可信计算，MPC与数字资产安全

弦冰科技首席科学家　何剑虹

2020 年 12 月 30 日

1 背景

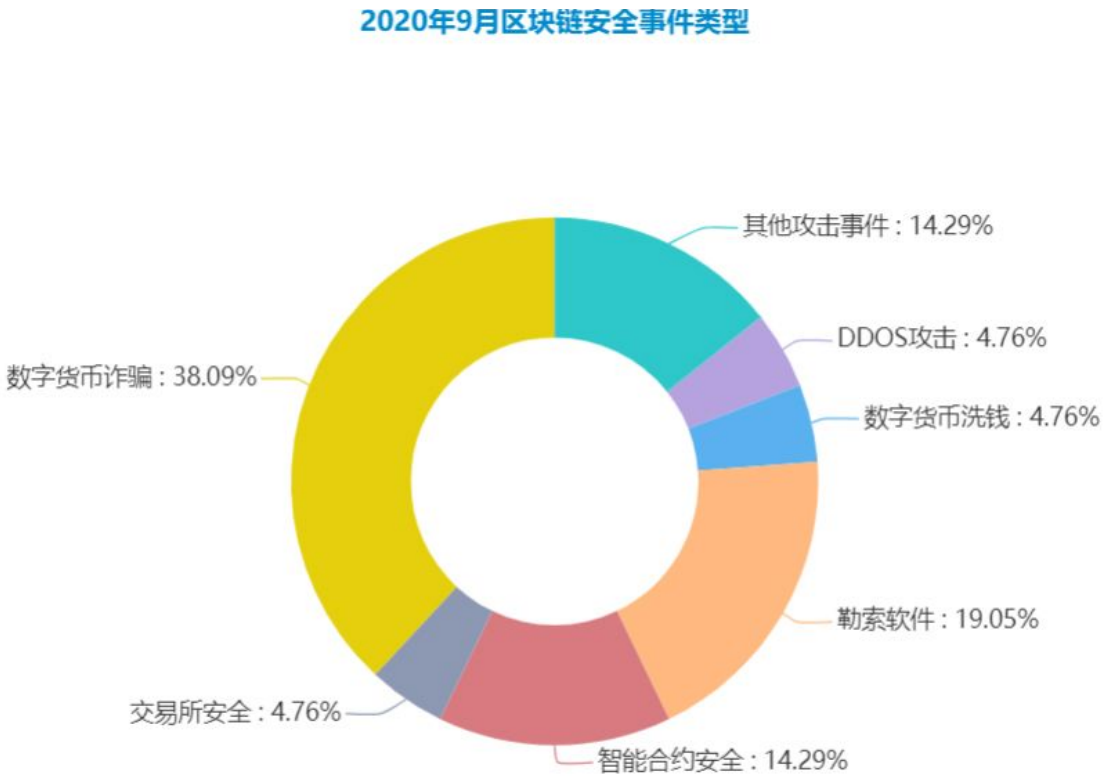
2 可信计算

3 安全多方计算

4 数字资产存管方案

5 MPC钱包演示

1 背景


2020年9月区块链安全事件类型
其他攻击事件 : 14.29%
DDOS攻击 : 4.76%
数字货币洗钱 : 4.76%
勒索软件 : 19.05%
智能合约安全 : 14.29%
交易所安全 : 4.76%
数字货币诈骗 : 38.09%

2 可信计算技术

- 2.1 定义可信计算
- 2.2 工业级可信计算硬件
- 2.3 核心概念
- 2.4 Intel SGX介绍
- 2.5 Intel SGX开发框架
- 2.6 TEE中的数据可迁移性
- 2.7 典型应用

2.1 定义可信计算

- 可信计算组织

如果一个实体的行为总是按照预期的方式和目标进行，那它就是可信的。

- IEEE可信计算技术委员会

可信是指计算机系统所提供的服务是可信赖的，而且这种可信赖是可论证的。

- 沈昌祥院士

可信计算系统是能够提供系统的可靠性、可用性、信息和行为安全性的计算机系统。

2.2 工业级可信计算硬件

厂商	可信计算硬件架构
Intel	SGX
AMD	SEV
ARM	TrustZone
AWS	Nitro

2.3 核心概念

- **安全输入输出(I/O)**

  安全输入输出是指电脑用户和他们认为与之交互的软件间受保护的路径。

- **内存屏蔽**

  内存屏蔽扩展为当前的内存保护技术，提供了对内存敏感区域(如放置密钥的区域)的全面隔离。

- **密封存储**

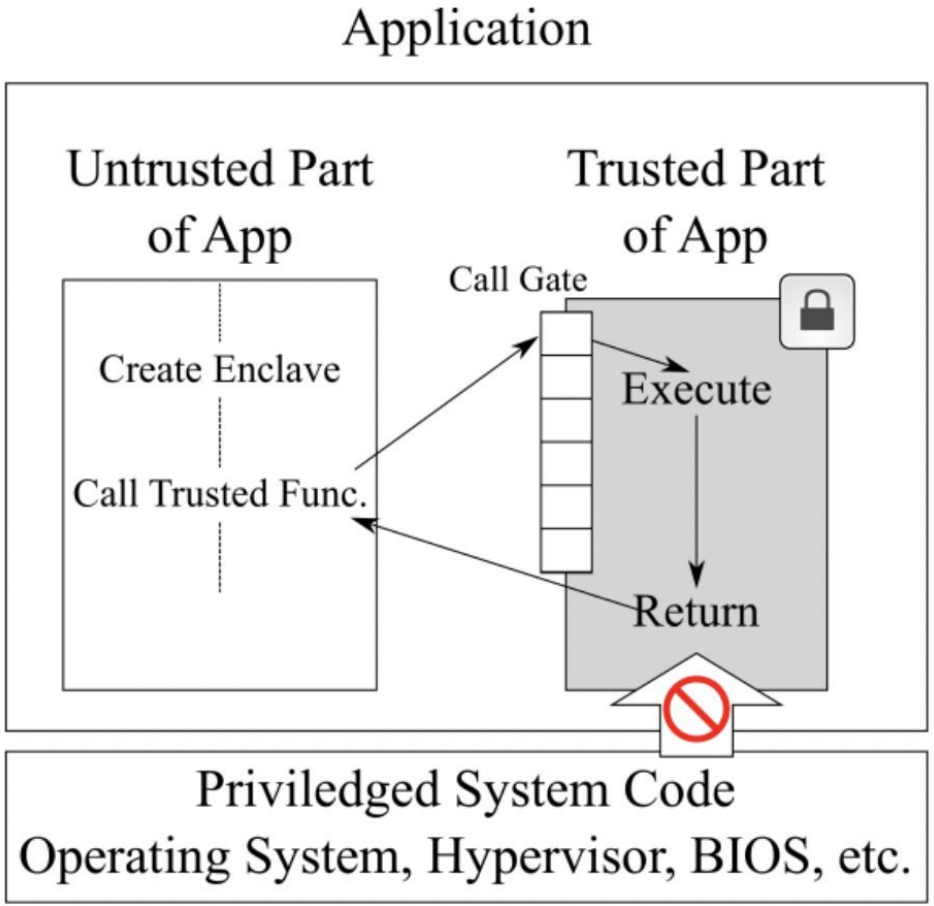
  密封存储从当前使用的软件和硬件配置派生出的密钥，并用这个密钥加密私有数据，从而实现对它的保护。

- **远程证明**

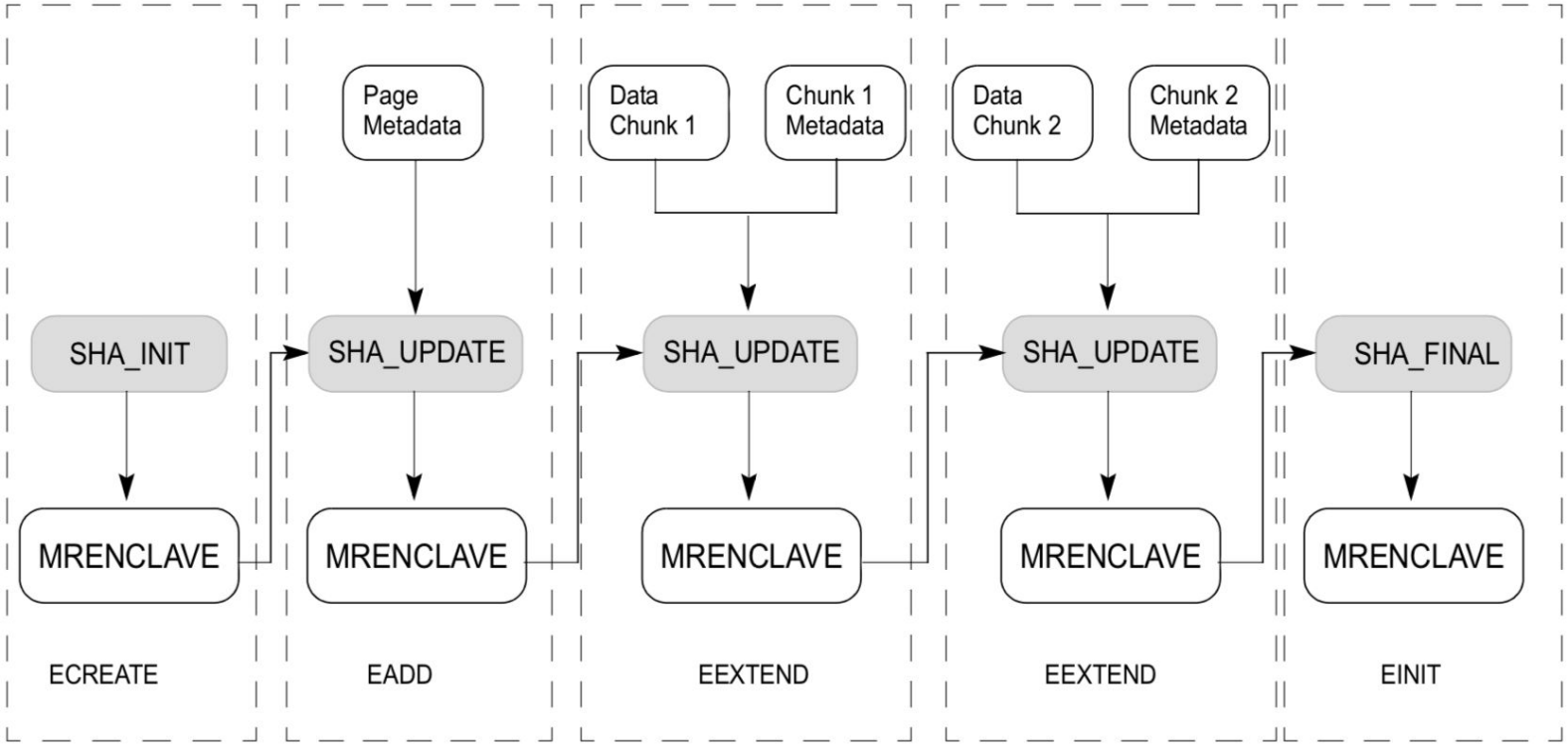
  远程证明使得用户或其他人可以检测到该用户的计算机的变化。

2.4 Intel SGX介绍

- SGX扩展指令集
- 可信部分
- 不可信部分

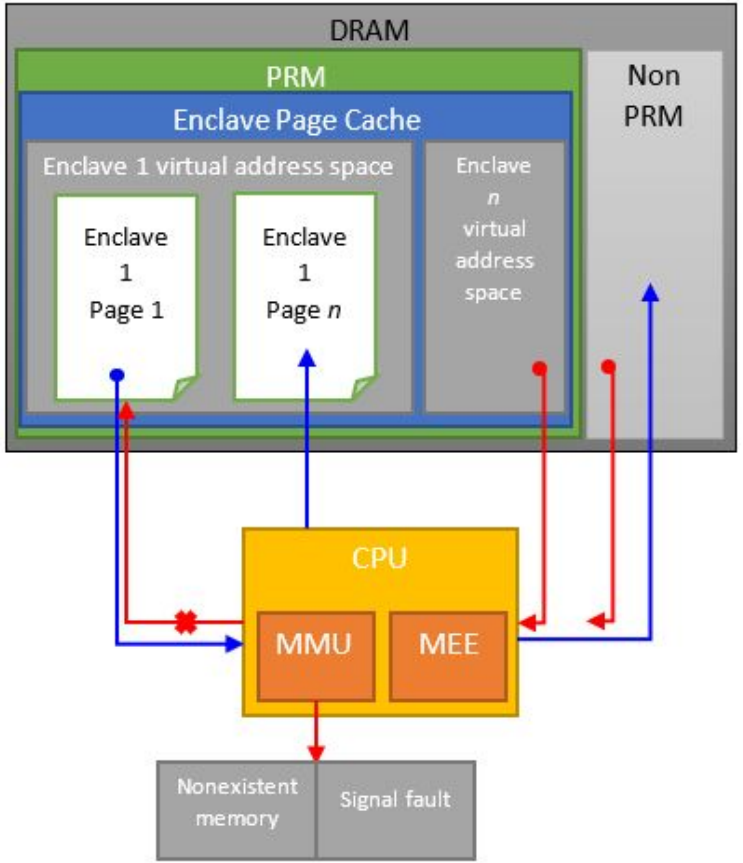


Enclave 度量


Page Metadata
Data Chunk 1
Chunk 1 Metadata
Data Chunk 2
Chunk 2 Metadata
SHA_INIT
SHA_UPDATE
SHA_UPDATE
SHA_UPDATE
SHA_FINAL
MRENCLAVE
MRENCLAVE
MRENCLAVE
MRENCLAVE
MRENCLAVE
ECREATE
EADD
EEXTEND
EEXTEND
EINIT

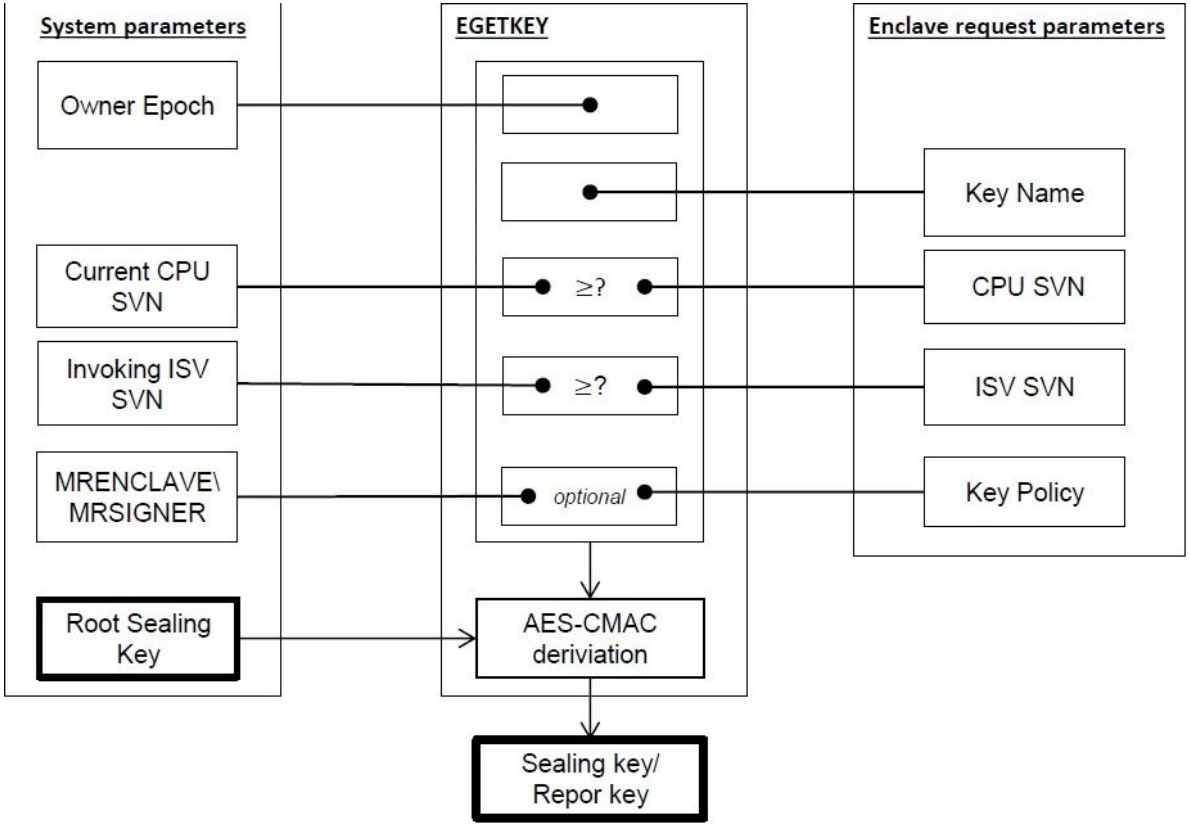
内存屏蔽

- 物理内存隔离
- MEE
- 进入CPU后，解密数据
- 离开CPU前，加密数据



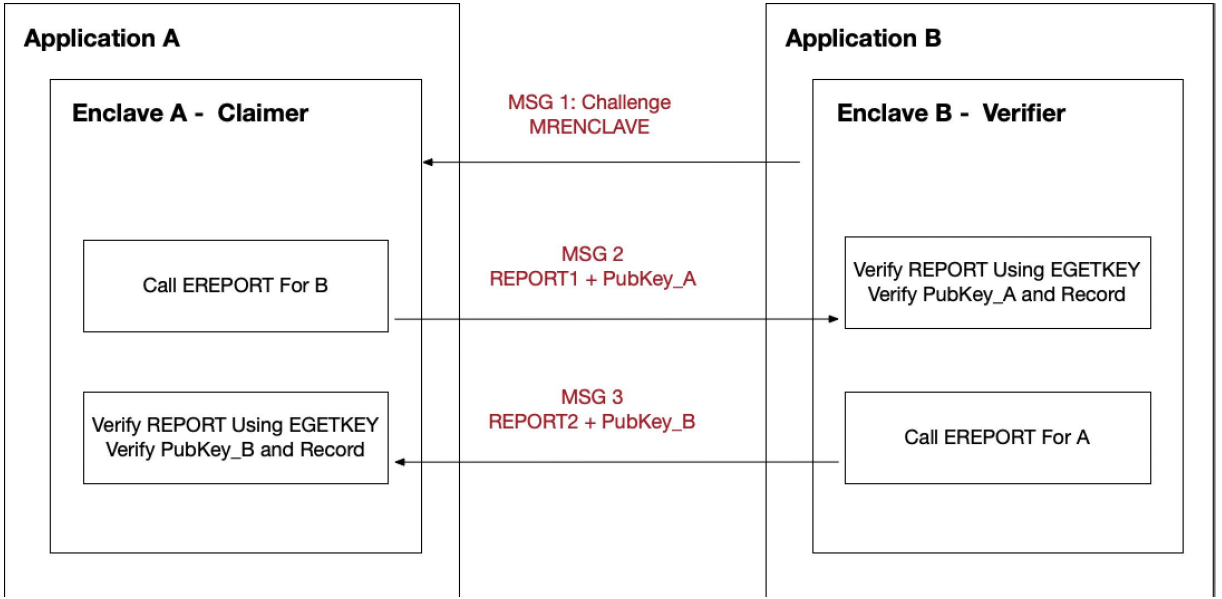
密封存储

- Root Seal Key
- 度量寄存器
  - MRENCLAVE
  - MRSIGNER



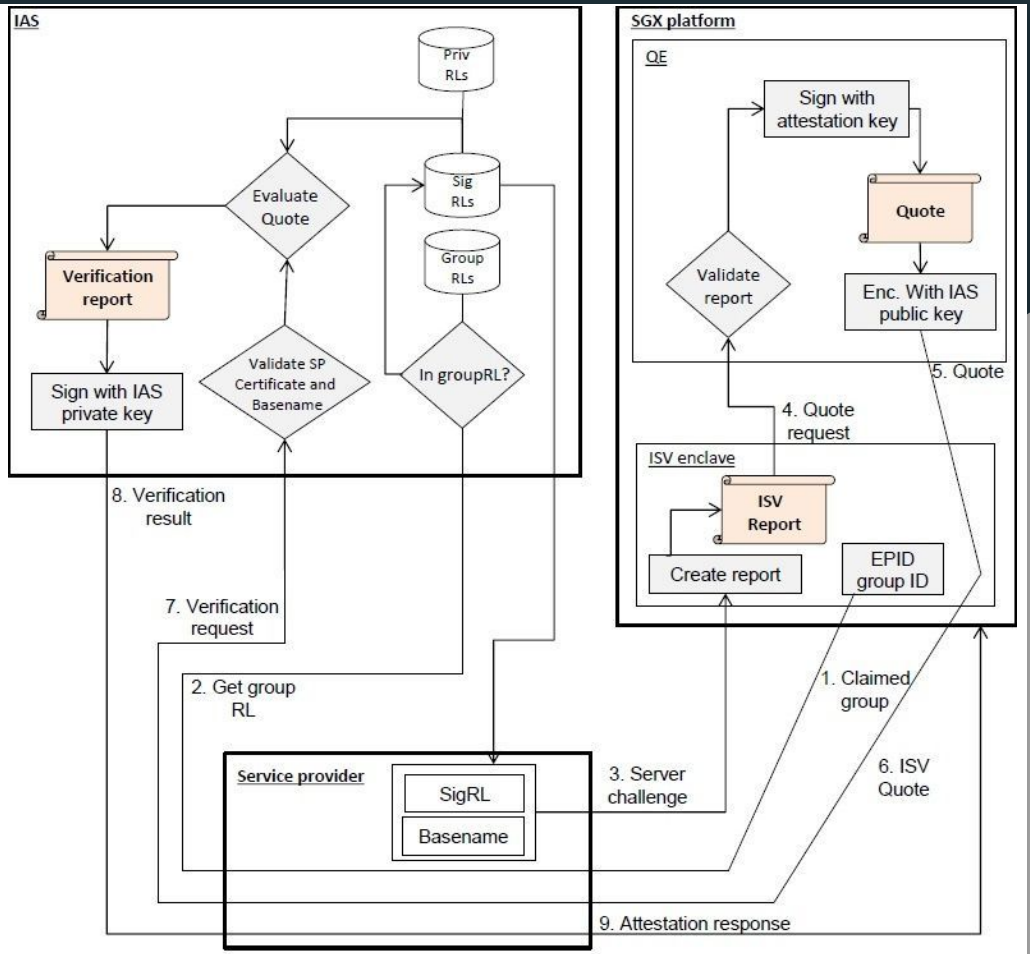
本地认证

- 同一平台
- EGETKEY
- EREPORT
- 挑战
- 认证



远程认证

- EPID
- Root Provision Key
- 度量寄存器
  - MRENCLAVE
  - MRSIGNER
- IAS
- QE



2.5 Intel SGX 开发框架

软件服务提供商	可信计算软件框架
Intel	SGX SDK
Microsoft Azure	OpenEnclave
Google	Asylo
Baidu	RustSGX
阿里蚂蚁	Occlum

2.5 Intel SGX 开发框架

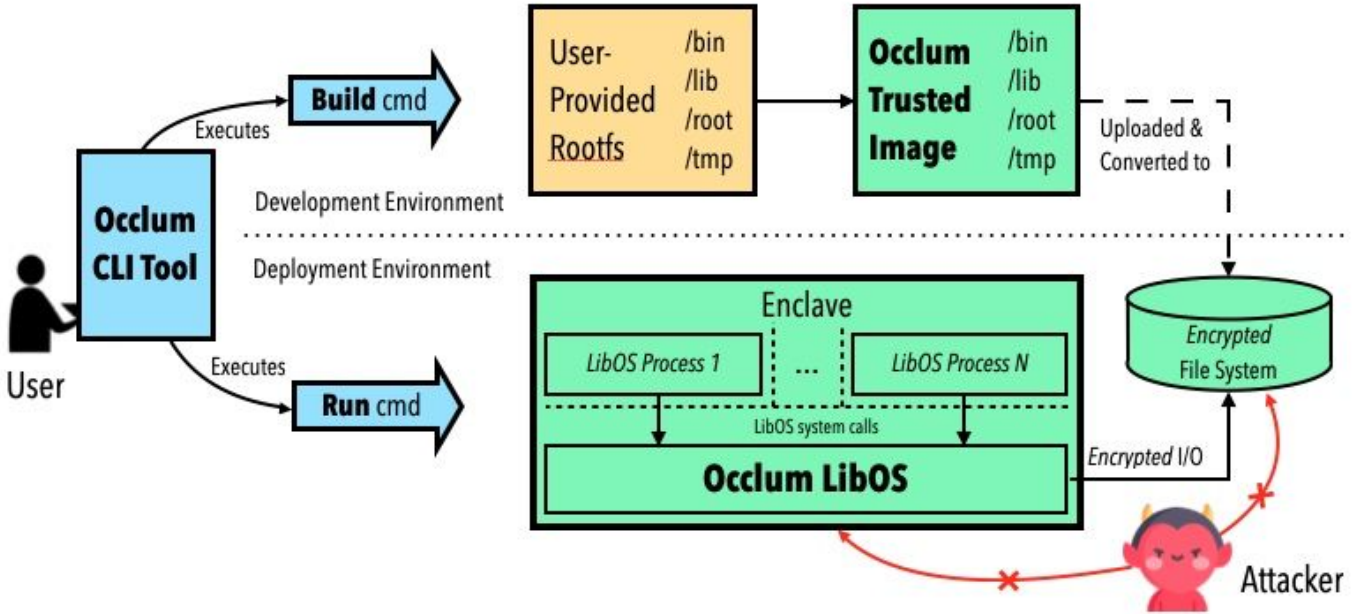


图. Occlum的系统架构

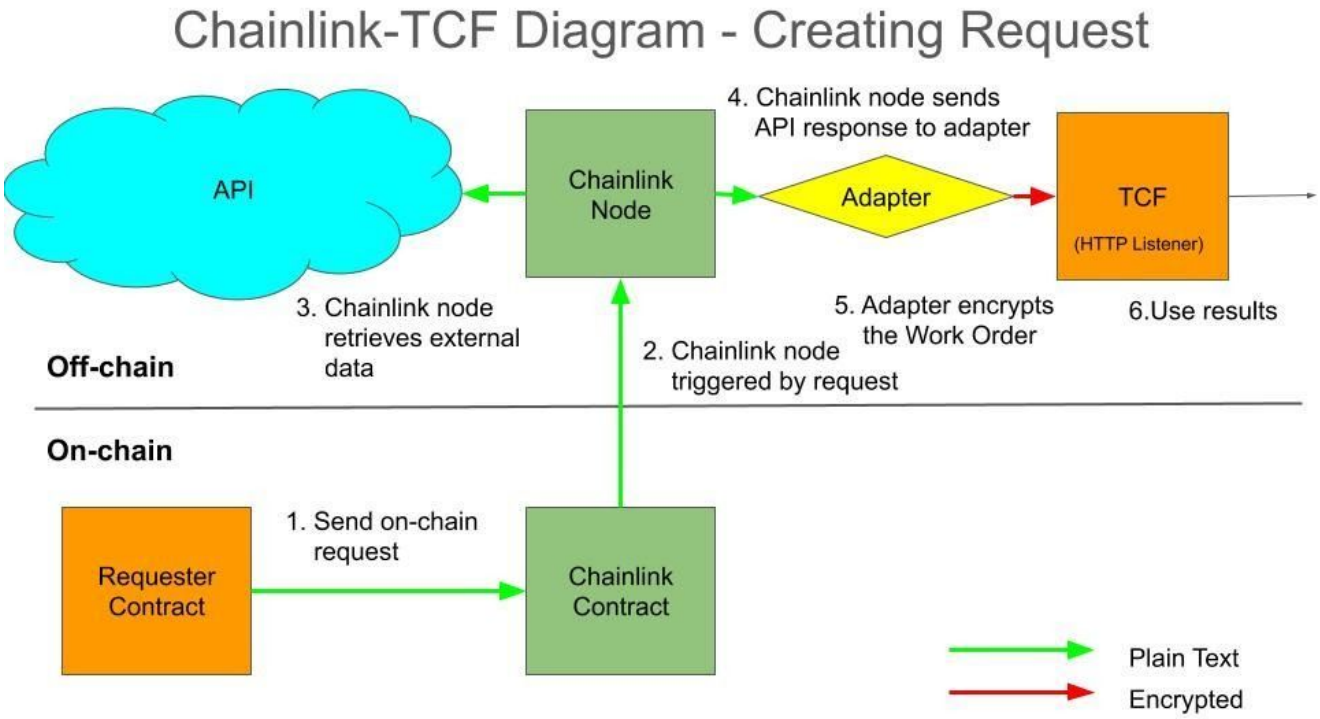
2.6 TEE中的数据可迁移性

同平台	同Enclave	数据密封模式	能否直接迁移
是	是	OE_SEAL_POLICY_UNIQUE	
是	是	OE_SEAL_POLICY_PRODUCT	
是	否	OE_SEAL_POLICY_UNIQUE	
是	否	OE_SEAL_POLICY_PRODUCT	
否	——	——	

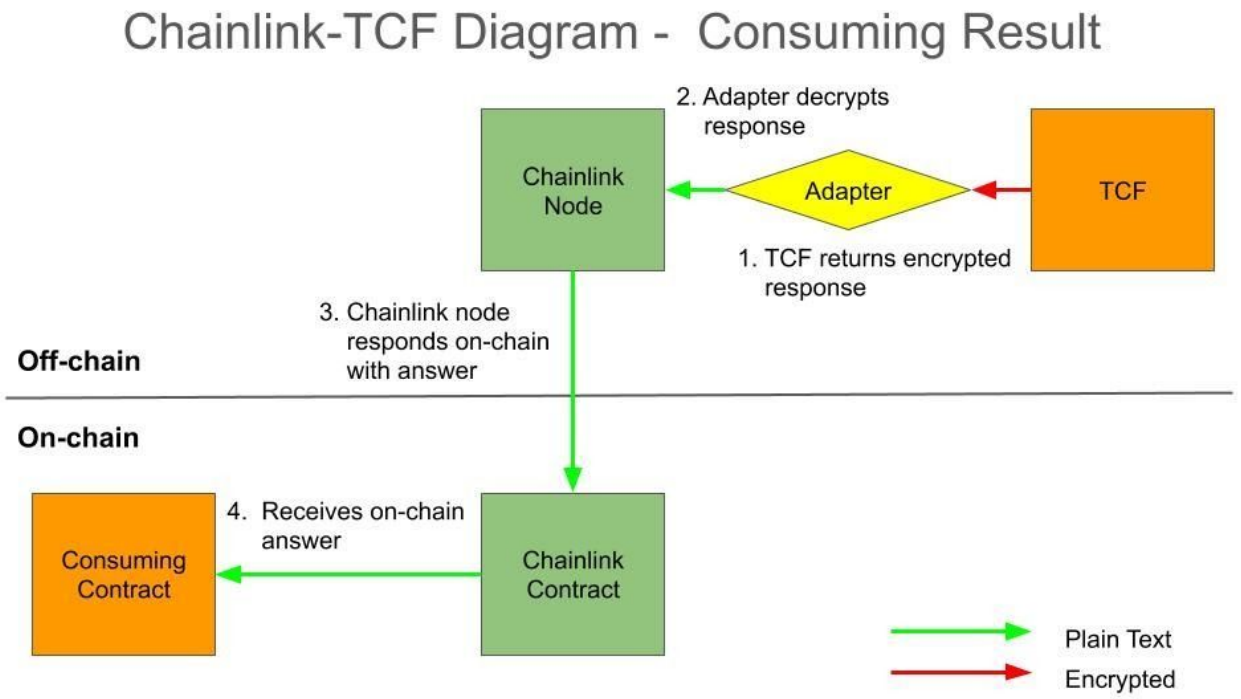
2.7 典型应用示例

- 预言机
  - Chainlink
- 区块链
  - Phala Network
  - OASIS
- 可信签名机

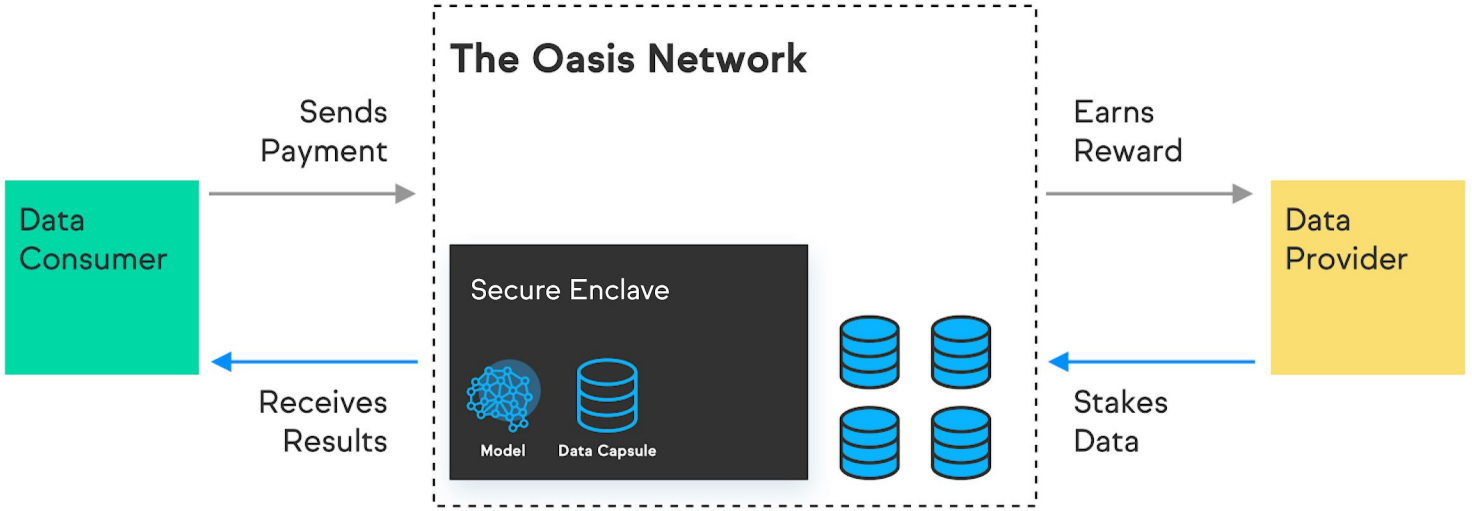
Chainlink


Chainlink-TCF Diagram - Creating Request
API
Chainlink Node
4. Chainlink node sends API response to adapter
Adapter
TCF
(HTTP Listener)
3. Chainlink node retrieves external data
5. Adapter encrypts the Work Order
6.Use results
2. Chainlink node triggered by request
Off-chain
On-chain
Requester Contract
1. Send on-chain request
Chainlink Contract
Plain Text
Encrypted

Chainlink

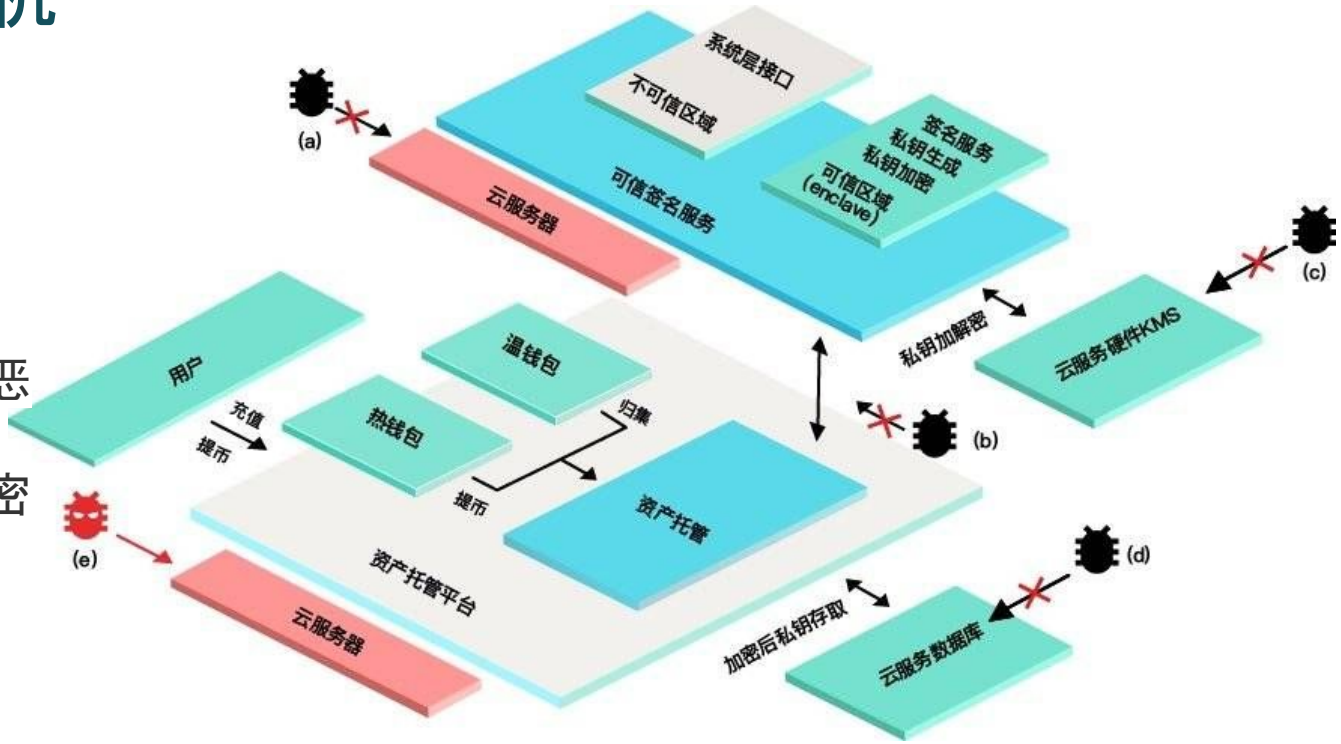

Chainlink-TCF Diagram - Consuming Result
2. Adapter decrypts response
Chainlink Node
Adapter
TCF
1. TCF returns encrypted response
3. Chainlink node responds on-chain with answer
Off-chain
On-chain
Consuming Contract
4. Receives on-chain answer
Chainlink Contract
Plain Text
Encrypted

Phala/Oasis


The Oasis Network
Sends Payment
Earns Reward
Data Consumer
Data Provider
Secure Enclave
Model
Data Capsule
Receives Results
Stakes Data

可信签名机

- 防代码篡改
- 抗内存嗅探
- 防止平台作恶
  - 二次加密
- 配置加密
- 横向扩展



3 安全多方计算(MPC)

- 3.1 百万富翁问题
- 3.2 定义安全多方计算
- 3.3 MPC主要特点
- 3.4 MPC核心概念
- 3.5 MPC多方签名
- 3.6 MPC分布式私钥生成
- 3.7 MPC多签流程
- 3.8 MPC多签的优点
- 3.9 MPC典型示例

3.1 百万富翁问题

两个百万富翁都想比较到底谁更富有，但是又都不想让别人知道自己有多少钱。在没有可信的第三方的情况下如何进行？

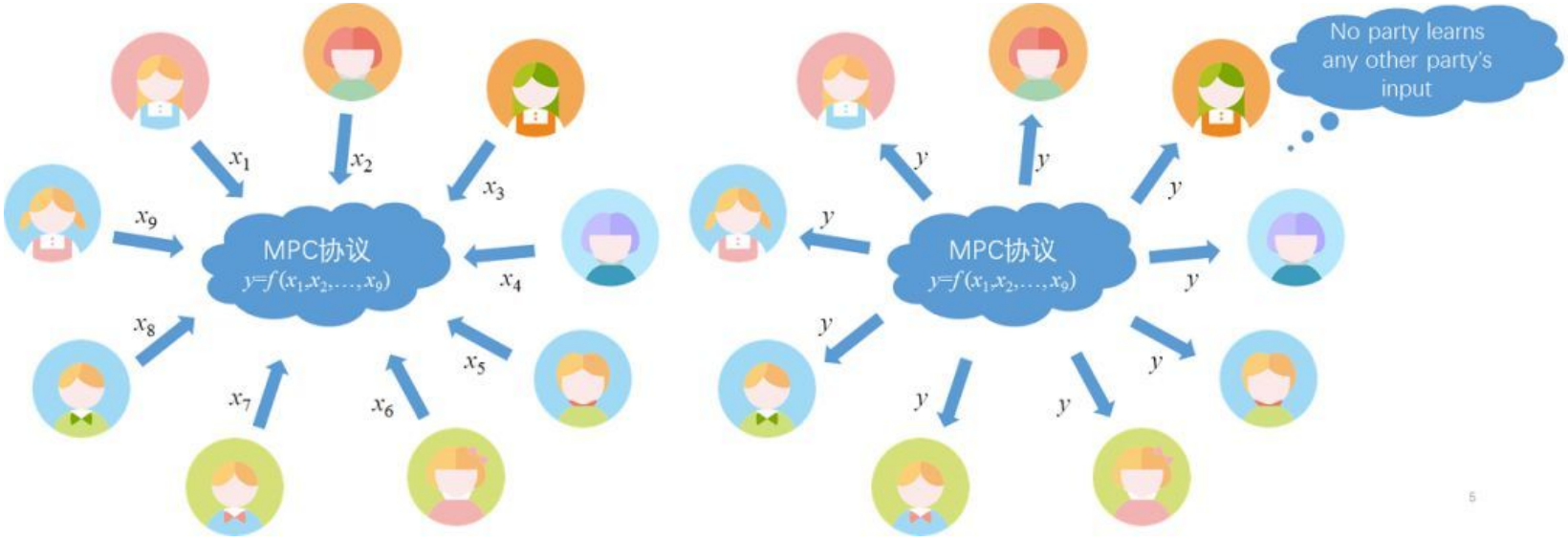
X > ? Y

财富 X

财富 Y

3.2 定义安全多方计算

针对无可信第三方的情况下，如何安全地计算一个约定函数的问题。



3.3 主要特点

- 输入隐私性

安全多方计算研究的是各参与方在协作计算时如何对各方隐私数据进行保护，重点关注各参与方之间的隐私安全性问题，即在安全多方计算过程中必须保证各方私密输入独立，计算时不泄露任何本地数据。

- 计算正确性

多方计算参与各方就某一约定计算任务，通过约定 MPC 协议进行协同计算，计算结束后，各方得到正确的数据反馈。

- ·去中心化

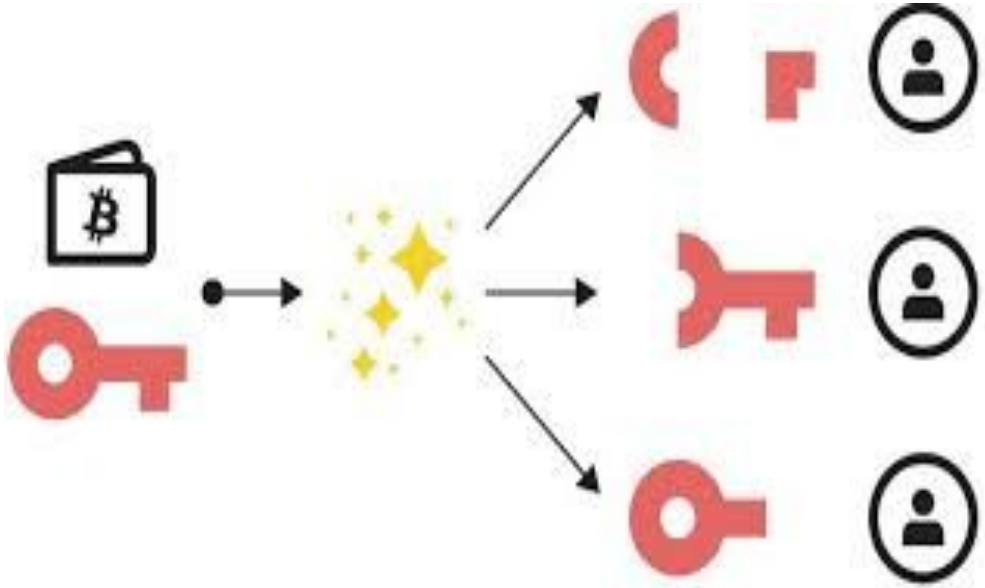
传统的分布式计算由中心节点协调各用户的计算进程，收集各用户的输入信息，而安全多方计算中，各参与方地位平等，不存在任何有特权的参与方或第三方，提供一种去中心化的计算模式。

3.4 MPC核心概念

- 密钥共享
- 混淆电路
- 不经意传输
- 零知识证明
- 同态计算

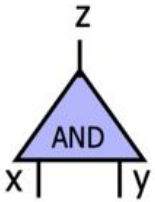
密钥共享

- Shamir‘ Secret Sharing
- Felldman’ Scheme
- Benaloh’s scheme



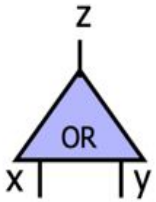
混淆电路

- 与非门
- 随机数标签



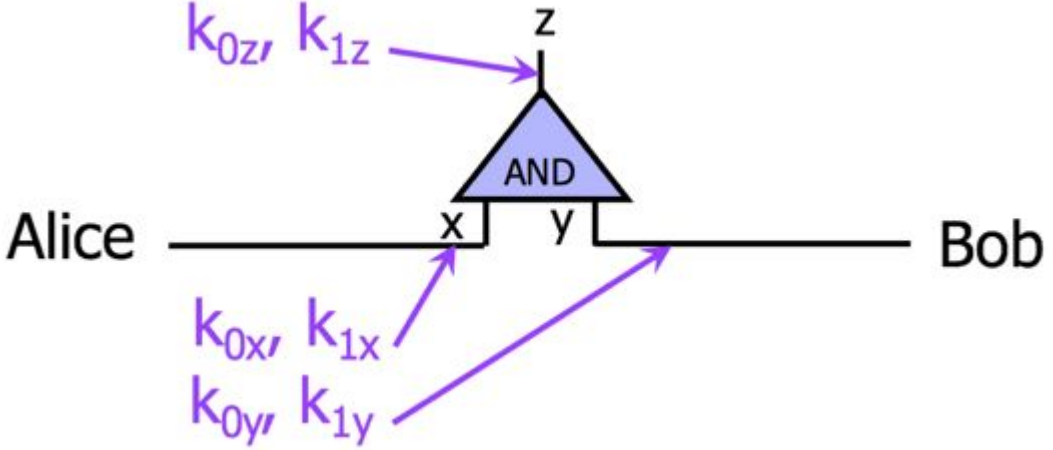
Truth table:

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1



Truth table:

x	y	z
0	0	0
0	1	1
1	0	1
1	1	1



混淆电路

- 加密真值表
- 打乱真值表

Encrypted truth table:

$$E_{k0x}(E_{k0y}(k_{0z}))$$
$$E_{k0x}(E_{k1y}(k_{0z}))$$
$$E_{k1x}(E_{k0y}(k_{0z}))$$
$$E_{k1x}(E_{k1y}(k_{1z}))$$

Garbled truth table:

$$E_{k1x}(E_{k0y}(k_{0z}))$$
$$E_{k0x}(E_{k1y}(k_{0z}))$$
$$E_{k1x}(E_{k1y}(k_{1z}))$$
$$E_{k0x}(E_{k0y}(k_{0z}))$$

不经意传输

1-out-2 OT

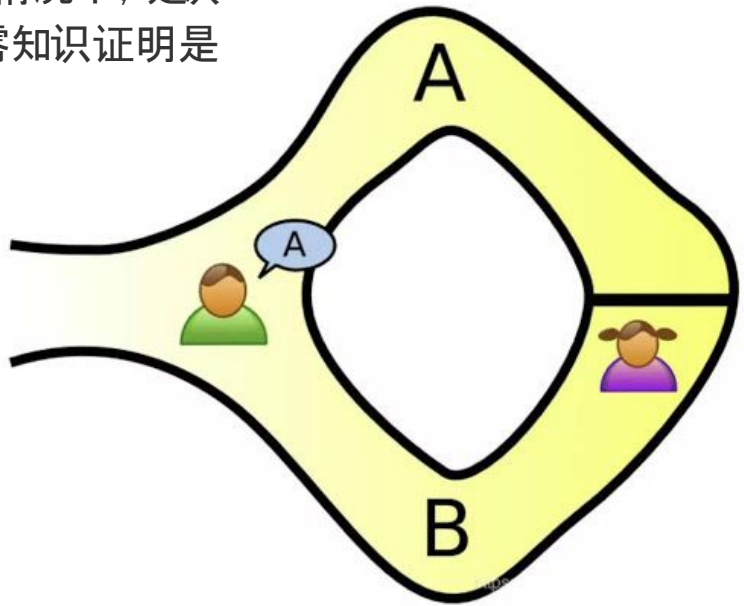
- 实现了发送方将潜在的许多信息中的一个传递给接收方，但对接收方所接收信息保持未知状态。
  - 1-2不经意传输
  - 1-N不经意传输
  - k-N不经意传输

1-out-n OT

零知识证明

证明者能够在不向认证者提供任何有用的信息的情况下，是认证者相信某个论断是正确的。和数学证明不同，零知识证明是概率证明。

- Schnorr Proof
- Range Proof
- Heg Proof



同态计算

- Pailliar 同态加密算法 $D(\mathrm{E}(x)) + \mathrm{D}(\mathrm{E}(y)) = \mathrm{D}(\mathrm{E}(x) \oplus \mathrm{E}(y))$
- ElGamal同态加密算法 $D(\mathrm{E}(x)) * \mathrm{D}(\mathrm{E}(y)) = \mathrm{D}(\mathrm{E}(x) \circledast \mathrm{E}(y))$

3.5 MPC多方签名

- 单签偏好型签名算法
  - Secp256k1（ECDSA）
  - Ed25519（EDDSA）
- 多签偏好型签名算法
  - Schnorr
  - BLS

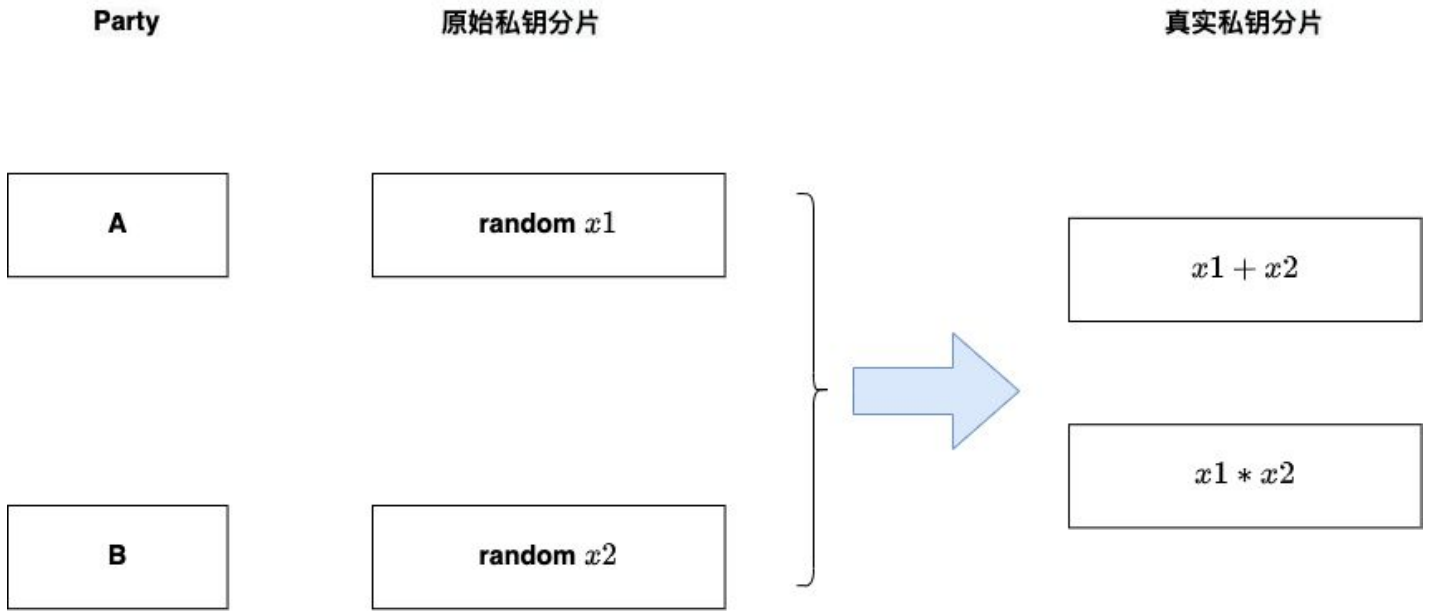
3.5 MPC多方签名

- 全员签名算法(n, n)
  - 2-2
  - 3-3
- 门限签名算法(t, n)
  - t - n => t - 2n
  - t - n => (2t + 1) - n

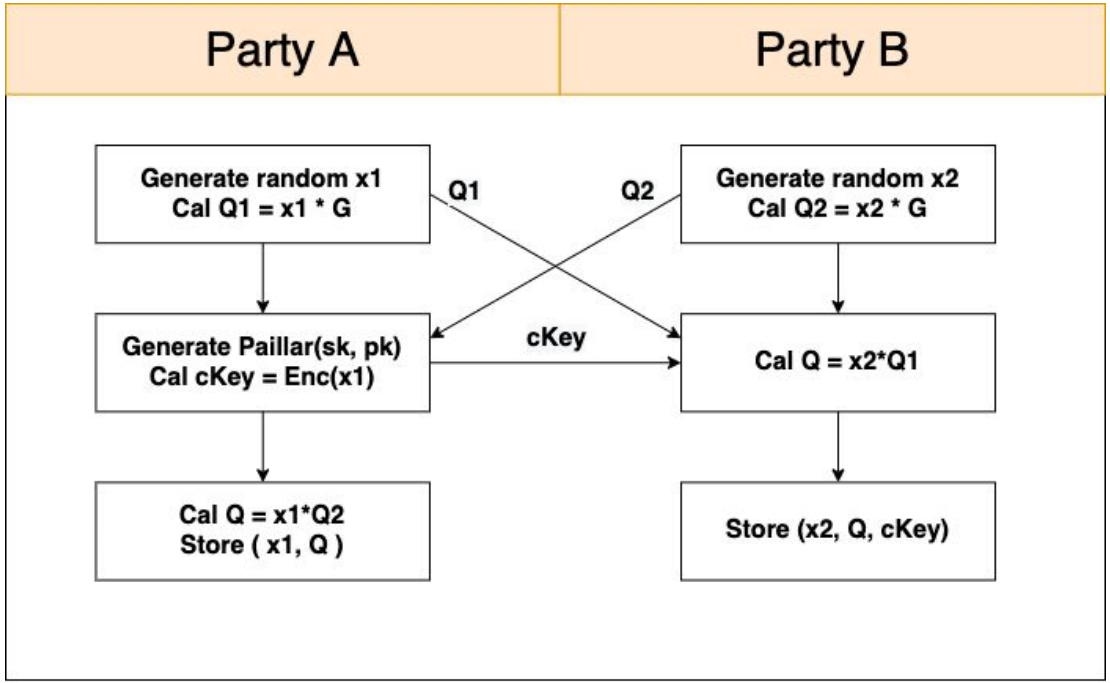
3.6 MPC分布式钱包生成

- 2 - 2 分布式钱包生成(全员多签)
- 3 - 3 分布式钱包生成(全员多签)
- 2 - 3 分布式钱包生成(门限多签)

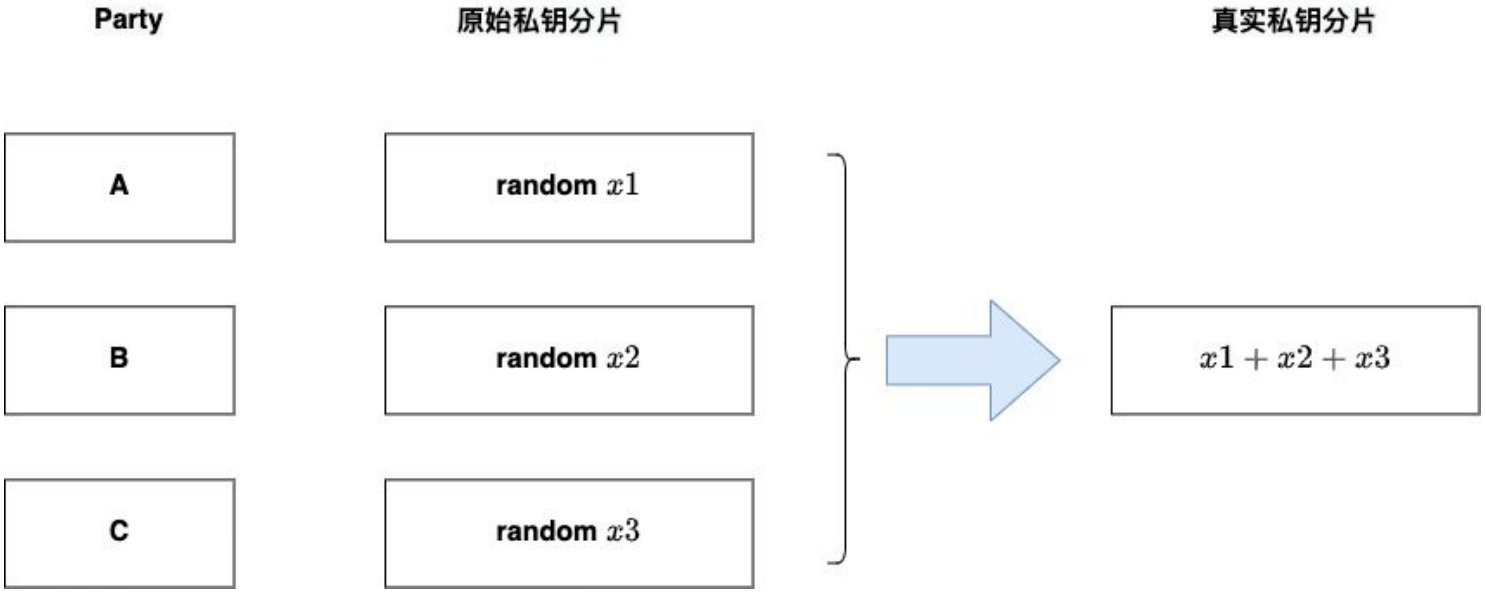
2-2 分布式钱包生成(全员多签)



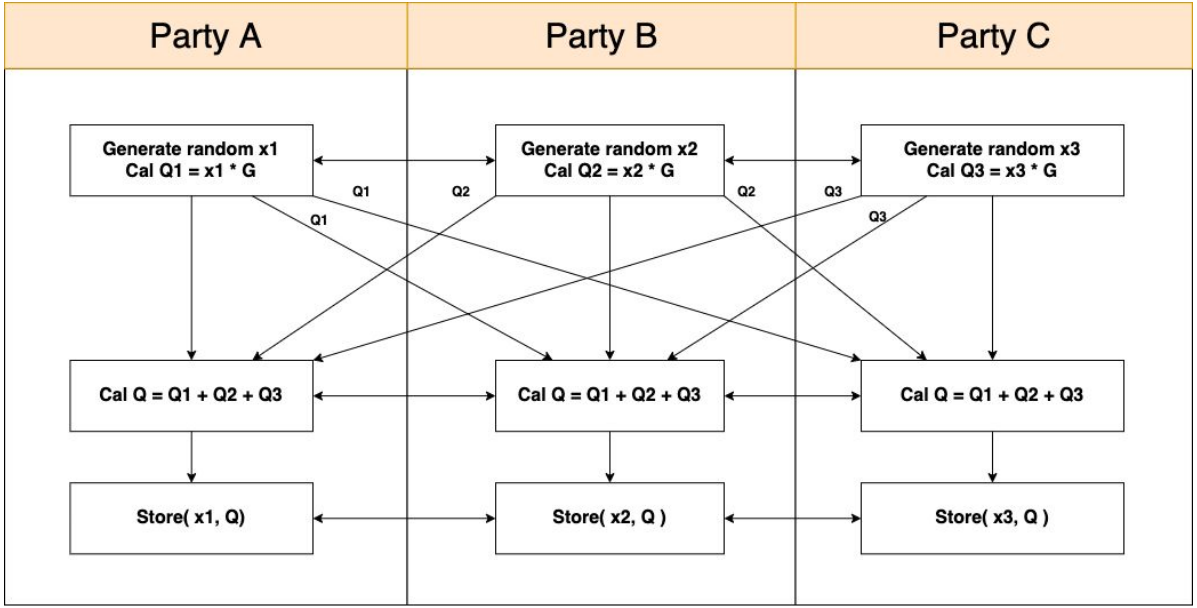
2-2 分布式钱包生成(全员多签)



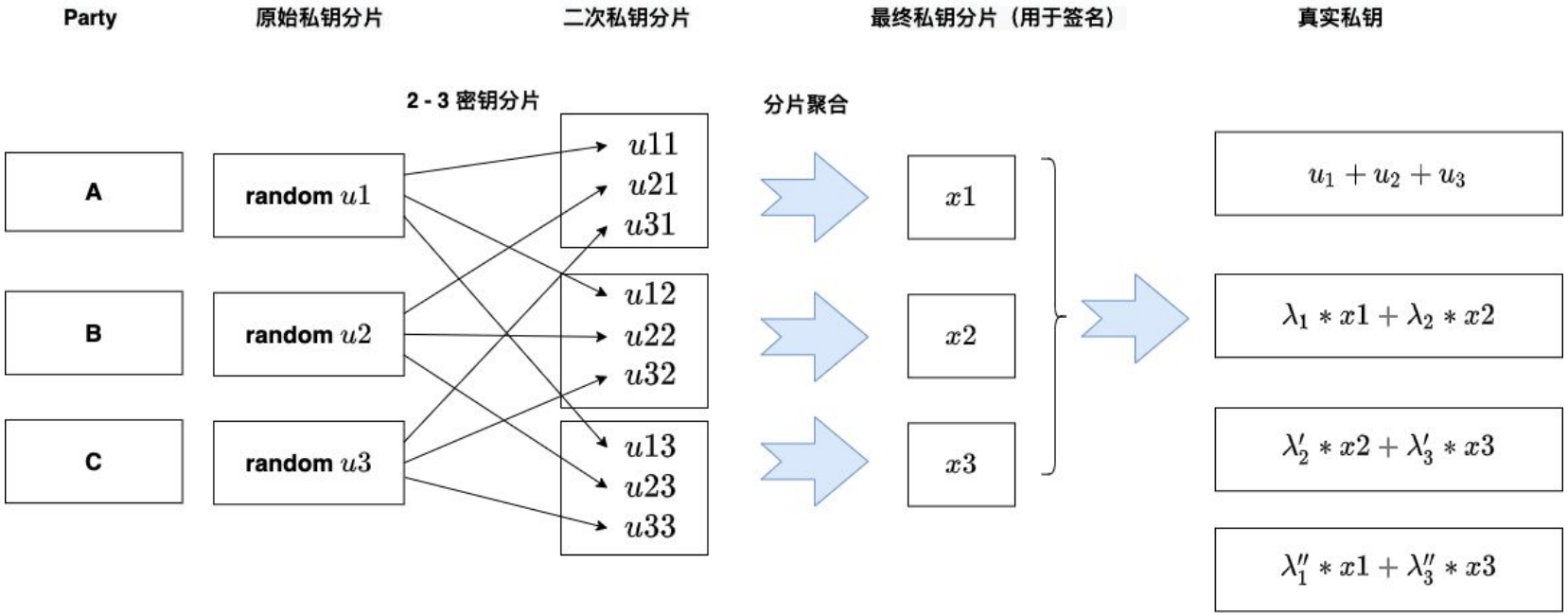
3-3 分布式钱包生成(全员多签)



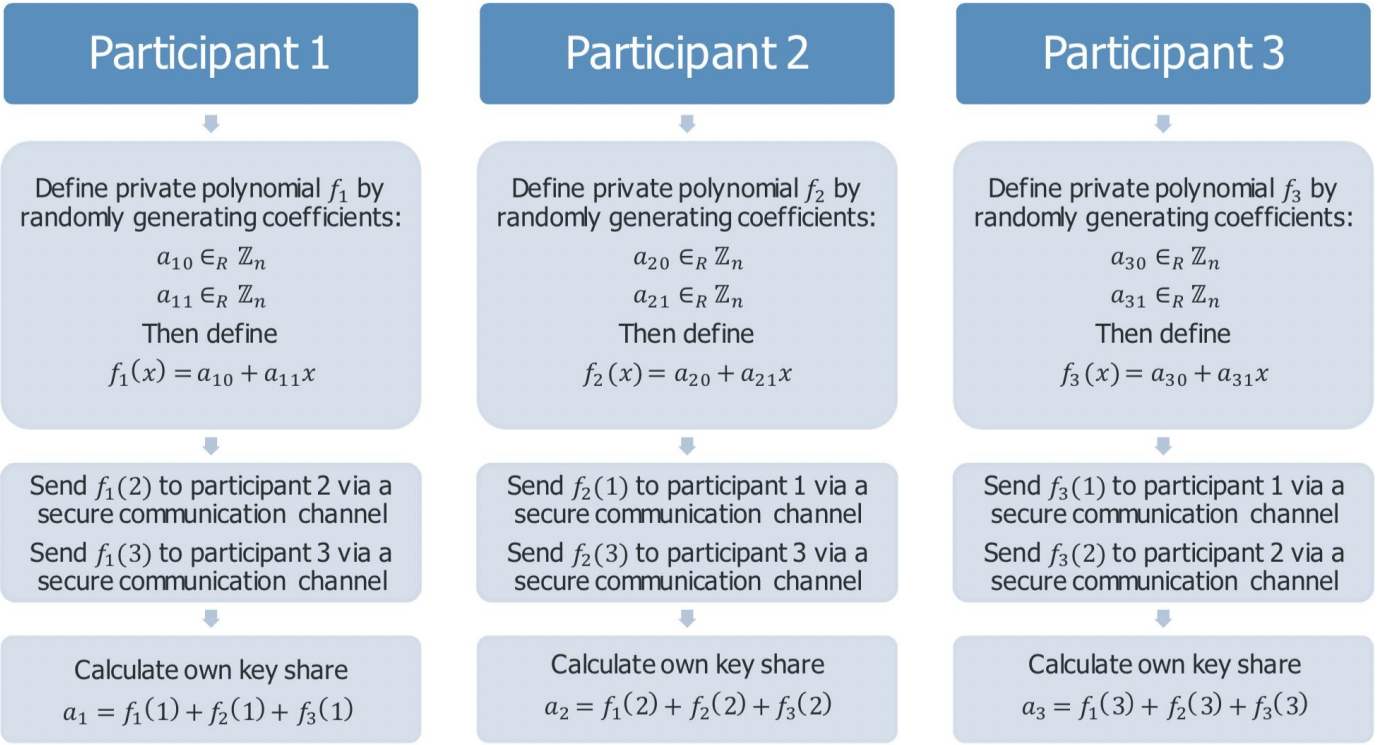
3-3 分布式钱包生成(全员多签)



2 - 3 分布式钱包生成(门限多签)



3-3 分布式钱包生成(全员多签)

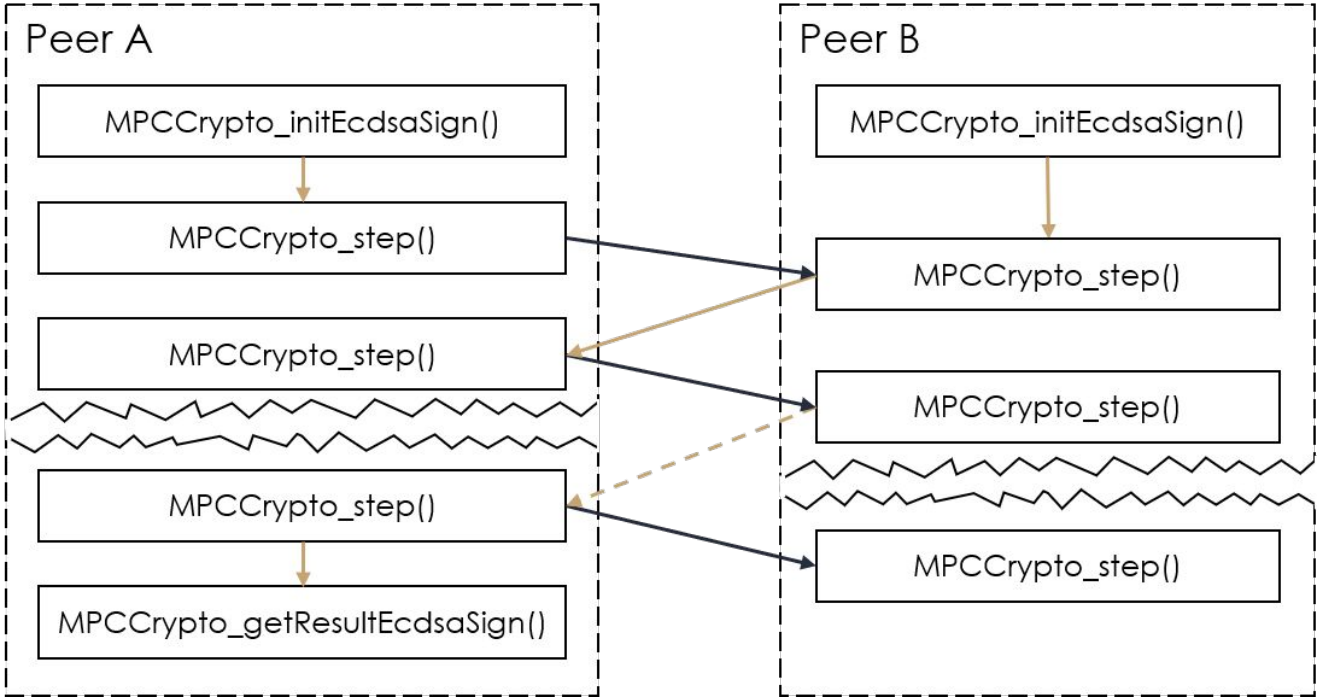


3.6 MPC多签流程

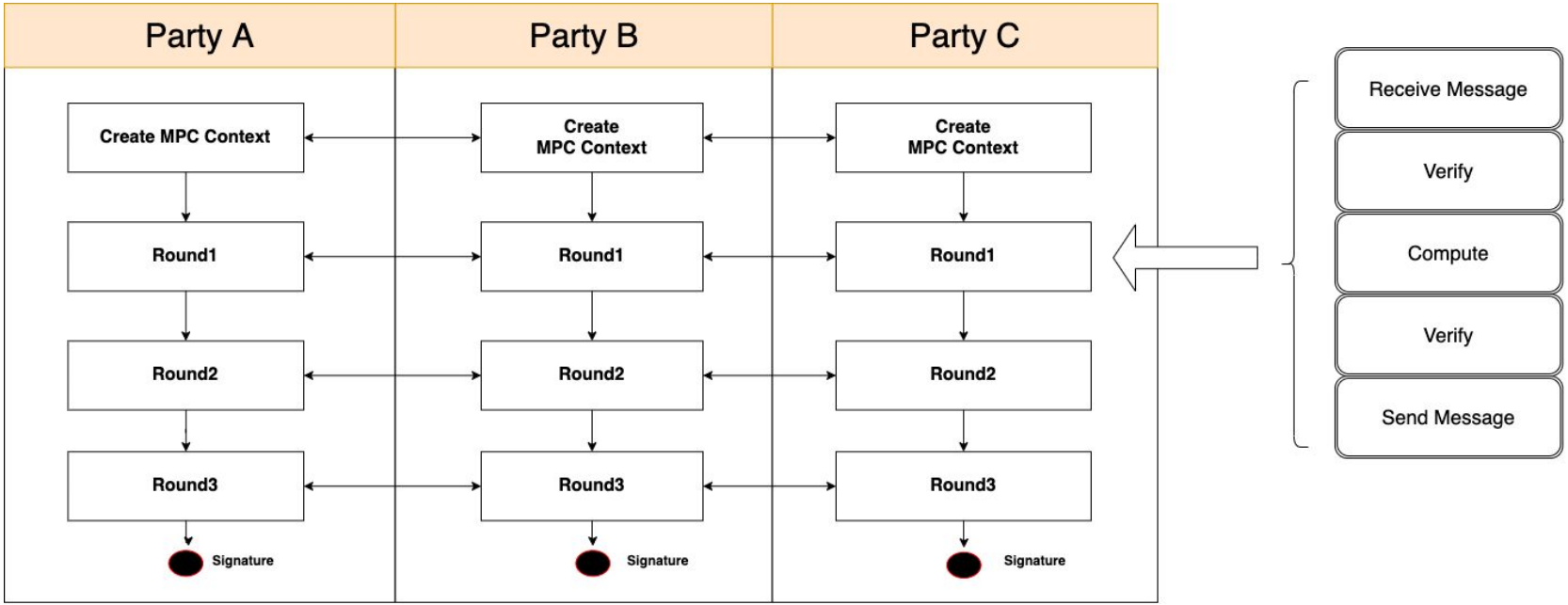
- 2-Party MPC

- N-Party MPC

典型MPC多签流程（2-Party）



典型MPC多签流程（N-Party）　　N >= 3

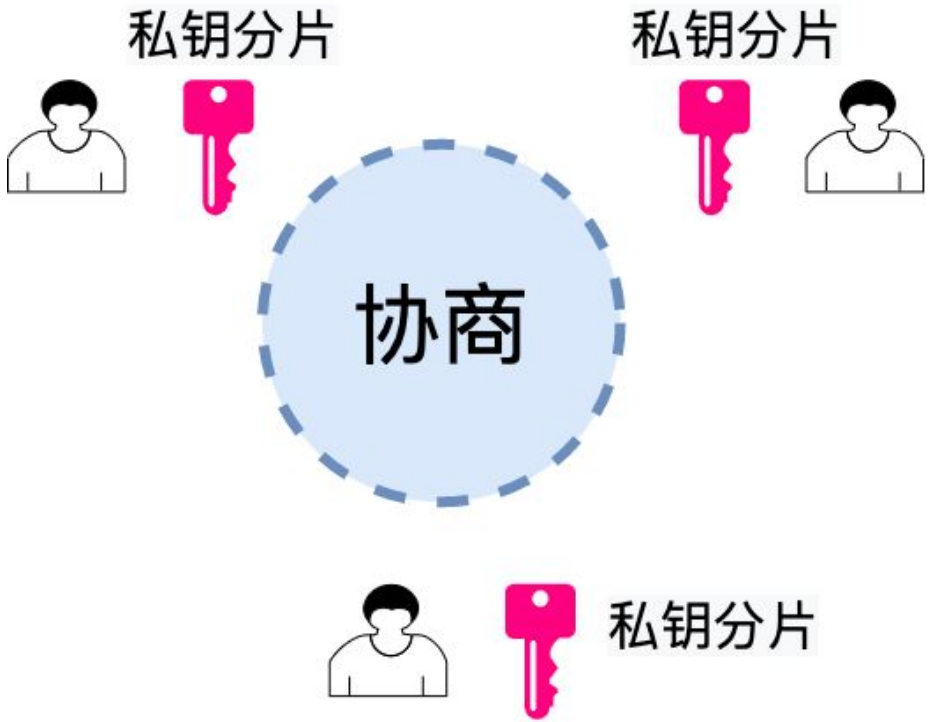


3.7 MPC多签的优点

- 去单点
- 通用性（与链无关）
- 灵活性
- 链上隐私+链下追溯
- 兼容HD派生

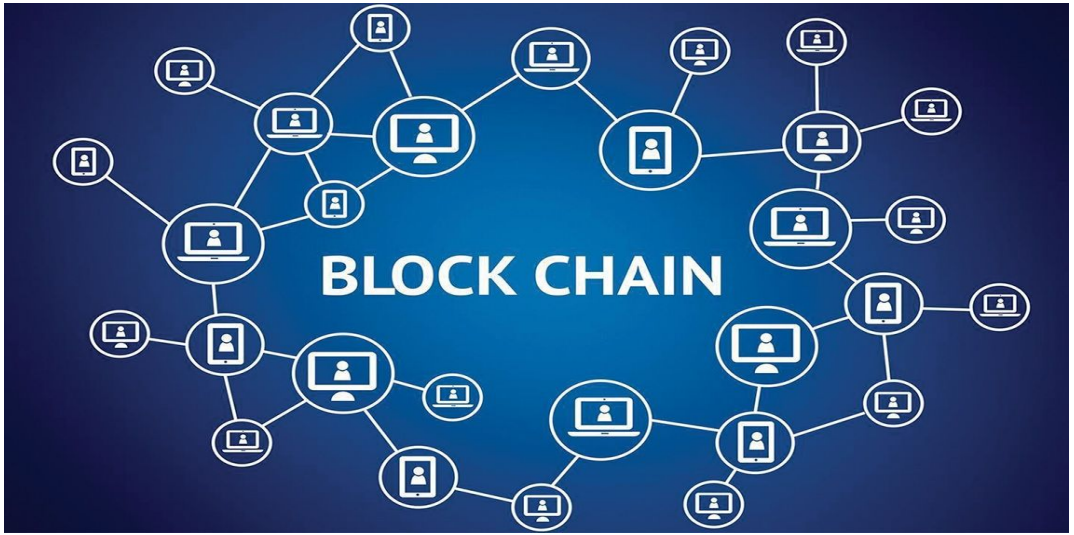
去单点

- 私钥分片
- 分布式生成
- 分布式签名
- 私钥从未出现



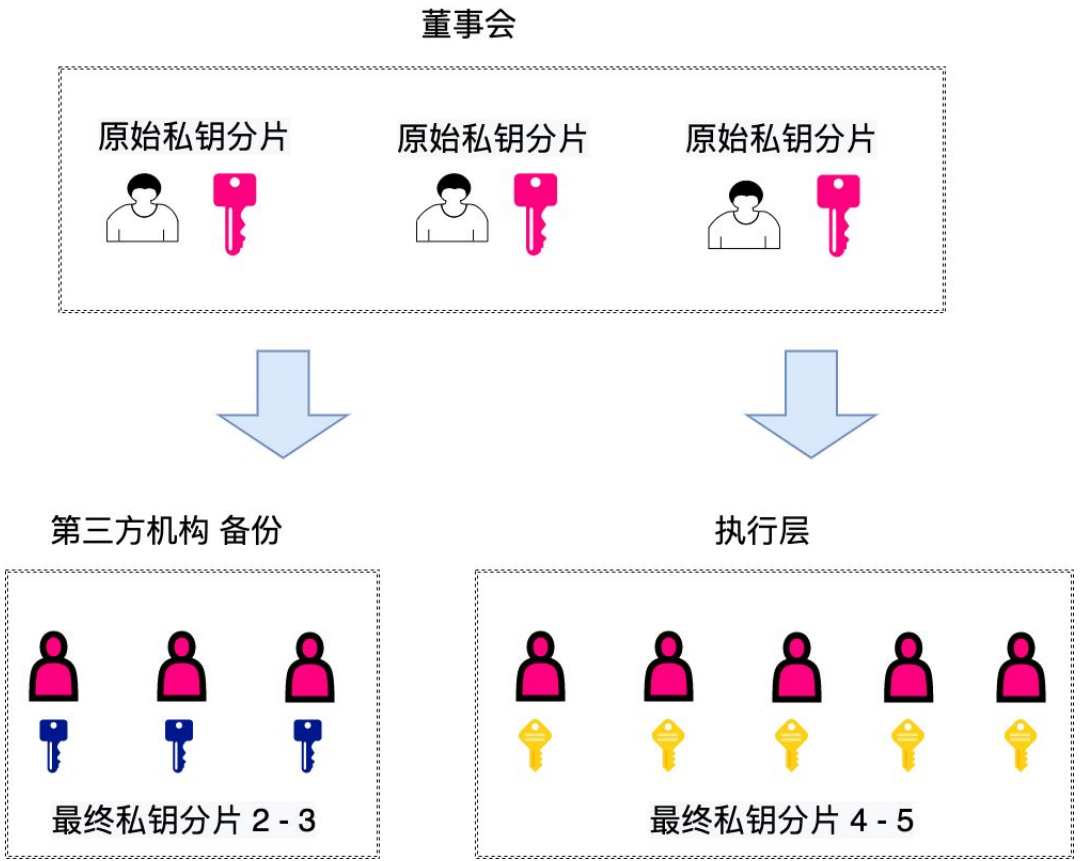
通用性

- 现有方式
  - 链原生多签
  - 合约多签
- 算法级兼容
- 不区分链

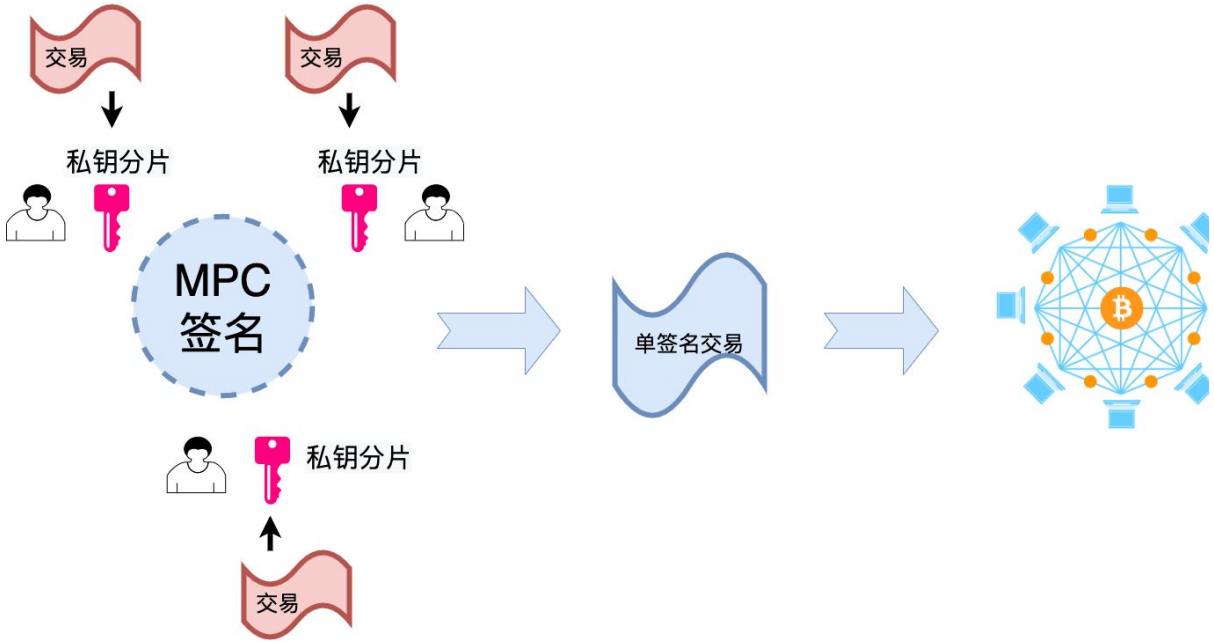


灵活性

- 增加新用户
  - 2 - 3 => 2 - 4
- 门限变更
  - 4 - 5 => 3 - 5
- 复杂管理



链上隐私 + 链下追溯



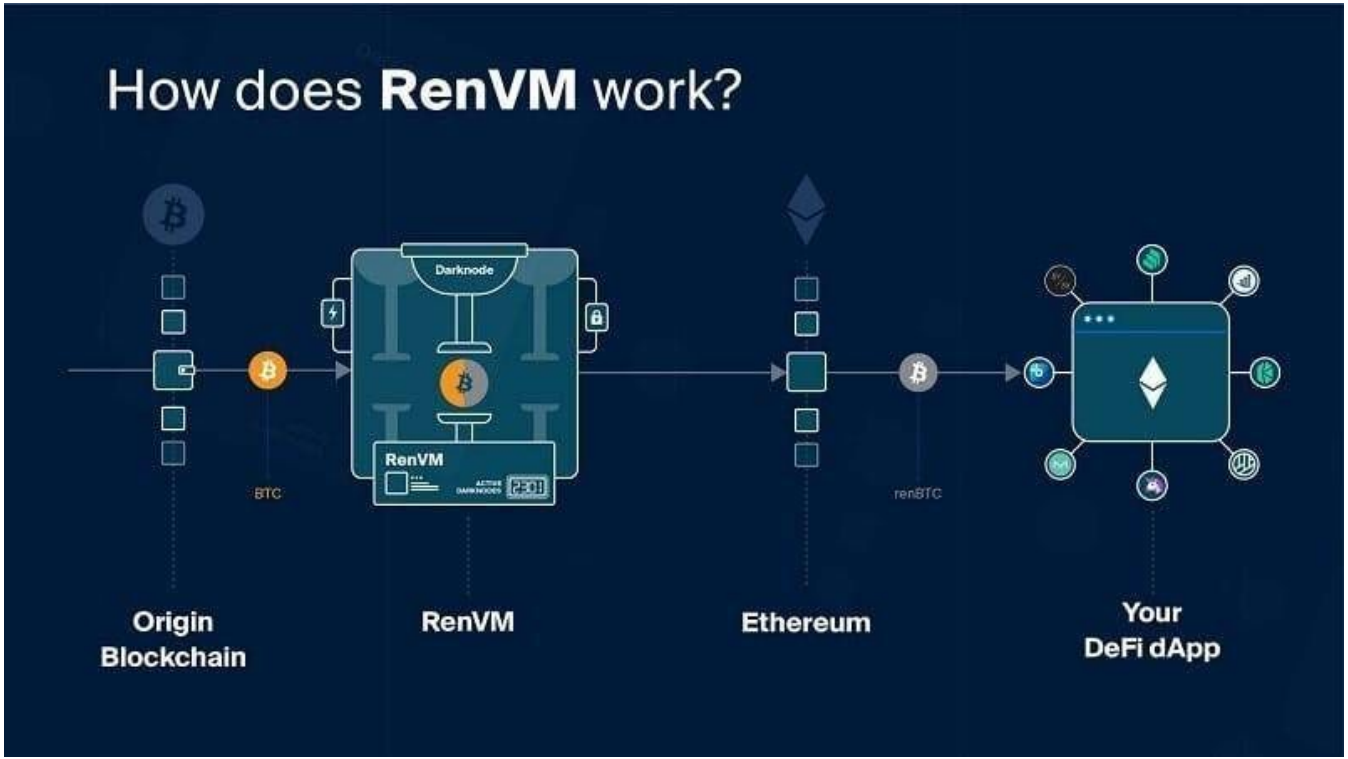
兼容HD 派生 —— BIP32 + BIP44

签名算法类型	签名门限	总分片数	Harden派生	非Harden派生	备注
全员多签	2	2			基于混淆电路
全员多签	n	n			n >= 3
门限多签	t	n			

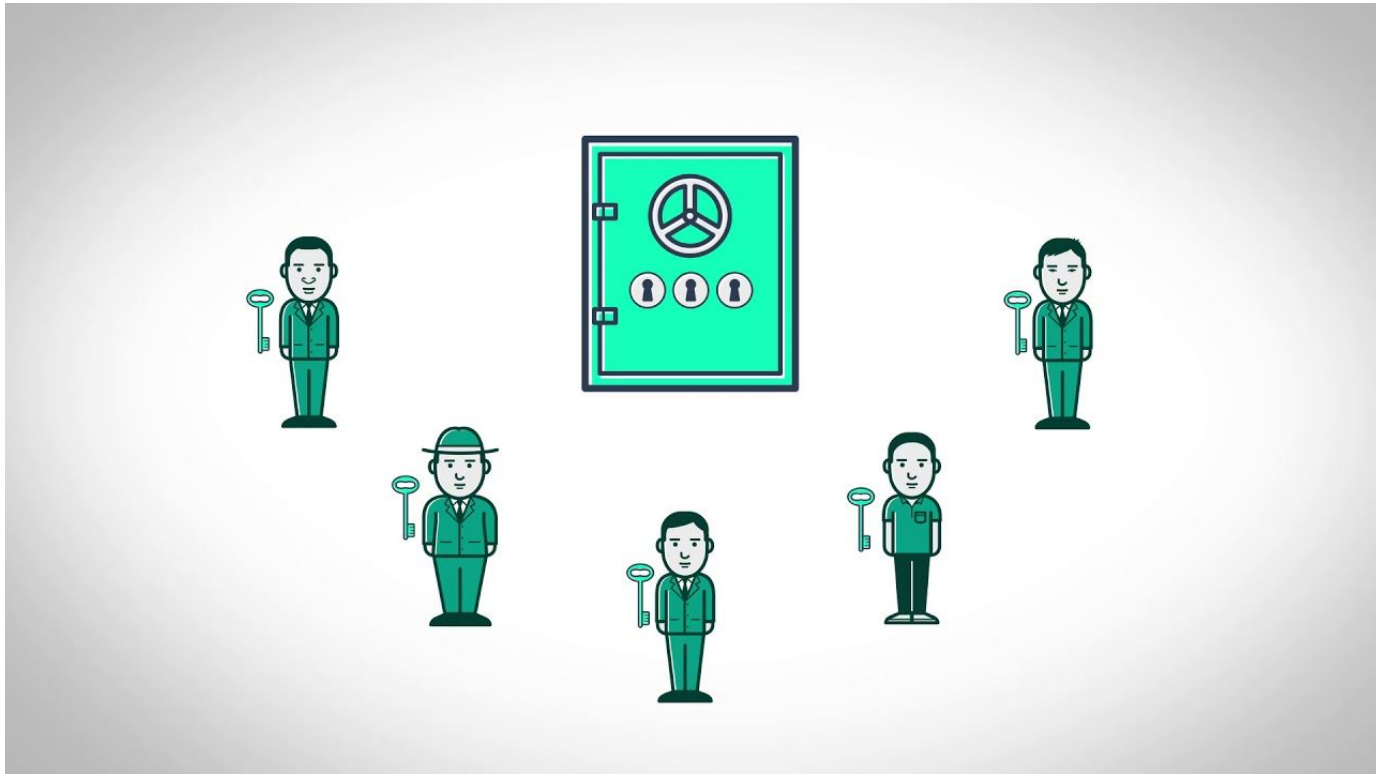
3.8 MPC多签的应用场景

- 资产锚定与跨链
- 通用数字资产存管
- Defi协议管理

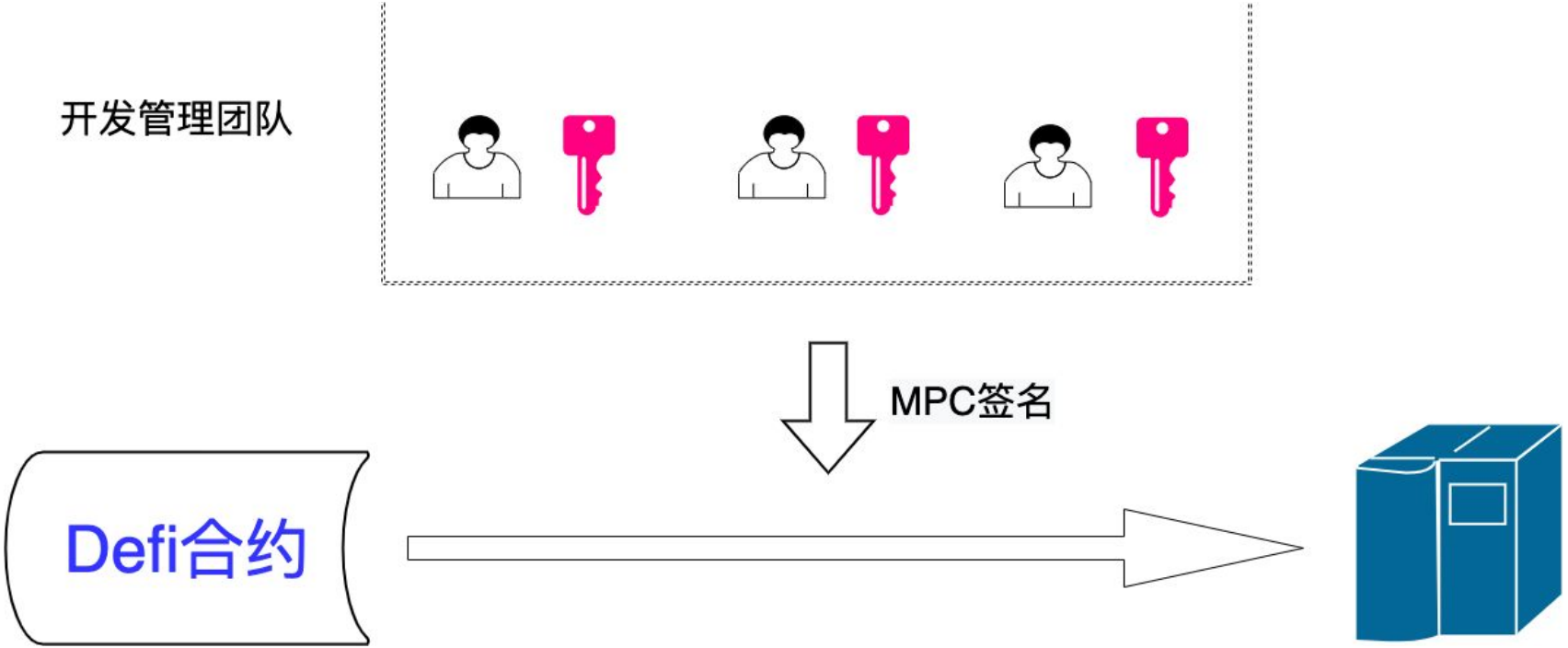
资产锚定与跨链——RenBTC



通用数字资产存管

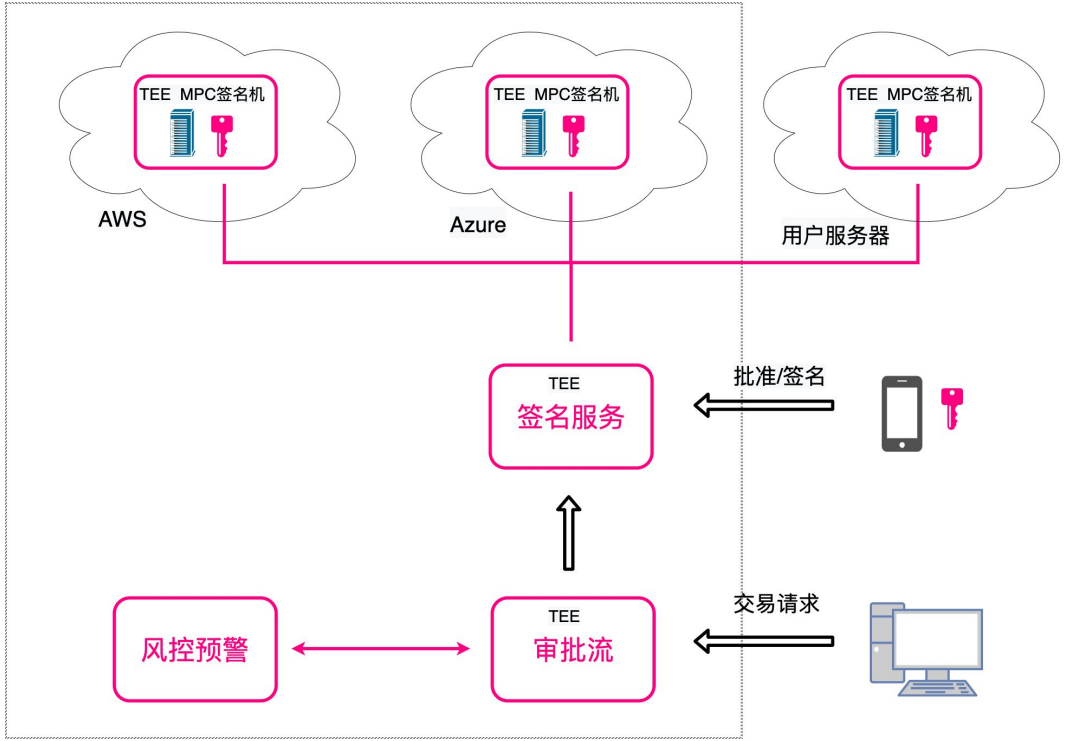


Defi合约管理



4 更安全的数字资产协同存管方案

- 多云
- 可信执行环境
- 私钥分片管理
- MPC签名



5 MPC钱包演示

- 分布式生成私钥分片
- 分布式协同签名
- 平台不掌握私钥分片
- 用户掌管私钥分片
- 完整私钥从未出现

联系方式

公司邮箱:market@istring.com

本人邮箱:hezong@istring.com

公　　司:弦冰科技有限公司

职　　位:首席科学家