共识算法。2016 年，R3 公司面向金融机构定制设计了分布式账本平台 Corda[37]，Corda 保证数据仅对交易双方及监管方可见。2016 年 2 月，BigChainDB 公司发布了基于企业级分布式数据库构建的 BigChainDB 区块链平台 [38]，既有高吞吐量、大容量、低延迟、丰富高效的查询语言和权限管理等传统分布式数据库的优点，又有去中心化控制、不可篡改、可创建移动数字资产等区块链的特性。2018 年华为云发布了《华为区块链白皮书》[39]。华为云区块链服务 BCS 是基于开源区块链技术和华为在分布式并行计算、PaaS、数据管理、安全加密等核心技术领域多年积累的基础上推出的企业级区块链云服务产品。

币科学公司（Coin Sciences）的一个团队于 2015 年 1 月创建了 Multichain 项目 [40]，通过对用户权限的综合管理解决了挖矿、隐私和公开性问题。腾讯在 2017 年 4 月发布了区块链平台 Tbaas[41]，致力于提供企业级区块链基础设施、行业解决方案，以及安全、可靠、灵活的区块链云服务。Tbaas 支持自适应的共识机制、MySQL 和 MariaDB 等多种数据库以及毫秒级确认交易及海量并发。2017 年 1 月，国内的众享比特团队发布了号称全球首个基于区块链的数据库应用平台 ChainSQL[42]，使用 Ripple 网络作为区块链网络的架设，底层支持 MySQL、SQLite 和 DB2 等多种传统数据库。

3.1 比特币中的智能合约

比特币脚本是智能合约的雏形，包括比特币在内的数字货币大多采用非图灵完备的简单脚本代码编程控制交易过程。脚本是简单的、从左向右处理的、基于堆栈的执行语言。脚本语言非常有限，只包含一些基本的算术、逻辑和加密操作（如数字签名的验证和哈希）[43]，但是它为区块链可编程提供了一个原型，后续一些可编程区块链项目都是基于脚本的原理发展起来的。例如，以太坊就增强了脚本机制，脚本机制中不再是简单的 OP 指令，而是支持脚本的一套图灵完备语言。

比特币交易依赖于两类脚本来验证，即解锁脚本（Signature Script）和锁定脚本（Pubkey Script）[44]。解锁脚本是比特币交易输入的一部分，通常含有用户私钥生成的数字签名。锁定脚本指定了今后花费这笔交易必须满足的条件。解锁脚本与锁定脚本对应，只有满足锁定脚本的条件才允许花掉这个脚本上对应的资产。任何解锁脚本和锁定脚本的组合结果为真（True）则为有效。

比特币网络处理的大多数交易都采用了接收者的公钥加密和私钥解密，其对应的是 P2PKH（Pay-to-public-key-hash）标准交易脚本。图 4 显示了 P2PKH 脚本在堆栈式计算引擎中检验交易有效性的过程，其中 <sig> 和 <PubK> 是解锁脚本，其余的是锁定脚本。具体过程如下：1）将 <sig> 和 <PubK> 压入堆

栈中，执行 DUP，复制堆栈顶的 <PubK> 并压入堆栈。2）执行 HASH160，弹出堆栈顶的 <PubK>，并用 HASH160 计算，将计算结果压入堆栈，然后将 <PubKHash> 压入堆栈。3）执行 EQUALVERIFY，比较堆栈顶的两个数值，如果不同，验证出错，交易不合法。如果验证通过，堆栈只剩下 <sig> 和 <PubK>，CHECKSIG 将二者弹出，验证该交易签名是否是由该公钥对应用户使用其私钥签署的。如果是，交易合法；否则，交易不合法。

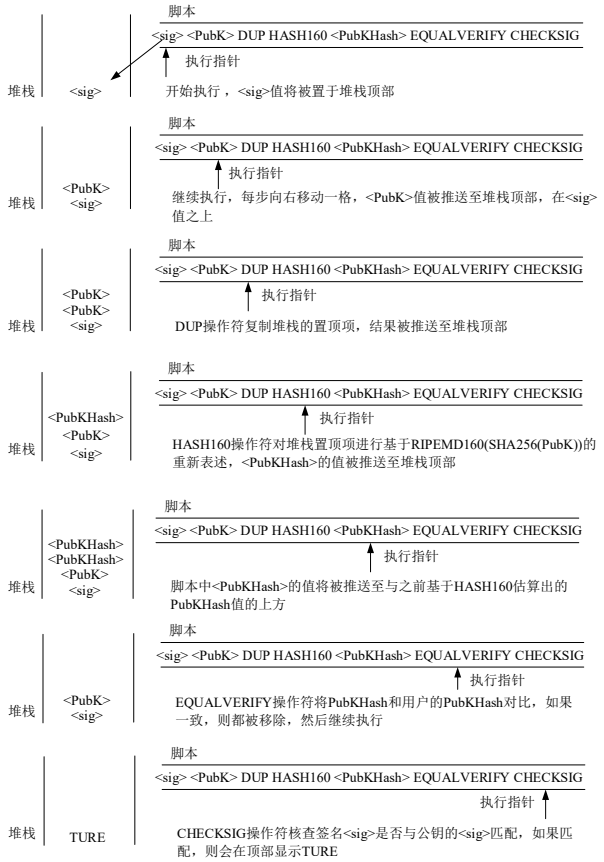

图 4 P2PKH 脚本检验交易有效性

比特币脚本系统还可以实现 *M*-*N* 多重签名，即至少提供 *N* 个私钥中的 *M* 个私钥（$N \leqslant M$）才能实现支付。多重签名可应用于公司会计账簿记账、中介担保、遗产分配等场景。

3.2 以太坊中的智能合约

以太坊的目的是基于脚本、竞争币和链上元协议（on-chain meta-protocol）概念进行整合和提高，使得开发者能够创建任意的基于共识的、可扩展的、标准化的、特性完备的、易于开发的和协同的应用 [1]。账户是以太坊的核心操作对象。以太坊中账户分为外部账户和合约账户两类，外部账户由公私钥对控制，是人为创建的能够存取货币的账户。合约账户由存储在账户中的代码控制，其地址是在创建合约时由合约创建者的地址和该地址发出过的交易数量计算得到的。

用于以太坊智能合约开发的语言主要有 Solidity、Serpent 和 LLL。在部署智能合约时，EVM 将用户编写的代码编译为基于堆栈的字节码语言，并存储在区块链上，在需要时通过 web3.js 库提供的 JavaScript API 接口来调用智能合约，并在 EVM 中运行。EVM 本身没有存储在区块链内，而是和区块链一样同时存储在各个节点计算机上。每个参与以太坊网络的校验节点都会运行 EVM，并将其作为区块有效性协议的一部分。每个节点都会对智能合约的部署和调用进行相同的计算，并存储相同的数据，以确保将正确的结果记录在区块链内。为了防止恶意用户部署无限循环的智能合约，以太坊要求用户为所部署的智能合约的每一步执行支付费用，费用的基础单位是 gas。

下面是用 Serpent 语言实现的一个令牌系统：从发送者中减去 *X* 个单位并将这 *X* 单位加到接收者中，前提条件是发送者在交易之前至少有 *X* 个单位并且发送者批准这笔交易。

```
from = msg.sender
to = msg.data[0]
value = msg.data[1]
If contract.storage[from] >= value:
contract.storage[from] >= value
contract.storage[to] = contract.storage[to] + value
```

3.3 超级账本中的智能合约

超级账本中共有 5 个项目支持智能合约，分别是 Burrow、Fabric、Indy、Iroha 和 Sawtooth。表 2 列出了超级账本项目使用的智能合约技术、类型及主要编程语言。

表 2 超级账本中的智能合约实现

项目	智能合约技术	智能合约类型	智能合约编程语言
Hyperledger Burrow	智能合约应用引擎	链上智能合约	本地语言代码
Hyperledger Fabric	链上代码	安装的智能合约	Go（大于 v1.0）、Javascript（大于 v1.1）
Hyperledger Indy	无	无	无
Hyperledger Iroha	链上代码	链上智能合约	本地语言代码
Hyperledger Sawtooth	交易集合	安装的智能合约和链上智能合约	JavaScript, C++, Java, Go, Python, Rust, Solidity

超级账本有两种不同类型的智能合约，即安装的智能合约和链上智能合约。

1）安装的智能合约。在网络启动之前，安装的智能合约在网络中的验证器上安装业务逻辑。

2）链上智能合约。链上智能合约将业务逻辑部署为交易，提交给区块链，并由后续交易调用。通过链上智能合约定义业务逻辑的代码将成为账本的一部分。

Fabric 是目前超级账本中最受关注的项目。Fabric 是以模块化架构为基础的分布式账本平台，提供高度的机密性（Confidentiality）、弹性（Resiliency）、灵活性（Flexibility）和可扩展性（Scalability）。Fabric 中智能合约被称为链上代码（Chaincode，以下简称链码），是使用 Go、JavaScript（node.js）编写的，并在 Java 等其他编程语言中实现了指定的接口。由应用程序调用并与分布式账本交互，其实质是在验证节点（Validating Node）上运行的分布式交易程序，用以自动执行特定的业务规则，并最终更新账本的状态。Fabric 使用轻量级 Docker 容器作为执行链码的环境，基于容器自身提供的隔离性和安全性保护宿主机不受容器中恶意合约的攻击，也防止了容器之间的相互影响。

Fabric 的链码分为应用程序链码（Application Chaincode）和系统链码（System Chaincode）。系统链码通常处理与系统相关的事务，如生命周期管理和策略配置，作用是简化节点和链码之间的 gRPC 通信成本，提升管理的灵活性。应用程序链码管理分类账本上的应用程序状态，包括数字资产或任意数据记录。由链码创建的状态仅限于该链码，不能由另一个链码直接访问。但是在同一网络中给定适当的权限，链码可以调用另一个链码来访问其状态。链码分为公开、保密和访问控制 3 种类型。公开链码可供任何一个成员调用，保密链码只能由验证成员（Validating Member）发起，访问控制型链码允许某些批准过的成员调用。

将链码、一个可选的实例化策略和拥有链码的实体签署的一组签名封装成链码包，链码包安装在交易方的网络节点上，节点通过向网络提交实例化事务激活链码，如果交易获得批准，链码进入活动（Active）状态，在此状态下，链码可以通过客户端应用程序接收来自用户的交易。任何经过验证的链码交易都会附加到共享账本中。链码实例化后可以随时通过更新事务进行升级。

4 智能合约存在的问题

虽然智能合约有很多显而易见的优点，但在实际应用中还存在一些不足并面临很多风险[45]。ALHARBY[46] 等人将智能合约现存的问题分为编码问题、安全问题、隐私问题和性能问题 4 类。

1）编码问题

智能合约的正确性意味着智能合约按其开发者的意图运行，因为这些智能合约具有有价值的货币单位[47]，如果智能合约未按预期执行，其部分货币单位将消失。例如，以太坊上的众筹项目 The DAO 的智能合约因递归调用漏洞而遭到攻击，约 1200 万个以太币被非法转移，后虽通过硬分叉追回了损失，但 The DAO 项目宣布失败[48]。为了解决这个问题，FRANTZ[49] 等人提出了一种建模方法，支持半自动化智能合约的创建，以便将人类可读的合约转换为智能合约规则。BHARGAVAN[50] 等人利用形式化方法分析和验证以太坊智能合约的正确性。BIGI[51] 等人将形式化方法与博弈论技术相结合来验证智能合约。DELMOLINO[47] 等人发布了教程，记录了编写智能合约时几类典型错误并给出指导方案，以帮助开发人员编写正确的智能合约。CHEN[52] 等人在智能合约中确定了 7 种导致额

外成本（如循环中不必要且昂贵的操作）的编程模式，并开发了一种名为 GASPER 的工具来检测受这些模式影响的智能合约，以避免用户额外的开销。

目前以太坊智能合约基于程序语言，在程序语言中，代码是作为一系列步骤执行的，程序员必须指定应该做什么以及如何去做。这使得使用这些语言编写智能合约的任务繁琐且容易出错 [53]。为了解决这个问题，IDELBERGER[53] 等人提议使用基于逻辑的语言而不是程序语言。超级账本项目支持 Java 语言和 Go 语言。

2）安全问题

NATOLI[54] 等人建议使用以太坊自带的函数（如 SendIfReceived）来强制执行事务顺序，如果没有以正确的顺序执行智能合约，可能会得出错误的结果。LUU[48] 等人构建了一个名为 OYENTE 的工具，可以检测以太坊智能合约是否存在被攻击的风险。一些智能合约需要区块链外部的信息（数据馈送），但不能保证外部提供的信息是可信的。ZHANG[55] 等人提出了一个 Town Crier（TC）解决方案，充当外部资源和智能合约之间的可信第三方，为智能合约提供经过验证的数据馈送。

3）隐私问题

在公有链系统中，所有交易和用户余额都可被公开查看，缺乏交易隐私。KOSBA[56] 等人构建了一个名为 Hawk 的工具，允许开发人员编写隐私保护智能合约，无需实施任何加密技术。WATANABE[57] 等人建议在将智能合约部署到区块链之前加密智能合约，只有合约的参与者才能使用解密密钥访问智能合约的内容。当智能合约要求数据馈送操作时，它会向提供这些馈送的一方发送请求，但该请求会暴露给公众，缺乏数据隐私。为了解决这个问题，ZHANG[55] 等人扩展了 TC 工具来支持私人请求。在发送请求之前，智能合约可以使用 TC 的公钥对请求进行加密，收到加密请求后，TC 可以使用其私钥对其解密，因此，可以保证请求的内容对区块链中的其他用户或者智能合约保密。

4）性能问题

在区块链系统中，智能合约按顺序执行，每秒可执行的智能合约数量将受到限制。随着未来智能合约数量的增加，区块链系统将无法扩展规模。VUKOLIĆ[58] 建议只要智能合约是独立的（如不更新相同变量）就可以并行执行智能合约。

5 结束语

以太坊是目前开发智能合约最常见的公有链平台，超级账本是最常见的联盟链平台。联盟链具有以下优势：1）采用多中心化，可以极大改善系统信任问题；2）可以联合多公司、多行业，对产业或国家的特定清算、结算用途意义重大，可以降低两地结算成本和时间，比现有的系统简单，效率更高；3）能够继承中心化的优点，容易进行控制权限设定；4）具有更高的可扩展性。因此未来在联盟链上开发智能合约应用会成为主流。由于智能合约有着执行准确、较低的人为干预风险、较低的运行成本等优点，未来将有广泛的应用场景，应用方向可能包括但不限于保险、电子政务、供应链、物流、交易清算等场景。（责编 潘海洋）

参考文献：

[1] BUTERIN V. A Next-generation Smart Contract and Decentralized Application Platform[EB/OL]. https://github.com/ethereum/wiki/wiki/White-Paper, 2018-5-11.

[2] UK Government Chief Scientific Adviser. Distributed Ledger Technology: Beyond Block Chain[EB/OL]. https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/492972/gs-16-1-distribute, 2018-5-11.

[3] NAKAMOTO S. Bitcoin: A peer-to-peer Electronic Cash System[EB/OL]. https://www.coindesk.com/bitcoin-peer-to-peer-electronic-cash-system, 2018-5-11.

[4] Department of Information and Software Services, Ministry of Industry and Information Technology. China Blockchain Technology and Application Development White Paper[EB/OL]. http://www.fullrich.com/Uploads/article/file/2016/1020/580866e374069.pdf.2016, 2018-5-11.

工业和信息化部信息化和软件服务业司. 中国区块链技术和应用发展白皮书 [EB/OL]. http://www.fullrich.com/Uploads/article/file/2016/1020/580866e374069.pdf.2016, 2018-5-11.

[5] XIE Hui, WANG Jian. Study on Block Chain Technology and Its Applications[J]. Netinfo Security, 2016, 16(9): 192–195.
谢辉，王健 . 区块链技术及其应用研究 [J]. 信息网络安全，2016，16(9)：192–195.
[6] SWAN M. Blockchain: Blueprint for a New Economy[EB/OL]. https://blog.vistart.me/book–blockchain–blueprint–for–a–new–economy, 2018–5–11.
[7] SZABO N. Smart Contracts[EB/OL]. http://virtualschool.edu/mon/Economics/SmartContracts.html, 2018–5–11.
[8] SZABO N. Formalizing and Securing Relationships on Public Networks[EB/OL]. https://www.researchgate.net/publication/220167894_Formalizing_and_Securing_Relationships_on_Public_Networks, 2018–5–11.
[9] YUAN Yong, WANG Feiyue. Block Chain: the State of the Art and Future Trends[J]. Acta Automatica Sinica, 2016, 42(4): 481–494.
袁勇，王飞跃 . 区块链技术发展现状与展望 [J]. 自动化学报，2016，42(4)：481–494.
[10] STARK J. Making Sense of Blockchain Smart Contracts[EB/OL]. http://www.coindesk.com/making –sense–smart–contracts, 2018–5–11.
[11] HAN Xuan, LIU Yamin. Research on the Consensus Mechanisms of Blockchain Technology[J]. Netinfo Security, 2017, 17(9): 147–152.
韩璇，刘亚敏 . 区块链技术中的共识机制研究 [J]. 信息网络安全，2017，17(9)：147–152.
[12] BOGNER A, CHANSON M, MEEUW A. A Decentralised Sharing APP Running a Smart Contract on the Ethereum Blockchain[EB/OL]. http://cocoa.ethz.ch/downloads/2017/08/2306_Sharing_App_Final_Publication.pdf, 2018–5–11.
[13] PARK L W, LEE S, CHANG H. A Sustainable Home Energy Prosumer–chain Methodology with Energy Tags over the Blockchain[J]. Sustainability, 2018, 10(3): 658.
[14] Al–BASSAM M. SCPKI: A Smart Contract–based PKI and Identity System[C]//ACM. The ACM Workshop on Blockchain, Cryptocurrencies and Contracts, April 2, 2017, Abu Dhabi, United Arab Emirates. New York: ACM, 2017: 35–40.
[15] YASIN A, LIU Lin. An Online Identity and Smart Contract Management System[C]//IEEE. 40th Annual Computer Software and Applications Conference, June 10–14, 2016, Atlanta, GA, USA. New Jersey: IEEE, 2016: 192–198.
[16] HUH S, CHO S, KIM S. Managing IoT Devices Using Blockchain Platform[C]//IEEE. 19th International Conference on Advanced Communication Technology, February 19–22, 2017, Bongpyeong, South Korea. New Jersey: IEEE, 2017: 464–467.
[17] HUCKLE S, BHATTACHARYA R, WHITE M, et al. Internet of Things, Blockchain and Shared Economy Applications[J]. Procedia Computer Science, 2016, 98(C): 461–466.
[18] ZHANG Yu, WEN Jiangtao. The IoT Electric Business Model: Using Blockchain Technology for the Internet of Things[J]. Peer–to–Peer Networking and Applications, 2017, 10(4): 983–994.
[19] XIA Qi, SIFAH E B, ASAMOAH K O, et al. MeDShare: Trust–Less Medical Data Sharing Among Cloud Service Providers via Blockchain[J]. IEEE Access, 2017, 5(99): 14757–14767.
[20] BENCHOUFI M, RAVAUD P. Blockchain Technology for Improving Clinical Research Quality[J]. Trials, 2017, 18(1): 335.
[21] CUNNINGHAM J, AINSWORTH J. Enabling Patient Control of Personal Electronic Health Records Through Distributed Ledger Technology[EB/OL]. https://www.researchgate.net/publication/322252917_Enabling_Patient_Control_of_Personal_Electronic_Health_Records_Through_Distributed_Ledger_Technology, 2018–5–11.
[22] JACYNYCZ V, CALVO A, HASSAN S, et al. Betfunding: A Distributed Bounty–based Crowdfunding Platform over Ethereum[EB/OL]. https://link.springer.com/chapter/10.1007%2F978–3–319–40162–1_44, 2018–5–11.
[23] HUANG Jiehua, GAO Lingchao, XU Yuzhuang, et al. The Design of Smart Contracts on Crowd Fuding Private Blockchain[J]. Journal of Information Security Research, 2017, 3(3): 211–219.
黄洁华，高灵超，许玉壮，等 . 众筹区块链上的智能合约设计 [J]. 信息安全研究，2017，3(3)：211–219.
[24] DERMODY R, KRELLENSTEIN A. Counterparty: Protocol specification(2014)[EB/OL]. http://counterparty.io/docs/protocol_specification, 2018–5–11.
[25] WOOD G. Ethereum: A Secure Decentralised Generalised Transaction Ledger[EB/OL]. http://gavwood.com/Paper.pdf, 2018–5–11.
[26] MORGAN J P. Enterprise–ready Distributed Ledger and Smart Contract Platforms[EB/OL]. https://github.com/jpmorganchase/quorum, 2018–5–11.
[27] MONAX. Monax: The Ecosystem Application Platform[EB/OL]. https://www.findbestopensource.com/product/monax–monax, 2018–5–11.
[28] DFINITY. Dfinity[EB/OL]. https://dfinity.network, 2018–5–11.
[29] GitHub. Hydrachain: Permissioned Distributed Ledger Based on Ethereum[EB/OL]. https://github.com/HydraChain/hydrachain, 2018–5–11.
[30] Ethcore. Parity: Next Generation Ethereum Browser[EB/OL]. https://ethcore.io/parity.html, 2018–5–11.
[31] Hyperledger. Blockchain Technologies for Business[EB/OL]. https://www.hyperledger.org, 2018–5–11.
[32] Hyperledger. Hyperledger fabric v0.6.0[EB/OL]. https://github.com/hyperledger/fabric/releases/tag/v0.6.0–preview, 2018–5–11.
[33] Hyperledger. Hyperledger fabric v1.0.0[EB/OL]. https://github.com/ hyperledger/fabric/releases /tag/v1.0.0–preview, 2018–5–11.
[34] GitHub. Sawtooth Lake[EB/OL]. https://github.com/hyperledger/sawtooth–core, 2018–5–11.
[35] Hyperledger. Hyperledger Burrow[EB/OL]. https://www.hyperledger.org/projects/hyperledger–burrow, 2018–5–11.
[36] Hyperledger. Hyperledger Iroha[EB/OL]. https://www.

hyperledger.org/projects/iroha, 2018−5−11.
[37] HEARN M. Corda: A Distributed Ledger[EB/OL]. https://docs.corda.net/_static/corda−technical−whitepaper.pdf, 2018−5−11.
[38] MCCINAGHY T. BigchainDB: A Scalable Blockchain Database[EB/OL]. http://coinreport.net/wp−content/uploads/2016/02/BigchainDB−Primer−20160210.pdf, 2018−5−11.
[39] Huawei Technologies Co., Ltd. Huawei Blockchain White Paper[EB/OL]. https://static.huaweicloud.com// upload/files/pdf/20180411/20180411144924_27164.pdf, 2018−5−11.
华为技术有限公司. 华为区块链白皮书 [EB/OL]. https://static.huaweicloud.com//upload/files/pdf/20180411/20180411144924_27164.pdf, 2018−5−11.
[40] MultiChain. MultiChain: Open Platform for Blockchain Applications[EB/OL]. https://www.multichain.com2018−5−11.
[41] Tencent Research Institute. Tencent Blockchain Solution White Paper[EB/OL]. https://cloud.tencent.com /product/tbaas, 2018−5−11.
腾讯研究院. 腾讯区块链方案白皮书 [EB/OL]. https://cloud.tencent.com/product/tbaas, 2018−5−11.
[42] Beijing PeerSafe Technology Co., Ltd. Blockchain−based Database Application Platform Technology White Paper[EB/OL]. http://www.peersafe.com, 2018−5−11.
北京众享比特科技有限公司. 基于区块链的数据库应用平台技术白皮书 [EB/OL]. http://www.peersafe.com, 2018−5−11.
[43] BARTOLETTI M, POMPIANU L. An Empirical Analysis of Smart Contracts: Platforms, Applications, and Design Patterns[C]// Springer. International Conference on Financial Cryptography and Data Security, April 3−7, 2017, Sliema, Malta. Heidelberg: Springer, 2017: 494−509.
[44] Andreas M. Mastering Bitcoin[EB/OL]. http://shop.oreilly.com/product/0636920049524.do, 2018−5−11.
[45] WANG Hao, SONG Xiangfu, KE Junming, et al. Blockchain and Privacy Preserving Mechanisms in Cryptocurrency[J]. Netinfo Security, 2017, 17(7): 32−39.
王皓,宋祥福,柯俊明,等. 数字货币中的区块链及其隐私保护机制 [J]. 信息网络安全, 2017, 17(7): 32−39.
[46] ALHARBY M, MOORSEL A V. Blockchain−based Smart Contracts: A Systematic Mapping Study[EB/OL]. http://cn.arxiv.org/ftp/arxiv/papers/1710/1710.06372.pdf, 2018−5−11.
[47] DELMOLINO K, ARNETT M, KOSBA A, et al. Step by Step Towards Creating a Safe Smart Contract: Lessons and Insights from a Cryptocurrency Lab[C]//Springer. International Conference on Financial Cryptography and Data Security, February 22−26, 2016, Christ Church, Barbados. Heidelberg: Springer, 2016: 79−94.
[48] LUU L, CHU D H, OLICKEL H, et al. Making Smart Contracts Smarter[C]//ACM. 2016 ACM Sigsac Conference on Computer and Communications Security, October 24−28, 2016, Vienna, Austria. New York: ACM, 2016: 254−269.
[49] FRANTZ C K, NOWOSTAWSKI M. From Institutions to Code: Towards Automated Generation of Smart Contracts[C]// IEEE. 1st International Workshops on Foundations and Applications of Self Systems, September 12−16, 2016, Augsburg, Germany. New Jersey: IEEE, 2016: 210−215.
[50] BHARGAVAN K, SWAMY N, ZANELLA−BÉGUELIN S, et al. Formal Verification of Smart Contracts: Short Paper[EB/OL]. https://www.microsoft.com/en−us/research/wp−content/uploads/2017/08/solidetherplas.pdf, 2018−5−11.
[51] BIGI G, BRACCIALI A, MEACCI G, et al. Validation of Decentralised Smart Contracts Through Game Theory and Formal Methods[EB/OL]. https://link.springer.com/chapter/10.1007%2F978−3−319−25527−9_11, 2018−5−11.
[52] CHEN Ting, LI Xiaoqi, LUO Xiapu, et al. Under−optimized Smart Contracts Devour Your Money[EB/OL]. http://pdfs.semanticscholar.org/7ae3/74185441ca58953d53cede25135e4ad1c2a3.pdf, 2018−5−11.
[53] IDELBERGER F, GOVERNATORI G, RIVERET R, et al. Evaluation of Logic−based Smart Contracts for Blockchain Systems[C]//Springer. International Symposium on Rules and Rule Markup Languages for the Semantic Web, July 6−9, 2016, Stony Brook, NY, USA. Heidelberg: Springer, 2016: 167−183.
[54] NATOLI C, GRAMOLI V. The Blockchain Anomaly[C]// IEEE. 15th International Symposium on Network Computing and Applications, October 31−November 2, 2016, Cambridge, MA, USA. New Jersey: 2016: 310−317.
[55] ZHANG Fan, CECCHETTI E, CROMAN K, et al. Town Crier: An Authenticated Data Feed for Smart Contracts[EB/OL]. http://www.cs.cornell.edu/~fanz/files/pubs/tc−ccs16−final.pdf, 2018−5−11.
[56] KOSBA A, MILLER A, SHI E, et al. Hawk: The Blockchain Model of Cryptography and Privacy−preserving Smart Contracts[C]//IEEE. 2016 IEEE Symposium on Security and Privacy, May 22−26, 2016, San Jose, CA, USA. New Jersey: IEEE, 2016: 839−858.
[57] WATANABE H, FUJIMURA S, NAKADAIRA A, et al. Blockchain Contract: A Complete Consensus Using Blockchain[C]//IEEE. IEEE 4th Global Conference on Consumer Electronics, October 27−30, 2015, Osaka, Japan. New Jersey: IEEE, 2016: 577−578.
[58] VUKOLIĆ M. Rethinking Permissioned Blockchains[C]// ACM. 2017 ACM Workshop on Blockchain, Cryptocurrencies and Contracts, April 2, 2017, Abu Dhabi, United Arab Emirates. New York: ACM, 2017: 3−7.