基于SGX的区块链闭环可信计算装置

1 相关技术背景（背景技术）

1.1背景技术

随着区块链概念与技术的兴起与发展，越来越多的应用场景使用区块链技术去满足信息存证、信息溯源、跨境支付等去中心化需求。目前这类应用场景主要利用区块链技术中的共识机制保证分布式账本的一致性与完整性，以便在弱信任环境下建立起可靠、可信、可恢复的多方交互协作计算模式。

区块链技术目标不仅限于为应用赋能一致、完整的分布式账本，而且应能够保证交易数据在整个交易生命周期的隐私性和计算可信性。针对这类新需求，目前有多种不同的方案与实现，主要是基于密码学的同态加密方案，具体可参考 1.2。该方案有独自的特点与适用场景，然而在计算效率、抵抗拜占庭恶意行为等方面不可兼顾，难以支撑起复杂、大规模的区块链应用可信计算需求。本专利基于 Intel SGX (Software Guard Extensions) 技术提出一种基于 SGX 的区块链闭环可信计算装置，从而保证区块链数据在整个交易闭环中的计算可信性和高效性。

1.2相关的现有技术一

1.2.1 现有技术一的技术方案

同态加密是基于数学难题的计算复杂性理论的密码学技术。对经过同态加密的数据进行处理得到一个输出，将这一输出进行解密，其结果与用同一方法处理未加密的原始数据得到的输出结果是一样的。用数学表达式来表示为：

$$E(m_1) * E(m_2) = E(m_1 * m_2) \quad \forall m_1, m_2 \in M$$

区块链用户首先对消息集M中的交易数据m_1, m_2进行加密，然后将加密后的密文$E(m_1), E(m_2)$上传到区块链智能合约进行同态计算，计算结果为$E(m_1) * E(m_2)$并与区块链用户的本地计算结果$E(m_1 * m_2)$比对，校验通过后将计算结果存储到区块链。如上整个交易生命周期中，用户隐私数据m_1, m_2未向其他服务或者环境进行暴露，达到了区块链隐私数据的可信计算目的。

目前，同态加密算法主要分为部分同态加密算法、层次同态加密算法、全同态加密算法。部分同态加密算法原理简单、易实现，但仅支持一种运算，即同态加法或者同态乘法；层次同态加密算法一般支持有限次数的加法与乘法运算，但随着运算次数增加，噪音会越来越大，导致密文无法再被解密；全同态加密支持无限次数、任意类型的计算，但计算效率过低。

1.2.2 现有技术一的缺点

- ✧ 为支撑不同类区块链应用的可信计算需求而采用全同态加密算法会导致计算效率过低，交易响应过慢，无法适应目前提升区块链吞吐量的发展趋势。
- ✧ 同态加密属于软件级数据隐私保护方案，若算法本身或所在系统存在漏洞，则极有可能导致密钥泄露或密文被破解。

2 本技术方案的详细阐述

2.1所要解决的技术问题

本方案提出一种基于 SGX 的区块链闭环可信计算装置，主要围绕解决区块链交易数据在传输过程、计算过程、存储过程的隐私问题和效率问题，为区块链用户的每笔交易提供可信、可靠、高效计算的环境。本方案通过 SGX 的 enclave 硬件可信执行环境执行应用智能合约来确保计算过程的可信，并结合 Intel EPID (Enhanced Privacy Identifier) 协议向区块链用户提供该 enclave 实例的可信证明材料；然后针对非 enclave 环境的区块链用户交易数据维护必要的密钥，进而从传输、计算、存储的交易全生命周期维度对用户敏感数据进行基于硬件的保护。由于 enclave 内存只能由 enclave 内的进程进行访问，其他如操作系统、特权进程、BIOS 等是无法进行访问的，因此出现拜占庭恶意行为的区块链节点也无法获取到用户敏感数据和 enclave 敏感数据。同时当运行 enclave 内智能合约的 CPU 访问对应实例 enclave 内存时是明文的，从而保证可信计算的高效性。

2.2提供的完整技术方案

本方案整体分为两部分，首先选取区块链交易环节的智能合约进行 enclave 化，即构建基于 enclave 的可信智能合约环境；然后基于第一部分 enclave 提供的“保险箱”功能对区块链非智能合约部分的数据流维护必要的密钥进行加密，即构建基于可信智能合约环境的可信

区块链。从 enclave 到智能合约、从智能合约到整个区块链的可信链条是源于 SGX 的密钥体系结构的，因此此处引入 SGX 密钥体系派生图，如图 1 所示，以便 2.2.1 部分和 2.2.2 部分的可信度分析。

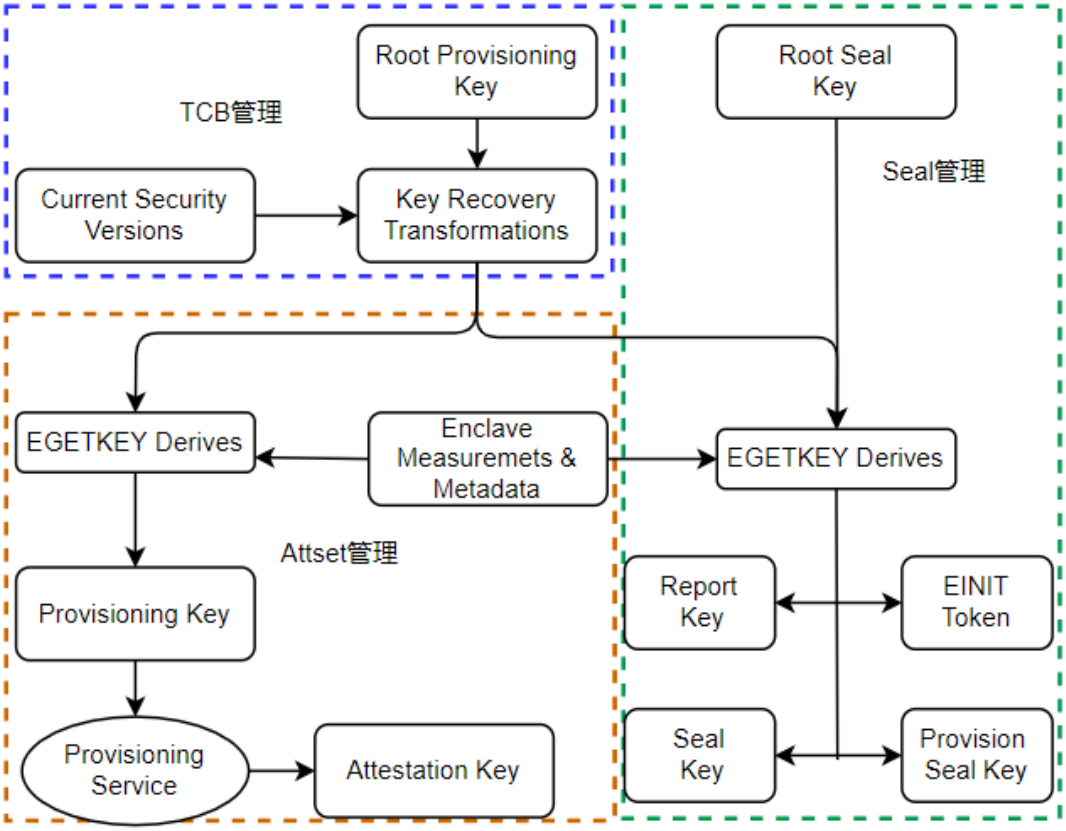

图 1 SGX 密钥体系派生图

在支持 SGX 特性的 CPU 上有两把固化在熔丝中的根密钥，分别为 Root Provisioning Key 和 Root Seal Key。前者首先由 Intel 离线密钥生成设施产生，Intel 和 CPU 制造厂商各保留一份。Intel 用此全球唯一标识此 CPU，CPU 制造厂商将此固化在硬件中。后者由 CPU 制造厂商生产 CPU 时生成并固化在硬件中，Intel 对此根密钥不可见。此根密钥主要解决 enclave 敏感数据的持久化问题。

TCB (Trusted Computing Base)管理部分主要是 TCB Key 的产生与更新，用来表示当前处理器逻辑的版本和可信计算环境的基础信息。关于 enclave 实例的可信证明材料涉及到 Attset 管理部分：在产生 enclave 实例时，通过 EGETKEY 指令产生 Provisioning Key，然后本地内置的 Provisioning enclave 通过 EPID 协议与 Intel Provisioning Service 交互产生 Attestation Key。Attestation Key 作为 EPID 组成员私钥仅仅由此 enclave 实例可知，Intel 不可知。关于 enclave 敏感数据的持久化问题涉及到 Seal 管理部分：在产生 enclave 实例时，通过 EGETKEY 指令派生出如图 4 个密钥。此时 EGETKEY 指令的参数包括 Intel 不可知的

Root Seal Key，进而 Intel 无法获取到这 4 个密钥。Report Key 用于本地不同 enclave 实例之间的相互可信认证，相对地 Attestation Key 用于远程可信认证；Provision Seal Key 用于加密持久化 Attestation Key；Seal Key 用于加密持久化 enclave 内用户敏感数据；EINIT Token 用于 EINIT 指令阶段。

2.2.1 构建基于 enclave 的可信智能合约环境

本部分的构建过程主要分为三个步骤，应用合约的 enclave 远程可信认证材料的生成、基于 IAS(Intel Attestation Service)的可信认证材料的验证、可信认证材料验证结果的广播。具体构建过程如图 2 所示。

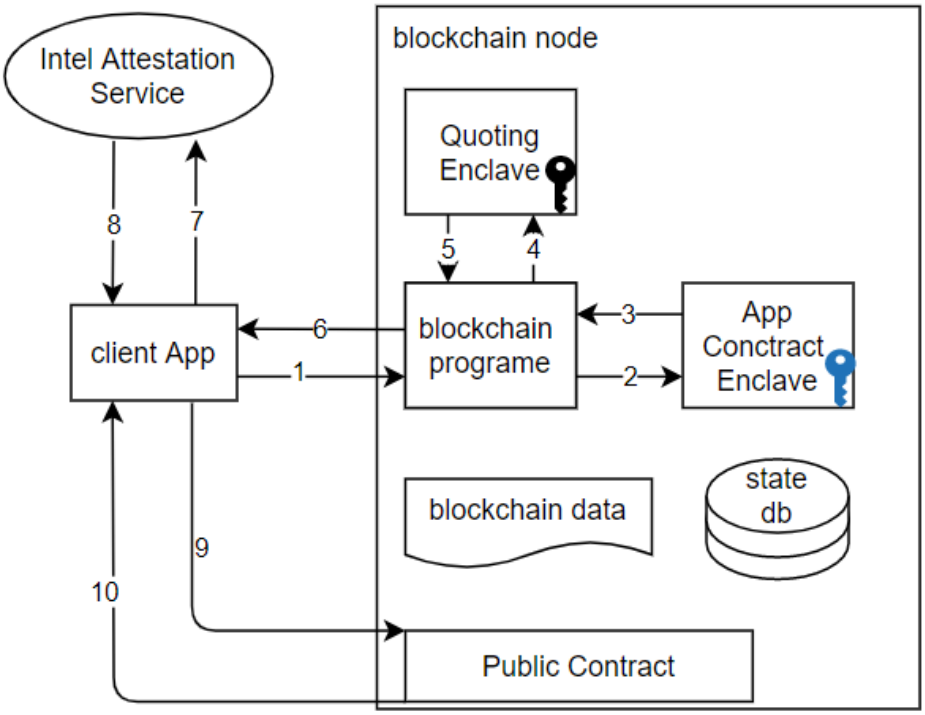

图 2 基于 enclave 的可信智能合约构建过程

图 2 中 1~6 步完成远程认证材料的生成，7~8 步完成基于 Intel IAS 的远程认证材料的验证，9~10 步完成验证结果的广播。整个构建流程不仅使本客户端用户确信其提交到应用智能合约的交易数据是在可信环境中计算的，而且使得同应用场景下的其他交互用户能够验证其证明材料的可信度。具体构建如下：

⑴ 客户端应用发起运行应用合约的请求；

⑵ 区块链程序通过调用 EINIT、ECREATE、EADD 等指令完成应用智能合约的实例化；

⑶ 应用智能合约利用其 enclave 实例的 Report Key 对本地可信认证报告材料签名；

⑷ SGX 内置 Quoting Enclave 对本地可信认证材料进行验证，验证通过后其用 Attestation Key 进行签名，完成本地可信认证报告到远程可信认证报告的转换；

⑸ 远程可信认证报告发送到区块链程序；

⑹ 客户端程序接收区块链程序返回的远程认证报告；

⑺ 客户端程序将远程认证报告发送到 IAS 服务处进行认证，认证内容主要包括远程认证报告中的 SPID 属性值是否匹配客户端用户的 SPID 值、EPID 组属性 GID 值是否过期；然后 IAS 服务利用对应 GID 的组公钥验证此远程认证报告的签名是否有效；

⑻ IAS 服务将验证结果响应给客户端程序；

⑼ 客户端程序将通过验证的报告 enclave 标识信息、验证结果通过公开合约进行发布。由于公开合约中的内容是公开的，即非用户、非 enclave 敏感数据，所以此公开合约可以按照普通的合约运行方式运行。

⑽ 客户端程序可以通过公开合约查询各个 enclave 的标识信息、验证结果，便于不同区块链参与方对 enclave 签名的验证。

通过以上步骤完成硬件可信到智能合约可信的传递与链接。

2.2.2 构建基于可信智能合约环境的可信区块链

本部分主要基于 2.2.1 节构建的可信智能合约环境完成对非 enclave 可信环境中的敏感数据进行加密处理。这些非可信环境的敏感数据包括用户提交的交易数据以及响应、区块链状态数据、持久化到磁盘的 enclave 敏感数据。通过在可信的智能合约“保险箱”中计算、产生这些加密所需的密钥，并且通过 SGX 密钥派生体系中的 Seal Key 完成对这些密钥的持久化存储。这样保证用户敏感数据在运行时、持久化时的隐私性。具体在 enclave 智能合约中产生的密钥如下：

⑴ 用以标识 enclave 实例的签名私钥和验证公钥，此处分别称为 esk 和 evk。当某条交易在某个 enclave 合约实例里运算后，用 esk 签名。验证公钥 evk 在 enclave 合约实例的远程认证报告验证通过后由 public 公开合约发布。

⑵ 用以加密用户交易数据的解密私钥和加密公钥，此处分别称为 csk 和 cpk。加密公钥 cpk 在 enclave 合约实例的远程认证报告验证通过后由 public 公开合约发布。公开合约中维护一个 key-value 映射表，其键值可以为 Hash(evk)，映射值为一个结构体{evk，cpk，meta}。

⑶ 用以加密状态数据的对称密钥，此处称为 sk。考虑到状态数据的读写性能影响，此处采用对称密钥。

在完成“保险箱”中安全产生这些敏感密钥数据、并通过 SGX Seal Key 安全保存这些敏感密钥数据后，我们通过图 3 所示的交易流程进一步阐述整套系统如何利用这些密钥完成敏感数据的保护，进而达到可信区块链的目标。关于不同服务之间的通信加密机制并未在图 3 中表现，默认开启 TLS 通信加密机制。

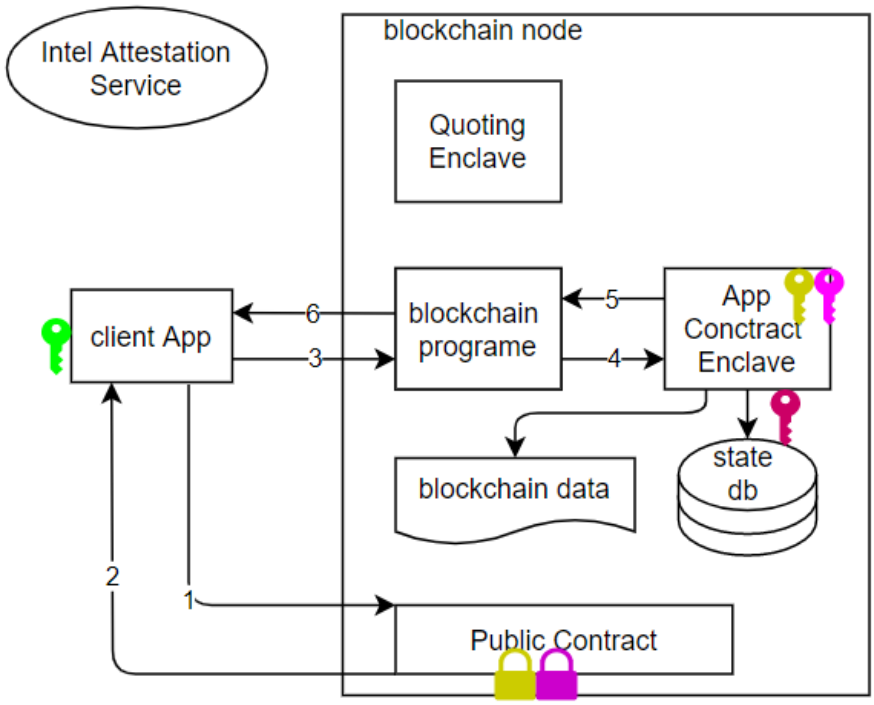

图 3 可信区块链交易流程示意图

图 3 所示为可信区块链系统交易流程示意图，具体如下：

⑴ 客户端应用发起交易请求前首先到 Public 公开合约查询目前可信智能合约的信息，主要是获取到对应的 cpk 公钥；

⑵ 客户端应用根据获取到的 cpk 公钥加密交易敏感数据，包括此请求响应将要用到的对称密钥。因为 2.2.1 节已经证明目标可信智能合约的可信性，所以可以安全地将此对称密钥传到合约 enclave 内部；

⑶ 区块链服务接收交易请求并处理非敏感数据部分；

⑷ 对于响应对称密钥等用户交易敏感数据交由目标可信合约进行可信处理；

⑸ 可信合约在 enclave 内部首先通过 csk 私钥解密交易请求中密文，然后以明文形式在 enclave 内进行高效运算；运算结果以 sk 加密存储到状态数据库、交易数据以密文形式存储在区块体中；同时，运算结果会在 enclave 内用用户响应对称密钥加密、用 esk 进行签名；

⑹ 区块链服务将加密并签名的交易结果转发到客户端；客户端一方面通过用户响应对称密钥解密交易结果，一方面通过公开合约验证此笔交易结果签名的有效性。当然其他参与方也

可通过公开合约验证此笔交易的有效性，即是否被有效可信合约签名。

通过以上步骤完成智能合约可信到区块链系统可信的传递与链接。至此，完成基于 SGX 的区块链闭环可信计算装置的构建。

2.3技术方案带来的有益效果

- ✧ 通过该装置，区块链系统不仅能够提供一致、完整的分布式账本，而且能够高效、安全地提供分布式可信计算能力，保证交易在整个闭环上的隐私性和安全性。特别是在多方协同计算应用场景下，各方都希望各自的数据在不泄露的情况下完成计算与交互。
- ✧ 由于系统整套的信任根固化在硬件中，对于出现拜占庭恶意行为的区块链节点也无法获取到用户交易敏感数据。因此，此方法进一步巩固系统整体的可靠性，为用户隐私数据提供强有力的保障。