关于专利的编写与申请

1. 专利概述

中国专利法规定可以获得专利保护的发明创造有发明、实用新型和外观设计三种，其中发明专利是最主要的一种。我们研究产生的专利属于发明专利。申请专利的做法属于知识产权保护。

发明是发明人运用自然规律而提出解决某一特定问题的技术方案。中国专利法实施细则指出“专利法所称的发明是指对产品、方法或其改进所提出的新的技术方案”。发明人只有将这种技术方案向专利局提出申请，并且通过一系列严格的审查，特别是新颖性、创造性和实用性的审查。

通过审查后，符合规定的发明专利申请会被授予专利权。申请人还应按期办理登记手续和缴纳当年年费，这项发明专利申请才能正式成为一项具有专利多种属性的发明专利。为了鼓励社会对发明创造的，提高发明创造的热情，我国大部分的地方政府对申请专利的个人和企业提供有专项的资金扶持，可到各个地方的科技局查询。

2. 专利点挖掘技巧

挖掘专利点并不难。专利点一般来源于从事科研实践过程中。在特定的技术或应用背景下，通过调研与阅读文献过程中不断地思考、凝练，获取你认为有创新有价值的想法，总结形成一种针对该领域特定问题的比较具体的方法、装置或系统，即一种有效解决方案设计。只要该方法、装置或系统具有至少三个创新点，就可以申请专利。

3. 专利编写

编写专利注意以下几个方面：

1. 专利取名：这个很重要。可以先做一个专利调研，参照一下该领域已经授权的专利的名称命名方式。一般的形式如“一种基于.....的方法”，“一种基于.....的方法与装置”，“一种.....方法与系统”，“一种基于。。。。。。的系统”等。

2. 专利内容结构

专利文档包含：权利要求书、说明书、说明书附图和说明书摘要。重点核心部分就是说明书与附图。该部分是专利的主体。其它部分是总结出来的。

专利说明书的内容包括技术领域、背景技术、发明内容、附图说明、具体实施方式。

1） 技术领域：写一句话即可。形式如“本发明涉及……安全技术领域，尤其涉及一种#########方法及系统。”

2） 背景技术：介绍发明涉及的创新方法的相关技术支撑。

3） 发明内容：介绍具体发明包含的内容构成，即发明是什么、具体过程怎样，所包含模块构成及具体描述，配以图文方式，后续要有附图说明。

4） 附图说明：对所涉及的图做一个整体说明。

5） 具体实施方式：举例介绍本方法具体实施中的一些注意细节。

4. 专利申请

专利稿子写完后，提交给专利代理机构，由代理机构完成后续向国家专利局的手续。学校指定的专利代理机构有两个，一个是天勤知识产权，一个是求是专利事务局。学校专利提交专利代理机构之前要在科研系统中申请，学院科研科确认，学校科研院审核后获得校内专利号，这个号报给专利代理机构。学校申请专利不用交钱，待到授权后，再叫专利代理费 1390 元（这个费用不用同学们交）。专利年费前三年（这个数据待确认）由学校代缴。三年后需要由实验室自己付每年年费大概就 3-400 元。

5. 专利样例

以下给出一个专利编写稿与两个最终提交的专利样板，供参考。这里给的样板是

专利代理机构处理后的格式。我们不需要严格按照这个来写。我们只要把类似说明书部分的内容写清楚，其他内容由代理机构来完成。

即：我们重点写清楚四个部分：发明背景与现有技术、本发明技术方案、本发明的技术保护点及本发明的有益效果。另外可以做一下检索情况（选做）和专利分析。其他内容，在你提交给专利代理机构之后，他们会帮我们弄好的。

以下给出一个编写稿样板，见“专利样板-一种“为大留大”异构资源的虚拟机在线调度方法与装置.pdf”。

附

专利 1：**Openstack 令牌访问保护机制的实现方法及系统**

专利 2：一种基于 SGX 的区块链用户密钥保护方法和装置

专利样例一

Openstack 令牌访问保护机制的实现方法及系统

权　利　要　求　书

1、一种 Openstack 令牌访问保护机制的实现方法，其特征在于，按照 memcache 的存储模式将 Openstack 的 token 表存储于 memcache 中，再利用软件防护扩展指令的保护机制对 token 表进行加密。

2、根据权利要求 1 所述的 Openstack 令牌访问保护机制的实现方法，其特征在于，包括以下步骤：

（1）将 token 表存储于 memcache 中，通过软件防护扩展指令为 memcache 分配可信空间，并生成用以验证可信空间访问权限的密钥；

（2）每次更新 token 表时，memcache 更新数据后，向 SGX 驱动器发起数据更新请求，通过密钥的验证后，将更新数据备份到可信空间中。

3、根据权利要求 2 所述的 Openstack 令牌访问保护机制的实现方法，其特征在于，将 token 表存储于 memcache 中，包括以下步骤：

（a）编辑/etc/keystone/keystone.conf 的 token 字段：

driver = keystone.token.backends.memcache.Token，将 token 字段的驱动修改为 memecache；

（b）重启 keystone，并启动 memcache，通过 memcache 对分布式存储的 token 表进行管理。

4、根据权利要求 2 或 3 所述的 Openstack 令牌访问保护机制的实现方法，其特征在于，步骤（1）中，通过软件防护扩展指令为 memcache 分配可信空间，并生成用以验证可信空间访问权限的密钥，具体包括：

（1-1）数据上载： 生成 memecache 的证书，将 memecache 和其证书上载到处理空间中；

（1-2）SGX 驱动器准备：通过 SGX 驱动器对上载的 memecache 和其证书进行参数测量，为可信空间分配地址空间和内存页，同时获取 memecache 的证书信息并传递给 SGX 硬件处理器；

（1-3）可信空间的建立：SGX 驱动器根据测量的参数创建可信空间，并将 memecache 上的数据信息复制到可信空间中，之后删除处理空间中的数据；

（1-4）密钥的生成：SGX 硬件处理器根据 memecache 的证书信息和 SGX 硬件处理器自身的特征数据生成可信空间的访问密钥，并通过密钥对可信空间进行加密。

5、根据权利要求 4 所述的 Openstack 令牌访问保护机制的实现方法，其特征在于，memecache 应用的证书信息包括 memecache 应用证书的哈希值和私钥。

6、一种 Openstack 令牌访问保护机制的系统，其特征在于，包括：

memcache 存储模块，将以 openstack 的存储方式进行存储的 token 表进行分布式存储并以 memcache 作为驱动；

SGX 加密模块，基于软件防护扩展指令生成可信空间，用以存储、操作 memcache 存储模块中的 token 数据，并生成用以验证可信空间访问权限的密钥。

说　　明　　书

Openstack 令牌访问保护机制的实现方法及系统

技术领域

本发明涉及云计算运行和存储过程中的安全技术领域，尤其涉及一种 Openstack 令牌访问保护机制的实现方法及系统。

背景技术

Openstack是一个开源的云计算管理平台项目，允许企业或服务提供者创建、运行自己的云计算和存储设施，具体包含五个重要构成部分：Nova（计算服务），Swift（存储服务），Glance（镜像服务），Keystone（认证服务）和Horizon（UI服务）。其中，Keystone为所有的Openstack组件提供认证和访问策略服务，它依赖自身REST（基于Identity API）系统进行工作，主要对（但不限于）Swift、Glance、Nova等进行认证与授权，通过对动作消息来源者请求的合法性进行鉴定。

Keystone采用两种授权方式，一种基于用户名/密码，另一种基于令牌（token）。因为用户名，密码以及tenant名更为直观，所以对于终端用户来说，很少会直接用Token进行操作，但对于自动化测试等操作来说，需要直接调用Openstack的各项api（应用程序编程接口），大量命令都依赖相关用户的token来完成，因此，获得用户的token意味着获得Openstack各项api的授权。

然而，每次管理Openstack的过程中都会产生一个新的token进行验证，使得Keystone库的token表可以增长至几十甚至上百GB，对于之后的数据库备份造成不便。为了更好地管理token表，常用的解决方案之一是将token存储于memecached中。

Memcache是一个高性能的分布式内存对象缓存系统，用于动态Web应用以减轻数据库负载。它通过在内存中缓存数据和对象来减少读取数据库的次数，从而提高动态、数据库驱动网站的速度。这是一套开放源代码软件，以BSD license授权协议发布。

然而Memcache在自身实现中缺乏足够的安全机制，使得数据可能被

未授权的用户访问或截获，因此当其应用于token存储过程中可能导致数据泄露。

软件防护扩展指令(Software Guard Extensions，SGX) 是Intel开发的新的处理器技术，可以在计算平台上提供一个可信的空间，将安全应用依赖的可信计算基TCB减小到仅包含CPU和安全应用本身，将不可信的复杂操作系统OS和虚拟机监控器VMM排除在安全边界之外，从而保障用户关键代码和数据的机密性和完整性。这种方式并不是识别和隔离平台上的所有恶意软件，而是将合法软件的安全操作封装在一个enclave（可信空间）中，保护其不受恶意软件的攻击，特权或者非特权的软件都无法访问enclave，也就是说，一旦软件和数据位于enclave中，即便操作系统或者和VMM（Hypervisor）也无法影响enclave里面的代码和数据。Enclave的安全边界只包含CPU和它自身。

发明内容

针对 memcache 存储令牌（token）表过程中缺乏安全保障的技术不足，本发明提供了一种基于软件防护扩展指令（Software Guard Extensions，SGX）的 Openstack 令牌访问保护机制的实现方法及系统，提高以 memcache 存储的 token 表的安全性。

一种 Openstack 令牌访问保护机制的实现方法，按照 memcache 的存储模式将 Openstack 的 token 表存储于 memcache 中，再利用软件防护扩展指令的保护机制对 token 表进行加密。

Openstack 是一中开源的云计算管理平台项目，允许企业或服务提供者创建、运行自己的云计算和存储设施；memcache 是一中高性能的分布式内存对象缓存系统；token 指 Openstack 的令牌。

在本发明的方法中，在保护openstack令牌时，通过SGX机制由计算机硬件对token信息进行加密，对其访问权限进行控制，使得只能在指定的物理资源（服务器等）上进行数据的读取和修改，从而保证了 token 信息的安全性。

作为优选，Openstack 令牌访问保护机制的实现方法，包括以下步骤：

（1）将 token 表存储于 memcache 中，通过软件防护扩展指令为 memcache 分配可信空间，并生成用以验证可信空间访问权限的密钥；

（2）每次更新 token 表时，memcache 更新数据后，向 SGX 驱动器发起数据更新请求，通过密钥的验证后，将更新数据备份到可信空间中。

进一步优选的，将 token 表存储于 memcache 中，包括以下步骤：

（a）编辑/etc/keystone/keystone.conf 的 token 字段：

driver = keystone.token.backends.memcache.Token，将 token 字段的驱动修改为 memecache；

（b）重启 keystone，并启动 memcache，通过 memcache 对分布式存储的 token 表进行管理。

进一步优选的，步骤（1）中，通过软件防护扩展指令为 memcache 分配可信空间，并生成用以验证可信空间访问权限的密钥，具体包括：

（1-1）数据上载： 生成 memecache 的证书，将 memecache 和其证书上载到处理空间中；

（1-2）SGX 驱动器准备：通过 SGX 驱动器对上载的 memecache 和其证书进行参数测量，为可信空间分配地址空间和内存页，同时获取 memecache 的证书信息并传递给 SGX 硬件处理器；

（1-3）可信空间的建立：SGX 驱动器根据测量的参数创建可信空间，并将 memecache 上的数据信息复制到可信空间中，之后删除处理空间中的数据；

（1-4）密钥的生成：SGX 硬件处理器根据 memecache 的证书信息和 SGX 硬件处理器自身的特征数据生成可信空间的访问密钥，并通过密钥对可信空间进行加密。

SGX 为软件防护扩展指令的缩写。

memecache 的证书信息包括 memecache 证书的哈希值和私钥。

本发明还提供了一种 Openstack 令牌访问保护机制的系统，包括：

memcache 存储模块，将以 openstack 的存储方式进行存储的 token 表进行分布式存储并以 memcache 作为驱动；

SGX 加密模块，基于软件防护扩展指令生成可信空间，用以存储、操作 memcache 存储模块中的 token 数据，并生成用以验证可信空间访问权限的密钥。

所述的 SGX 加密模块包括用户空间、SGX 驱动器和 SGX 硬件处理器，

用户空间，包括用于加载 memecache 和其证书的处理空间以及用于为 memecache 分配的可信空间；

SGX 驱动器，对 memecache 进行参数测量并为其分配可信空间，同时获取 memecache 的证书信息并将其传递给 SGX 硬件处理器；

SGX 硬件处理器，对 memecache 的证书和可信空间的完整性进行验证，根据 memecache 的证书的哈希值和其自身特征数据的哈希值生成可信空间的访问密钥，并通过密钥对可信空间进行加密。

所述的 SGX 驱动器属于操作系统；SGX 硬件处理器属于硬件构架。

密钥是由客户 memecache 和物理机硬件信息交叉生成，保证了后续验证步骤的安全性和有效性。

与现有技术相比，本发明的有益效果为：

通过 SGX 机制由计算机硬件对 token 信息进行加密，对其访问权限进行控制，使得只能在指定的物理资源（服务器等）上对 token 信息进行读取和修改，从而保证了

token 信息的安全性。

附图说明

图 1 为本发明的 Openstack 令牌访问保护机制的实现方法的流程控制示意图；

图 2（a）为数据上载阶段工作原理示意图；

图 2（b）为软件防护扩展指令驱动器准备阶段工作原理示意图；

图 2（c）为可信空间建立阶段工作原理示意图；

图 2（d）为密钥生成阶段工作原理示意图。

具体实施方式

下面结合附图和实施例对本发明作进一步详细描述。

本发明通过 2 个软件模块实现：memcache 存储模块以及 SGX 加密模块，其流程控制如图 1 所示。

memcache 存储模块的作用是将以 openstack 的存储方式进行存储的 token 表进行分布式存储，并以 memcache 作为驱动。具体步骤如下：

（1）编辑/etc/keystone/keystone.conf 的 Token 字段：

driver = keystone.token.backends.memcache.Token，将其驱动修改为 memecache；

（2）重启 keystone，并启动 memcache，通过 memcache 对分布式存储的 token 表进行管理。

SGX 加密模块的作用是生成可信空间以存储、操作相应数据，并生成用以验证访问权限的密钥。其工作原理具体如下：

（1）数据上载阶段：如图 2（a）所示，创建 memcache 并生成其证书，其中，memcache 证书信息包括其哈希值和私钥，并将 memcache 和证书上载到处理空间中；

（2）SGX 驱动器准备阶段：如图（b）所示，SGX 驱动器对上载数据进行参数测量，用以为可信空间分配地址空间和内存页，同时 SGX 驱动器获取 memcache 生成的证书信息并将其传递给底层 SGX 硬件处理器；

（3）可信空间数据建立阶段：如图（c）所示，SGX 驱动器将根据对 memcache 进行的参数测量，创建可信空间，并将 memcache 上数据信息复制到可信空间中，之后删除处理空间中的数据。通过 SGX 硬件处理器对证书和可信空间的完整性进行验证；

（4）密钥生成阶段：如图（d）所示，SGX 硬件处理器根据证书中哈希值和 SGX 硬件处理器自身特征数据的哈希值生成可信空间访问密钥，并通过密钥对可信空间进行加密，之后要访问可信空间中的数据必须获得此密钥，从而使得可信空间中存储的 memcache 数据得到保护。

每次更新 token 表时，memcache 更新数据后，向 SGX 驱动器发起数据更新请求，通过密钥的验证后，将更新数据备份到可信空间中。

说　明　书　附　图

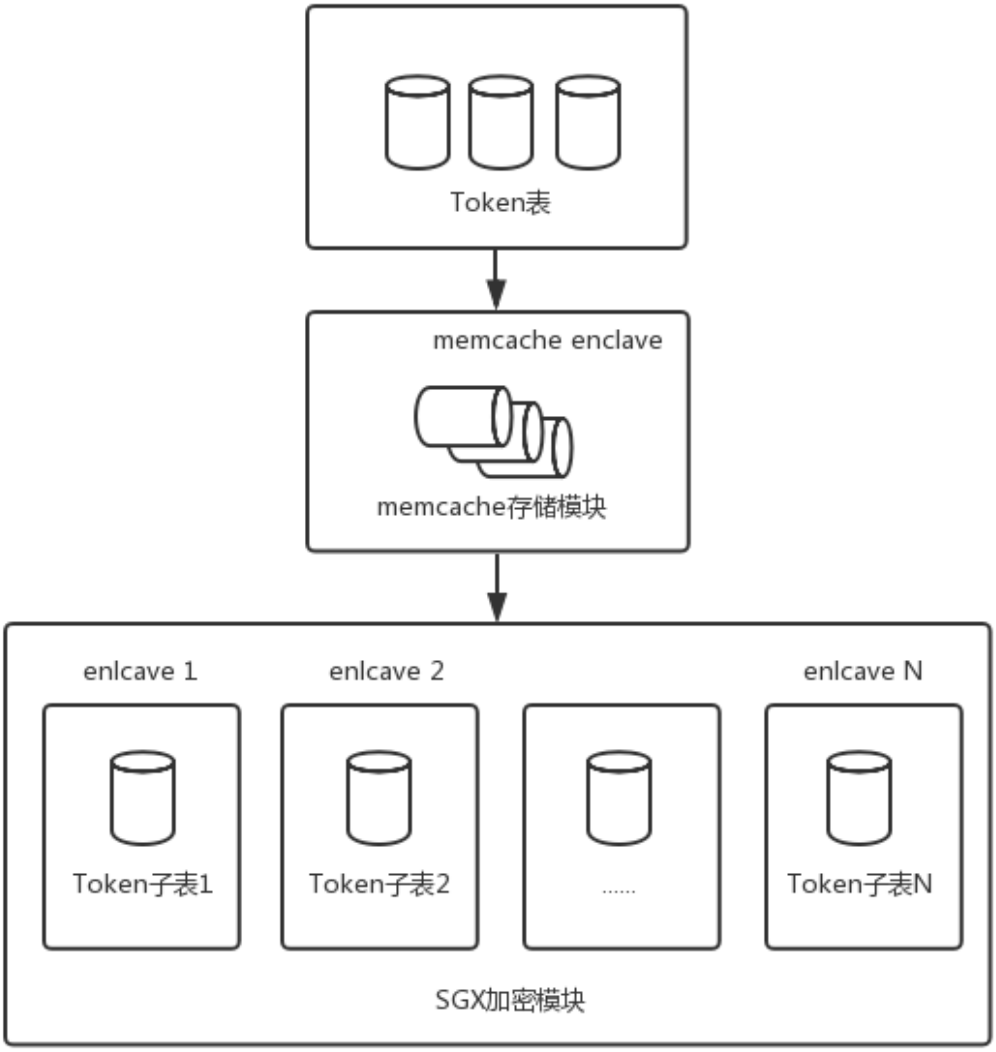

Token表
memcache enclave
memcache存储模块
enlcave 1
enlcave 2
enlcave N
Token子表1
Token子表2
.....
Token子表N
SGX加密模块

图 1

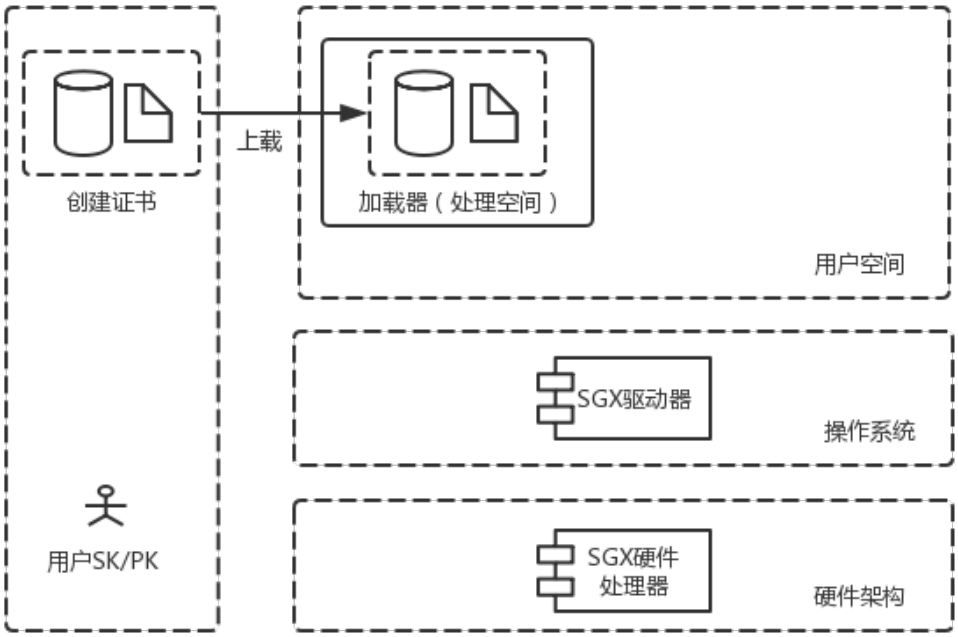

上载
创建证书
加载器（处理空间）
用户空间
SGX驱动器
操作系统
用户SK/PK
SGX硬件
处理器
硬件架构

图 2（a）

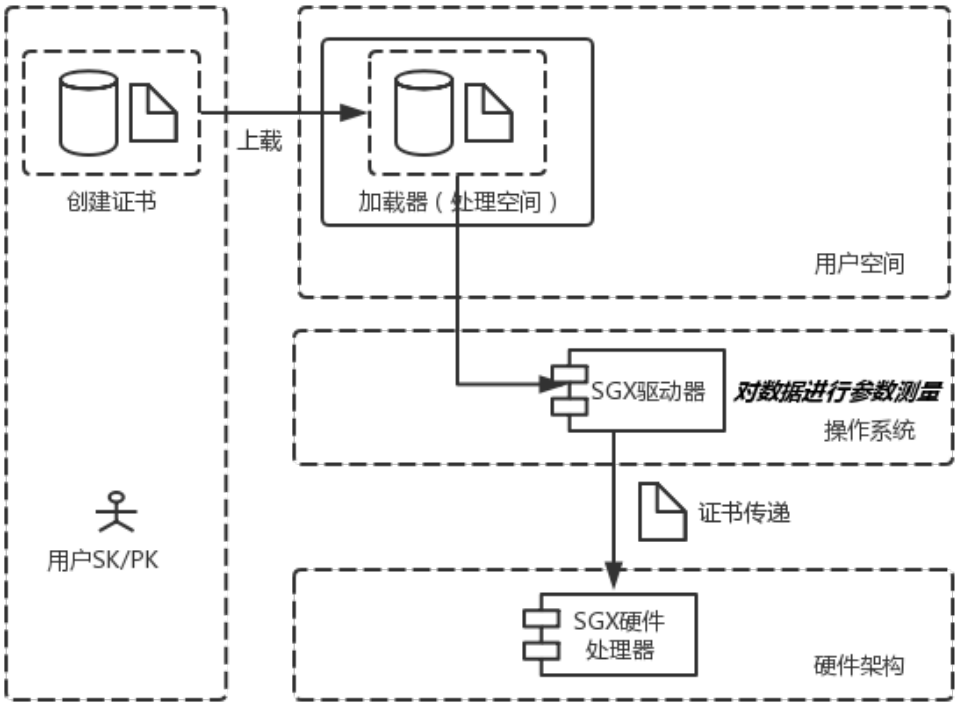

上载
创建证书
加载器（处理空间）
用户空间
SGX驱动器
对数据进行参数测量
操作系统
证书传递
用户SK/PK
SGX硬件
处理器
硬件架构

图 2（b）

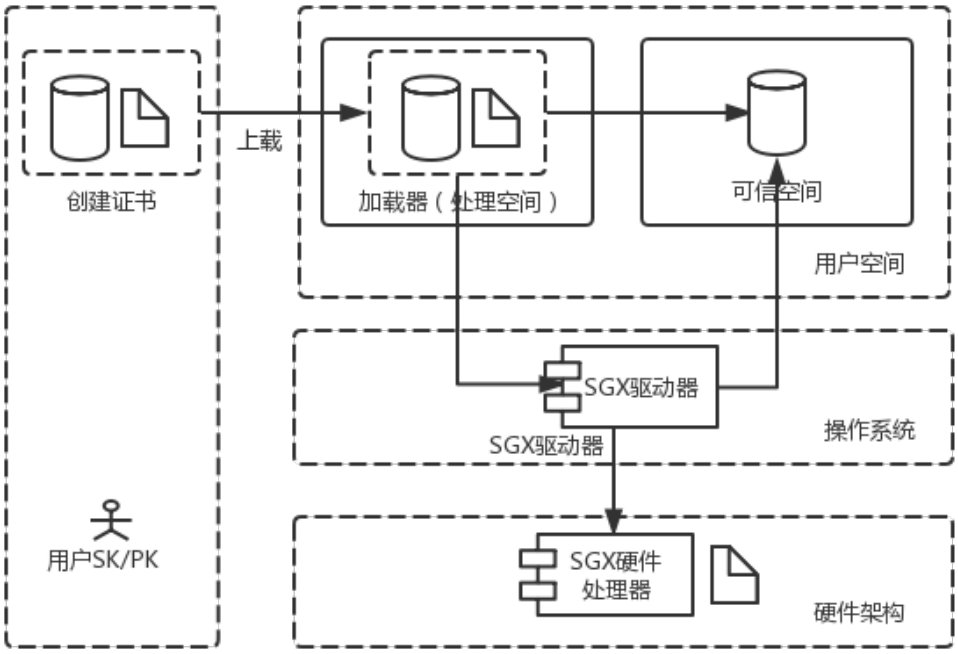

创建证书
上载
加载器（处理空间）
可信空间
用户空间
SGX驱动器
SGX驱动器
操作系统
用户SK/PK
SGX硬件
处理器
硬件架构

图 2（c）

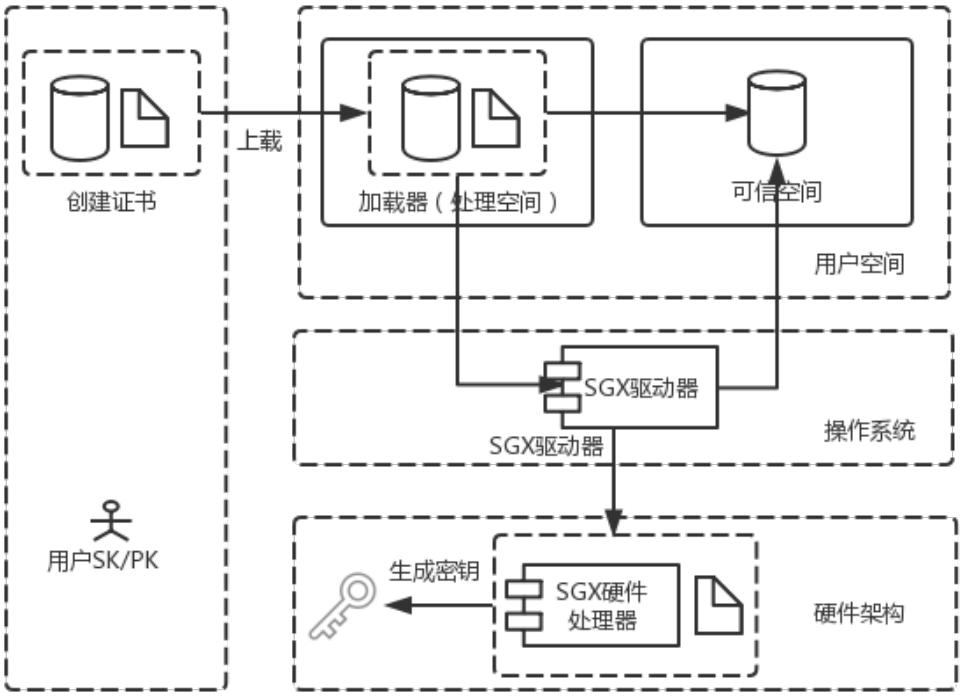
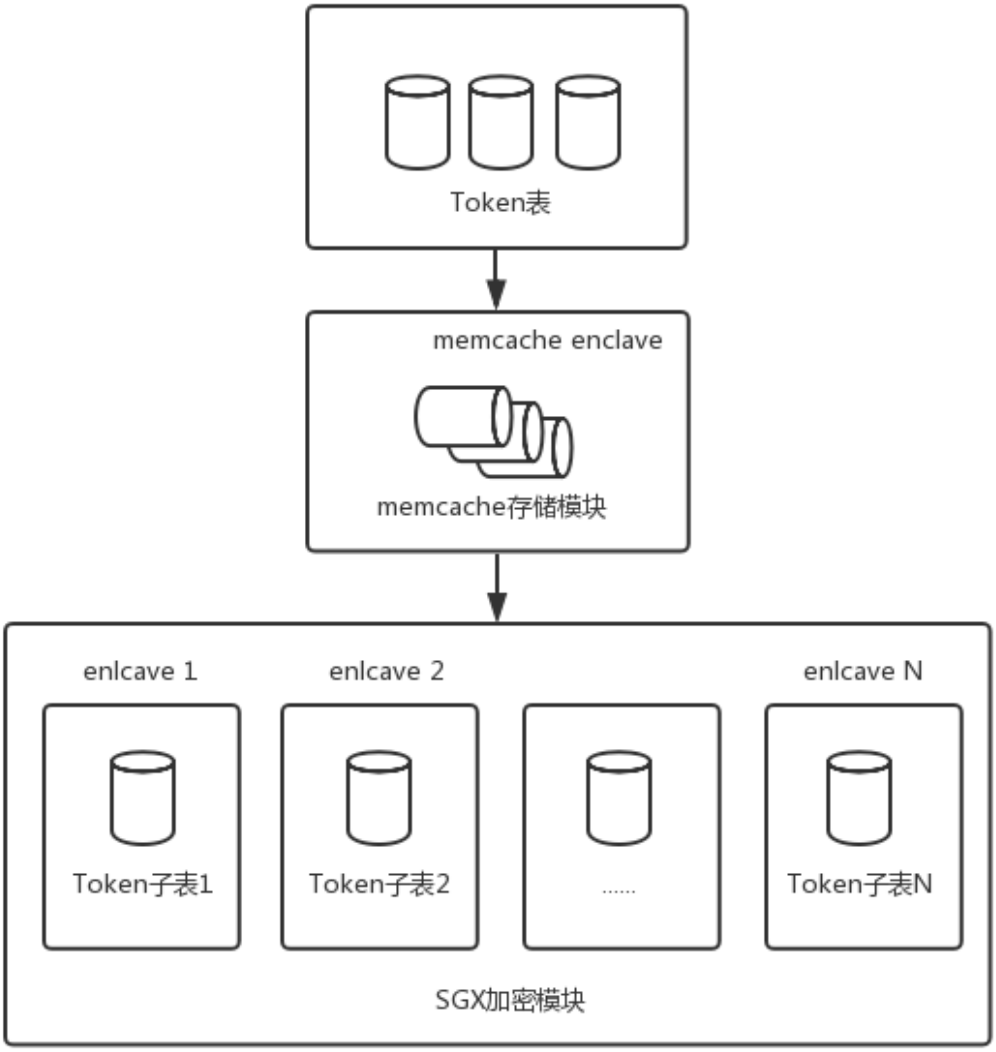

创建证书
上载
加载器（处理空间）
可信空间
用户空间
SGX驱动器
SGX驱动器
操作系统
用户SK/PK
生成密钥
SGX硬件
处理器
硬件架构

图 2（d）

说　明　书　摘　要

本发明公开了一种 Openstack 令牌访问保护机制的实现方法及系统，其中方法为：按照 memcache 的存储模式将 Openstack 的 token 表存储于 memcache 中，再利用软件防护扩展指令的保护机制对 token 表进行加密。通过 SGX 机制由计算机硬件对 token 信息进行加密，对其访问权限进行控制，使得只能在指定的物理资源（服务器等）上对 token 信息进行读取和修改，从而保证了 token 信息的安全性。

摘　要　附　图


Token表
memcache enclave
memcache存储模块
enlcave 1
enlcave 2
enlcave N
Token子表1
Token子表2
.....
Token子表N
SGX加密模块

专利样例二

权　利　要　求　书

1、一种基于 SGX 的区块链用户密钥保护装置，其特征在于，包括：

SGX 加密模块，基于软件防护扩展指令生成可信空间，并生成用于验证所述可信空间访问权限的访问密钥；所述可信空间用于存储区块链网络的用户密钥和密钥操作函数；

交易共识模块，接收来自区块链网络的交易，通过所述访问密钥访问 SGX 加密模块，调用其中的密钥操作函数，实现对交易的验证共识；

交易构造模块，根据用户的意图发起交易，通过所述访问密钥访问 SGX 加密模块，调用其中的密钥操作函数，实现交易信息的填充与合法化，并向区块链网络广播该交易。

2、根据权利要求 1 所述的区块链用户密钥保护装置，其特征在于，所述的 SGX 加密模块包括：

用户空间，包括处理空间和可信空间；所述处理空间用于加载用户密钥和密钥操作函数的证书信息，所述可信空间用于存储用户密钥和密钥操作函数；

SGX 驱动器，通过对用户密钥和密钥操作函数的证书信息进行测量，为所述的用户密钥和密钥操作函数分配可信空间，将用户密钥和密钥操作函数的证书信息传递给 SGX 硬件处理器；

SGX 硬件处理器，对用户密钥和密钥操作函数证书信息和可信空间的完整性进行验证，根据用户密钥和密钥操作函数证书的哈希值和 SGX 硬件处理器特征数据的哈希值生成可信空间的访问密钥，通过访问密钥对可信空间进行加密。

3、一种基于 SGX 的区块链用户密钥保护方法，其特征在于，包括：

（1）获取用户密钥并存入 SGX 加密模块；

（2）处理相关的密钥操作，包括处理由区块链网络传入交易引起的密钥操作和由用户发起交易引起的密钥操作。

4、根据权利要求 3 所述的区块链用户密钥保护方法，其特征在于，步骤（1）包括：

（1-1）以安全途径获取区块链认可的用户密钥；

（1-2）对用户密钥进行备份；

（1-3）从用户密钥生成公钥并存储于本地；

（1-4）将用户密钥存入 SGX 加密模块。

5、根据权利要求 4 所述的区块链用户密钥保护方法，其特征在于，步骤（1-4）包括：

（a）生成用户密钥和密钥操作函数的证书，将用户密钥和密钥操作函数与所述证

书一起上载到处理空间中；

所述密钥操作函数包括密钥验证函数和密钥签名函数；

（b）通过 SGX 驱动器对上载的用户密钥、密钥操作函数以及其证书进行参数测量，为可信空间分配地址空间和内存页，创建可信空间，并将用户密钥和密钥操作函数复制到可信空间中，之后删除处理空间中的数据；

（c）SGX 驱动器获取用户密钥和密钥操作函数的证书信息并传递给 SGX 硬件处理器；SGX 硬件处理器根据证书信息的哈希值以及 SGX 硬件处理器自身的哈希值生成可信空间的访问密钥，通过访问密钥对可信空间进行加密；

（d）重复步骤（a）～（c），建立多个可信空间，将备份的多份用户密钥分别存入不同的可信空间中。

6、根据权利要求 3 所述的区块链用户密钥保护方法，其特征在于，步骤（2）中，处理由区块链网络传入交易引起的密钥操作包括以下步骤：

（ⅰ）客户端接收区块链网络传入的交易信息；

（ⅱ）调用 SGX 安全区外的验证函数对交易中不需要用户密钥的部分进行验证，计算交易的优先级并进行排序；

该部分的验证限于不需要用户密钥的验证，例如验证交易的版本是否正确、交易资产量是否合法等。同时视具体区块链的架构选择性完成交易的优先级计算与排序等操作。

（ⅲ）客户端向 SGX 驱动器发起请求，通过访问密钥验证后，调用某一可信空间内的密钥验证函数对交易中需要用户密钥的部分进行验证；

需要用户密钥的部分的验证限于需要用户密钥的验证，例如确定交易目标是否为自己等。

（ⅳ）检查对密钥验证函数的调用结果，若调用结果不正常，则调用另一个可信空间内的密钥验证函数，直至调用结果正常；之后将验证信息打包入验证结果，完成交易的验证；

（ⅴ）根据验证结果向区块链网络发送反馈；

若验证成功，则向区块链网络广播该交易信息，供其余用户节点继续验证；

若验证失败，则停止传播该交易，或向区块链网络反馈交易非法。

7、根据权利要求 3 所述的区块链用户密钥保护方法，其特征在于，步骤（2）中，处理由用户发起交易引起的密钥操作包括以下步骤：

（Ⅰ）客户端在 SGX 安全区外构建交易；

该步骤中，客户端需要按照相应区块链的规则合法构造交易，为交易补充目标地址、交易额等必要信息；同时在该步，客户端无需提供自己的签名信息；

（Ⅱ）客户端向 SGX 驱动器发起请求，通过访问密钥验证后，调用某一可信空间内的密钥签名函数获取用户密钥签名；

（Ⅲ）检查对密钥签名函数的调用结果，若调用结果不正常，则调用另一个可信

空间内的密钥签名函数，直至调用结果正常；之后将签名信息打包入交易信息，完成交易的构建；

（Ⅳ）向区块链网络广播交易。

说　　明　　书

一种基于 SGX 的区块链用户密钥保护方法和装置

技术领域

本发明涉及区块链应用领域，尤其涉及一种基于 SGX 的区块链用户密钥保护方法和装置。

背景技术

云计算、互联网及大数据等新兴技术的发展，使得传统中心化系统信息数据的共享需求日益迫切。然而，中心化系统缺乏一种安全可信的机制，一直是现实的痛点。随着比特币的兴起和应用深入，底层区块链技术为解决现实痛点问题提供了有效手段，近年来引起了广泛关注。区块链技术作为比特币底层技术与基础架构而诞生，其本质上是一个去中心化的记账系统，类似一个分布式共享账本。新型区块链系统采用P2P网络技术、密码学、共识算法、智能合约和分布式数据库等技术，具有数据不可篡改、系统集体维护、信息公开透明等传统账本系统望尘莫及的特性，这使之成为一种极具潜力的、去中心化的资产管理工具与技术。

以比特币为代表的公有区块链，使得区块链具有发行价值资产的重要特性。用户密钥是区块链系统中的重要数据。密钥的安全是用户权益的重要保障。在区块链系统中，数字资产（如比特币）的所有权是通过数字密钥和数字签名来确立的。数字密钥实际上并不存储于区块链网络，而是由各个用户负责保管，对他人保密。区块链的安全性正是依赖于密钥的分散控制。同时，区块链的去中心化信任与控制、所有权认证等特性也都是基于现代密码学的密钥机制实现的。按照密码学原理，只有有效的密钥才能生成有效的数字签名，只有有效的数字签名才能使链上的交易有效。这从根本上杜绝了区块链上伪造交易的可能，保障了用户的利益。

然而，如果存在第三方拥有一个用户账户的密钥或其副本，则该第三方用户就能实际控制对应于该密钥的账户资产。因此，对于一个用户来说，一旦遗失或损坏密钥，用户可能会彻底失去自己对应的全部资产，并且没有任何找回手段。对于一个健壮的区块链网络，单个用户在密钥管理上的

疏漏并不会影响整个网络的稳定，但用户本人却要承担全部后果。此外，现代通用操作系统并不十分安全，亦不适合以文件形式存储密钥信息。特别随着互联网技术的普及，我们的电脑长时间通过互联网暴露在外，期间恶意软件或程序可以隐秘而便捷地盗取本地数据；或由于操作者的疏忽，使恶意软件混入并安装于本地，进而威胁到所有重要的本地数据。

为保护密钥，主要有两种常用方法。第一种是将密钥加密后存储。这是一种常见的安全手段，不过该方法的效果受具体加密算法与手段的影响。因为将数据简单加密并存于操作系统本地，恶意软件仍有机会持续访问，并可以在期间尝试解密并获取密钥，复杂的恶意软件甚至可以直接从内存中提取机密数据，威胁密钥安全；过于复杂的加密方法则为用户的维护带来极大负担，甚至可能因遗忘或丢失部分加密细节而“遗失”密钥。另一种相对更为安全的方法是使用物理存储，即将密钥或加密的密钥信息记录于纸、塑料、金属等物理媒介，并备份多个分别加以安全保存（如锁进保险箱）。但是，这种方法会导致使用密钥的便捷性下降。因为一旦密钥的使用频率上升（特别是对部分专业用户，或是区块链应用本身交易频繁），用户便不得不一一处理繁琐的写入操作。

为兼顾安全性与可用性，专业级的硬件防篡改“钱包”技术应运而生。不像易于受到攻击的常用软、硬件，硬件钱包只提供非常有限的接口，甚至内置密钥的生成系统及相关验证程序，对外只输出私钥的签名，从而为非专业用户提供几近万无一失的安全等级，同时也具有相当的易操作性。为规避损坏与遗失的风险，硬件钱包通常十分坚固，并为用户提供物理备份私钥的途径。不过其高度专业化的架构往往需要与具体的区块链应用严密对接，从而使之难以设计成一款通用设备。目前并没有泛型区块链密钥存储硬件，仅有针对特定区块链应用（如比特币）的专业硬件钱包（如Trezor）。

英特尔SGX（Software Guard Extensions）是一套 CPU 指令，可支持应用创建安全区（enclave）：应用地址空间中受保护的区域，它可确保终端操作系统环境上信息内容的机密性和完整性。试图从软件角度访问enclave的内存内容是不被允许的，即使是高级特权软件（如主操作系统，虚拟机监控器等）都不允许访问。enclave的安全边界只包含CPU和它自身。

SGX创建的enclave也可以理解为一个可信执行环境TEE。SGX技术中一个CPU可以运行多个安全enclaves，支持并发执行。

发明内容

本发明提供了一种基于 SGX 的区块链用户密钥保护装置，在区块链应用系统的客户端节点引入 SGX，该区块链用户密钥保护装置与各式区块链网络相互兼容，可作为区块链网络的节点管理组件，为区块链网络用户提供密钥保存与验证服务。

一种基于 SGX 的区块链用户密钥保护装置，包括：

SGX 加密模块，基于软件防护扩展指令生成可信空间，并生成用于验证所述可信空间访问权限的访问密钥；所述可信空间用于存储区块链网络的用户密钥和密钥操作函数；

交易共识模块，接收来自区块链网络的交易，通过所述访问密钥访问 SGX 加密模块，调用其中的密钥操作函数，实现对交易的验证共识；

交易构造模块，根据用户的意图发起交易，通过所述访问密钥访问 SGX 加密模块，调用其中的密钥操作函数，实现交易信息的填充与合法化，并向区块链网络广播该交易。

优选的，所述的 SGX 加密模块包括：

用户空间，包括处理空间和可信空间；所述处理空间用于加载用户密钥和密钥操作函数的证书信息，所述可信空间用于存储用户密钥和密钥操作函数；

SGX 驱动器，通过对用户密钥和密钥操作函数的证书信息进行测量，为所述的用户密钥和密钥操作函数分配可信空间，将用户密钥和密钥操作函数的证书信息传递给 SGX 硬件处理器；

SGX 硬件处理器，对用户密钥和密钥操作函数证书信息和可信空间的完整性进行验证，根据用户密钥和密钥操作函数证书的哈希值和 SGX 硬件处理器特征数据的哈希值生成可信空间的访问密钥，通过访问密钥对可信空间进行加密。

所述的 SGX 驱动器属于操作系统；SGX 硬件处理器属于硬件构架。

可信空间的访问密钥是由用户的区块链网络用户密钥、密钥操作函数和 SGX 硬件处理器的物理硬件信息交叉生成，保证了相关验证步骤的安全性和有效性。

本发明还提供了一种基于 SGX 的区块链用户密钥保护方法，该方法通过引入 Intel 的 SGX 硬件到区块链网络用户的客户端节点，利用 SGX 的 enclave（可信空间）机制，在本地构建安全的用户密钥存储空间与相应的存取操作，实现本地用户密钥的安全存储与使用。

一种基于 SGX 的区块链用户密钥保护方法，包括：

（1）获取用户密钥并存入 SGX 加密模块；

（2）处理相关的密钥操作。

步骤（1）包括：

（1-1）以安全途径获取区块链认可的用户密钥；

对于需要用户自主生成用户密钥的情况，生成用户密钥的步骤应在安全的设备上（如从未联网的安全 PC）由密码学完备的种子与算法按照规定格式生成。

对于需要区块链网络颁发的用户密钥，则通过严格保护的通信渠道，以安全的加密方式获取；或由线下的安全方式获取。

（1-2）对用户密钥进行备份；

为规避日后由于各种原因导致密钥遗失的风险，无论采取何种密钥保管方式，都建议用户对获取的用户密钥预先进行安全的备份。建议采用物理存储或离线安全设备存储等方式对用户密钥进行备份；并备份多份，分别予以安全存储。

（1-3）从用户密钥生成公钥并存储与本地；

一般情况下，用户需由用户密钥生成公钥，用公钥生成自己的区块链地址（如比特币），以便他人指定自己为交易目标。故而需要在存储用户密钥前执行该步骤以生成与用户密钥对应的公钥，存储于本地。

公钥生成的地址可向全网公开，无需加密。对于由区块链网络负责处理用户公钥与地址的区块链系统（如各用户可向区块链网络询问每个用户的地址），该步骤可由区块链网络完成。

（1-4）将用户密钥存入 SGX 加密模块。

步骤（1-4）包括：

（a）生成用户密钥和密钥操作函数的证书，将用户密钥和及密钥操作函数与所述证书一起上载到处理空间中；

所述密钥操作函数包括密钥验证函数和密钥签名函数；

（b）通过 SGX 驱动器对上载的用户密钥、密钥操作函数以及其证书进行参数测量，为可信空间分配地址空间和内存页，创建可信空间，并将用户密钥和密钥操作函数复制到可信空间中，之后删除处理空间中的数据；

（c）SGX 驱动器获取用户密钥和密钥操作函数的证书信息并传递给 SGX 硬件处理器；SGX 硬件处理器根据证书信息的哈希值以及 SGX 硬件处理器自身的哈希值生成可信空间的访问密钥，通过访问密钥对可信空间进行加密；

（d）重复步骤（a）~（c），建立多个可信空间，将备份的多份用户密钥分别存入不同的可信空间中。

步骤（2）中，处理相关的密钥操作包括处理由区块链网络传入交易引起的密钥操作和由用户发起交易引起的密钥操作。

处理由区块链网络传入交易引起的密钥操作包括以下步骤：

（ⅰ）客户端接收区块链网络传入的交易信息；

（ⅱ）调用 SGX 安全区外的验证函数对交易中不需要用户密钥的部分进行验证，计算交易的优先级并进行排序；

该部分的验证限于不需要用户密钥的验证，例如验证交易的版本是否正确、交易资产量是否合法等。同时视具体区块链的架构选择性完成交易的优先级计算与排序等操作。

（iii）客户端向 SGX 驱动器发起请求，通过访问密钥验证后，调用某一可信空间内的密钥验证函数对交易中需要用户密钥的部分进行验证；

需要用户密钥的部分的验证限于需要用户密钥的验证，例如确定交易目标是否为自己等。

（iv）检查对密钥验证函数的调用结果，若调用结果不正常，则调用另一个可信空间内的密钥验证函数，直至调用结果正常；之后将验证信息打包入验证结果，完成交易的验证；

（v）根据验证结果向区块链网络发送反馈；

若验证成功，则向区块链网络广播该交易信息，供其余用户节点继续验证；

若验证失败，则停止传播该交易，或向区块链网络反馈交易非法。

处理由用户发起交易引起的密钥操作包括以下步骤：

（Ⅰ）客户端在 SGX 安全区外构建交易；

该步骤中，客户端需要按照相应区块链的规则合法构造交易，为交易补充目标地址、交易额等必要信息；同时在该步，客户端无需提供自己的签名信息；

（Ⅱ）客户端向 SGX 驱动器发起请求，通过访问密钥验证后，调用某一可信空间内的密钥签名函数获取用户密钥签名；

（Ⅲ）检查对密钥签名函数的调用结果，若调用结果不正常，则调用另一个可信空间内的密钥签名函数，直至调用结果正常；之后将签名信息打包入交易信息，完成交易的构建；

（Ⅳ）向区块链网络广播交易。

与现有技术相比，本发明的有益效果为：

（1）本发明的基于 SGX 的区块链用户密钥保护客户端同时兼备安全性、可用性和通用性，能有效抵御恶意软件对用户本地密钥的嗅探与破解，保护用户的区块链资产不受侵害；

（2）客户端的操作分为两个独立部分，每个部分使用不同的密钥调用函数，提高了客户端的运作效率。同时两个部分对密钥的操作均放在 Enclave（可信空间）内执行，外部程序无法得知密钥信息与相关操作。密钥也不会以明文方式出现于任何不可信内存，能够抵御内存泄漏攻击；

（3）给出了一种多 Enclave 机制，进一步提升了密钥保存的安全性。即使单个 Enclave 损坏，客户端仍能正常运行，用户同时可以收到 Enclave 损坏的反馈，采取进一步的针对性措施维护客户端，保护密钥；

（4）对于未受到 SGX 保护的密钥预写入给出了一套安全的操作流程，能有效降低用户密钥在被 SGX 保护前受到的安全威胁。

附图说明

图 1 为基于 SGX 的区块链用户密钥保护客户端的结构及工作流程示意图；

图 2 为获取用户密钥并存入 SGX 加密模块的流程示意图；

图 3 为处理由区块链网络传入交易引起的密钥操作的流程示意图；

图 4 为处理由用户发起交易引起的密钥操作的流程示意图。

具体实施方式

下面结合附图和实施例对本发明作进一步详细描述。

本实施例的基于 SGX 的区块链用户密钥保护装置，即区块链客户端包括 3 个软件模块：交易共识模块、交易构造模块以及 SGX 加密模块，其流程控制如图 1 所示，具体如下：

（1）预备处理：在客户端运行之前需要预先进行区块链密钥的获取与写入 SGX，其流程如图 2 所示。

（1-1）以安全途径获取区块链认可的用户密钥；

对于需要用户自主生成用户密钥的情况，生成用户密钥的步骤应在安全的设备上（如从未联网的安全 PC）由密码学完备的种子与算法按照规定格式生成。

对于需要区块链网络颁发的用户密钥，则通过严格保护的通信渠道，以安全的加密方式获取；或由线下的安全方式获取。

（1-2）对用户密钥进行备份；

为规避日后由于各种原因导致密钥遗失的风险，无论采取何种密钥保管方式，都建议用户对获取的用户密钥预先进行安全的备份。建议采用物理存储或离线安全设备存储等方式对用户密钥进行备份；并备份多份，分别予以安全存储。

（1-3）从用户密钥生成公钥并存储与本地；

一般情况下，用户需由用户密钥生成公钥，用公钥生成自己的区块链地址（如比特币），以便他人指定自己为交易目标。故而需要在存储用户密钥前执行该步骤以生成与用户密钥对应的公钥，存储于本地。

公钥生成的地址可向全网公开，无需加密。对于由区块链网络负责处理用户公钥与地址的区块链系统（如各用户可向区块链网络询问每个用户的地址），该步骤可由区块链网络完成。

（1-4）将用户密钥存入 SGX 加密模块。

步骤（1-4）包括：

（a）生成用户密钥和密钥操作函数的证书，将用户密钥和及密钥操作函数与所述证书一起上载到处理空间中；

所述密钥操作函数包括密钥验证函数和密钥签名函数；

（b）通过 SGX 驱动器对上载的用户密钥、密钥操作函数以及其证书进行参数测量，为可信空间分配地址空间和内存页，创建可信空间，并将用户密钥和密钥操作函数复制到可信空间中，之后删除处理空间中的数据；

（c）SGX 驱动器获取用户密钥和密钥操作函数的证书信息并传递给 SGX 硬件处理器；SGX 硬件处理器根据证书信息的哈希值以及 SGX 硬件处理器自身的哈希值生成可信空间的访问密钥，通过访问密钥对可信空间进行加密。

（d）重复步骤（a）~（c），建立多个可信空间，将备份的多份用户密钥分别存入不同的可信空间中。

其中将密钥存入 SGX 使用了 SGX 加密模块，其作用是基于软件防护扩展指令生成的 enclave 来存储用户的密钥信息与密钥操作函数，并生成用以验证可信空间访问权限的密钥，供后续访问操作使用。同时该过程建立了多个 enclave（本例中为 3 个，如图 1），以保障在极其特殊情况下单个 enclave 损坏时区块链客户端仍能正常运行，并给予用户反馈以修复损坏的 enclave，避免用户在不知情的情况下丢失密钥。

（2）交易共识处理：该部分使用交易共识模块和 SGX 加密模块，共同完成交易共识处理，其流程如图 3 所示。交易共识模块的作用是接收来自区块链网络的交易流，并进行格式验证、交易数额验证等验证功能，验证过程中通过调用 SGX 加密模块完成需要用户密钥的验证，最终实现对网络中交易的验证共识。一旦安全区内的验证函数返回异常，便调用另一个安全区内的相同验证函数，直至完成验证。SGX 加密模块在这里的作用是为共识处理提供能够安全调用用户区块链账户密钥的验证函数。

包括以下步骤：

（ⅰ）客户端接收区块链网络传入的交易信息；

（ⅱ）调用 SGX 安全区外的验证函数对交易中不需要用户密钥的部分进行验证，计算交易的优先级并进行排序；

该部分的验证限于不需要用户密钥的验证，例如验证交易的版本是否正确、交易资产量是否合法等。同时视具体区块链的架构选择性完成交易的优先级计算与排序等操作。

（ⅲ）客户端向 SGX 驱动器发起请求，通过访问密钥验证后，调用某一可信空间内的密钥验证函数对交易中需要用户密钥的部分进行验证；

需要用户密钥的部分的验证限于需要用户密钥的验证，例如确定交易目标是否为自己等。

（ⅳ）检查对密钥验证函数的调用结果，若调用结果不正常，则调用另一个可信空间内的密钥验证函数，直至调用结果正常；之后将验证信息打包入验证结果，完成交易的验证；

（ⅴ）根据验证结果向区块链网络发送反馈；

若验证成功，则向区块链网络广播该交易信息，供其余用户节点继续验证；

若验证失败，则停止传播该交易，或向区块链网络反馈交易非法。

（3）交易构造处理：该部分使用交易构造模块和 SGX 加密模块，共同完成交易构造与发起，其流程如图 4 所示。交易构造模块按照用户的意图发起交易，通过调用 SGX 加密模块实现交易信息的填充与合法化，并向全网广播该交易。一旦安全区内的签名函数返回异常，便调用另一个安全区内的相同签名函数，直至完成交易构建。SGX 加密模块在这里的作用是为交易构造提供能够安全调用用户区块链账户密钥的签名函数。

包括以下步骤：

（Ⅰ）客户端在 SGX 安全区外构建交易；

该步骤中，客户端需要按照相应区块链的规则合法构造交易，为交易补充目标地址、交易额等必要信息；同时在该步，客户端无需提供自己的签名信息；

（Ⅱ）客户端向 SGX 驱动器发起请求，通过访问密钥验证后，调用某一可信空间内的密钥签名函数获取用户密钥签名；

该步骤的签名用于证实自己发起交易中的资产有效。

（Ⅲ）检查对密钥签名函数的调用结果，若调用结果不正常，则调用另一个可信空间内的密钥签名函数，直至调用结果正常；之后将签名信息打包入交易信息，完成交易的构建；

若调用结果不正常，说明该可信空间内的用户密钥损坏，则调用另一个可信空间内的签名函数重新获取签名。

（Ⅳ）向区块链网络广播交易。

以上所述的实施例对本发明的技术方案和有益效果进行了详细说明，应理解的是以上所述仅为本发明的具体实施例，并不用于限制本发明，凡在本发明的原则范围内所做的任何修改、补充和等同替换等，均应包含在本发明的保护范围之内。

说　明　书　附　图

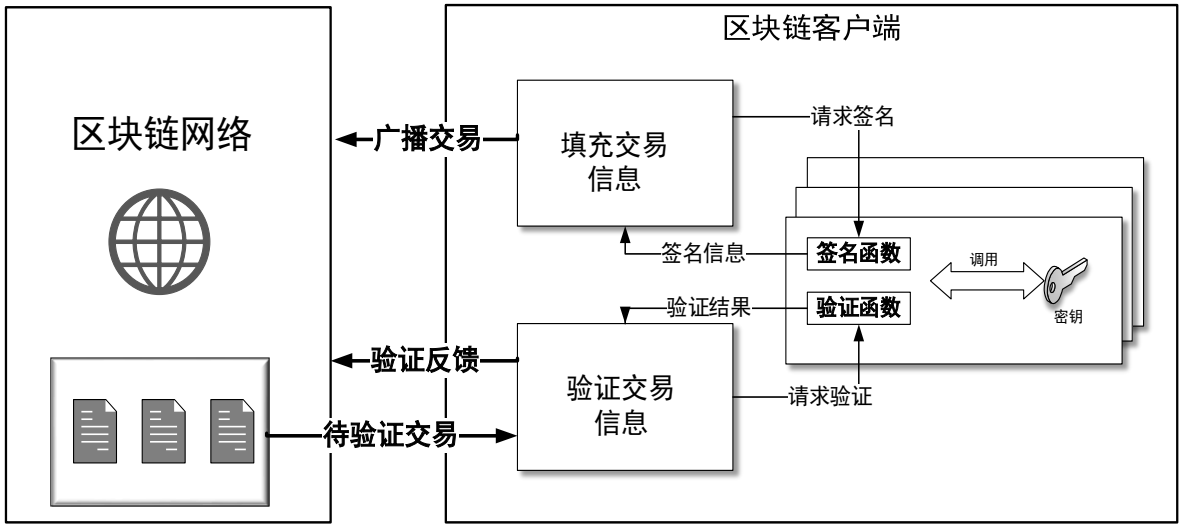


图 1（图 1 中的字体不清楚，图中不要填充颜色）

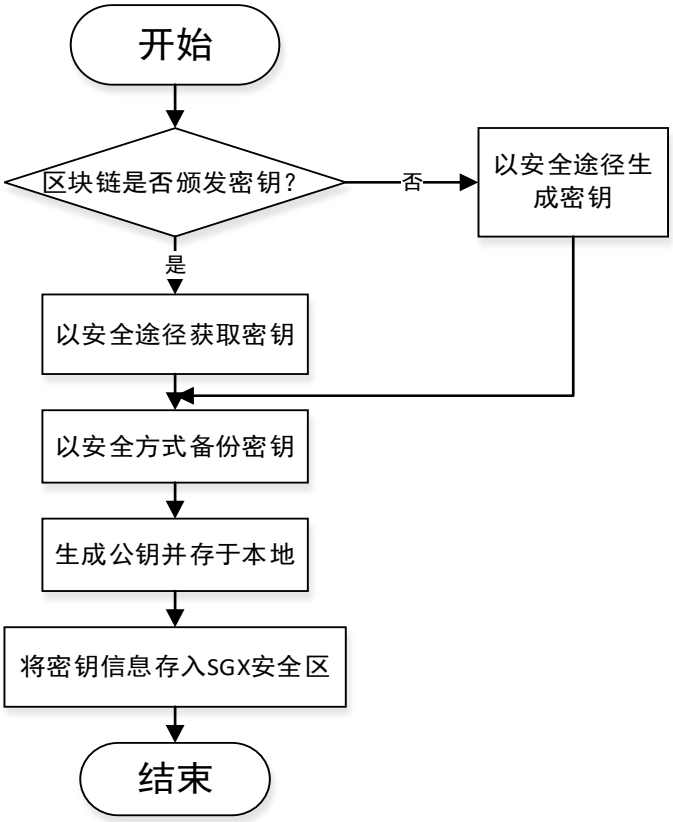


图 2（请去掉框图中填充的颜色）

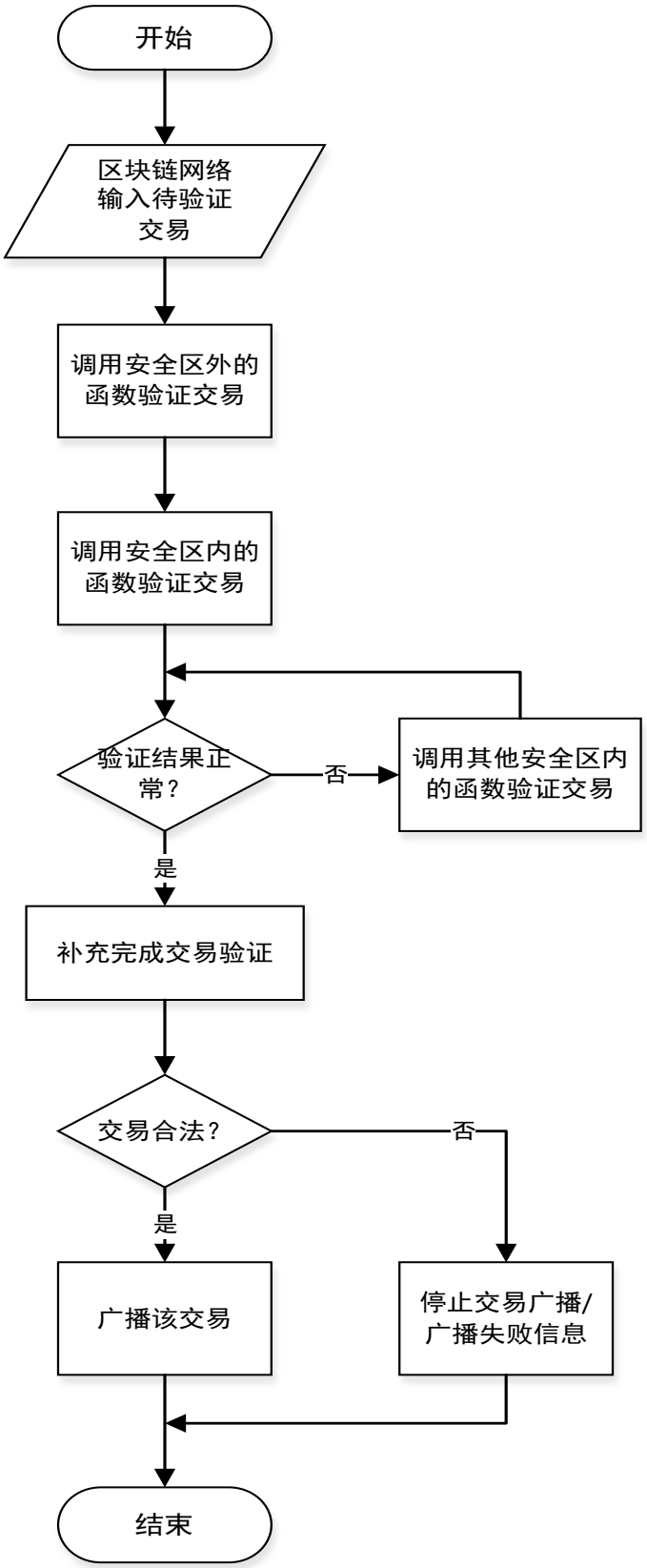


图 3（请去掉框图中填充的颜色）

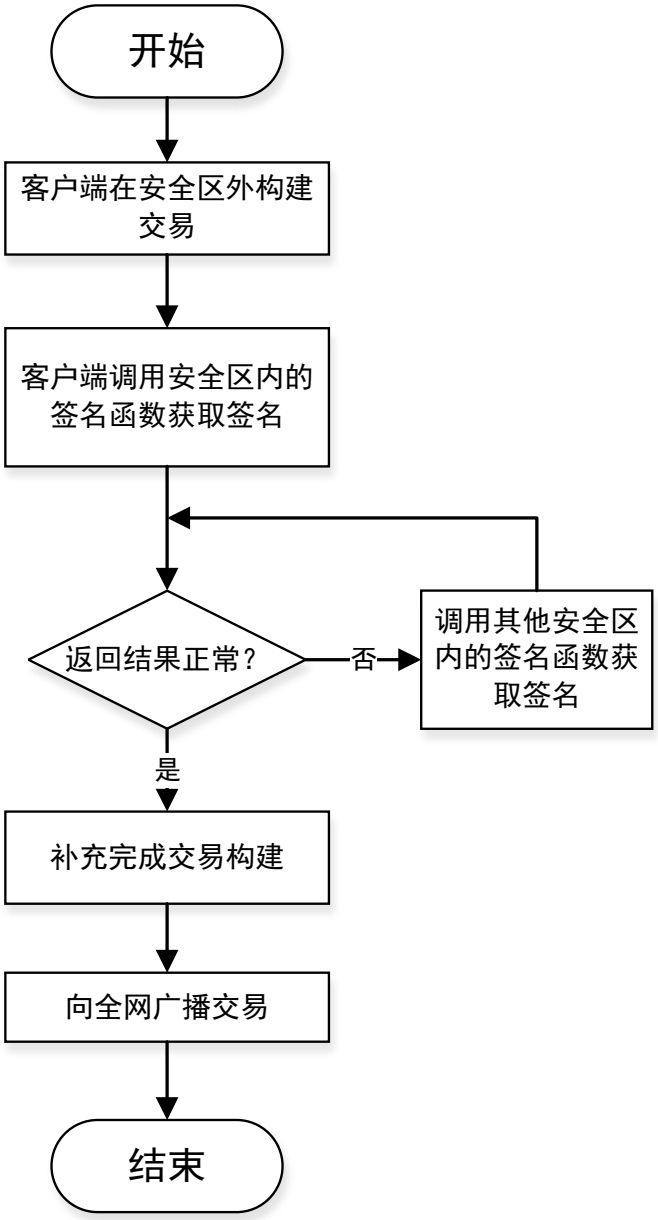
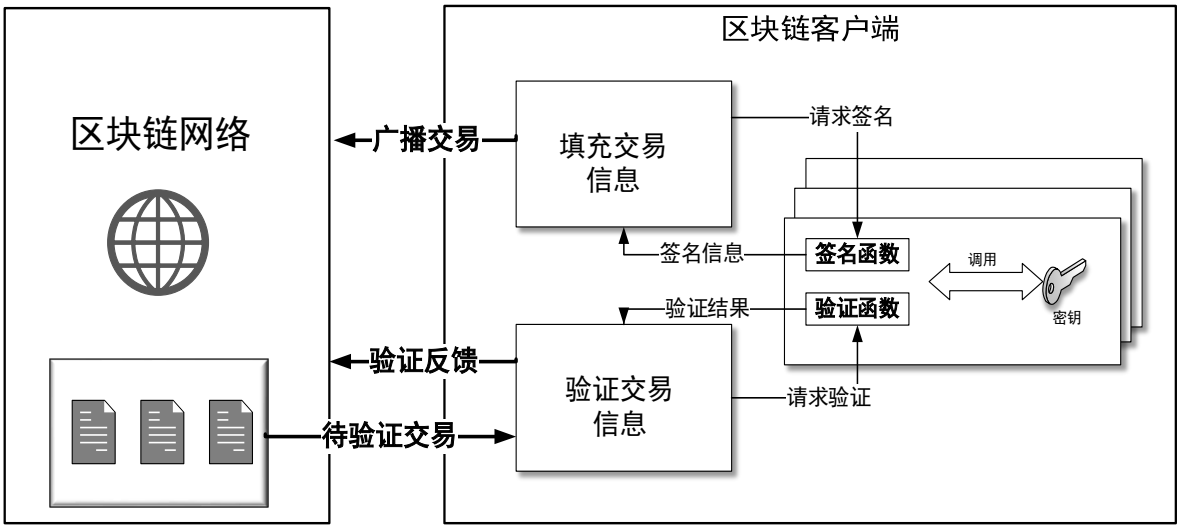


图 4（请去掉框图中填充的颜色）

说　明　书　摘　要

本发明公开了一种基于 SGX 的区块链用户密钥保护装置，包括：SGX 加密模块，基于软件防护扩展指令生成可信空间，并生成用于验证所述可信空间访问权限的访问密钥;所述可信空间用于存储区块链网络的用户密钥和密钥操作函数;交易共识模块，接收来自区块链网络的交易，通过所述访问密钥访问 SGX 加密模块，调用其中的密钥操作函数，实现对交易的验证共识；交易构造模块，根据用户的意图发起交易，通过所述访问密钥访问 SGX 加密模块，调用其中的密钥操作函数，实现交易信息的填充与合法化，并向区块链网络广播该交易。本发明还公开了区块链用户密钥保护方法。该方法能有效抵御恶意软件对用户本地密钥的嗅探与破解，保护用户的区块链资产不受侵害。

摘　要　附　图


区块链网络
区块链客户端
广播交易
填充交易
信息
请求签名
签名信息
签名函数
调用
密钥
验证结果
验证函数
验证反馈
验证交易
信息
请求验证
待验证交易