一种基于 TEE 的隐私保护智能合约平台

高丰[1]， 顾叶明[2]

[1]（上海数据科学重点实验室，上海 201203）

[2]（江南计算技术研究所，江苏 无锡 214083）

通信作者：高丰，E-mail: gflet@163.com

摘 要： 智能合约是一种能实现自我执行和自我验证的计算机协议，智能合约已经成为区块链的重要组成部分，基于智能合约的去中心化的应用（DAPP）为区块链应用场景提供了重要的技术支撑。零知识证明和多方安全计算是智能合约中隐私保护的重要方法，但这些方法由于效率较低，并不能满足很多应用场景的需求。本文分析了基于可信计算环境（TEE）和区块链技术相结合的隐私保护智能合约平台技术。虽然 TEE 本身也面临着安全挑战，但采用基于秘密共享的密钥管理方案，在多个节点共同认证和授权下，安全节点才能获取密钥并执行智能合约，保障了系统的安全。本文的方案具备抗击内部攻击和外部攻击的能力，具备很好的安全特性。

关键词： 智能合约;可信计算环境;区块链;秘密共享

中图法分类号： TP311

中文引用格式

英文引用格式

A TEE Based Confidentiality-Preserving Smart Contracts Platform

Gao Feng[1], Gu Ye-Ming[2]

[1](Shanghai Key Laboratory for Data Science, Shanghai 201203, China)

[2](Jiangnan Institute of Computing Technology, Wuxi 214083, China)

Abstract: Smart contract is a kind of computer protocol that can realize self-execution and self-verification. Smart contracts are an important part of Blockchain technology now. Smart contract based Decentralized applications (DAPP) are provides important technical support in Blockchain scenarios. The zero-knowledge proof and secure multiparty computation are important methods for the confidentiality protection in smart contract, but these methods cannot meet the requirements of many application scenarios due to their significant performance overhead. In this paper we analyzes the confidentiality protection in smart contract platform based on trusted computing environment (TEE). Although TEE faces security challenges, the secret sharing based key management scheme is adopted in this paper. Under the joint authentication and authorization of multiple nodes, the security node can acquire the key and execute the smart contract. The scheme in this paper has the ability to resist internal and external attacks，so it has good security features.

Key words: Smart Contracts; Trust Executing Environment; Blockchain; Secret Share

在计算机科学领域，智能合约是指一种计算机协议，这类协议一旦制定和部署，就能实现自我执行（self-executing）和自我验证（self-verifying），不再需要人为的干预。智能合约作为一种程序，可以自主地执行全部或部分和合约相关的操作，并产生相应的可以被验证的证据，来说明执行合约操作的有效性。

智能合约通常在区块链上部署，而区块链的共识机制通常要求数据是公开和可见的。因此，现有的智能合约缺少机密性和隐私性，因此并不能满足对数据机密性要求高的场景。面对保密性和隐私性挑战，密码学基于零知识证明[1]和安全多方计算[2]进行了很多有益的探索。但是，目前因为密码理论的制约，这些方法计算开销很大，性能受到很大影响，仅能适用于少数的应用场景。可信执行环境（TEE）构建了一种全隔离的安全硬件环境，能够保护内部的数据和执行状态不会被篡改或者被探测，因此，基于 TEE 安全硬件和区块链可以构建一种更通用和更高效的强隐私保护智能合约执行平台。

从系统安全的三方面要素：机密性、完整性和可用性的角度，区块链的去中心化的共识机制，优点在于保障系统的完整性和可用性，但不利于保障系统的机密性；而 TEE 的优点保障系统的机密性，不利于保障系统的完整性和可用性。因此区块链和 TEE 具备强互补性，基于 TEE 构建去中心化的区块链系统，可以从机密性、完整性和可用性三个方面，对系统的安全提供很好的保障。

1 智能合约中的隐私保护及研究进展

20 世纪 90 年代，从事数字合约和数字货币研究的计算机科学家尼克萨博（Nick Szabo）第一次提出了“智能合约”的概念[3]，其初衷是将已有的合约法律法规以及相关的商业实践转移到互联网上来，使得陌生人通过互联网就可以实现以前只能在线下进行的商业活动，并实现真正的完全的电子商务。在部署智能合约之前，与合约相关的所有条款的逻辑流程就已经被制定好了。智能合约通常具有一个用户接口（interface），以供用户与已制定的合约进行交互，这些交互行为都严格遵守此前制定的逻辑。得益于计算机网络和密码学，这些交互行为能够被严格地验证，以确保合约能够按照此前制定的规则顺利执行，从而防止出现违约行为。

智能合约具有高效实时更新、准确执行、人为干预风险低等优点，从 2017 年之后，基于区块链各类应用的推广和普及，智能合约数量大幅增加，仅 2018 年第一季度，以太坊上新部署的智能合约的数量超过 50 万，达到 536120[4]。智能合约已经成为区块链中重要的组成部分。但与此同时，现有的区块链也在很大程度上制约了智能合约，特别是在隐私保护方面。现有的基于假名机制的区块链对隐私保护是非常薄弱的[5,6,7,8]。为了在共识机制中验证状态迁移的合法性，区块链上的数据通常是公开的。因此，智能合约也缺少保密性和隐私性机制，一些敏感数据并不能在区块链上存储和运算。以太坊作为现有的最大的去中心化的智能合约平台，几乎只能适用于 token 类的应用。

隐私保护一直是区块链领域重要的研究内容。Hawk[1]基于以太坊的智能合约平台进行开发，将智能合约分为公有合约部分和私有合约部分。Hawk 采用链下的方式提供智能合约的隐私保护，将零知识证明的结果上链形成共识。Hawk 中采用的多方计算的效率比较低，而且 Hawk 仍然不能保证合约代码的隐私性，攻击者可以通过反编译等手段获得合约代码。Hyperledger 提供了 PDO 技术[9,10]（Private Data Objects）将分布式账本和可信硬件结合在一起，PDO 使用 Intel SGX 中执行的智能合约用于调解不同的利益方对数据的访问权限；微软的 Coco Framework[11]，理论上可以用来保护任意区块链系统的隐私性，它提供了一个信任的基础，可以整合现有的链协议（如 Ethereum 等），提供完整的企业级的区块链隐私解决方案。

2 可信执行环境（TEE）及其安全缺陷

2.1 Intel可信执行环境

智能合约需要涉及诸如密码、账号、健康信息、加密密钥和信用卡等私人信息。这些敏感数据只能由指定交易方才能访问。按英特尔 SGX 术语来讲，这些隐私信息被称为应用机密。

操作系统的任务对计算机系统实施安全策略，以避免这些机密信息无意间暴露给其他用户和应用。操作系统会阻止用户访问其他用户的文件（除非访问已获得明确许可），阻止应用访问其它应用的内存，阻止未授权用户访问操作系统资源（除非通过受到严格控制的界面进行访问）。应用通常还会采用数据加密等其它安全保护措施，以确保发送给存储器或者通过网络连接而发送的数据不会被第三方访问。

虽然有很多措施可保护应用免受未授权用户访问，但大部分计算机系统仍然面临着一项重大安全隐患：几乎没有一种措施可保护应用免受拥有更高权限的应用或者操作系统本身的入侵。复杂的恶意软件可以锁定应用的保护方案，提取加密密钥，甚至直接从内存提取机密数据。一旦获取管理权限，恶意软件可不受限制地访问所有系统资源。

为对这些机密信息提供高级别的保护，同时抵御恶意软件的攻击，英特尔公司设计了 Intel SGX，ARM 公司设计了 TrustZone[12] 等可信执行环境（TEE）。以英特尔 SGX 为例，SGX 是一套 CPU 专用指令，可支持应用创建安全区：应用地址空间中受保护的区域，它可确保数据的机密性和完整性。即便有获取权限的恶意软件，也不能篡改或是窥探隐私数据的执行流程。

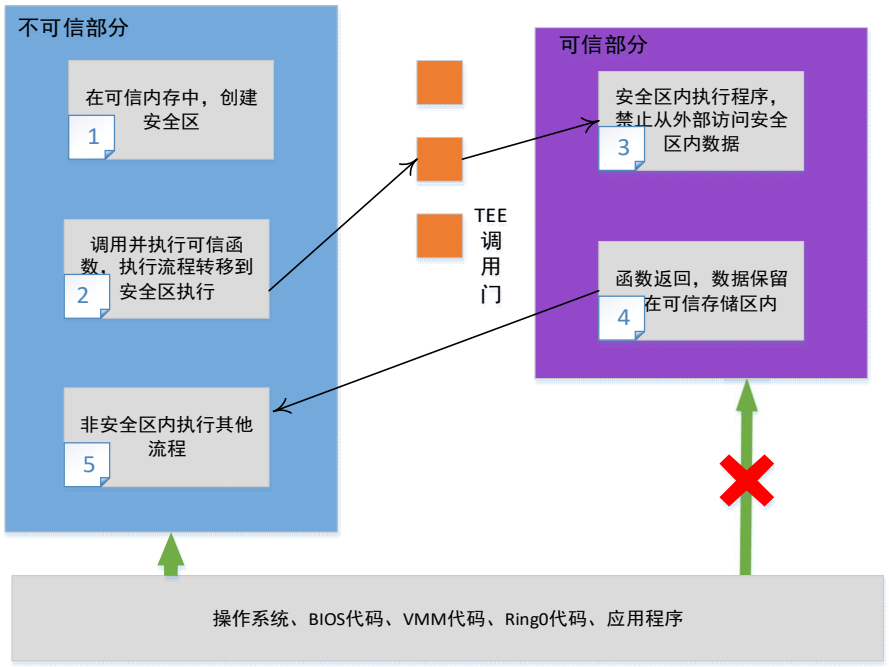

Fig. 1 the program framework based on a trusted execution environment (TEE)

图 1 基于可信执行环境（TEE）的程序框架

如图 1 所示，英特尔 SGX 技术的应用设计要求将应用分成两个部分：

可信部分：安全区内执行的代码。可信代码访问应用机密。应用程序可以拥有一个以上可信部分。

不可信部分：它包括应用的剩余部分及其所有模块。从安全区的角度来看，操作系统和虚拟机显示器都作为不可信部分。

可信部分应尽量保持最小，仅限于需要最高等级保护的数据以及必须直接作用于其上的操作。具有复杂界面的大型安全区不仅仅会消耗更多受保护内存，还会产生更大攻击面。

2.2 TEE的安全挑战

尽管 TEE 是以保护程序被篡改和被探测为初衷，但 TEE 本身也不可避免的存在安全问题。当前研究者认为 TEE 主要存在拒绝服务类和侧边带攻击两类安全性问题：

拒绝服务类攻击：TEE 的安全区（enclave）的进出 IO 是性能瓶颈，攻击者如果控制了主机，那么攻击者可以通过创立大量的 TEE 程序堵塞 enclave 的输入输出单元。由于 TLB 中缓存 enclave 中的内存访问缘故，因而进出 enclave 需要进行 TLB flush。另外执行 enclave 代码时，非 TLB 的内存访问也会造成额外的一些检查，导致更大的额外开销。同时也存在 TEE 代码编写漏洞风险，如果 enclave 编写代码本身有漏洞的话，enclave 是无法保护程序的安全，目前就有针对缓冲区溢出的 ROP 攻击能够控制 enclave。

侧边带类攻击[13]（Side-Channel Attacks）：2018 年 3 月，威廉玛丽学院、加州大学河滨分校等的研究人员已经确定并证实了一种名为“BranchScope”的 Side-Channel 新型攻击方式。BranchScope 是以定向分支预测单元（BPUs）为攻击目标。分支预测单元是 CPU 用于通过猜测分支指令的执行路径来提高流水线处理器的性能，BPU 包括了一个分支目标缓冲区（BTB）和一个方向预测器。当两个进程在同一个物理 CPU 内核上执行时，它们共享一个 BPU，恶意进程操纵如果能控制分支目标缓冲区或者是方向预测器，都可以从内存中获取敏感数据。

在本系统中的方案，其威胁模型包括某些节点试图主动或被动窃取 TEE 内部状态的风险，即 TEE 存在被拒绝服务或是侧边带攻击的可能。本文的方案中，对密钥管理基于时间和空间进行划分，只有被多数信任的节点，才能获得短期密钥，从而将节点被攻击的威胁控制在可控的范围内。

3. 系统模型及工作流程

3.1 系统模型

系统模型中包括 3 种实体：用户、共识节点和安全节点。

用户：智能合约的终端用户，用户创建智能合约并将其提交到平台。用户授权给安全节点执行智能合约。

安全节点：安全节点由智能合约执行及密钥管理两个模块所组成，多个安全节点构成安全层。智能合约执行和密钥管理都是基于 TEE 的安全区内执行，保障智能合约、用户数据和密钥体系的安全。智能合约执行模块在执行用户请求的合约后，会产生认证消息（attestation），并将 attestation 提供给共识节点，用于区块链记账。

共识节点：共识节点维护不可篡改的区块链账本。通过安全节点和共识节点之间的协议，共识节点负责检查安全节点所提供的 attestation 的正确性，合约的状态和运行的认证结构会记录在区块链账本中。

3.2 基于秘密共享的密钥管理

由于 TEE 本身可能存在安全缺陷，而且区块链去中心化的特性，使得系统不能依赖于某个 TEE（或者固定几个 TEE）安全节点，系统需要有完善的密钥管理的机制，使得加密和解密可以在不同的安全节点上处理。为了使密钥传播而带来的风险降到最低，本文采用基于多项式的(t, n)门限秘密共享的密钥管理机制，实现密钥管理委员会机制，安全节点通过和委员会的加密通信，获取密钥的 share 并恢复密钥。

传统上秘密共享是一种分割式保存方式，它是将一个密钥分割成多份，每个用户保存一份，所有用户合作可以恢复密钥。传统方式可以较好解决密钥的安全问题，但是必须所有用户合作才能恢复密钥，灵活性较差。为了解决上述密钥管理方式的不足，Shamir 等人[14,15]提出了新的秘密共享的机制。它的是将一个密钥分割成多份share，然后将每一份share发送给一个用户保存，只要大于规定门限数量的用户集合合作可以恢复秘密。这种方式既安全的保存了密钥，提高了密钥管理系统的可靠性，也可以根据需求让不同的用户集合恢复秘密，具有较好灵活性和实用性。随着秘密共享的发展，为了解决对密钥管理的不同需求，秘密共享发展出很多分支，如门限秘密共享，动态秘密共享，可验证的秘密共享，多秘密共享，公平秘密共享等。

3.3 工作流程

当用户创建一个合约，会将合约代码发送给相应的安全节点，安全节点将合约代码加载到 TEE 的安全区并开始初始化智能合约执行模块。执行模块首先产生合约相关的 id（记为 cid）从密钥管理模块获取合约相关的公私钥对$(pk_{cid}^{in}, sk_{cid}^{in})$和会话密钥$k_{cid}^{state}$，执行模块首先产生一个加密的初始状态$\mathrm{Enc}(k_{cid}^{state}, \vec{0})$并产生认证消息$\sigma_{TEE}$用于证明初始状态的正确性。认证消息$\sigma_{TEE}$会和加密合约、公钥、加密的初始状态等形成一个条目（$\mathrm{Enc}(k_{cid}^{state}, Contract), pk_{cid}^{in}, Enc(k_{cid}^{state}, \vec{0}), \sigma_{TEE}$）并发送给共识节点。共识节点首先验证认证消息的正确性，如果正确则将该条目记录在区块链上，并返回用户该交易的凭证。

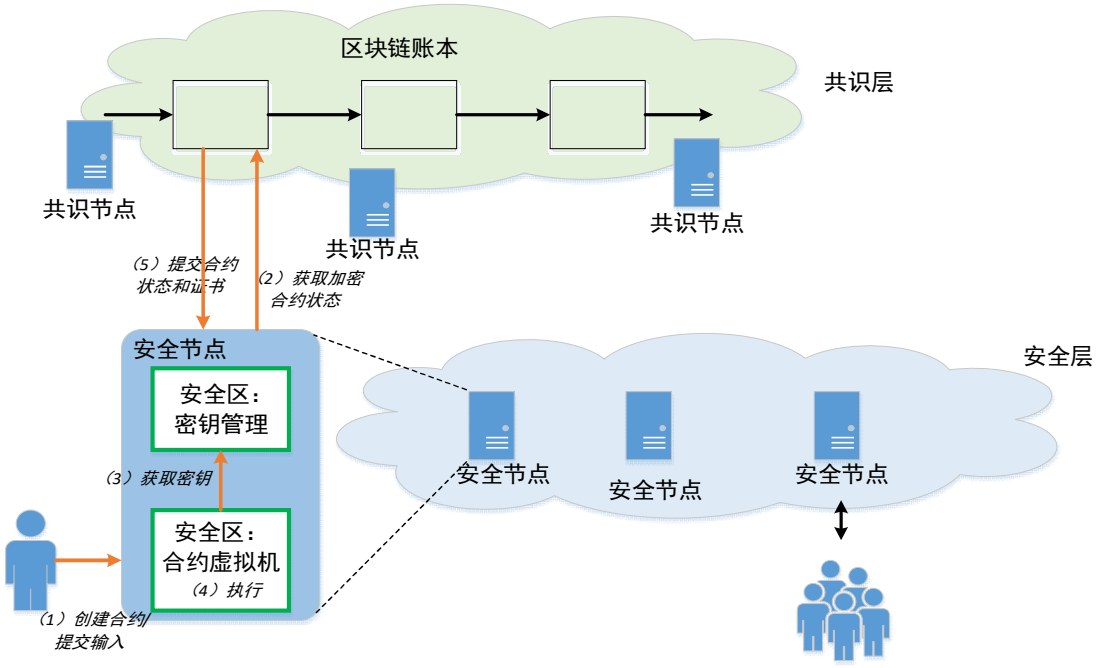

Fig 2. Workflow of the confidentiality-preserving smart contract platform

图 2 基于可信执行环境的智能合约执行流程

在合约初始化完成后，用户可以调用执行智能合约。执行的流程：

用户在根据合约初始化时获取的凭证，用户访问区块链获取 cid 和交易相关公钥pk_{cid}^{in}，用户首先将输入数据 inp用pk_{cid}^{in}加密，得到数据密文$inp_{ct} = Enc(pk_{cid}^{in}, inp)$，并将密文形成一个消息（cid, inp_{ct}）发送给安全节点。

安全节点采用秘密共享的方案，从多个安全节点的密钥管理模块恢复k_{cid}^{state}。安全节点将区块链中的所保存的$\mathrm{Enc}(k_{cid}^{state}, Contract)$及$\mathrm{Enc}(k_{cid}^{state}, st_{prev})$加载到 TEE 的安全区内，分别解密获取合约代码及该合约的上一个状态st_{prev}。

同理，安全节点从密钥管理模块获取 cid 相关的私钥sk_{cid}^{in}，将用户的加密数据inp_{ct}加载到 TEE 的安全区，由私钥解密，并获取明文的输入数据。TEE 的执行模块执行智能合约，产生运行结果、合约状态和签名消息的三元组（$outp$, $\mathrm{Enc}(k_{cid}^{state}, st_{new})$, σ_{TEE}），其中运行结果和合约状态处于加密的状态。

安全节点将将新的状态通过原子化的协议传递给共识节点，当且仅当共识节点接受了新的状态，计算结果返回给用户。用户得到了新的计算结果。

4. 安全性评估

在本文的方案中，密钥管理模块（Key Manager，简称 KM）是可信安全的 share 分发的实体。KM 选取大素数 p，有限域上的随机数 r 及对称多项式等基本参数，同时公布哈希函数 Hash(x)及秘密 s 的 Hash(s)。本文假定安全节点在初始化合约的阶段可以安全获取 share 和哈希函数等信息。对于协议的安全性，以假设攻击模型为主要研究方法，其中主要考虑外部攻击和内部攻击两种情形。

4.1 外部攻击评估

在外部攻击的情况下，用户(U_i, U_j)通过私有密钥S_{ij}加密 Component 后互相发送，因此，外部攻击者即使截取了通信信息，仍需要破解S_{ij}才能获取 Component，而S_{ij}本质上是由 KM 秘密选取的对称多项式生成的，攻击者在未掌握对称多项式相关信息的情况下是无法获取S_{ij}的。因此针对外部攻击是有效的。

内部攻击

多个组内用户合谋获取组密钥，即合谋者通过互相发送 Component 来恢复秘密，进而排除其他的合法用户参与，假定合谋者最多不超过 t-1 个，基于(t, n)门限秘密共享的原理，本文方案可以抵御 t-1 个内部用户合谋攻击，攻击者无法获取密钥。

5. 总 结

智能合约是一种能实现自我执行和自我验证的计算机协议，智能合约已经成为区块链的重要组成部分，基于智能合约的去中心化的应用为区块链应用场景提供了重要的技术支撑，智能合约的机密性和隐私性是区块链的重要研究课题。本文分析了基于可信硬件和区块链技术相结合的隐私保护智能合约平台技术，虽然 TEE 面临着安全挑战，但采用基于秘密共享的密钥管理方案，可以在多

个节点共同认证下，节点才能获取状态密钥和私有密钥。本文的方案具备抗击内部攻击和外部攻击的能力，基于很好的安全特性。

References:

[1] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, “Hawk: The blockchain model of cryptography and privacy-preserving smart contracts,” in Security and Privacy (SP), 2016 IEEE Symposium on. IEEE, 2016, pp. 839–858.

[2] G. Zyskind, O. Nathan et al., “Decentralizing privacy: Using blockchain to protect personal data,” in Security and Privacy Workshops (SPW), 2015 IEEE. IEEE, 2015, pp. 180–184.

[3] Tapscott Don；Tapscott Alex（May 2016）.The Blockchain Revolution：How the Technology Behind Bitcoin is Changing Money，Business，and the World.pp.72，83，101，127.

[4] https://hackernoon.com/ethereum-smart-contracts-most-of-them-are-rarely-used-f45749730d3e

[5] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, “A fistful of Bitcoins: characterizing payments among men with no names,” in Proceedings of the 2013 conference on Internet measurement conference. ACM, 2013, pp. 127–140.

[6] M. Moser and R. B ohme, “The price of anonymity: empirical evidence ¨ from a market for Bitcoin anonymization,” Journal of Cybersecurity, 2017.

[7] F. Reid and M. Harrigan, “An analysis of anonymity in the Bitcoin system,” in Security and privacy in social networks. Springer, 2013, pp. 197–223.

[8] D. Ron and A. Shamir, “Quantitative analysis of the full Bitcoin transaction graph,” in International Conference on Financial Cryptography and Data Security. Springer, 2013, pp. 6–24.

[9] “Hyperledger Private Data Objects,” https://github.com/ hyperledger-labs/private-data-objects.

[10] M. Bowman, A. Miele, M. Steiner, and B. Vavala, “Private data objects: an overview,” arXiv preprint arXiv:1807.05686, 2018.

[11] Microsoft, “The Coco Framework: Technical Overview,” https://github. com/Azure/coco-framework/.

[12] TrustZone, https://developer.arm.com/technologies/trustzone

[13] Y. Xu, W. Cui, and M. Peinado, “Controlled-channel attacks: Deterministic side channels for untrusted operating systems,” in IEEE Symposium on Security and Privacy, SP, 2015, pp. 640–656.

[14] D. Schultz, B. Liskov, and M. Liskov, “MPSS: Mobile proactive secret sharing,” ACM Transactions on Information and System Security (TISSEC), vol. 13, no. 4, p. 34, 2010.

[15] L．Harn，Changlu Lin，Authenticated Group Key Transfer Protocol Based on Secret Sharing，IEEE Transactions on Computers，v01.59，no.6，PP.842—846，June 2010.