

Intel Software Guard Extensions -SGX技术简介

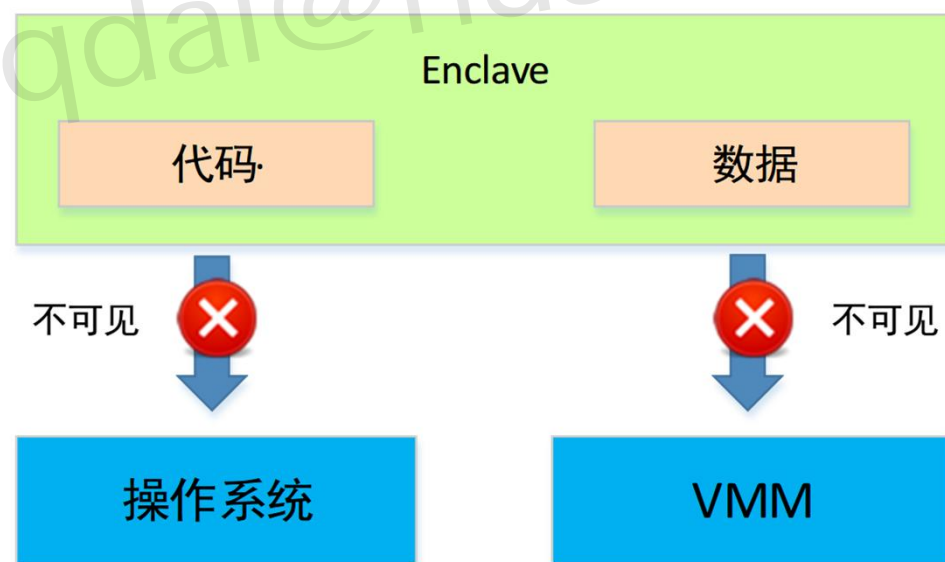


華中科技大學

WUZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

SGX概述

SGX全称**Intel Software Guard Extensions**，顾名思义，其是对因特尔体系（IA）的一个扩展，用于增强软件的安全性。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**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。



SGX设计的八大目标

- ①：允许应用开发人员保护敏感数据免受来自具有高级权限的流氓软件的越权访问以及修改
- ②：允许保持敏感数据的完整性以及机密性，但是不会干扰合法系统软件安排以及管理平台资源的应用
- ③：保证计算机用户对他们机器的控制权，并且保证用户能够自由安装卸载应用以及服务
- ④：允许平台测量应用所信任的代码并提供一个签名证书，证书根植于处理器

SGX设计的八大目标

- ⑤：能够让可信的应用程序调用外部的进程或者其他工具
- ⑥：允许受信任程序去规划底层处理器的功能，例如中断
- ⑦：允许软件供应商按照他们思路运行可信软件以及对可信软件进行更新
- ⑧：即使攻击者物理上控制了平台，并且能够进行直接的内存攻击，也能够保证可信空间的代码以及数据的机密性

SGX结构

在SGX的结构中，其底层依赖于操作系统和驱动的支持，一个SGX程序包含两个部分即应用程序（不可信部分）和Enclave（可信部分），位于不可信部分的代码和数据对于主机操作系统来说是可见的，可信部分对外部不可见，两者通过Enclave Definition Language（EDL）接口进行数据传输，而位于Enclave的代码和数据则是只能够被和Enclave关联的应用程序访问。



SGX运行过程

运行SGX程序时，首先运行应用程序（不可信部分），应用程序创建其对应的Enclave，Enclave事先已经写好了数据处理代码，只需要将被处理的数据通过EDL文件接口传入Enclave之内即可，处理过程对外部不可见，之后Enclave将处理结果通过EDL文件接口返回给应用程序。



可信硬件对比

SGX VS ARM TrustZone

- TrustZone划分出两个隔离环境（安全世界和正常世界），安全世界中可以包含多个应用程序的代码，这些代码不存在隔离
- SGX也划分出了安全世界（Enclave）和正常世界，安全世界中每个应用程序的代码位于一个Enclave之内，代码存在隔离
- TrustZone应用在手机端
- SGX应用在PC端



可信硬件对比

SGX VS TPM

- 两者都具有较高的安全性
- SGX比TPM性能更强
- SGX比TPM更加方便使用，SGX提供了简单的编程模型。

高安全
高性能
使用方便

SGX

高安全
低性能
使用繁琐

TPM

可信硬件对比

为什么SGX比TPM性能更强？

- ✓ 本质上来说，**Enclave是CPU内存的一部分**，这部分称之为Enclave Page Cache (EPC)，EPC内存是SGX创建Enclave所使用的内存，每个EPC页面的大小都是**4K**，所有的Enclave都在EPC之内，正是因为SGX可信部分的代码和数据都在CPU之上直接运行，因此性能更强。
- ✓ SGX的EDL接口传参本质上是将数据**从不可信部分的内存拷贝到Enclave对应的内存**，相同的代码在Enclave内运行的时间与在CPU非可信部分运行的时间一致，SGX性能开销集中在**数据拷贝**，这部分开销极低。
- ✓ **TPM并没有集成到CPU之上**，CPU需要将数据拷贝到TPM之内，TPM负责运行代码以及数据，性能远不比CPU！

SGX本地认证

SGX核心功能包括两个大功能，其功能如下：

SGX认证功能：

SGX本地认证：同一台电脑的多个Enclave能够互通数据

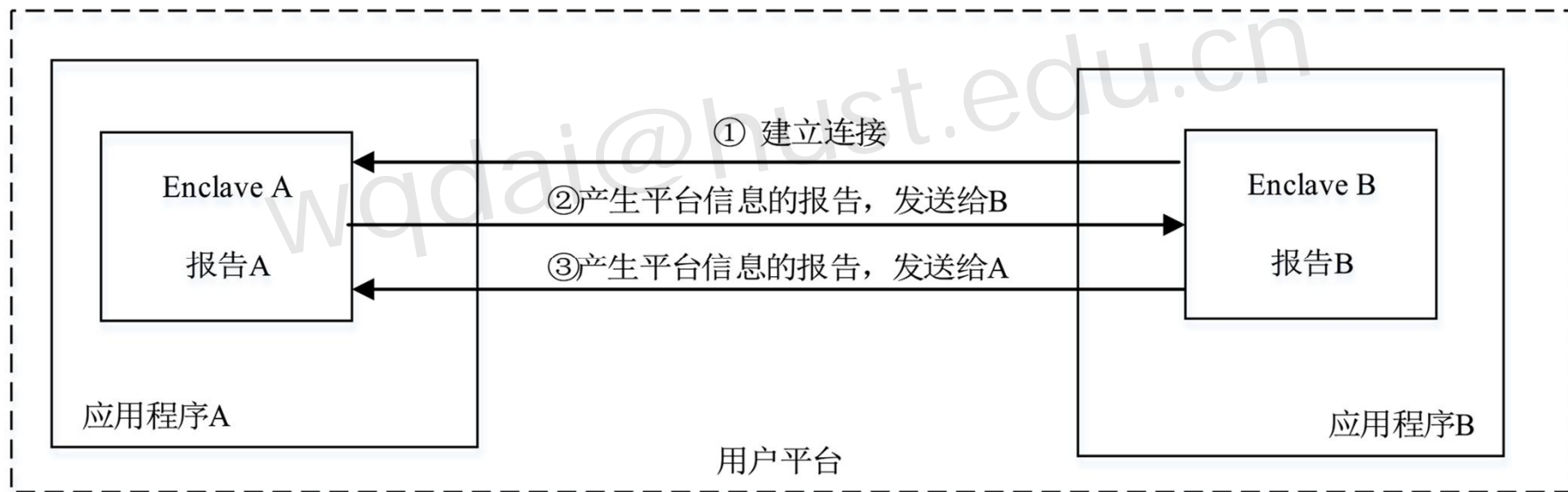
SGX远程认证：不同电脑的多个Enclave能够互通数据

SGX密封功能：

将Enclave数据使用SGX的加密方案保存到本地，以便后续使用

SGX核心功能

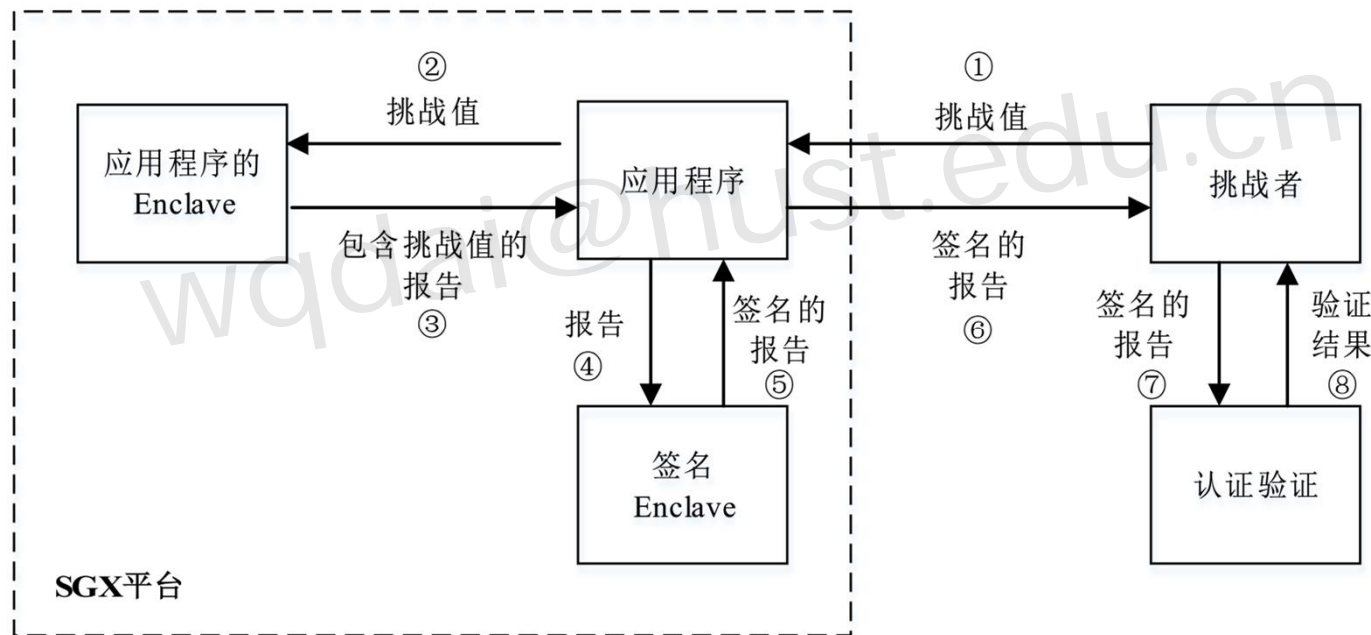
在实际的开发过程中，可能需要同一平台多个程序进行协同合作，但是每个程序都有一个Enclave，并且每个Enclave都互相隔离，为了使多个Enclave之间能够协同合作，SGX提出了本地认证这一功能，大致过程如下：



其中信息报告包含创建Enclave时对其代码的hash，SGX版本等信息

SGX核心功能

在实际的开发过程中，可能需要不同平台多个程序进行协同合作，例如用户A想在用户B的电脑上处理数据，但是不知道用户B是否具有SGX安全环境，此时可以用SGX远程认证功能：



其中报告包含的是SGX版本等信息，签名Enclave是SGX自带的Enclave，在挑战者收到签名的报告之后，需要将报告提供给Intel官方网站进行验证。

SGX核心功能

SGX创建的Enclave在应用程序运行完之后会自动销毁，如果不对数据进行保存就**无法下次使用**，为了解决这个问题，SGX提出了数据的密封以及解密封功能：

密封过程：

- 检查待密封数据的**有效性**，例如指针指向的内存是否位于**Enclave**之内
- 使用密封数据的**API将数据密封**（API会调用SGX硬件才能访问的密钥加密数据）
- **保存**此次的密封信息到**外部存储器**（例如密封数据的类型，结构等信息）

解密封过程：

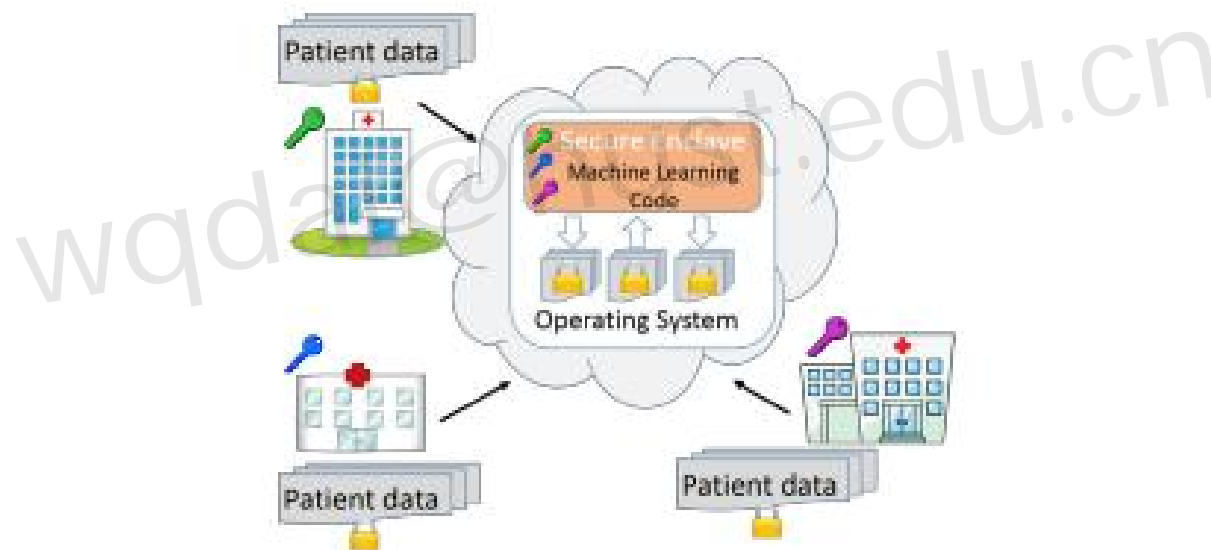
- 将保存在外部的密封信息读取密封
- 在**Enclave**内申请相应的内存空间来存放解密封的数据
- 使用密封数据的API将数据解密封（API会调用SGX硬件才能访问的密钥解密数据）

SGX文章

顶级期刊于会议上都有较多的SGX文章，这也间接证明了SGX的价值， 以下文章是比较有代表性的文章：

USENIX Security

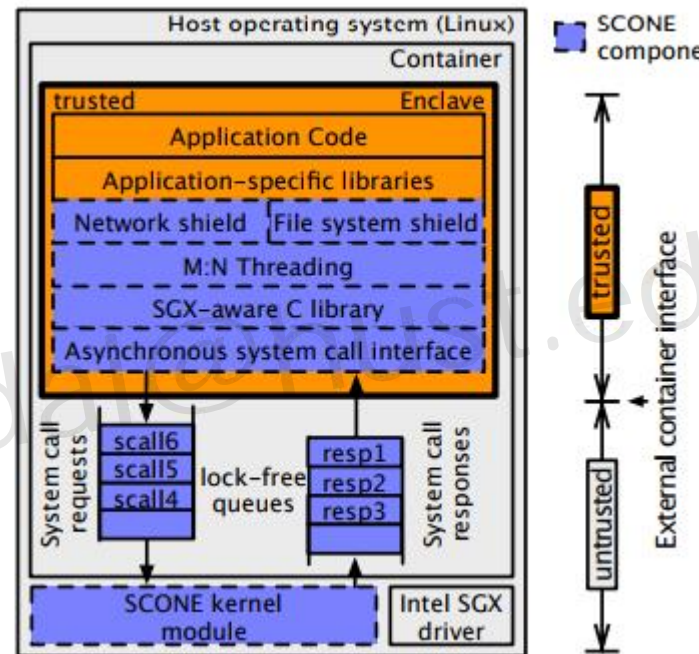
Oblivious Multi-Party Machine Learning on Trusted Processors.



多方机器学习的代码置于**SGX**创建的可信环境中，多个不互信的机构掌握有不同的数据，在进行机器学习的过程中，互不可信的多方将自己的数据发送至**SGX**的可信环境中一同分析，在隐私数据不泄露的情况下扩大了数据分析的数据基础使得数据分析结果更加可靠

OSDI

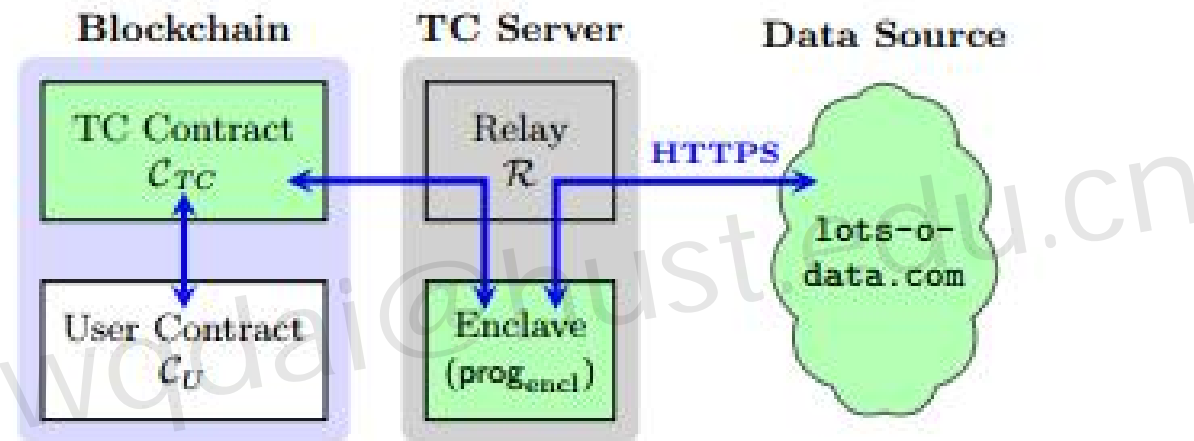
SCONE: Secure Linux Containers with Intel SGX.



使用了**SGX**保护**Docker**容器环境，进而保证容器内代码以及数据的安全性。**Docker**容器性能相对于传统的虚拟机而言启动迅速部署简单，但是由于和主机公用内核，因此其安全性略弱，通过使用**SGX**保护容器技术，使得**Docker**容器在保障性能的同时大大提高了其安全性。

CCS

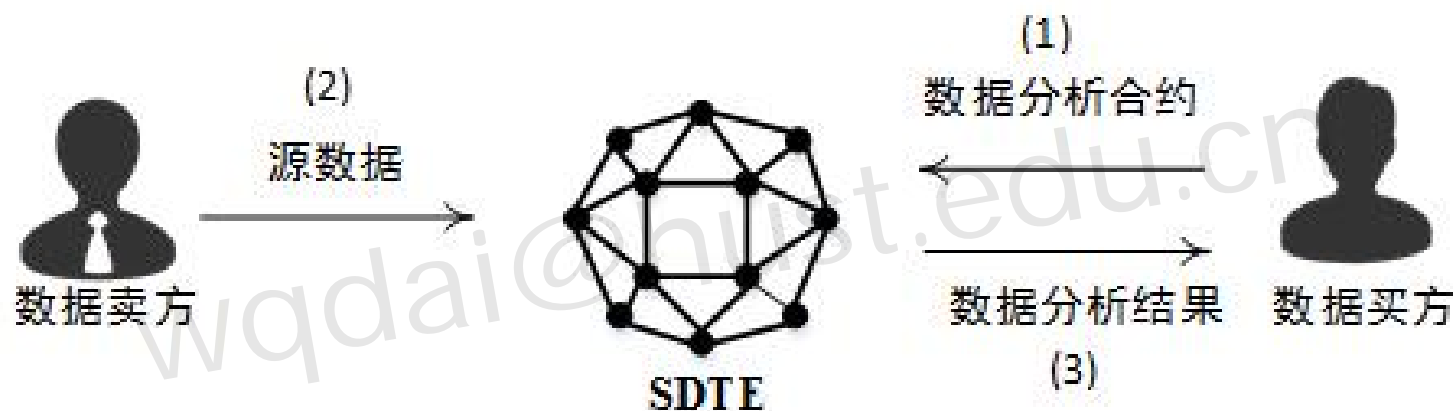
Town Crier: An Authenticated Data Feed for Smart Contracts.



在某些情况下，智能合约需要人们输入某些收集的数据，数据的收集过程的可信程度就决定了智能合约执行结果的可信程度，Town Crier(TC)使用SGX保护智能合约的数据收集过程。在TC中，数据源会将数据通过HTTPS的方式发送给TC服务器的Enclave之中，根据Enclave的安全特性，数据一旦进入到Enclave之内就不会被更改，因此里面的数据得到了安全保护。

TIFS (Accept with Minor) (我们自己的文章，这里重点讲一下)

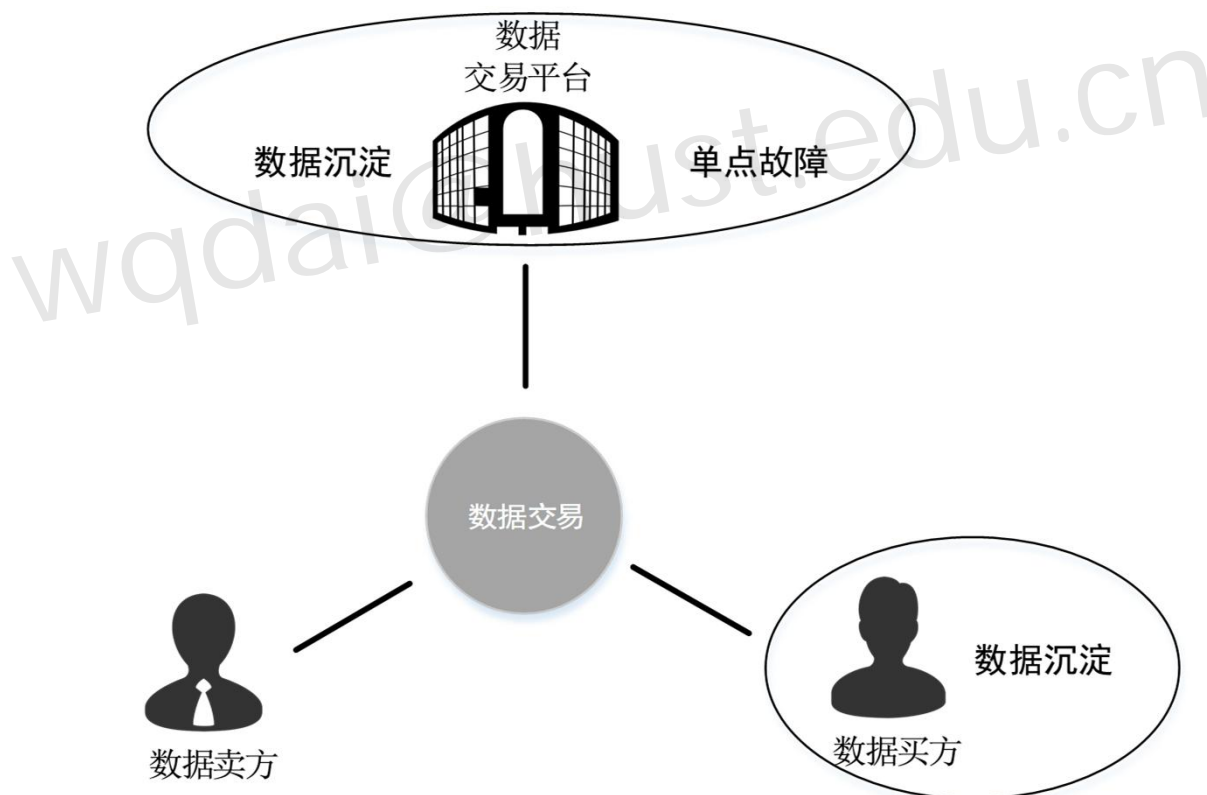
SDTE: Secure Blockchain-based Data Trading Ecosystem.



将数据分析过程移植到基于区块链的数据交易平台，并且使用**SGX**保护智能合约执行环境，进而防止数据分析过程中隐私数据的泄露，实现了数据的安全交易

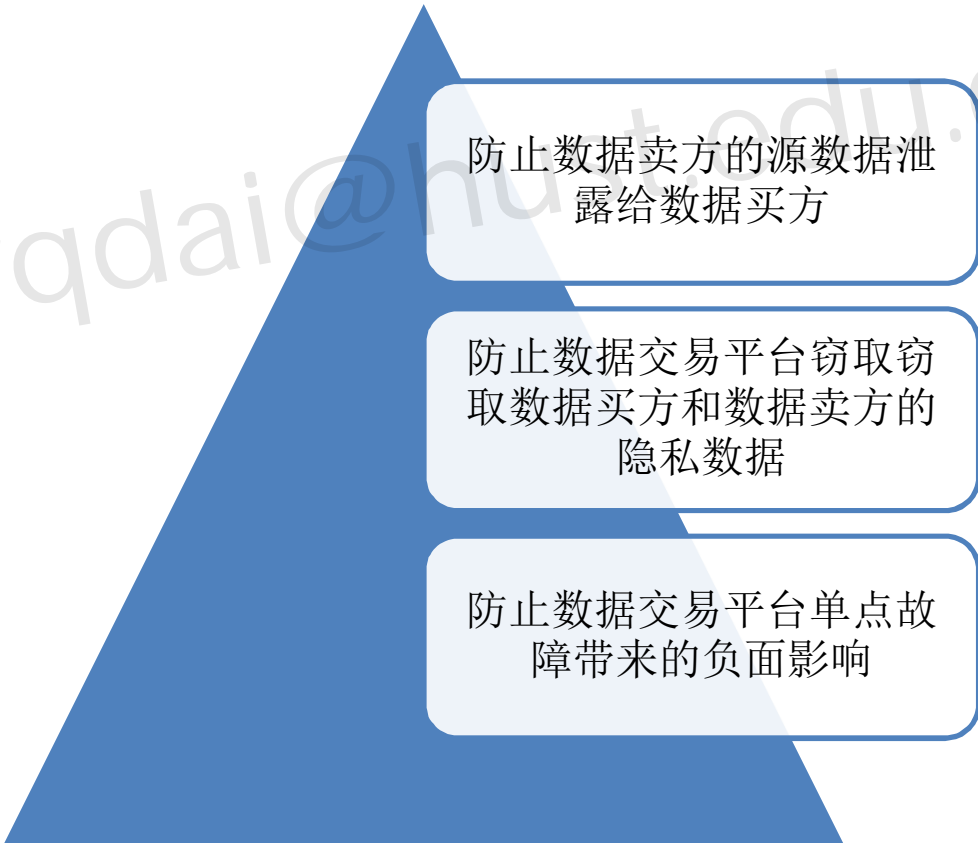
论文出发点

现有的数据交易存在以下三个问题中的一个或者多个: 1) 数据买方的数据沉淀问题; 2) 数据交易平台的数据沉淀问题; 3) 数据交易平台的单点故障问题;



设计目标

系统的设计目标包含三个：1) 防止数据卖方的源数据泄露给数据买方；2) 防止数据交易平台窃取数据买方和数据卖方的隐私数据；3) 防止数据交易平台单点故障带来的负面影响；



防止数据卖方的源数据泄
露给数据买方

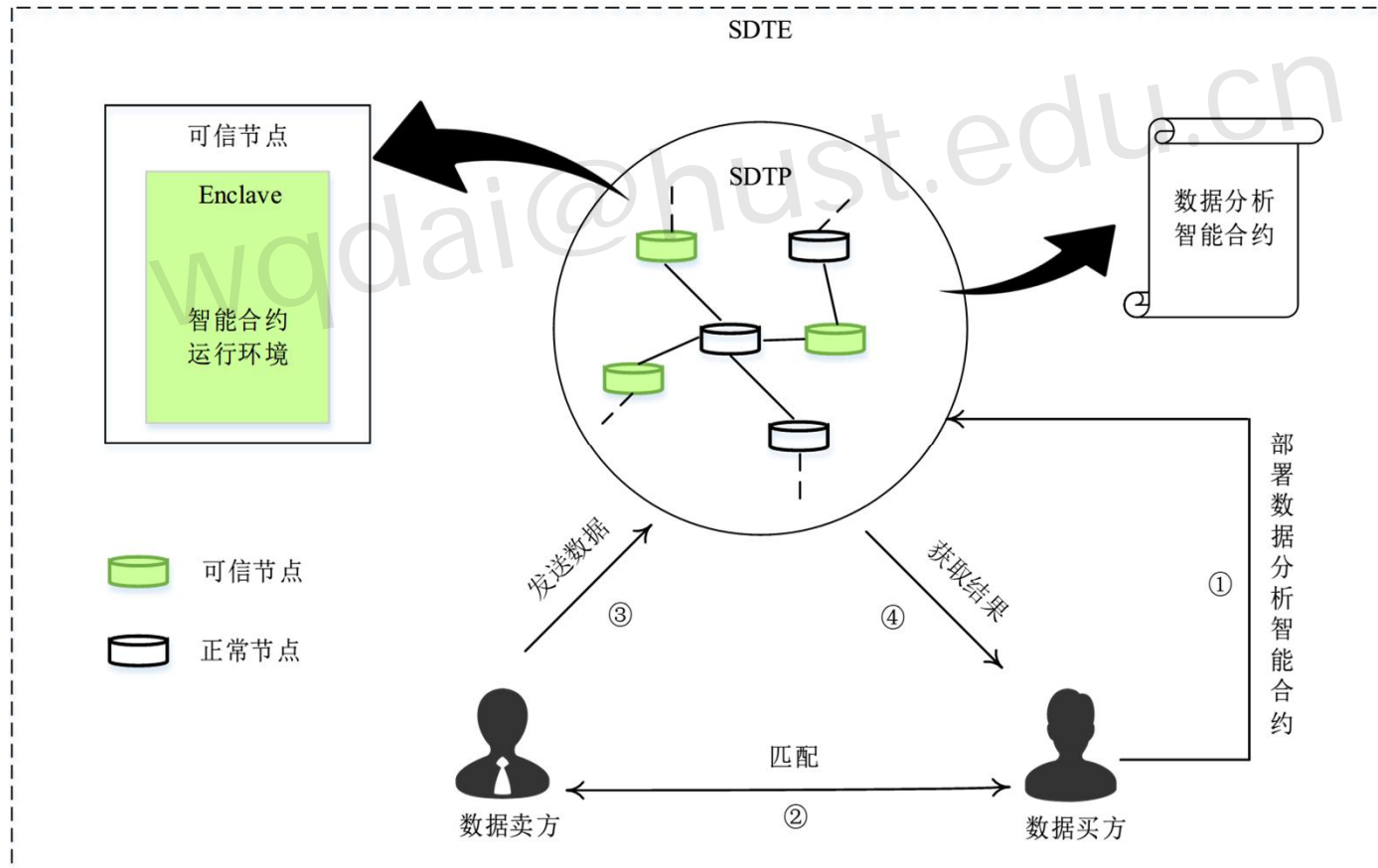
防止数据交易平台窃取窃
取数据买方和数据卖方的
隐私数据

防止数据交易平台单点故
障带来的负面影响

解决方案

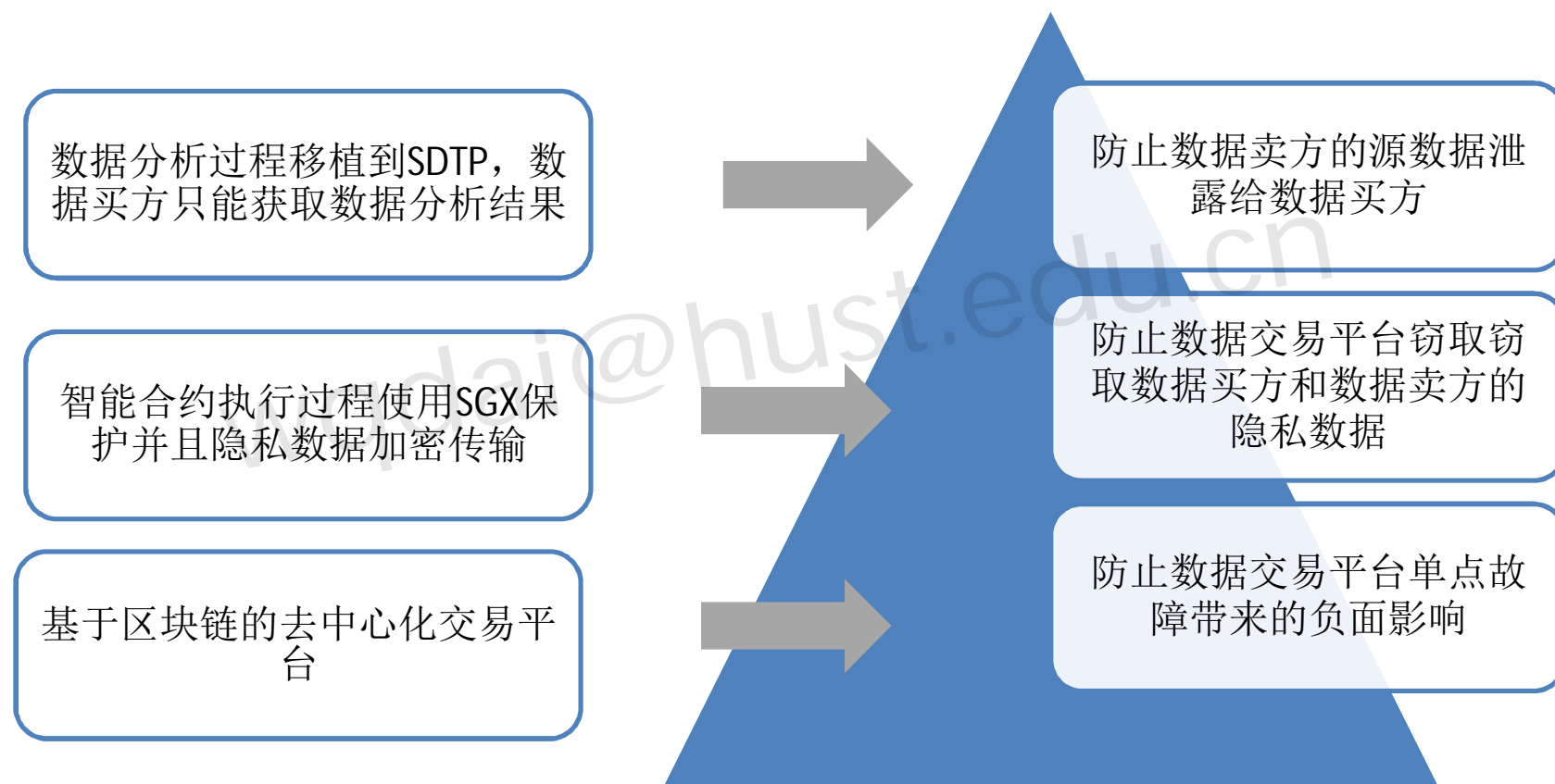
基于区块链的数据交易生态系统（SDTE）包含以下角色：

- SDTP: 基于区块链的数据交易平台，运行数据分析合约
- 数据买方: 在SDTP部署数据分析智能合约
- 数据卖方: 向数据分析智能合约提供源数据



课题解决方案

上述解决方案实现了设计目标：

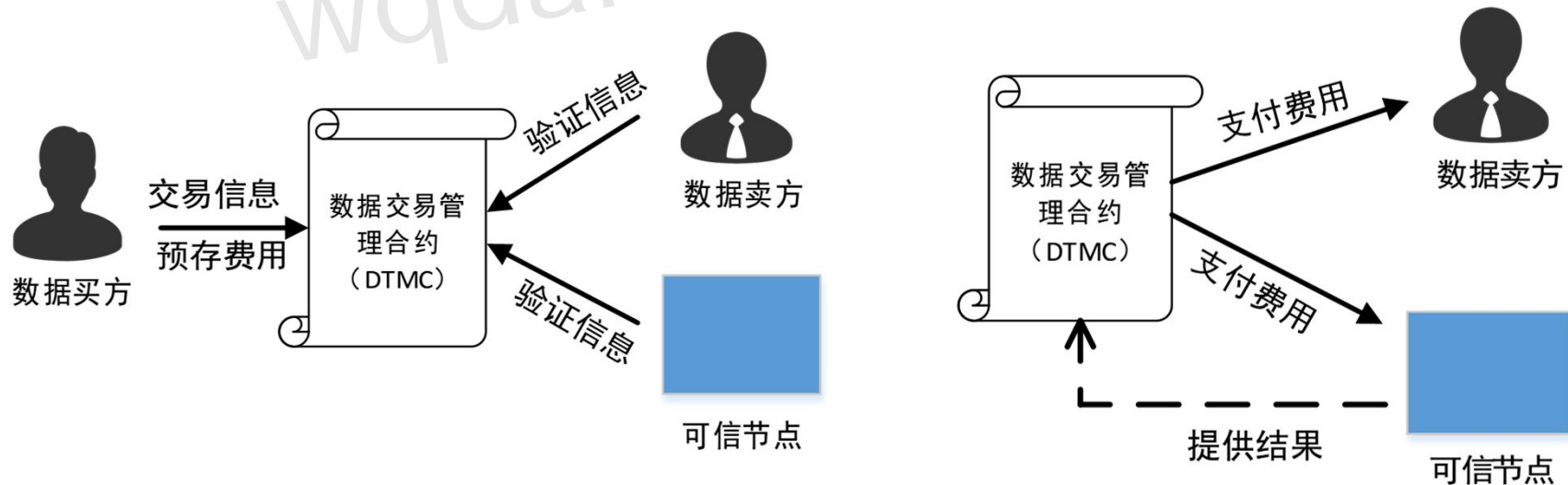


但是引入了新的问题

角色抵赖和欺诈

角色可能在数据交易过程中进行抵赖和欺诈，例如数据买方可能会对支付的各种费用进行抵赖等等

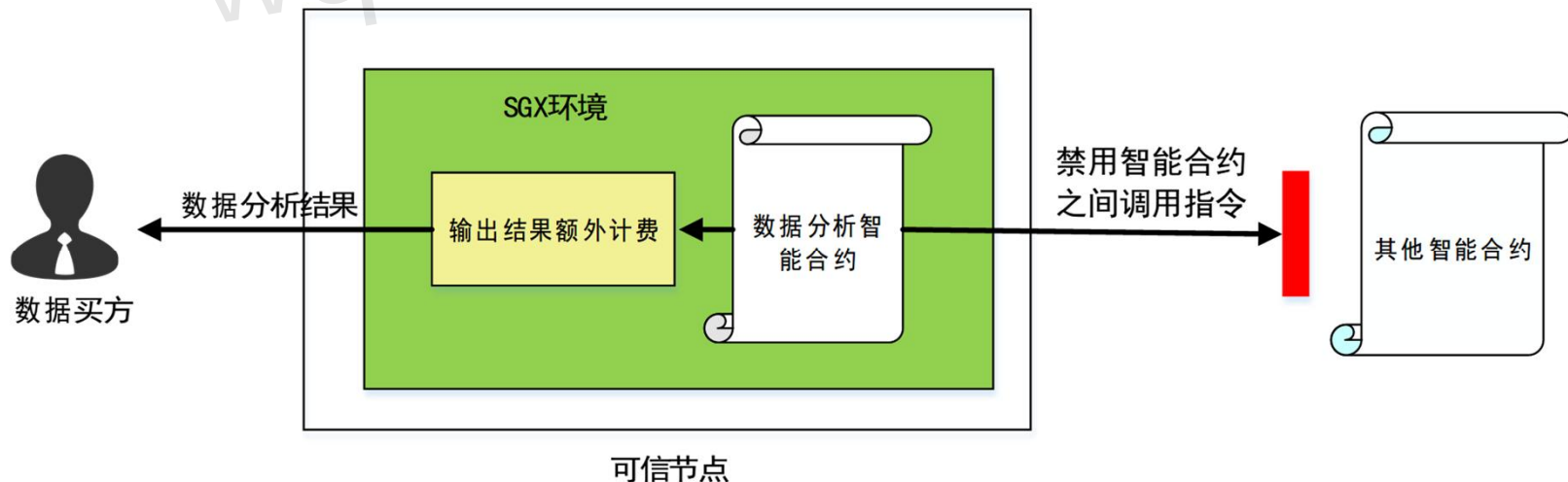
解决方案：SDTE部署了数据交易管理智能合约（DTMC），DTMC本身不会作弊，可以充当诚实第三方监督交易中的角色。



恶意智能合约窃取源数据

恶意智能合约对源数据的窃取有两种方式：1) 智能合约之间的调用来窃取源数据；2) 智能合约直接输出源数据

解决方案：**禁用智能合约之间调用**，进而防止智能合约之间的调用窃取源数据。**对输出结果按照数据大小进行额外计费**，海量的源数据数据买方需要支付巨额的费用，进而防止智能合约直接输出源数据。



OSDI

Ryoan: A Distributed Sandbox for Untrusted Computation on Secret Data.

提出可基于**SGX**的分布式沙盒，他允许用户将他们的服务放在使用**SGX**保护的**可信沙盒**之中并对输入数据的次数进行限制，进而保障用户的隐私信息。

WWW

SGX-PySpark: Secure Distributed Data Analytics.

使用**SGX**保护公有云的数据分析过程，进而防止公有云对用户隐私数据的窃取

SP

VC3:Trustworthy Data Analytics in the Cloud Using SGX.

使用**SGX**保护**MapReduce**处理海量数据的过程，进而防止隐私数据的泄露

OSDI

A Shielding Applications from an Untrusted Cloud with Haven.

使用**SGX**保护**Apache**以及**SQL**服务

SGX缺陷

SGX虽然安全性以及性能方面都可圈可点，但是**SGX**本身也存在一些缺陷：

首先**SGX**的使用依赖于**SGX SDK**，但是**SGX SDK**目前只支持**C++**编程语言，对**JAVA**、**Go**等语言暂不支持。

其次，**SGX SDK**支持的函数库有限，**SGX**应用程序只能够链接**SGX SDK**支持的第三方函数库，但是目前诸如**BOOST**主流函数库支持，都不支持。

然后，**SGX**目前支持的**CPU**型号并不是主流**CPU**型号，入手**SGX**安全硬件存在一些困难。

最后，**SGX**并不是绝对的安全，目前已有文章对**SGX**进行侧信道攻击成功。

SGX相关网站

Intel SGX 官方网站:

<https://software.intel.com/zh-cn/sgx>

SGX开发指南:

<https://software.intel.com/en-us/node/702968>

Intel SGX新技术学习研究引导手册:

<http://www.vonwei.com/post/IntelSGXGuide.html>

SGX SDK Windows 下载:

<https://registrationcenter.intel.com/en/forms/?productid=2614>

SGX SDK Linux 下载:

<https://01.org/zh/intel-software-guard-extensions/downloads?langredirect=1>

SGX SDK 相关文档:

<https://download.01.org/intel-sgx/linux-2.0/docs/>

SGX SDK Github:

<https://github.com/intel/linux-sgx>

结论

- 就概念来说SGX并没有特别独到之处，可以说是Trusted computing的reengineering。其创新之处在于SGX把这些功能都实现在了CPU上，因此性能上获得了很大提升，并且提供了比较简单的编程模型。这一点是TPM+TXT所不能比拟的。
- SGX提供得认证功能实现了多个Enclave之间的联动，SGX提供的密封功能实现了Enclave内数据的寿命延长。
- SGX目前也存在一些不足，这些限制了SGX的应用范围。