

Intel SGX原理及其区块链应用研究

李佳轩

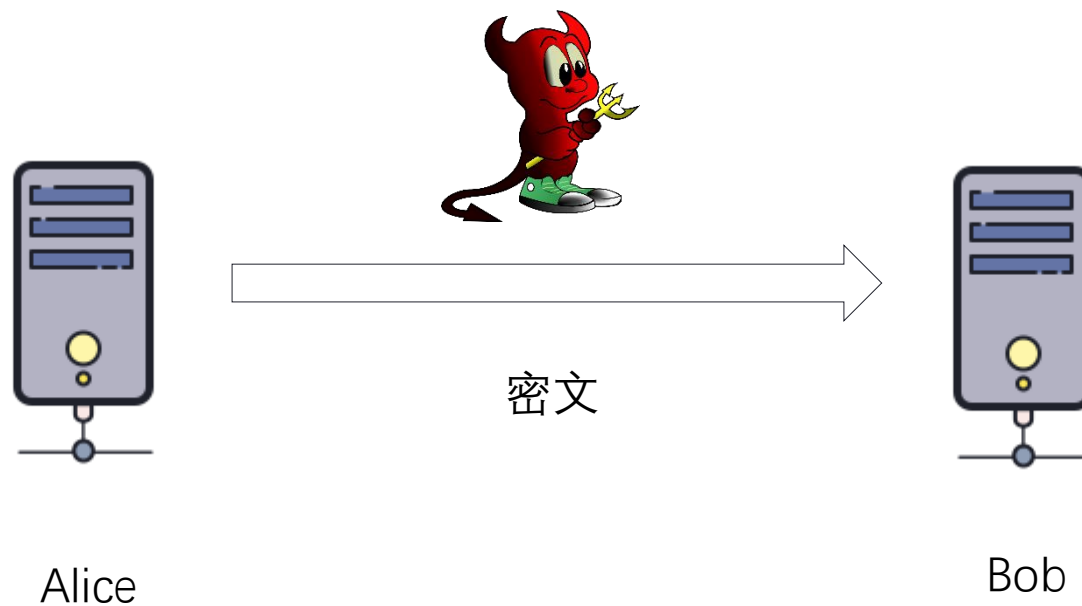
2020年7月20日

南方科技大学

SGX 研究背景

目前现状

- 目前密码学研究的重点倾向于过程安全。即密文过程中是否可以被监听，被篡改。
- 终端权限有操作系统保证。

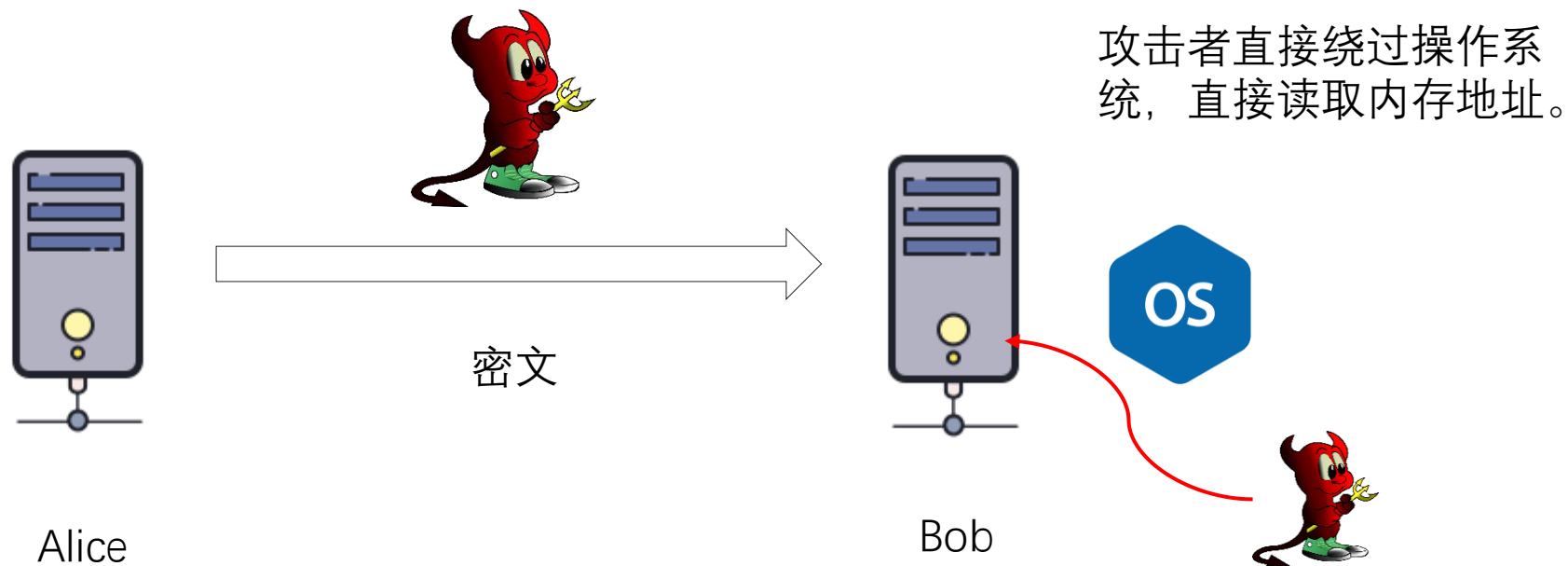


Bob自己为了
获得密文，需
要本地解密。

SGX 研究背景

安全问题

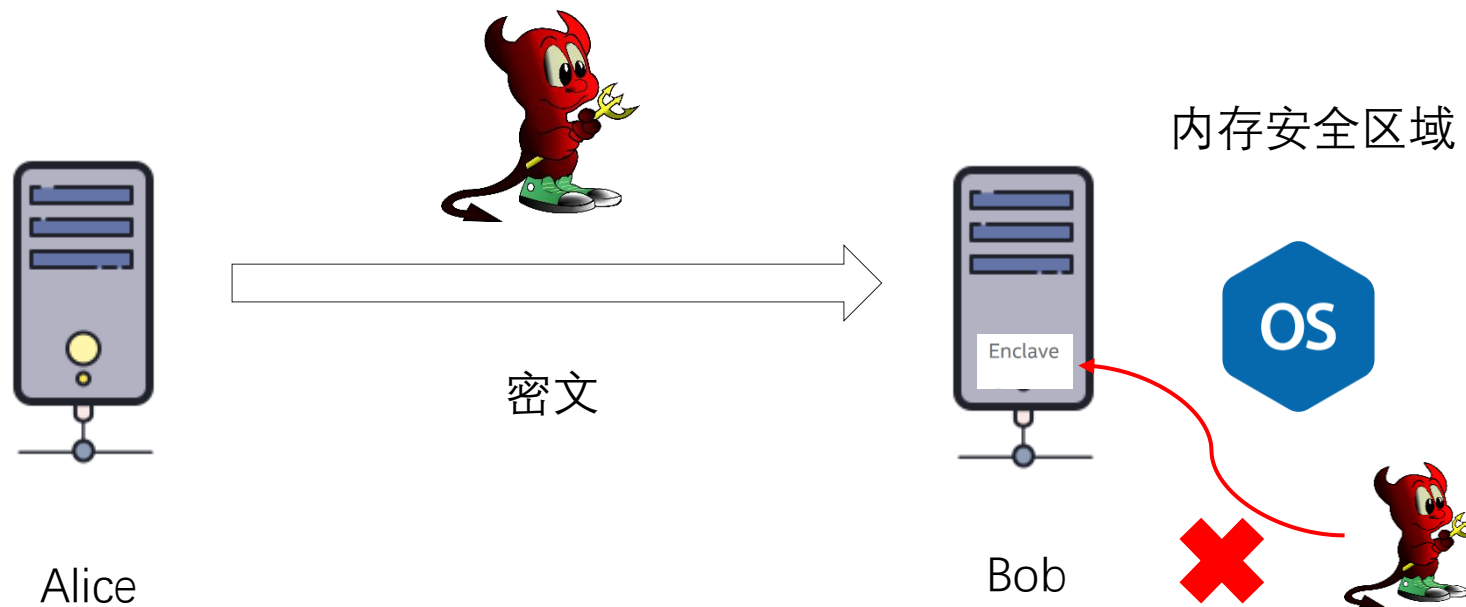
- 操作系统可能会被攻击者攻破。
- 攻击者可能直接绕过操作系统



SGX 目标

SGX目标

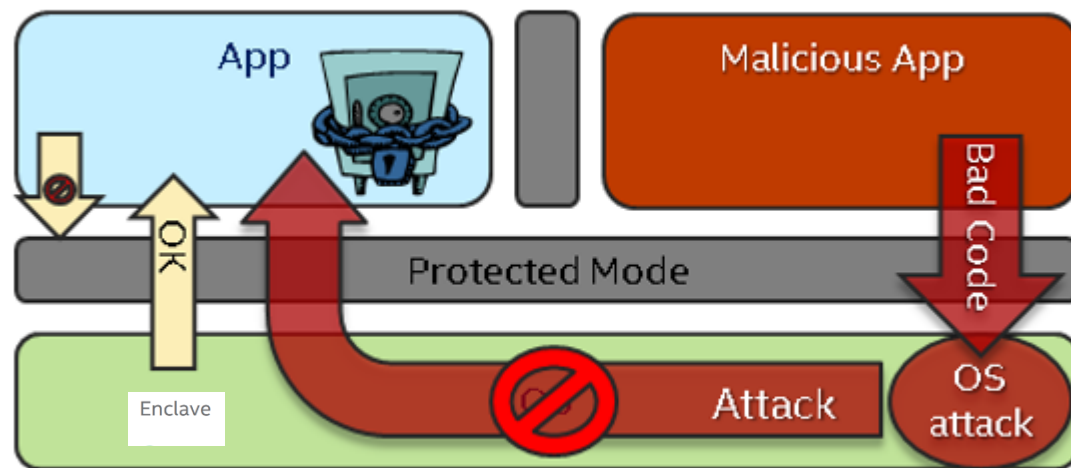
- CPU指令创建安全区域，仅特定的条件可以访问。



SGX 保护了什么？

- 来自操作系统/管理程序的攻击
- 来自BIOS，固件，驱动的攻击
- 来自程序系统管理模块的攻击
- 来自管理引擎的攻击
- 任何远程攻击

减少了被攻击面



图片来源: 引用1

Enclave设计

- 什么是 Enclave ?
 - Enclave 的目标
- Enclave 技术架构
- Enclave 生命周期

区块链应用

- Ekiden 项目

01

系统概念

- 技术架构
- 物理内存
- 虚拟内存
- 工作原理

02

03

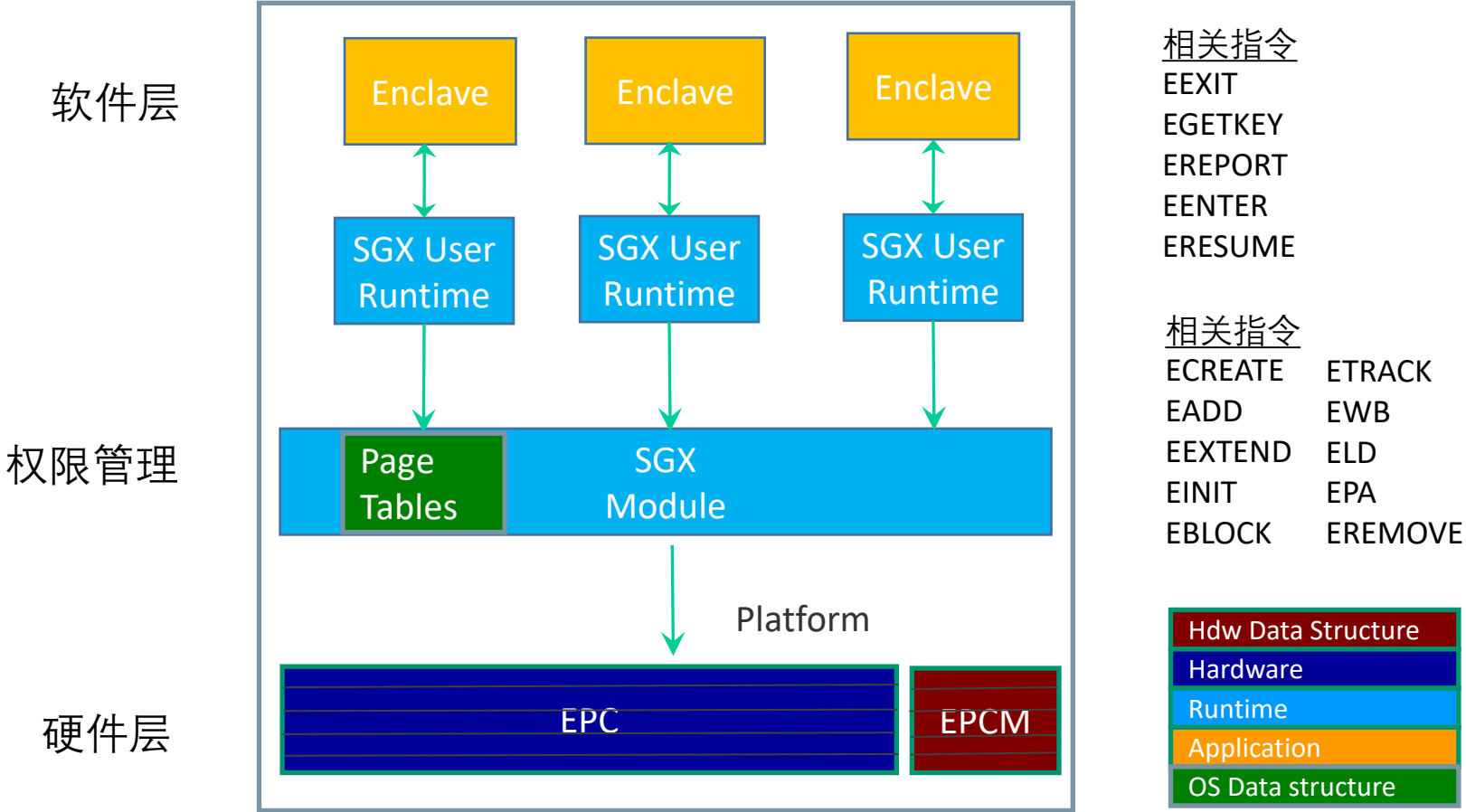
密码协议

- 工作流程
- 本地断言
- 远程断言

04



Enclave 技术架构



SGX内存条目

Structure	Description
Enclave Page Cache (EPC)	Contains protected code and data in 4K pages
Enclave Page Cache Map (EPCM)	Contains meta-data of enclave page
SGX Enclave Control Store (SECS)	Meta data for each enclave
Thread Control Structure (TCS)	Meta data for each thread
VA Page	Version Array of evicted pages
SIGSTRUCT	enclave's signature structure, the sealing identity

EPC有特有软件管理软件和硬件共同控制。

EPC 是enclave实际的生存区域。

EPCM和EPC是一对一的映射关系。

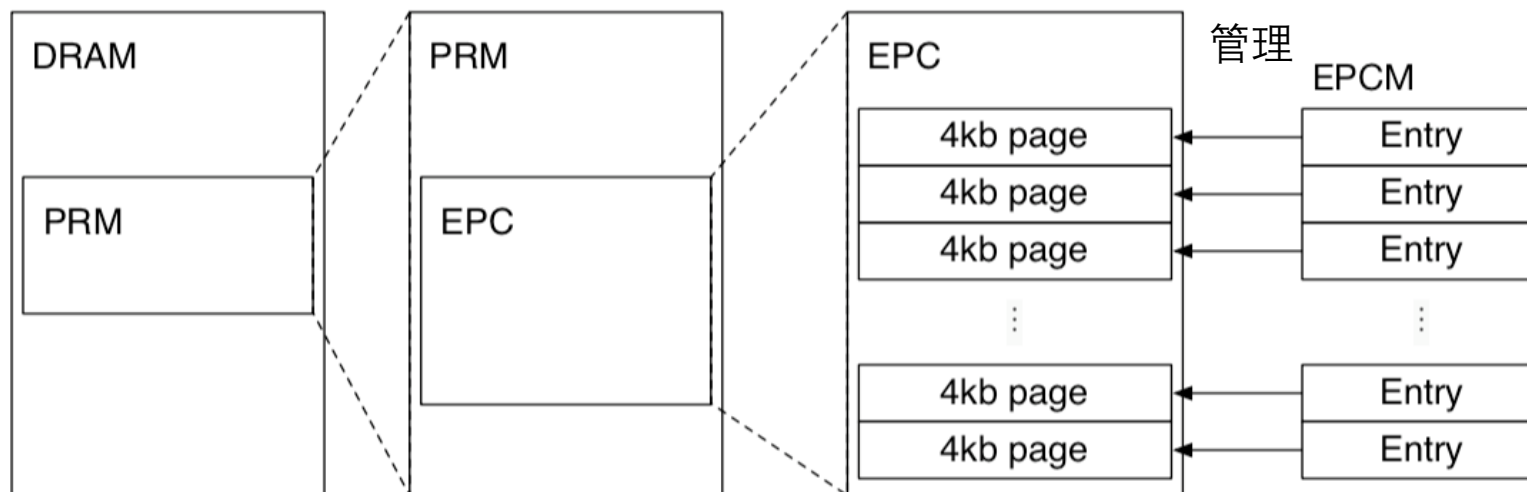
SECS存储了enclave相关的元数据，其消耗1页的EPC。

TCS占用EPC的一页，并且包含硬件用于控制每个逻辑处理器进入enclave的元数据。

VA进行版本控制，不属于任何Enclave。

SIGSTRUCT 提供更灵活的密封身份和证明。

物理内存的组织



相关概念

Dynamic random-access memory (DRAM)

Processor Reserved Memory (PRM)

Enclave Page Cache (EPC)

Enclave Page Cache Map (EPCM)

Enclave Page Cache (EPC)

EPC用来保存缓存Enclave页面和SGX的结构。

EPC分为4KB的块，称为EPC页面。

EPC页面可以被标识为有效或者无效。

每个EPC有EPCM进行管理。

每个EPC的页属于一个单独的enclave instance

物理内存的组织

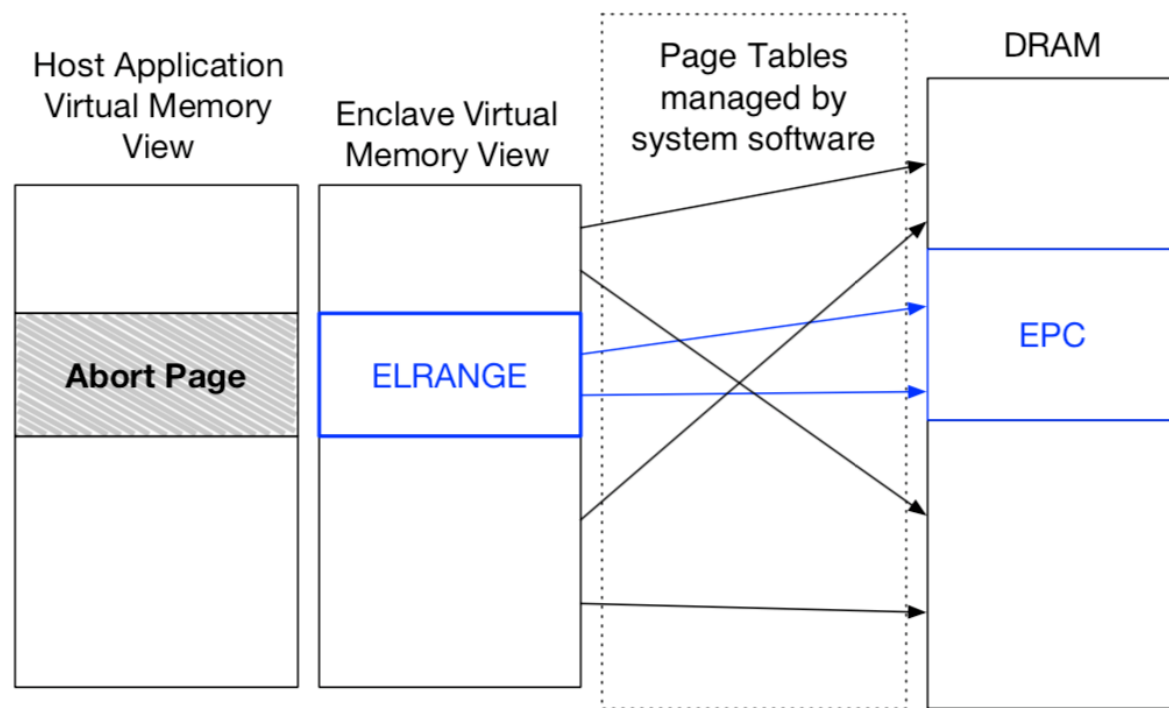
Enclave Page Cache Map (EPCM) 是处理器用来跟踪 EPC 内容的受保护结构。每个 EPCM 条目都包含以下信息：

- EPC 页面有效还是无效
- 拥有该页面的安全区实例。
- 页面的类型 (REG, TCS, VA, SECS)
- 允许飞地访问页面的虚拟地址x飞地在该页面上指定的读/写/执行权限
- 是否可以访问页面 (阻止或解锁)

Field	Bits	Description
VALID	1	0 for un-allocated EPC pages
PT	8	page type
ENCLAVESECS		identifies the enclave owning the page

SGX 将每个 enclave 元数据存储在 与每个 enclave 相关的安全区控制结构 SGX Enclave Control Structure (SECS) 中。

虚拟内存的组织



SGX设计可确保ELRANGE内部的安全区内内存访问服从虚拟内存抽象，而ELRANGE外部的内存访问不接受任何保证（因为无法信任系统软件）。因此，安全区必须将其所有代码和私有数据存储在ELRANGE内，并且必须将ELRANGE外部的内存视为与外界的不可信接口。

Enclave Linear Address Range (ELRANGE)

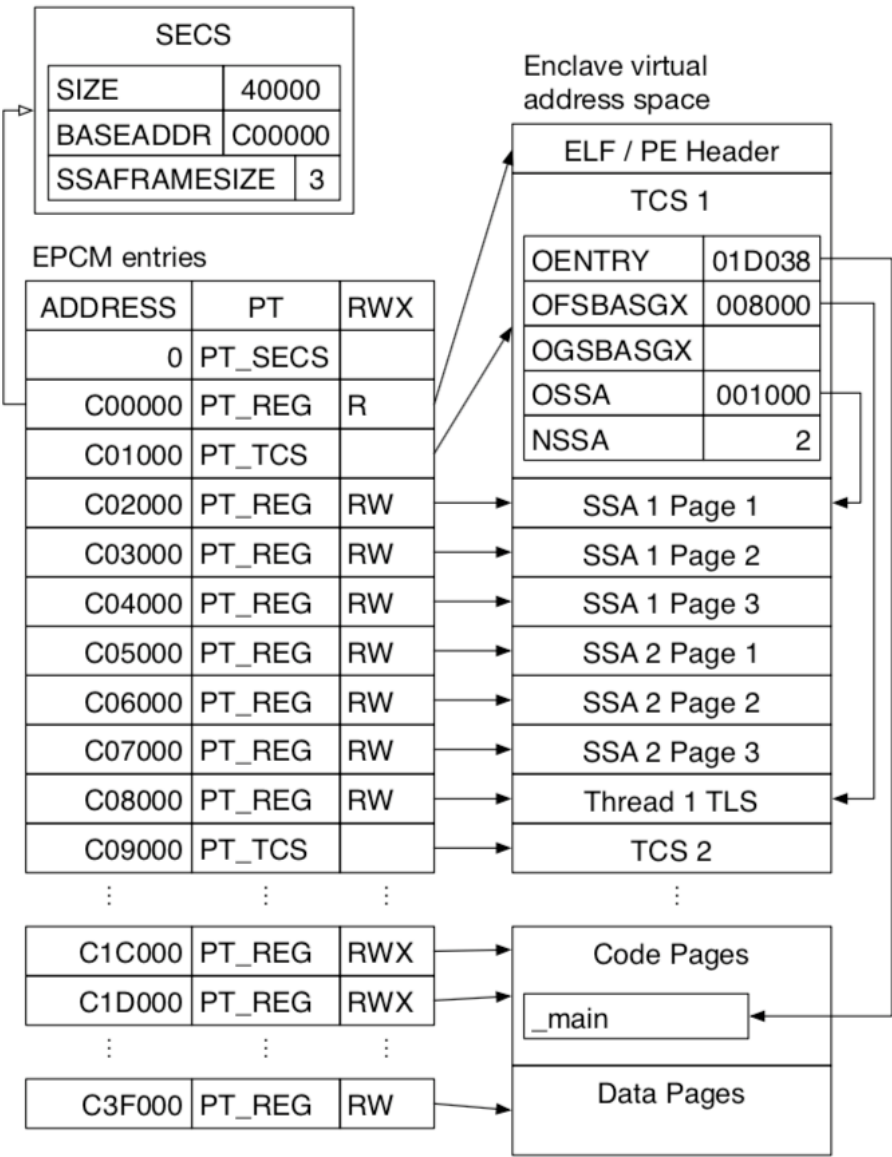
虚拟内存的组织

SGX设计完全支持多核处理器。多个处理器可以通过不同的线程同时执行同一安全区的代码。

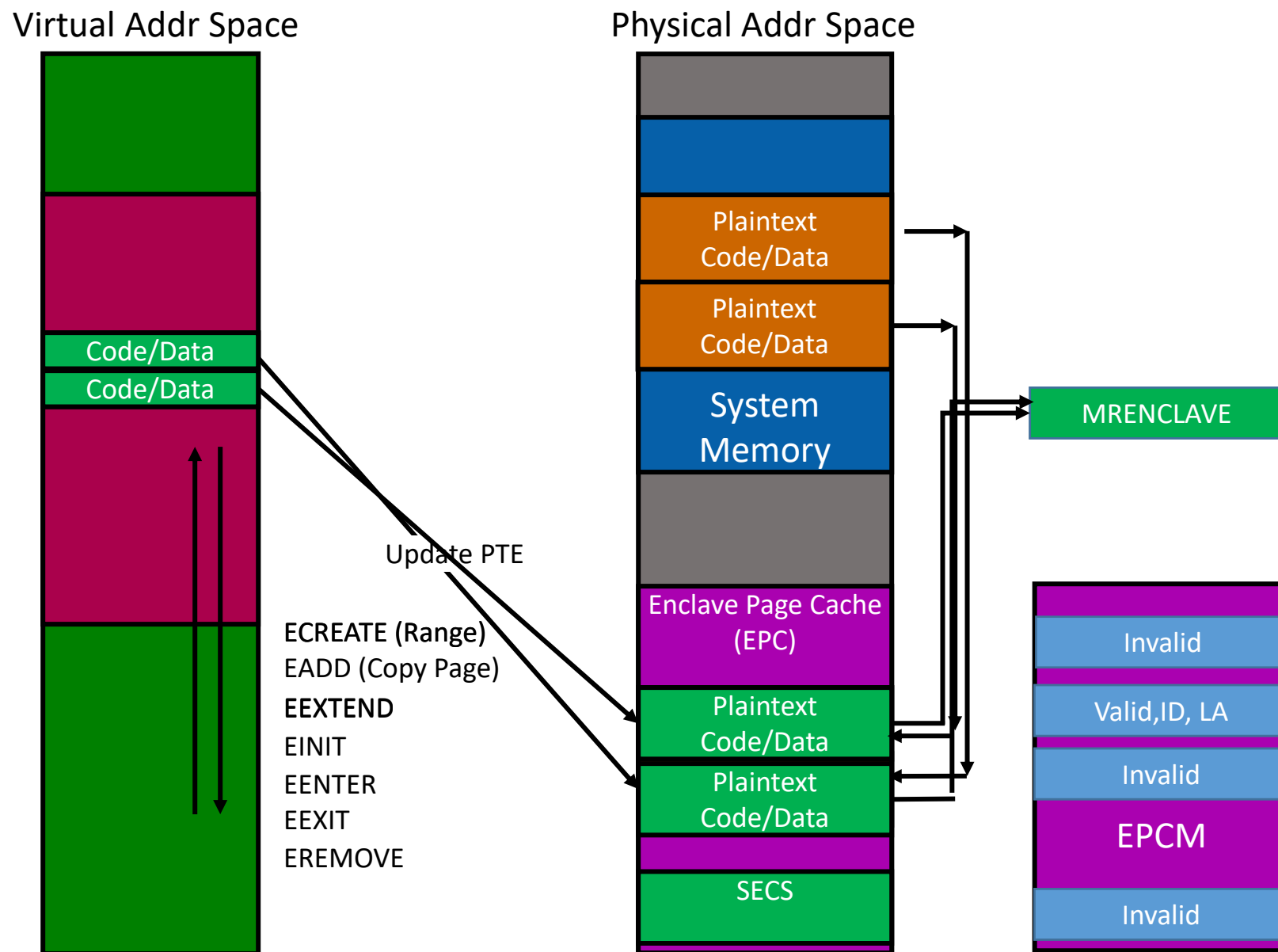
线程控制结构（TCS）控制不同的处理器。

State Save Area (SSA) 用于保存异常的Enclave中断。

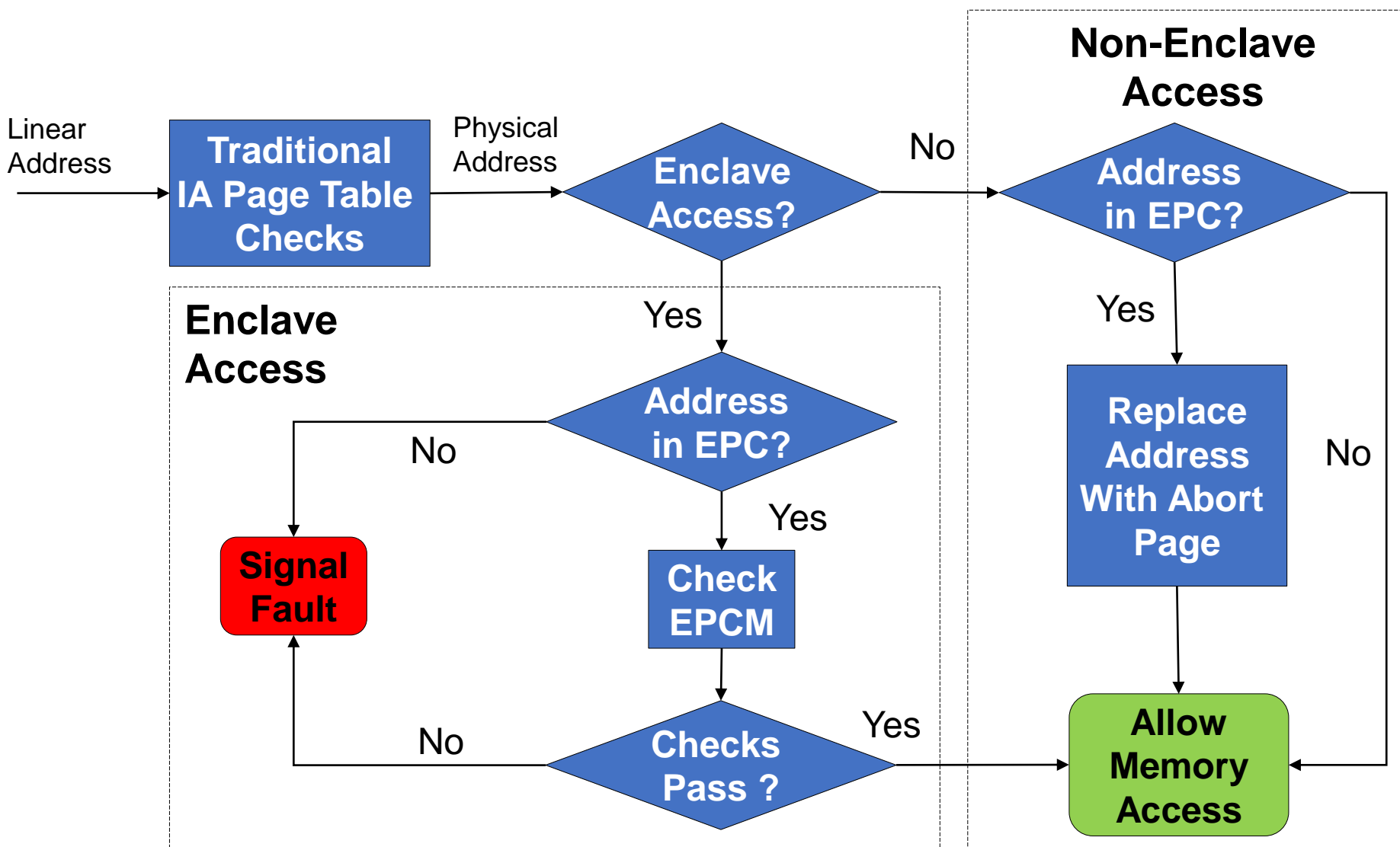
SGX Enclave Attributes 用户保存相关的属性。



映射关系



CPU如何保护EPC ?



Enclave设计

- 什么是 Enclave ?
- Enclave 的目标
- Enclave 技术架构
- Enclave 生命周期

区块链应用

- Ekiden 项目

01

系统概念

- 系统概览
- 物理内存
- 虚拟内存
- 工作原理

02

03

密码协议

- 工作流程
- 本地断言
- 远程断言

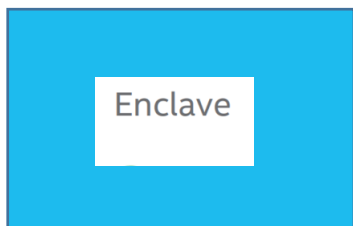
04



什么是 Enclave (飞地) ?

Intel is developing the Intel® Software Guard Extensions (Intel® SGX) technology, an extension to Intel® Architecture for generating protected **software containers**. The container is referred to as an enclave. Inside the enclave, software's code, data, and stack are protected by hardware enforced access control policies that prevent attacks against the enclave's content.

- Ittai Anati



被保护的容器，容器内可执行代码，
容器内的东西外部不可见。

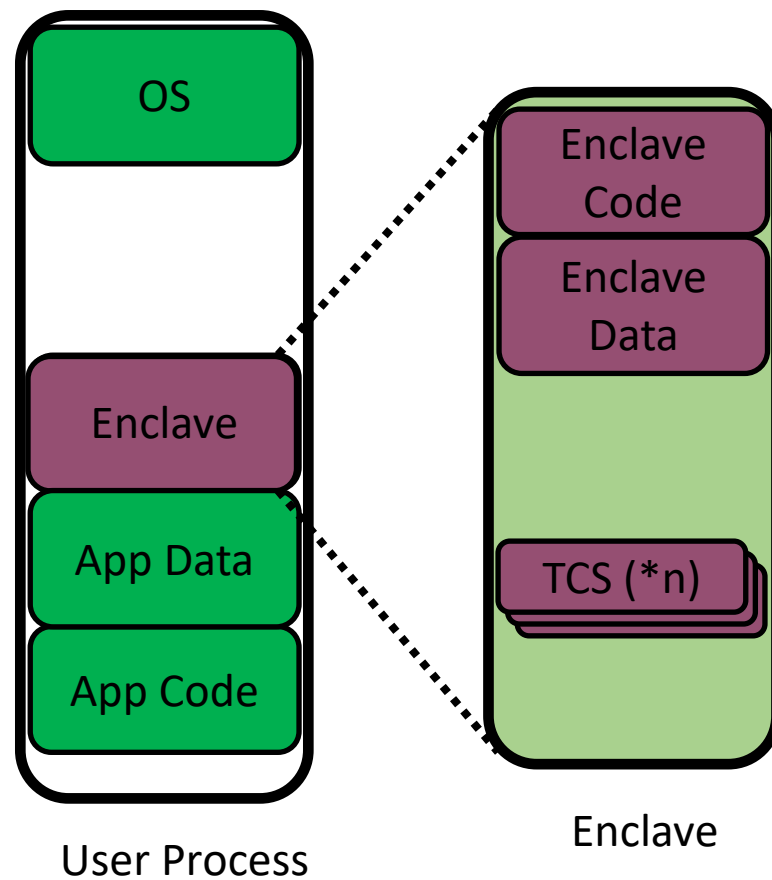
SGX与Enclave概览

什么是 Enclave (飞地) ?

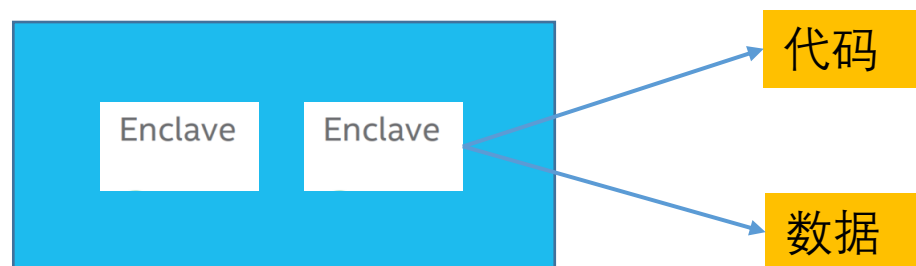
SGX基础设施CPU



配备SGX处理器的机器



Enclave 的目标



程序在Enclave内部是透明状态

Enclave 实例可以检测到被篡改的代码/数据。

Enclave 实例提供软件攻击的代码/数据的机密性。

Enclave 实例之间提供隔离。

Enclave 可以防止软件重放攻击。

Enclave 创建和销毁指令

创建的相关指令

Instruction	Description
ECREATE	Declare base and range, start build
EADD	Add 4k page
EEXTEND	Measure 256 bytes
EINIT	Declare enclave built
EREMOVE	Remove Page

销毁的相关指令

Instruction	Description
EENTER	Enter enclave
ERESUME	Resume enclave
EEXIT	Leave enclave
AEX	Asynchronous enclave exit

ECREATE启动安全区创建过程，并初始化安全区控制结构（SECS），该结构包含有关安全区的全局信息。EADD将EPC页面提交到一个飞地并记录该承诺，但不记录SECS中的内容。飞地的内存内容由EEXTEND显式测量。EINIT完成了创建过程，该过程最终确定了飞地测量并建立了飞地标识。在执行EINIT之前，不允许进入Enclave。

EINIT创建两个身份：Enclave identity 和 Sealing identity。

Enclave Identity 和 Sealing Identity

英特尔®SGX架构负责建立用于证明和密封的身份。对于每个Enclave，它都提供两个测量寄存器：MRENCLAVE和MRSIGNER。MRENCLAVE在构建时提供围场代码和数据的标识，而MRSIGNER在围场上提供权限的标识。这些值在构建安全区时记录下来，并在开始执行安全区之前完成。只有TCB有权写入这些寄存器，以确保在证明和密封时可以准确反映身份。

飞地具有用于数据保护的第二个身份，称为“密封身份”。同一个SGX的 Sealing Identity 相同，可以共享和迁移其密封的数据。



Enclave设计

- 什么是 Enclave ?
 - Enclave 的目标
- Enclave 技术架构
- Enclave 生命周期

区块链应用

- Ekiden 项目

01

系统概念

- 系统概览
- 物理内存
- 虚拟内存
- 工作原理

02

03

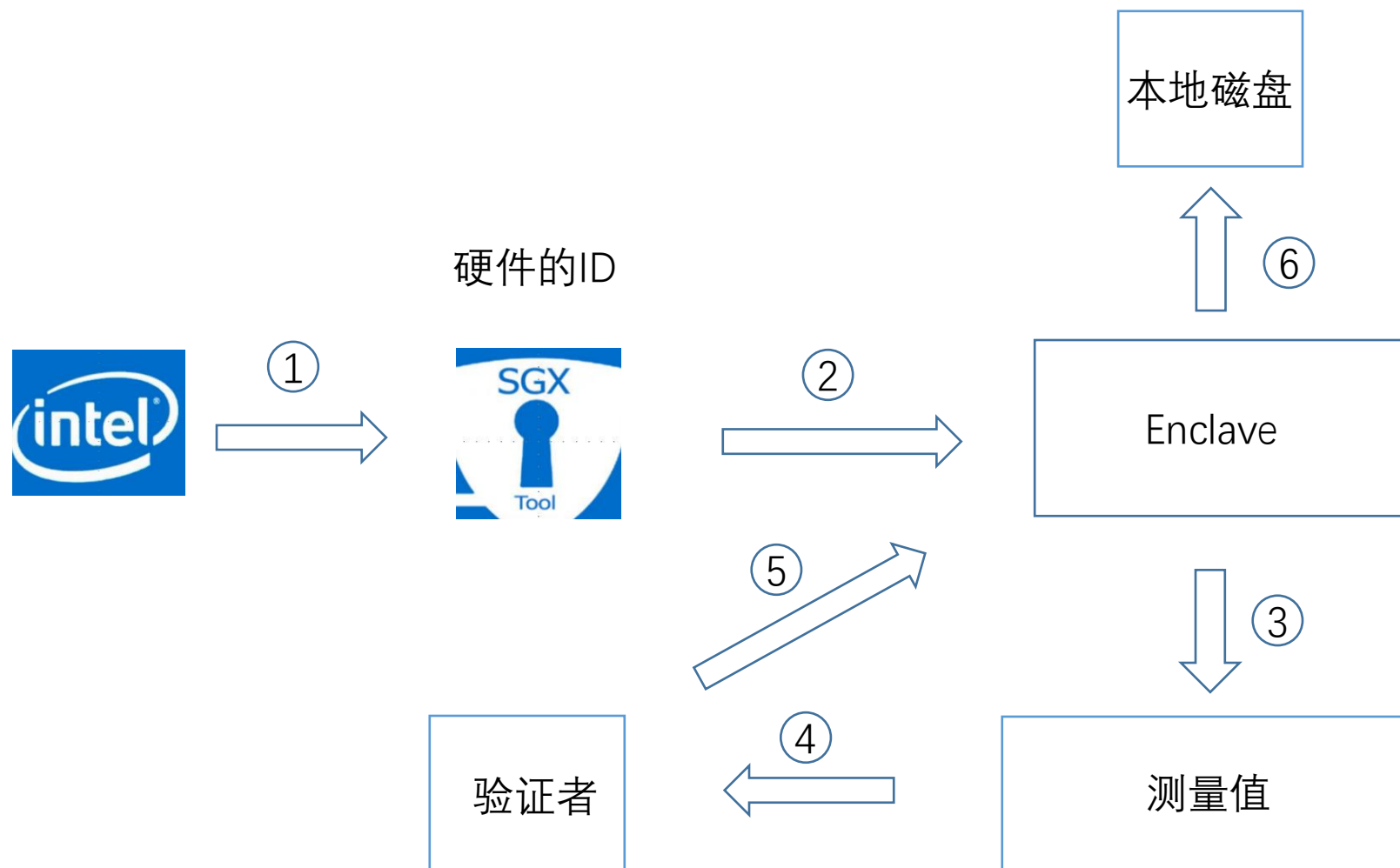
密码协议

- 工作流程
- 本地断言
- 远程断言

04

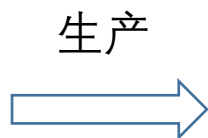
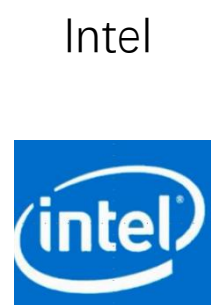


工作流程



1. Intel 签发SGX硬件ID
2. SGX 创建飞地
3. 对飞地进行测量
4. 本地/远程断言
5. 建立安全管道
6. 导出到本地

Intel 签发硬件ID

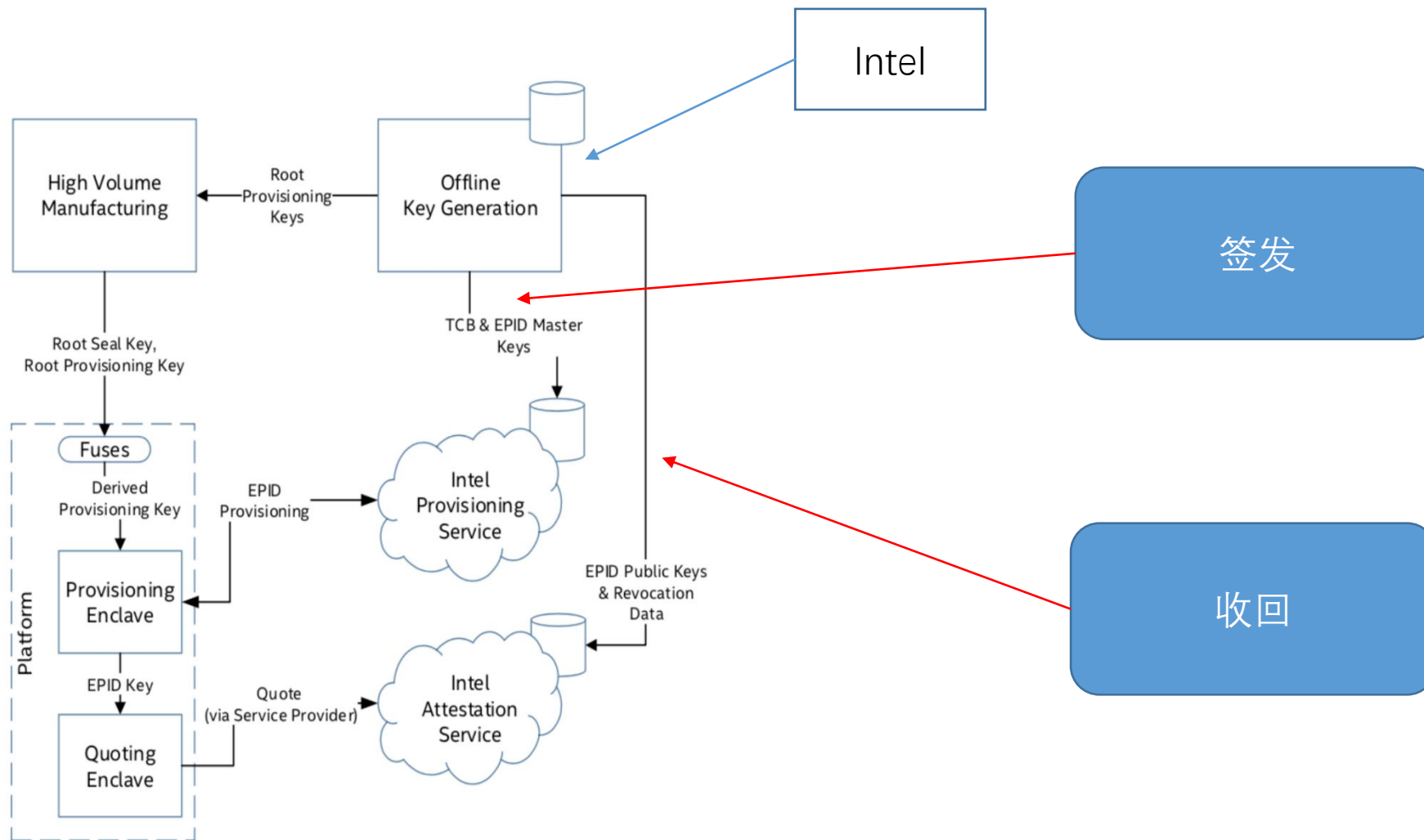


硬件的ID



- 工作流程
- 硬件ID的隐私签发
- 硬件ID的隐私回收
- 实际应用案例


硬件签发工作流程



硬件ID的隐私签发



生产



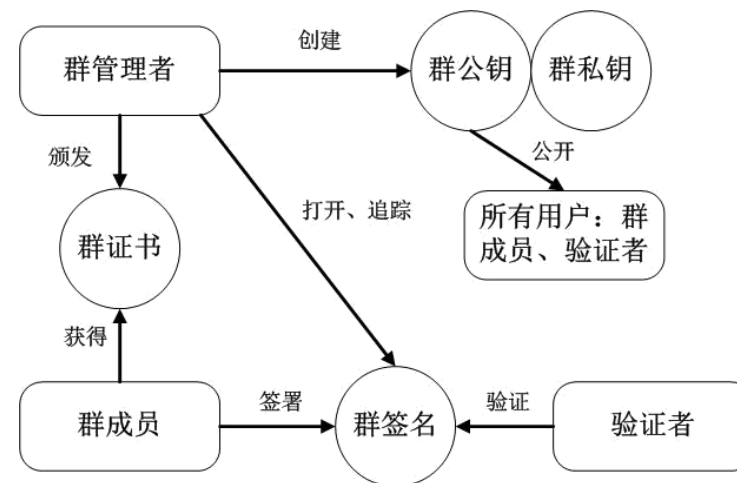
硬件的ID



正常情况下，Intel
直接进行签名

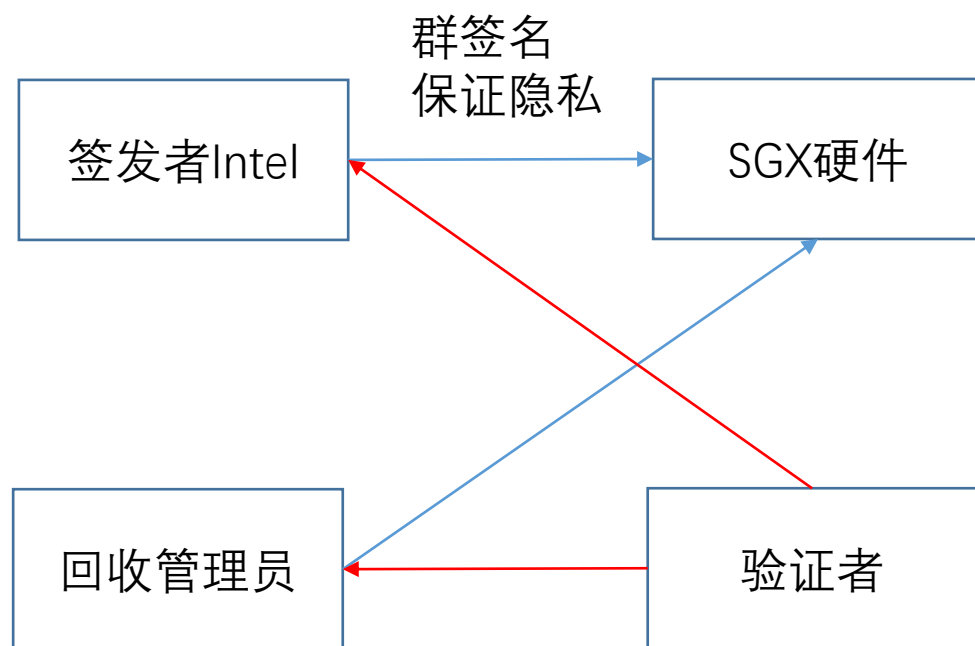
造成隐私的问题，
Intel 知道了所有人

实际上使用的是群签名。



Each EPID group contains a million CPUs of the same type (e.g. Core i3, i5, or i7). Verifying a signature does not enable you to identify the signer but instead verifies the signer as a member of a valid EPID group. Thus the signature has the attribute of **anonymity**.

1.2 硬件ID的回收



- 基于私钥的回收
- 基于签名的回收
- 验证是否有效
- 验证是否被回收

IEEE International Conference on Social Computing / IEEE International Conference on Privacy, Security, Risk and Trust

Enhanced Privacy ID from Bilinear Pairing for Hardware Authentication and Attestation

Ernie Brickell
Intel Corporation
Hillsboro, USA
Email: ernie.brickell@intel.com

Jiangtao Li
Intel Labs
Hillsboro, USA
Email: jiangtao.li@intel.com

Abstract—Enhanced Privacy ID (EPID) is a cryptographic scheme that enables the remote authentication and attestation of a hardware device while preserving the privacy of the device. EPID can be seen as a direct anonymous attestation scheme with enhanced revocation capabilities. In EPID, a device can be revoked if the private key embedded in the hardware device has been extracted and published widely so that the revocation manager finds the corrupted private key. In addition, the revocation manager can revoke a device based on the signatures the device has created, if the private key of the device is not known. In this paper, we introduce a new security notion of EPID including the formal definitions of anonymity and unforgeability. We also give a construction of an EPID scheme from bilinear pairing. Our EPID scheme is efficient and provably secure in the random oracle model under the strong Diffie-Hellman assumption and the decisional Diffie-Hellman assumption.

Index Terms—hardware authentication; trusted computing; privacy; anonymity; direct anonymous attestation; cryptographic protocol

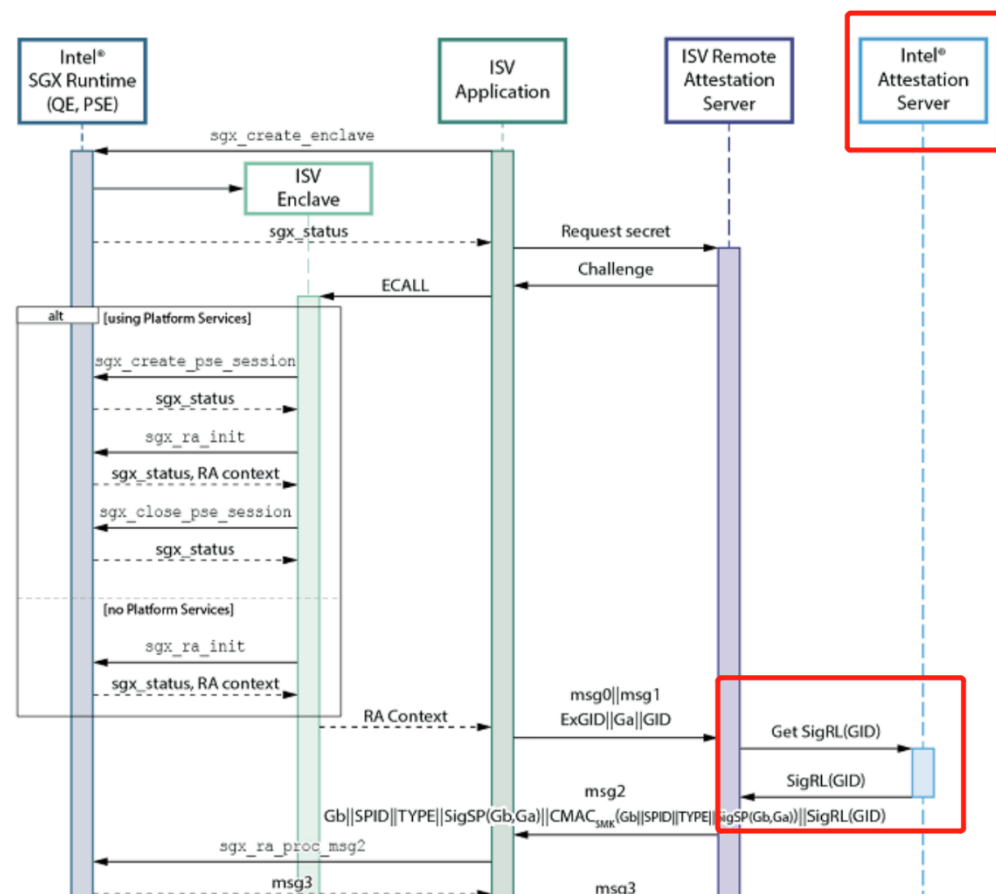
I. INTRODUCTION

Consider the following authentication problem: a hardware device (e.g., a graphics chip, a trusted platform module, a mobile device, a smart phone, etc.) wants to authen-

in DAA. If the named base option in DAA is used, it can allow revocation based on signatures for all uses of the same named base, but it has the unfortunate property of removing the anonymity for all uses with the same named base. To get around the problem of the limited revocation properties of DAA, Brickell and Li [12] introduced the notion of Enhanced Privacy ID (EPID). In EPID, the revocation manager can revoke a hardware device based on the signatures that were signed by the private key of the device, without reducing the anonymity properties. The EPID scheme will have broader applicability beyond attestation and the TCG application. More motivations about EPID can be found in [13].

In an EPID scheme, there are four types of entities: an issuer, a revocation manager, platforms, and verifiers. The issuer could be the same entity as the revocation manager. The issuer is in charge of issuing membership to platforms, i.e., each platform obtains a unique private key from the issuer through a join process. A platform can prove membership to a verifier by creating a signature using its private key. The verifier can verify membership of the platform by verifying the

回收使用案例



在使用之前要确认此SGX对应的ID是否已经被回收。

Intel 担任了回收管理员的角色
(中心化的问题)

SGX 创建飞地

硬件的ID



飞地的ID/ Sealing ID



2.1. SGX创建飞地过程

2.2 SGX CPU 主密钥

2.2 SGX 密钥来源

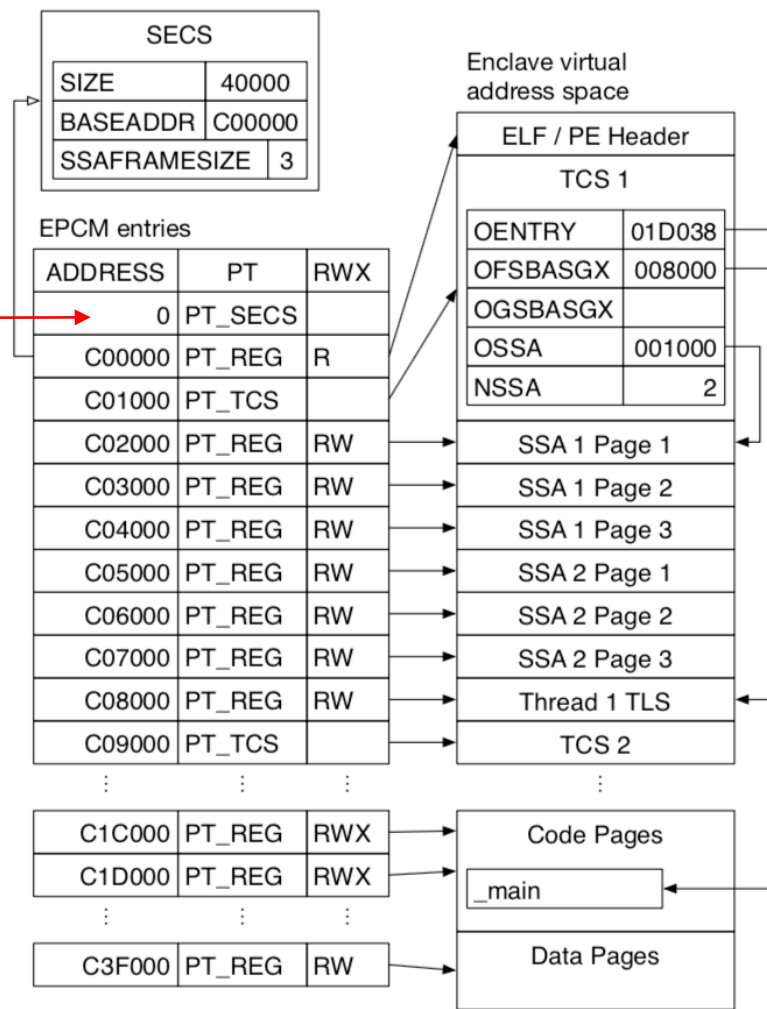
创建的步骤

初始化SECS页面

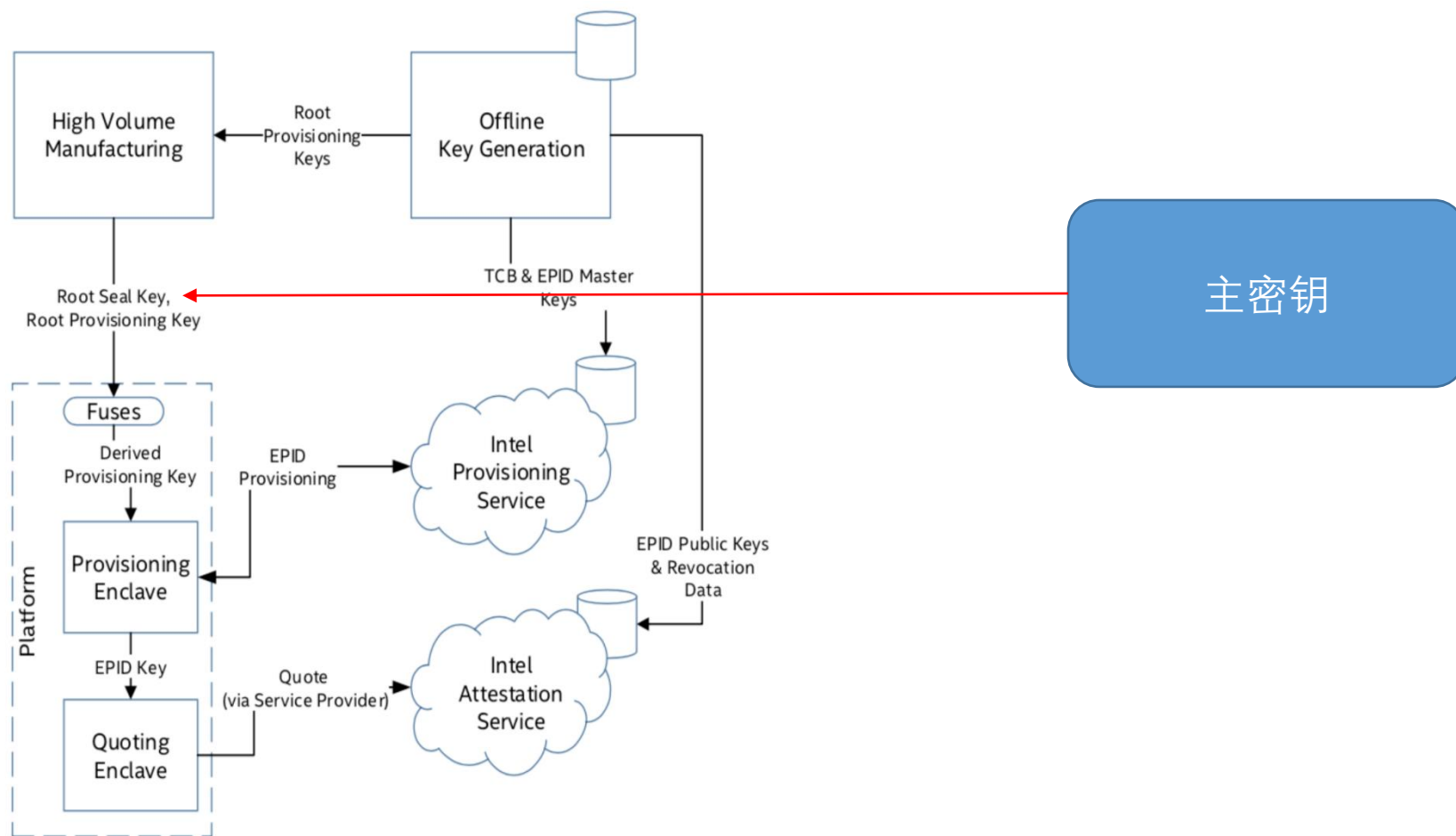
空闲的EPC页面转换为SECS页面并初始化结构。
部分的EPC页面转换给SECS页面。

EINIT创建，创建成功后建立两个身份

- Enclave identity
- Sealing identity



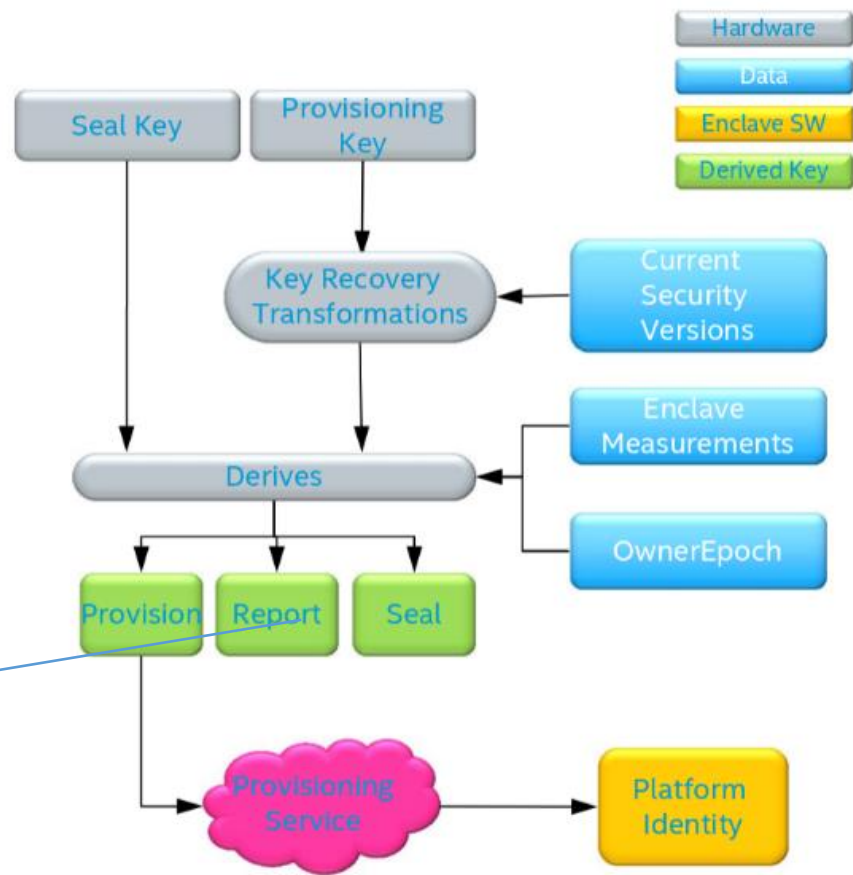
SGX CPU 主密钥



SGX CPU 主密钥

在生产环境中，有两个设备根密钥已融合到SGX CPU

- 根供应密钥（RPK）该密钥是有专用硬件安全模块（HSM）随机生成的。
- 根密封密钥（RSK）在生产过程中，此密钥是在CPU内部自动随机生成的，在统计上因部件而异。英特尔宣布将尝试擦除该密钥的所有生产线残留，以便每个平台都应假定其RSK值既独特又仅为自身所知。



不同的KEY不同的用途
不同的KEY可能与周期有一定的关系

SGX 密钥来源

在生产环境中，有两个设备根密钥已融合到SGX CPU

- 根供应密钥（RPK）英特尔负责维护由HSM生成的所有密钥的数据库。RPK交付给不同的工厂设施，这些设施被英特尔的正式出版物称为“大批量生产系统”，可以集成到处理器的保险丝中。英特尔存储所有RPK，因为它们都是SGX处理器通过在线配置协议展示其真实性质的基础。
- 根密封密钥（RSK）在生产过程中，此密钥是在CPU内部自动随机生成的，在统计上因部件而异。英特尔宣布将尝试擦除该密钥的所有生产线残留，以便每个平台都应假定其RSK值既独特又仅为自身所知。Enclave的受信界面提供的大多数密钥都是基于平台的RSK派生的，因此其他任何一方都无法知道这些密钥。

私钥有Intel保存

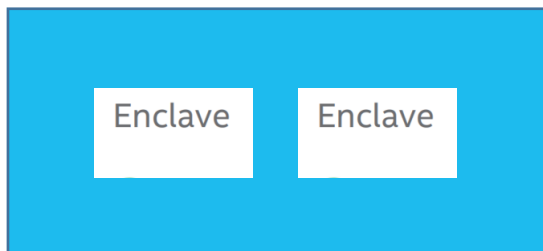
硬件随机数产生

3.对飞地进行测量

硬件的ID



飞地的ID/ Sealing ID



- 可信计算基
- 测量相关的指令

可信计算基 (TCB)

可信计算基（英语：Trusted computing base, TCB）是指为实现计算机系统安全保护的所有安全保护机制的集合，机制可以硬件、固件和软件的形式出现。

在Enclave 创建时生成了生成日志，该日志记录构建安全区时完成的所有活动。该日志包含以下信息：

1. 页面的内容（代码，数据，堆栈，堆）。
2. 区域中页面的相对位置。
3. 与页面关联的所有安全标志

The “Enclave Identity”, which is a 256-bit hash digest of the log, is stored as **MRENCLAVE** as the enclave’s software TCB

测量相关指令

- EREPORT 指令：

飞地中代码和数据的测量。

飞地初始化时显示的ISV证书中公钥的哈希值。

用户数据。

其他与安全性有关的状态信息（此处未描述）。

上述数据的签名块，可以由生成报告的同一平台进行验证。

- EREPORT指令：

该指令生成一个称为REPORT的加密结构，该结构将MRENCLAVE绑定到目标安全区的REPORT KEY。

- REPORT KEY指令：

由EREPORT用来对在该特定平台上生成的所有报告进行签名。（所有的REPORT 共享）

测量相关指令

- EGETKEY 指令:

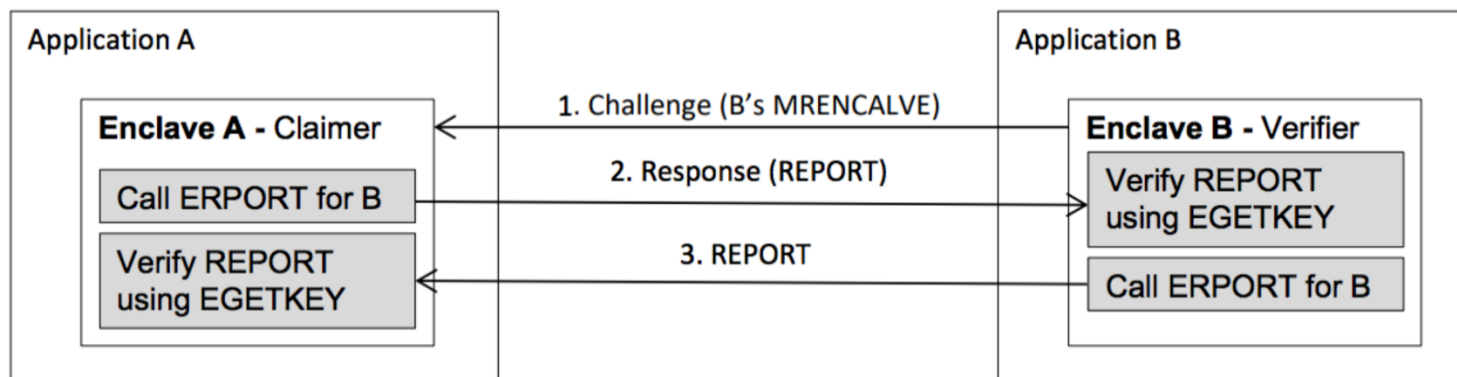
Enclaves可以使用EGETKEY指令来获取设备密钥的派生类。EGETKEY会根据调用安全区属性和请求的密钥类型来产生用于不同目的对称密钥。有五种不同的密钥类型。两个是所有Enclaves均可使用的密封Sealing 和报告Report 密钥。其余仅限于SGX飞地。

EGETKEY使用请求区域指定的安全版本号 (SVN) 定义请求的密钥特征。CPU SVN反映处理器微码版本，或ISV SVN反映飞地软件版本。EGETKEY对照存储在SIGSTRUCT中的值检查这些值，并且仅允许获得SVN值小于或等于调用区域的SVN值的密钥，以便同一软件的升级版本可以检索以前版本创建的密钥。

- SIGSTRUCT 指令:

飞地的证书称为SIGSTRUCT，是发起任何飞地的强制性补充。SIGSTRUCT将飞地的MRENCLAVE与其他飞地属性一起保存。SIGSTRUCT由ISV用其私钥签名，该私钥最初是由SGX启动机构签发的。英特尔被认为是主要的飞地发射机构。

本地断言

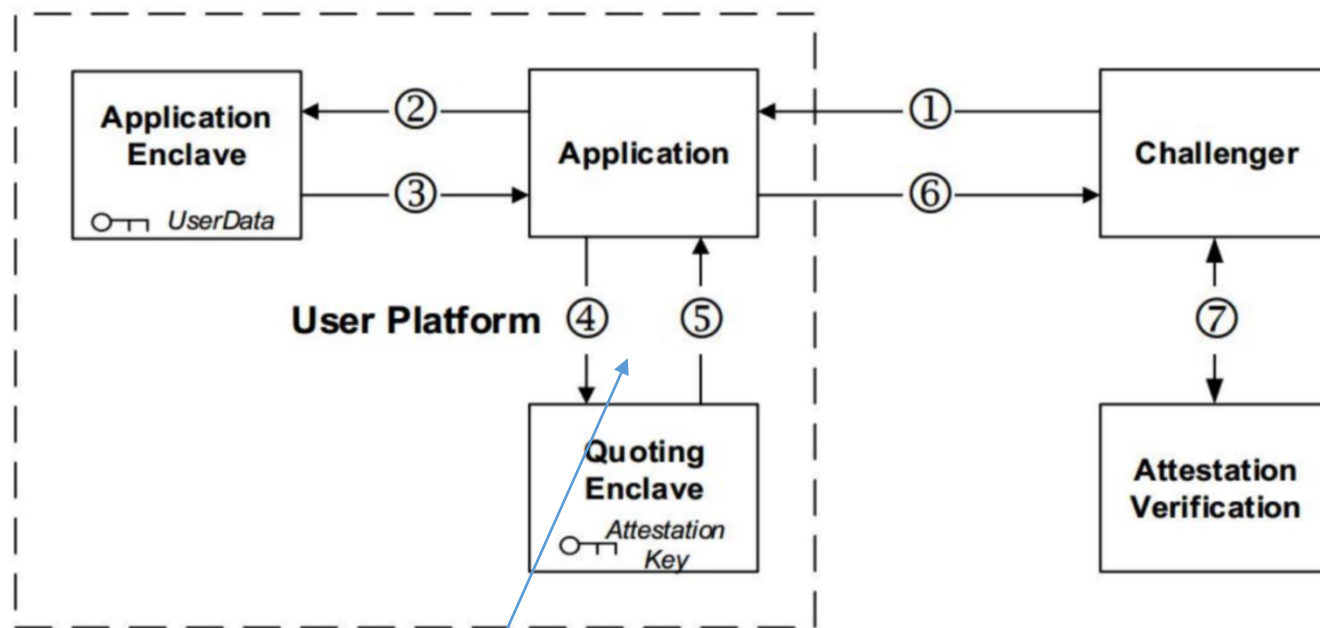


在同一平台上有两个**ENCLAVE**，分别称为安全区A和安全区B。我们假设它们已经在彼此之间建立了通信路径，并且该路径不需要被信任。我们假设log要求B要求A证明其与B在同一平台上运行。

- B检索其**MRENCLAVE**值，并通过不受信任的通道将其发送给A。
- A使用**EReport**指令使用B的**MRENCLAVE**为B生成报告。然后，A将此报告发送回B。A还可以将Diffie-Hellman密钥交换数据包括在REPORT中，作为将来创建受信任通道的用户数据。
- B从A接收到REPORT后，B调用**EGETKEY**指令获取REPORT KEY来验证REPORT。如果可以使用REPORT KEY验证REPORT，则B确保A与B在同一平台上，**因为REPORT KEY特定于平台**。
- B使用从A的**报告中**接收到的**MRENCLAVE**为A创建另一个REPORT，并将该REPORT发送给A。
- A也可以与步骤4相同，以确保Verify B与A在同一平台上。

通过利用**REPORT**的用户数据字段，A和B可以使用Diffie-Hellman密钥交换创建安全通道。信息交换可以通过共享对称密钥进行加密。

远程断言（组成）



Provisioning Enclave (PvE)

组成部分

服务提供（挑战者）

带有飞地和QE的应用程序

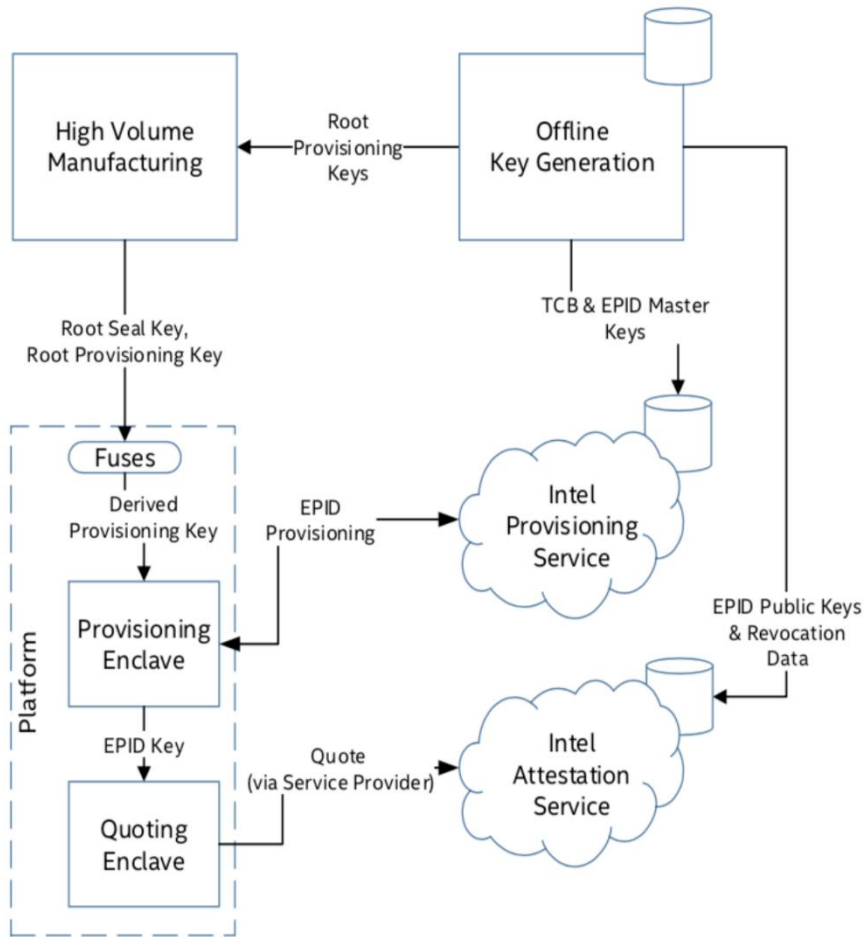
验证飞地的英特尔认证服务（IAS）

远程认证Provisioning过程

为了将本地REPORT转换为可远程验证的QUOTE, **Quoting Enclave** 使用平台唯一的非对称证明密钥。然后, 远程方可以使用相应的公钥来验证QUOTE。那么QE首先如何获得该证明密钥?

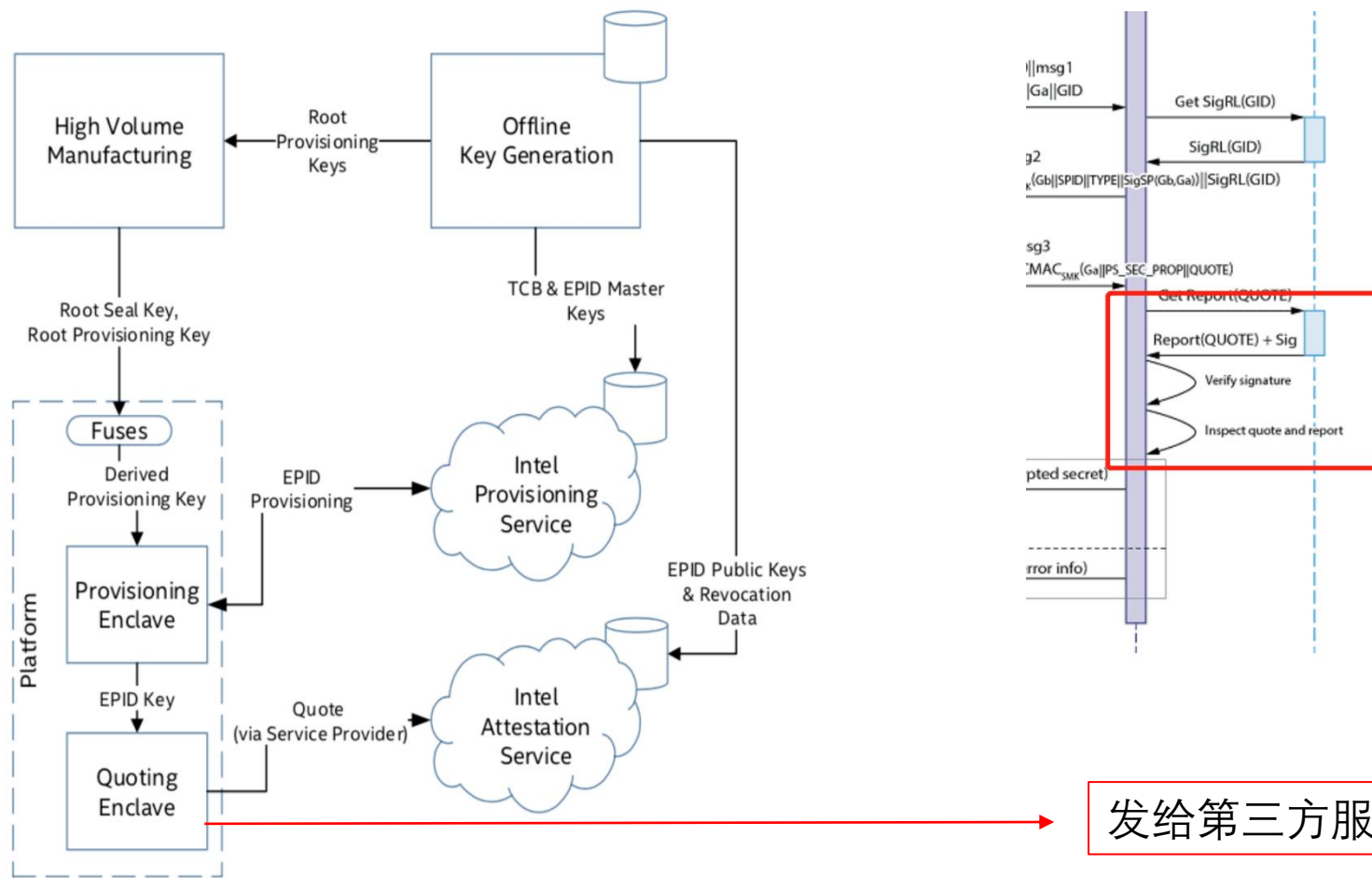
Provisioning 是指SGX设备向英特尔展示其真实性以及其CPU SVN和其他系统组件属性的过程, 以便获得真实反映其SGX和TCB版本的适当证明密钥。通常, **Provisioning** 是在平台初始设置阶段完成的, 但是由于存在漏洞, 由于更新了关键的系统组件 (例如固件, BIOS或微码), 因此也可以在购买后进行重新**Provisioning**。在这种情况下, 可以替换证明密钥以反映平台更新的TCB安全级别。

4.4 Provisioning Enclave (PvE)



PvE通过使用几种SGX特权密钥类型证明了其真实性，这些类型只能通过SGX体系结构区域通过EGETKEY指令进行访问。这些密钥中的两个是预配密钥（PK）和预配密封密钥（PSK）。PvE和QE的独特性基于英特尔（MRSIGNER）签署的SIGSTRUCT证书。因此，这些飞地被授权使用特权属性启动，以便稍后通过执行EGETKEY指令获得特殊键

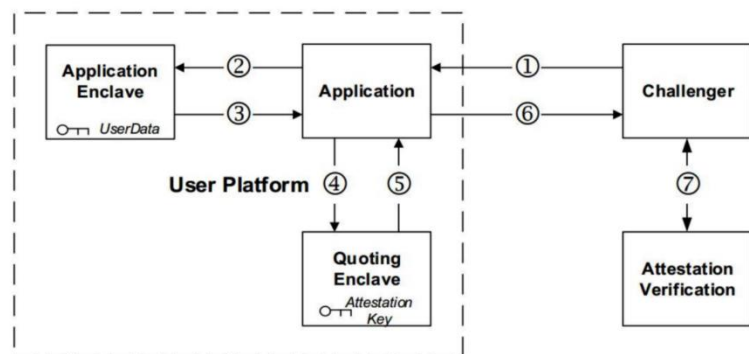
Provisioning Enclave (PvE)



<https://software.intel.com/zh-cn/articles/code-sample-intel-software-guard-extensions-remote-attestation-end-to-end-example>

发给第三方服务，可以进行验证

远程断言（步骤）



具体步骤

1和2. 服务提供者构造一个质询消息，该质询消息包括其SPID，生动的随机随机数，更新的SigRL和可选的基名参数（如果需要假名签名）。

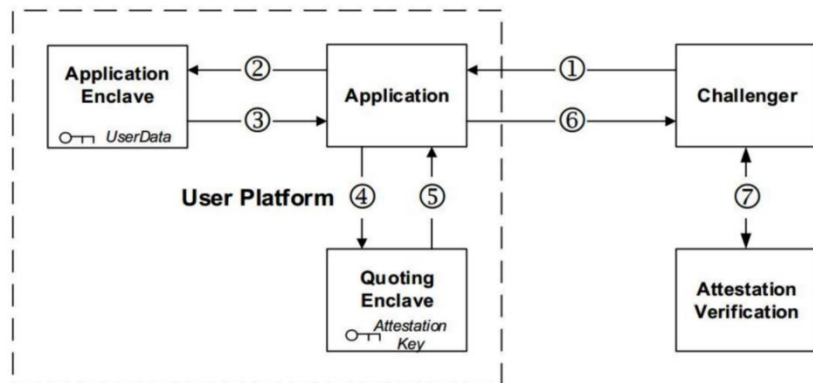
3.如果Enclave 支持请求的签名模式，它将调用EREPORT指令以创建针对平台QE的本地可验证报告。为了在Enclave和服务提供商之间建立经过身份验证的安全通道，可以将新生成的临时公共密钥添加到报告的用户数据字段中。该报告和SP的挑战已发送给QE。

4 QE调用EGETKEY以获得REPORT KEY并验证报告。如果成功，则QE再次调用EGETKEY来接收平台的Provisioning Seal密钥，以解密平台的远程证明密钥（EPID私钥）。

5首先根据要求的证明模式，通过对质询的基本名称或随机值进行签名，将证明密钥用于生成身份签名。

6.然后，将证明密钥用于计算平台的身份签名（MRENCLAVE）上的两个知识签名。第一个证明身份签名是由英特尔认证的密钥签名的。第二个是不可撤销的证明，证明用于身份签名的密钥不会创建被质询的SigRL中列出的任何身份签名。然后，使用IAS的公钥生成最终的QUOTE并对其进行加密，并在QE中对其进行硬编码，然后将结果发送回证明区域。QUOTE包含证明区域的标识，执行模式详细信息（例如SVN级别）和其他数据。

远程断言（步骤）



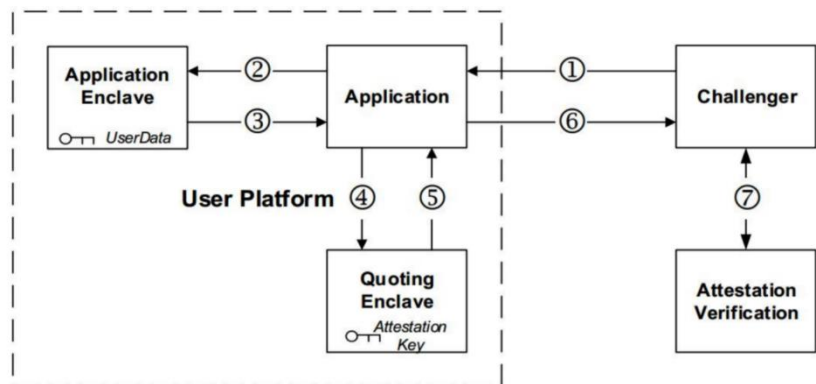
具体步骤

7.然后，安全区将QUOTE转发给SP进行验证。由于QUOTE是加密的，因此只能由Intel进行验证。因此，服务提供商只需将QUOTE转发给IAS进行验证。

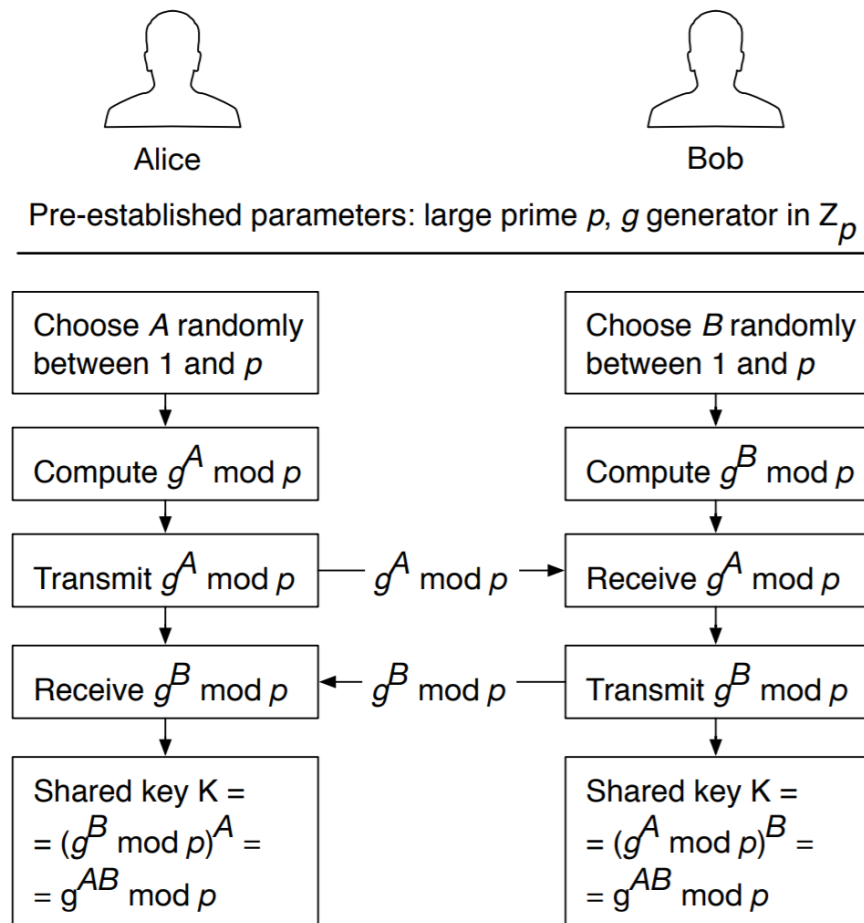
IAS通过首先根据其身份签名验证其EPID证明来检查QUOTE。然后，通过为列表中的每个私钥计算QUOTE基名上的身份签名，并验证它们均不等于QUOTE的身份签名，从而验证平台未在Priv-RL组中列出。这样就完成了平台的有效性检查，然后IAS创建一个新的证明验证报告作为对SP的响应。证明验证报告包括平台为证明区域提供的QUOTE结构。

最后，SP将确认飞地在正版intel SGX处理器上运行了特定代码。然后，SP负责验证ISV（独立软件供应商）安全区身份并向平台提供适当的响应。

Diffie-Hellman密钥交换（远程断言）



第七步完成之后开始，
Challenger 和 Application
之间开始建立联系



ENCLAVE 转移到磁盘

英特尔SGX提供了操作“密封密钥”的指令。该指令可以导出“密封密钥”加密后的密文，可以保存在外部（磁盘）。



图片来源: 引用1

Enclave设计

- 什么是 Enclave ?
 - Enclave 的目标
- Enclave 技术架构
- Enclave 生命周期

区块链应用

- Ekiden 项目

01

系统概念

- 系统概览
- 物理内存
- 虚拟内存
- 工作原理

02

03

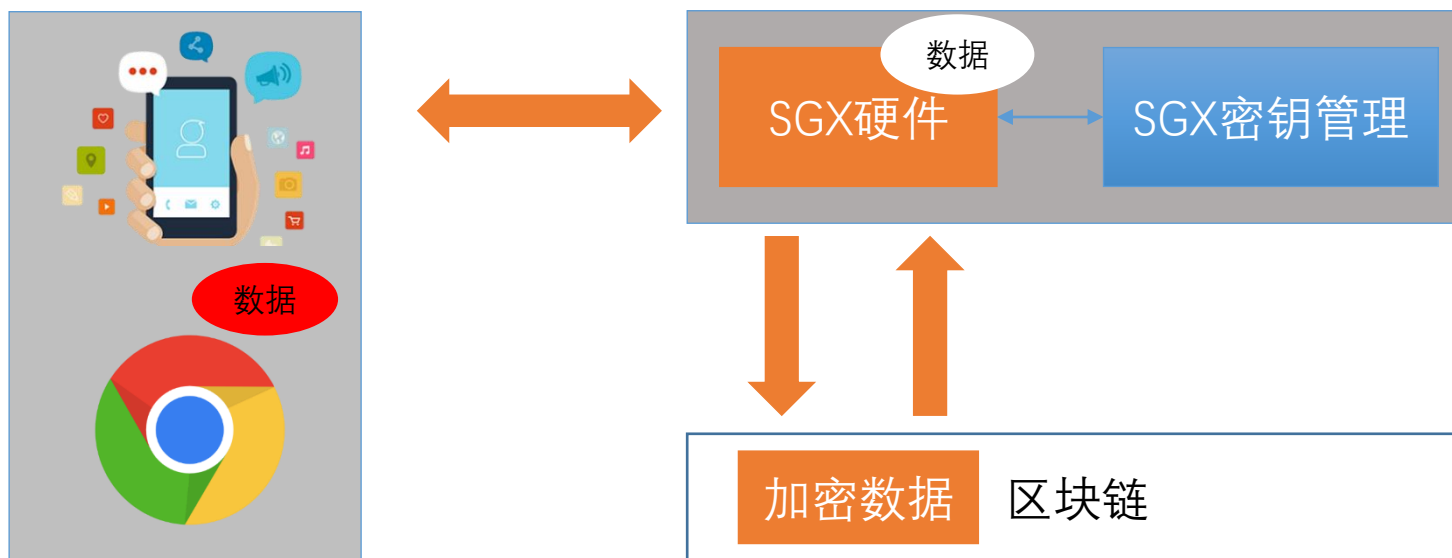
密码协议

- 工作流程
- 本地断言
- 远程断言

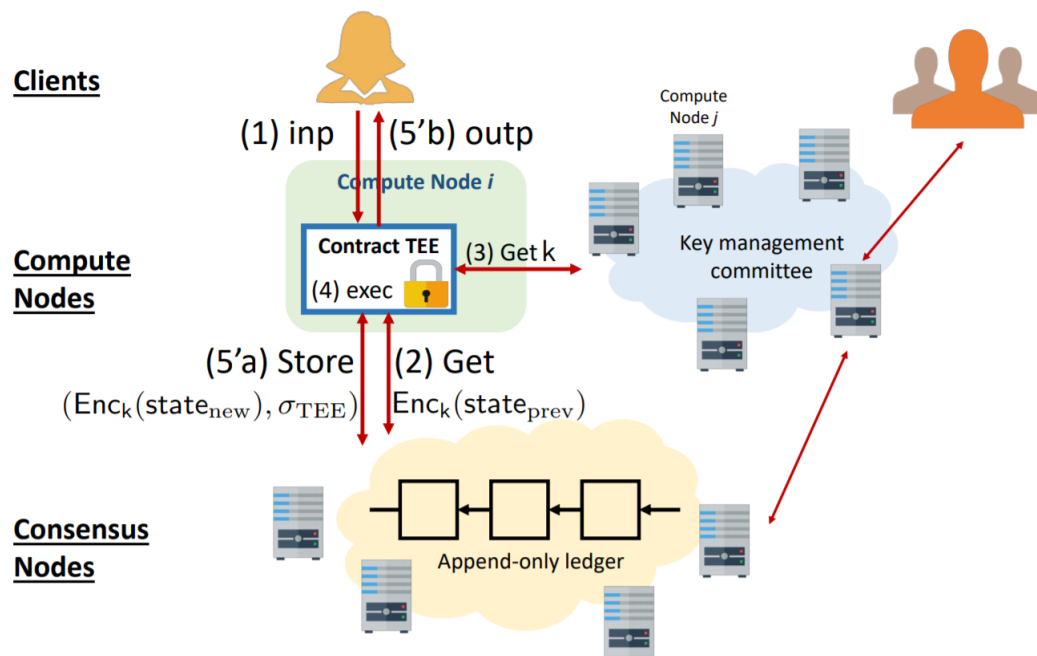
04



SGX和区块链的结合



案例：Ekiden 项目



客户读取从Enclave获取加密数据

计算节点配备SGX，负责创建Enclave

共识节点负责执行共识算法

Key committee 负责私钥的管理

相关引用

1. <https://software.intel.com/sites/default/files/managed/c3/8b/intel-sgx-product-brief-2019.pdf>
2. https://bournetocode.com/projects/GCSE_Computing_Fundamentals/pages/3-4-4-sys_arc.html
3. http://www.sgx101.com/portfolio/attestation_primitives/
4. <https://software.intel.com/en-us/blogs/2016/12/20/overview-of-an-intel-software-guard-extensions-enclave-life-cycle>
5. <https://software.intel.com/en-us/forums/intel-software-guard-extensions-intel-sgx/topic/754783>
6. <https://software.intel.com/sites/default/files/managed/48/88/329298-002.pdf>
7. <https://software.intel.com/zh-cn/articles/code-sample-intel-software-guard-extensions-remote-attestation-end-to-end-example>
8. <https://software.intel.com/sites/default/files/article/413939/hasp-2013-innovative-technology-for-attestation-and-sealing.pdf>
9. http://www.sgx101.com/portfolio/remote_attestation/
10. <https://www.blackhat.com/docs/us-16/materials/us-16-Aumasson-SGX-Secure-Enclaves-In-Practice-Security-And-Crypto-Review.pdf>
11. <https://intel-epid-sdk.github.io/download/enhanced-privacy-id-from-bilinear-pairing-hardware.pdf>

Intel SGX原理及其区块链应用研究

李佳轩

2020年7月20日

南方科技大学