IBM

Open by design™

IBM开源技术微讲堂

区块链和HyperLedger系列

第七讲

HyperLedger中的隐私与安全

更多信息，请访问：http://ibm.biz/opentech-ma

“区块链和HyperLedger” 系列公开课

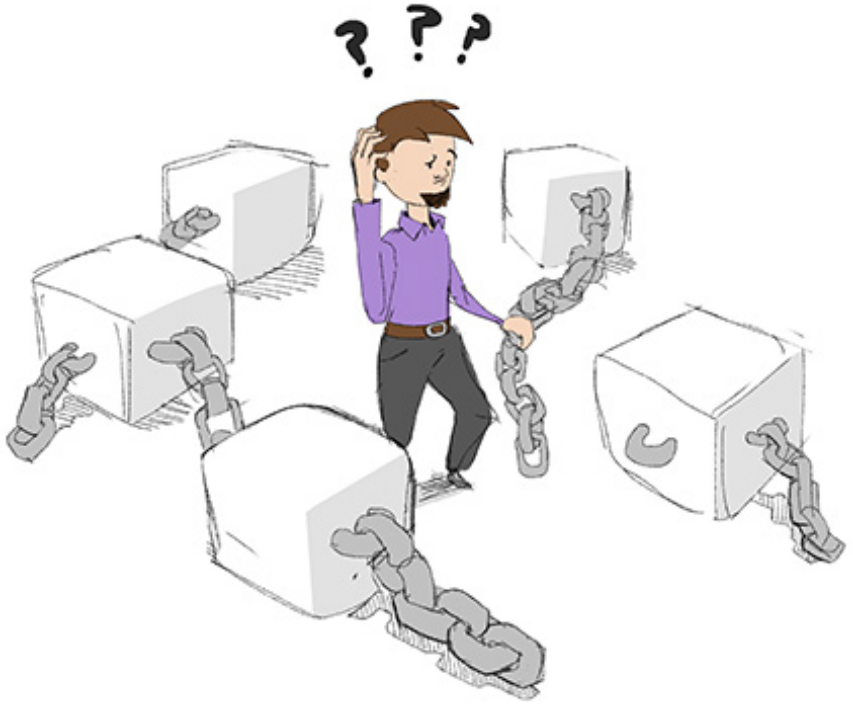
- 每周四晚8点档
  - 区块链商用之道
  - HyperLedger review
  - HyperLedger架构解读
  - HyperLedger 中的共享账本
  - HyperLedger中的共识管理
  - **HyperLedger中的隐私与安全**
  - HyperLedger应用案例赏析

讲师介绍—赵冬路

- IBM中国实验室服务团队
- Hyperledger开源社区爱好者
- CDL Blockchain Community 负责人
- 信息安全资深专家
- 参与国内金融保险行业Blockchain技术实施

议程

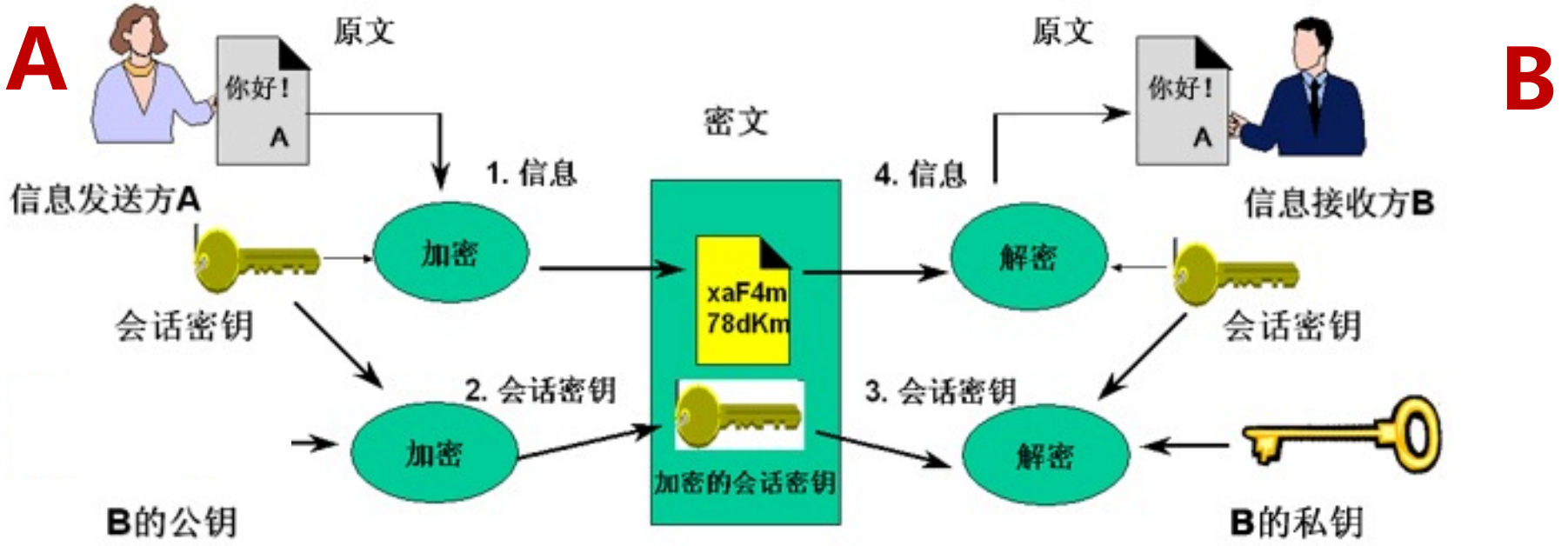
- PKI等密码学技术基础
- 区块链的基本数据模型
- 如何保障交易数据的不可更改
- 如何保障交易的私密性
- 如何保障交易的可监管能力
- 如何保护隐私



PKI（Public Key Infrastructure）公钥体系的一些基本概念

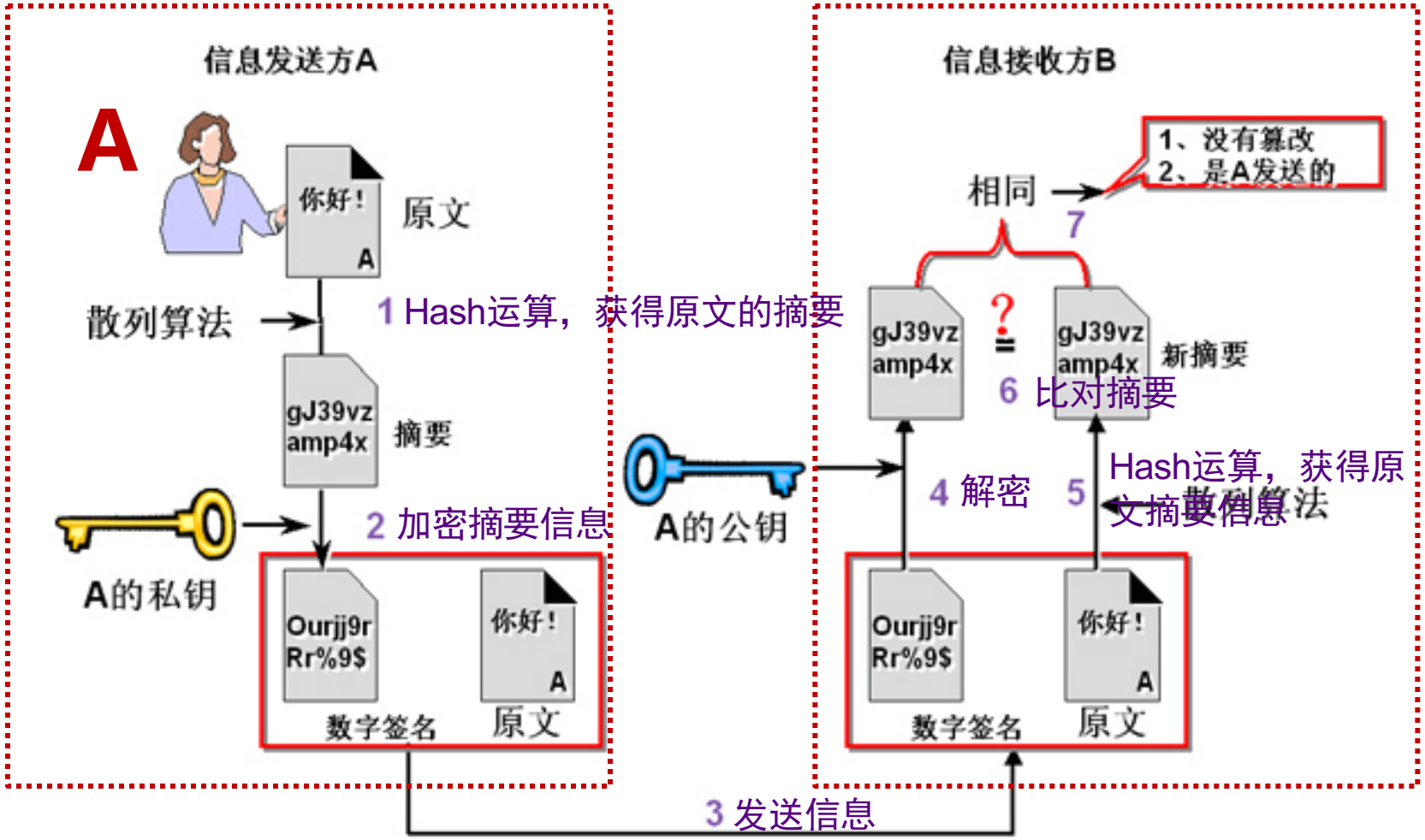
– 对称密码算法，典型算法：DES, AES, ...
  - 加解密方共用一个密钥
  - 加/解密速度快，但密钥分发比较困难
– 哈希或散列函数（Hash），典型算法SHA,MD5
  - 如果两个散列值是不相同的，那么这两个散列值的原始输入也是不相同的
  - 用于信息压缩，并发现信息是否发生变化
  - 计算速度快，特定算法其结果长度统一
  - 目前至少使用SHA256

– 非对称密码算法（公钥体系），典型算法：RSA, ECC
  - 加解密时，通讯一方有一对密钥（公钥和私钥）
  - 公钥可以公开，分发给任何人
  - 私钥不可以公开，严格持有，例如U盾中存放私钥等
  - 公钥加密，只能用私钥解密，反之亦然
  - 加/解密速度较慢，但无密钥分发问题
  - 区块链主要使用ECC椭圆曲线算法
– 公钥体系与对称密钥相结合的加密方式
  - 公钥体系通常运算性能低，做大量数据加解密力不从心
  - 通常利用公钥体系实现对称密钥的安全交换



PKI --- 基于公钥体系的签名和验签机制

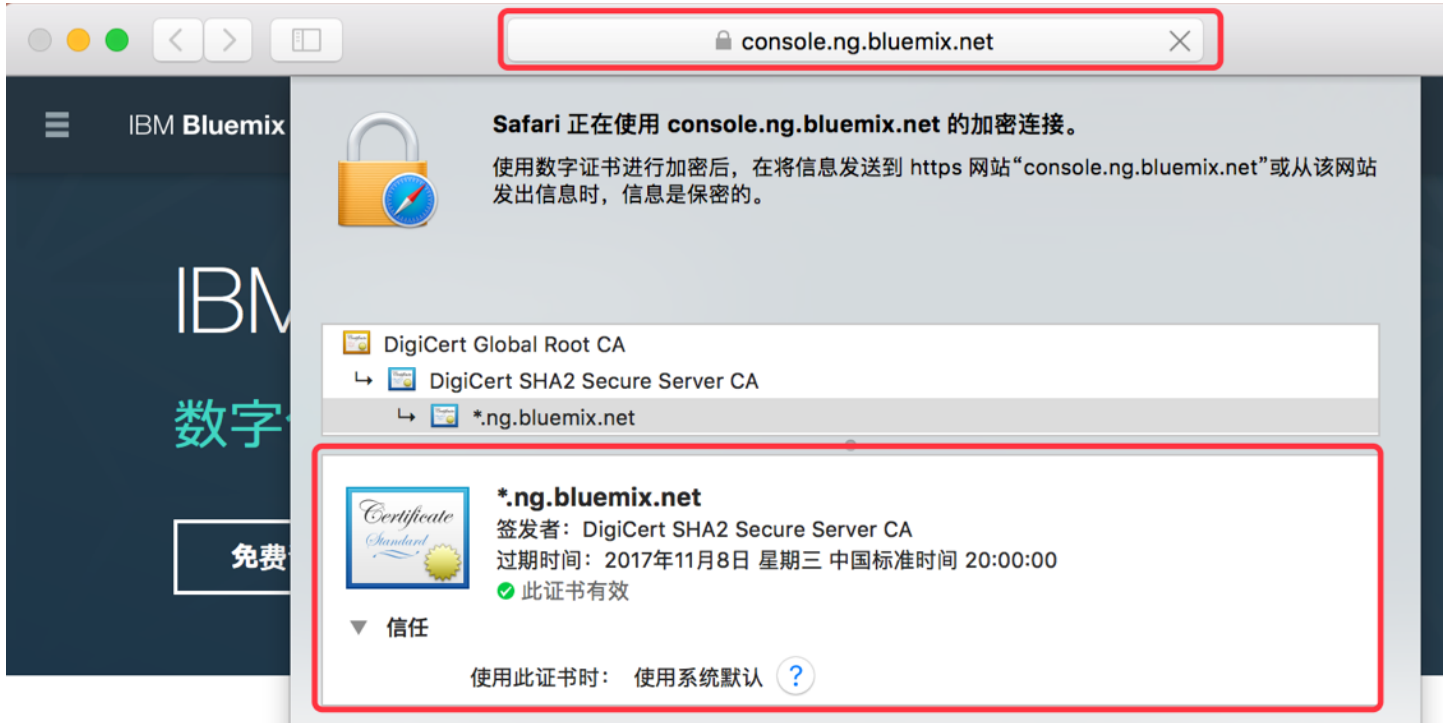
– 数字签名的目的：检测数据未经授权的修改，签名者的身份识别和抗抵赖。

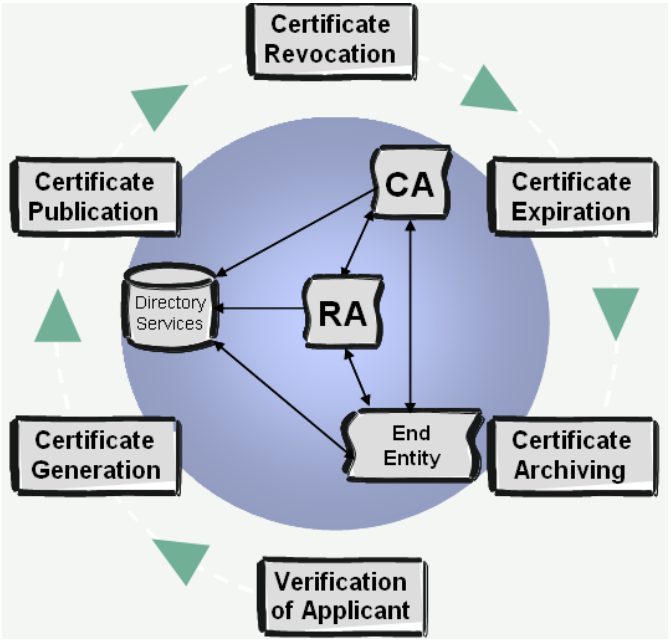


– Hash的意义：
  - 缩小存储或传输的数据量
  - 规避公钥体系加解密性能低下的问题
– 如果发送的信息没有篡改，那么也只有使用A的公钥才可以通过相应的验签
– 此验签过程，若通过，则表示信息一定是“公钥A”的持有者制作
– 数字证书体系也是以此为核心；
– 如果“张三有数字身份”，则：
  - 张三有自己的一对公钥和私钥
  - 数字身份的发证机关，证明了这个公钥对应的持有人是张三；

PKI --- 数字证书(Certificate) 和CA（认证中心，数字证书发证系统）

- 数字证书（Digital Certificate），又叫“数字身份证”、“网络身份证”，是由认证中心发放并经认证中心数字签名的，包含公开密钥拥有者以及公开密钥相关信息的一种电子文件，可以用来证明数字证书持有者的真实身份
- 数字证书采用公钥体制:
  - **数字证书是”公钥+证书名称信息+签发机构对证书的数字签名”、匹配的私钥**
  - **数字证书遵从X.509国际标准**
- 每一个用户有一个各不相同的名称，一个可信的认证中心CA（Certificate Authority）给每个用户分配一个唯一的名称并签发一个包含用户名称和公钥的证书。
- 证书可以存储在网络中的数据库中。用户可以利用网络彼此交换证书。当证书撤销后，签发此证书的CA仍保留此证书的副本，以备日后解决可能引起的纠纷。



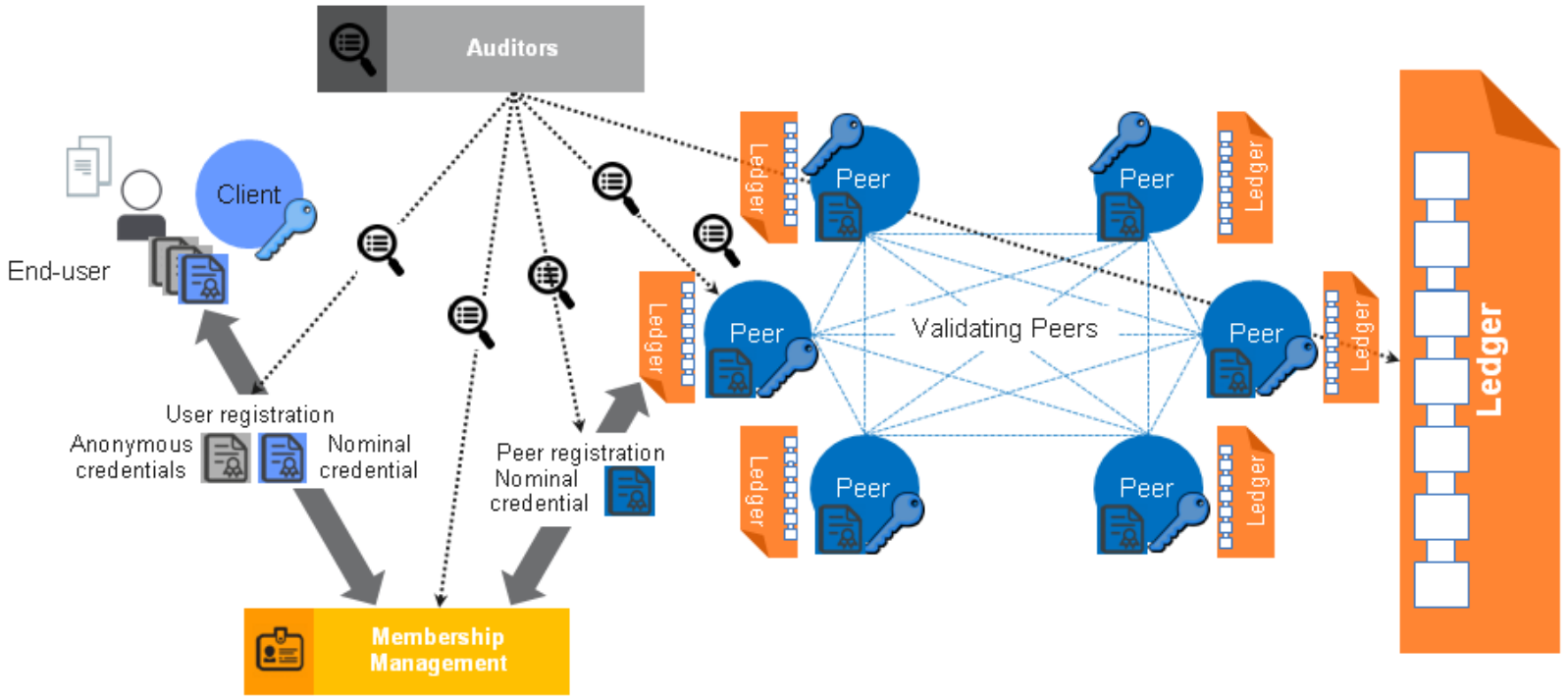


区块链的业务安全需求

- **不可更改的加密交易数据（分布式账本）**
- **可追责、不可陷害**
- **隐私保护：交易匿名、交易不可关联**
- **监管和审计支持**

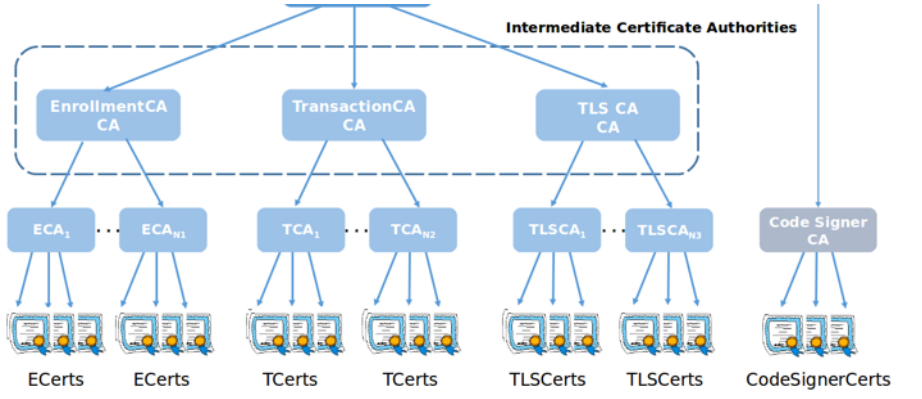
Business Security Requirements

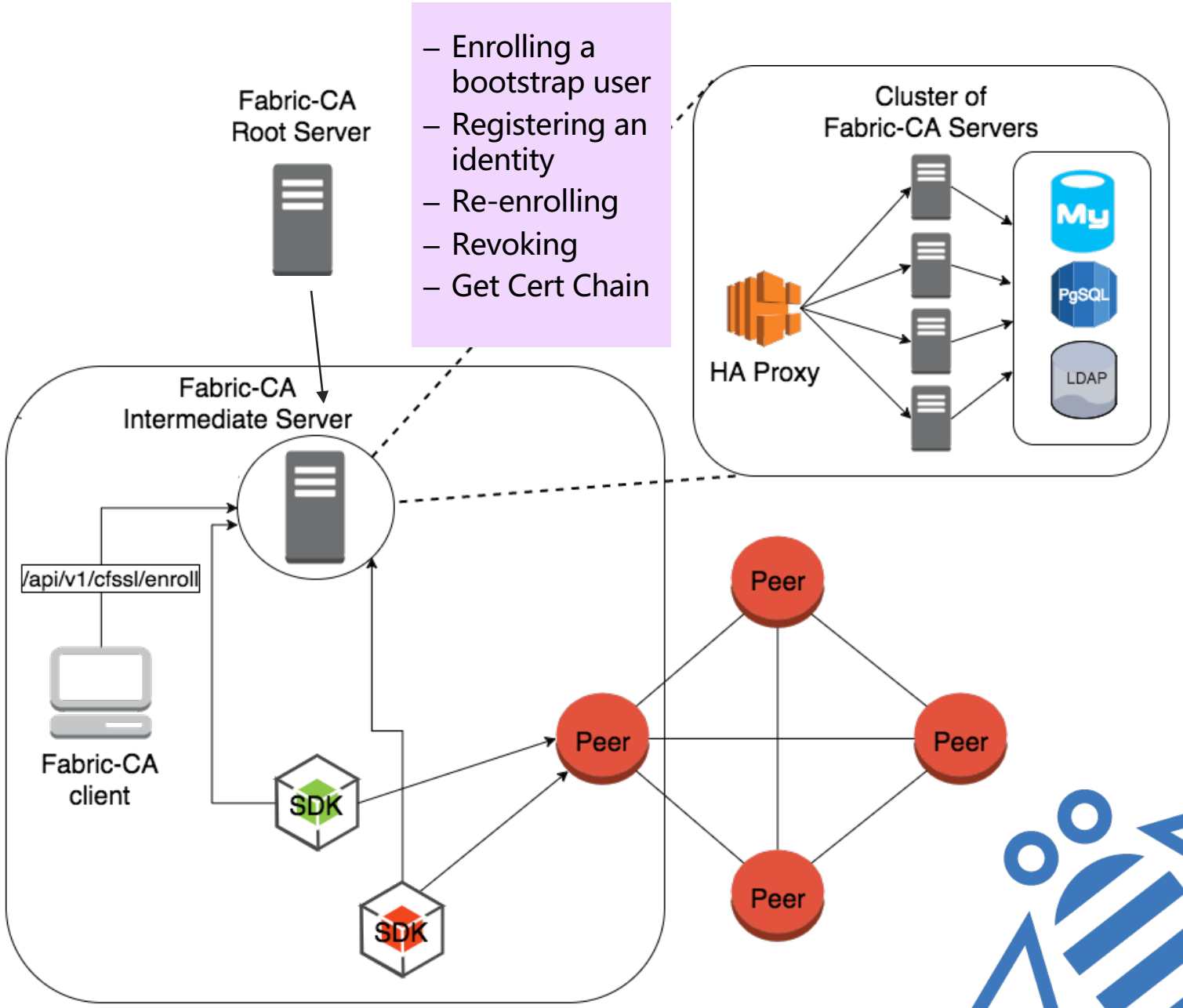
- **Accountability** and **non-frameability** are two reasons that identity management is a critical component
- Transaction Privacy
  - **Transaction anonymity**, where the owner of a transaction is hidden among the so called anonymity set, which in the fabric, is the set of users.
  - **Transaction unlinkability**, where two or more transactions of the same user should not be linked as such.
- **Audit support**



HyperLedge Fabric的CA支撑环境

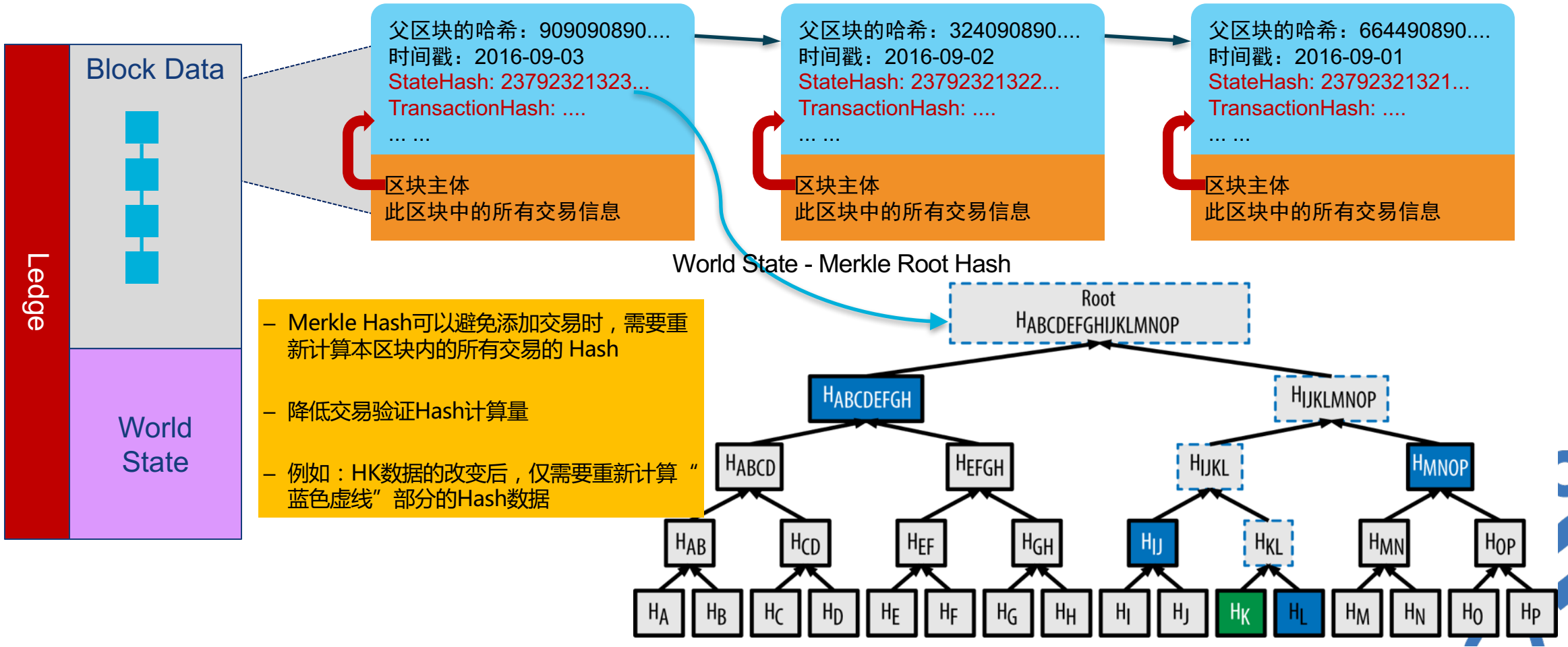
– CA是Membership的重要组件之一
– 满足于Fabric的安全需求，为参与各方实现：
  • 用户注册
  • 证书签发
  • 证书吊销
  • 发布证书链
– 基于RESTful/CLI等多种接口方式，服务于Blockchain的各个环节，包括：
  • T-Cert – Transaction Certificate (交易证书证书)，执行交易时使用
  • E-Cert – Enrollment Certificate (注册证书)，携带实体信息的证书
  • CSR – 证书吊销列表





区块数据的结构 --- 区块链

– 把一段时间内生成的信息（包括数据或代码）打包成一个区块，盖上时间 戳，与上一个区块衔接在一起，每下一个区块的页首都包含了上一个区块的索引数据，然后再在本页中写入新的信息，从而形成新的区块，首尾相连，最终形成了区块链。



区块数据的构成 --- 事务（交易）数据

– 每个区块中包含一系列的事务数据
– 事务中包含事务发起方的数字签名（TCert – 交易证书）
– 每个Block中包含所有事务的Hash, 用于共识时检查事务信息是否与其它节点一致
– 每个Block中包含World State的Hash, 用于共识时检查State信息是否与其它节点一致

Block #N

```
message Block {
  version = 1;
  google.protobuf.Timestamp timestamp = 2;
  bytes transactionsHash = 3;
  bytes stateHash = 4;
  bytes previousBlockHash = 5;
  bytes consensusMetadata = 6;
  NonHashData nonHashData = 7;
}

message BlockTransactions {
  repeated Transaction transactions = 1;
}
```

包括

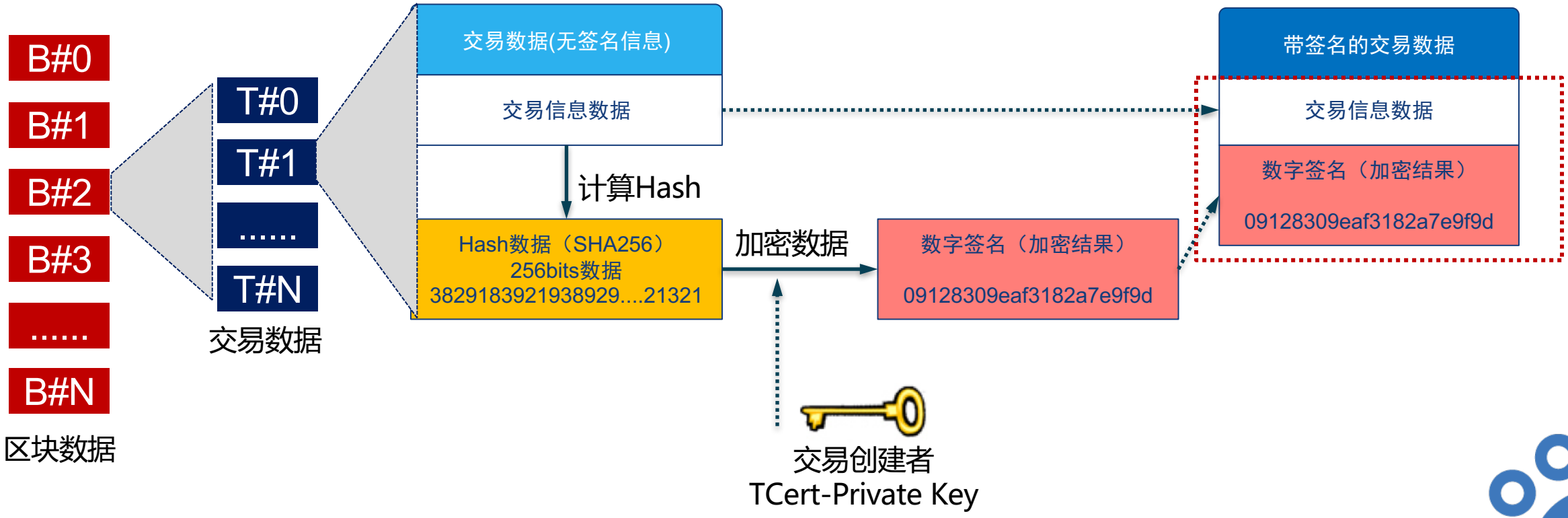
Transaction #N

```
message Transaction {
    enum Type {
        UNDEFINED = 0;
        CHAINCODE_DEPLOY = 1;
        CHAINCODE_INVOKE = 2;
        CHAINCODE_QUERY = 3;
        CHAINCODE_TERMINATE = 4;
    }
    Type type = 1;
    string uuid = 5;
    bytes chaincodeID = 2;
    bytes payloadHash = 3;

    ConfidentialityLevel confidentialityLevel = 7;
    bytes nonce = 8;
    bytes cert = 9;
    bytes signature = 10;

    bytes metadata = 4;
    google.protobuf.Timestamp timestamp = 6;
}

message TransactionPayload {
    bytes payload = 1;
}

enum ConfidentialityLevel {
    PUBLIC = 0;
    CONFIDENTIAL = 1;
}
```

– PKI相关的密码学在BlockChain中的应用保障单个Peer上数据的完整性
  - 数字签名
    – 不可抵赖，防篡改
  - Hash
  - 加密
    – 事务隐私保护
    – 数据访问控制
– 共识机制及BlockChain数据分布化，可以防止某个Peer造假，达到高度自治

区块数据的构成 --- 如何做到交易抗抵赖

– 由交易“提交方”使用自己的“数字证书”对每个交易做 “数字签名”来确保交易无法伪造
– 这笔交易确实是你提交的，别人无法伪造你的交易！
– 既然无法伪造一个交易，所以如果存在一个你的“交易”，那么你也无法抵赖
– 这里的“你”是指用户的数字身份，数字身份就是“某个数字证书的持有者”

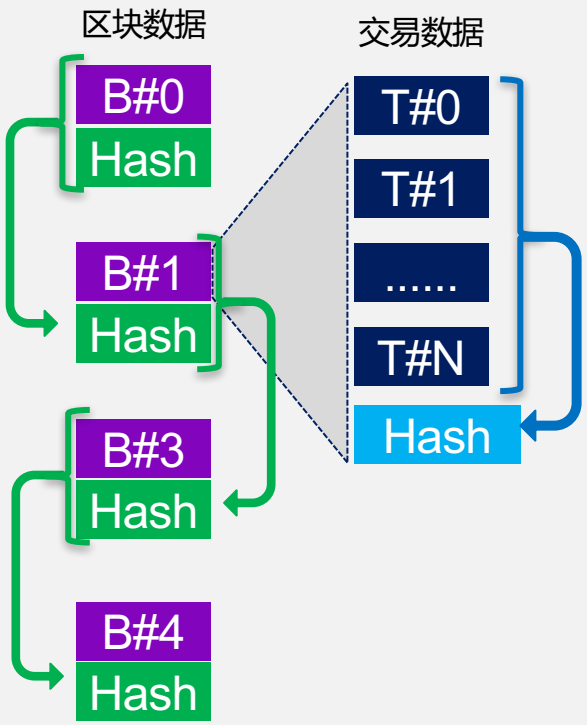


在PKI/CA体系完备性保障的情况下，数字签名保障了我们无法伪造他人的交易。
既然无法伪造他人的交易，因此交易的数字签名显示是你做的，那么必定是你做的。

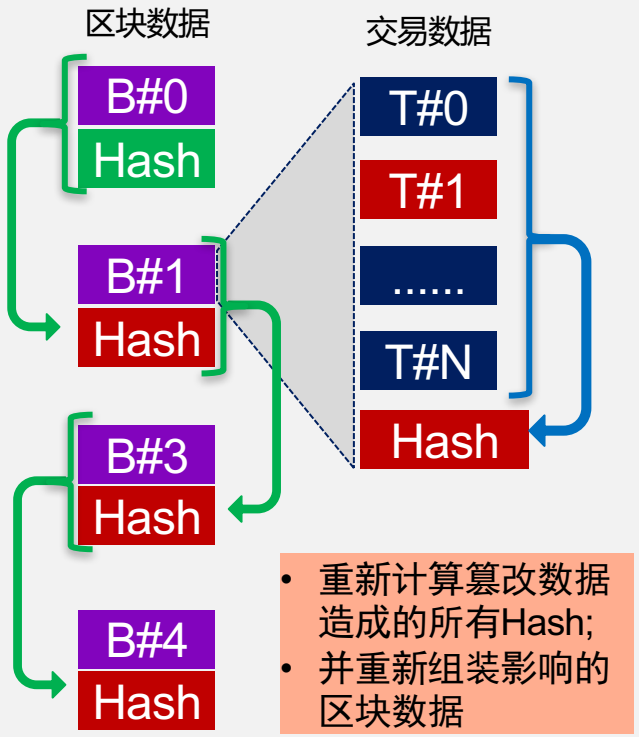
如何防止分布式账本的伪造

– 利用数字签名，伪造一个他人的单个交易非常困难，除非能够获得他人数字证书的私钥
– 另外分布式账本可以防止如下类型的篡改：
  • 删除历史交易
  • 伪造自己的历史交易

篡改前的正常区块数据（某个Peer）



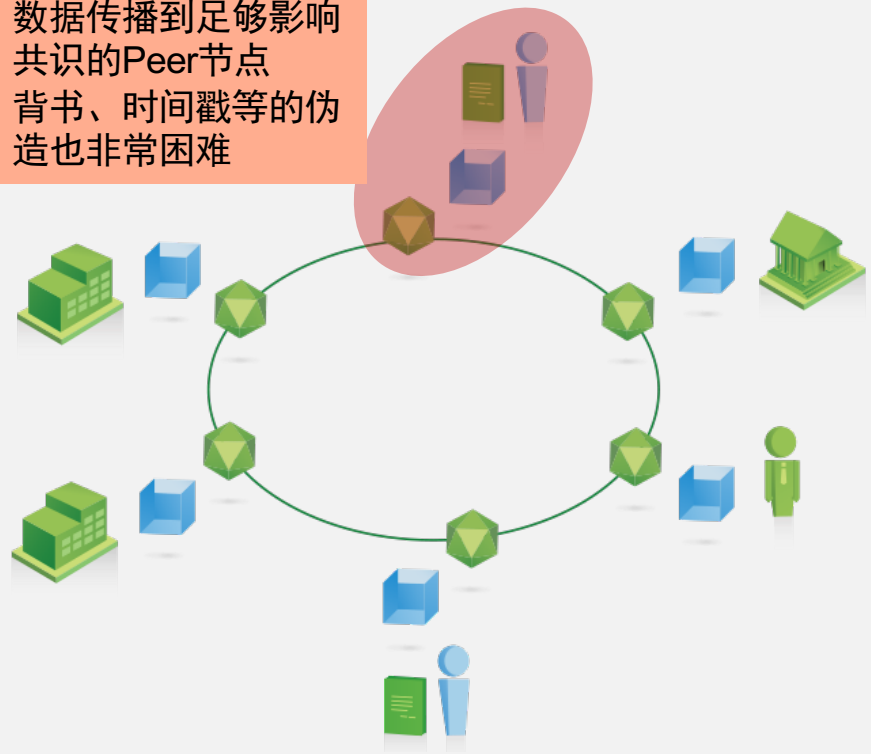
篡改：删除区块B#1的T#1交易(某个Peer)



- 重新计算篡改数据造成的所有Hash;
- 并重新组装影响的区块数据

在整个区块链网络中生效篡改

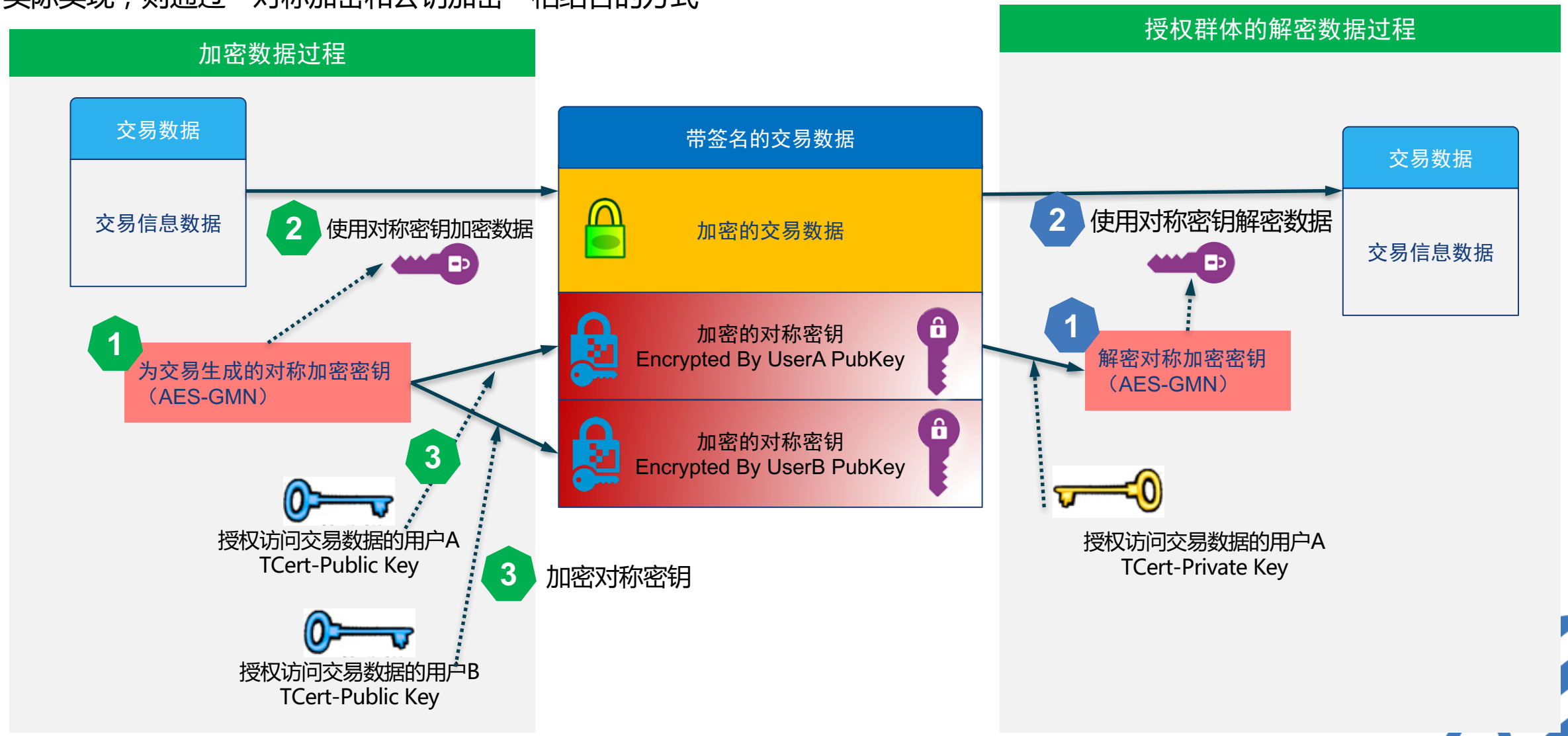
- 需要将篡改后的区块数据传播到足够影响共识的Peer节点
- 背书、时间戳等的伪造也非常困难



注：上述举例为逻辑层面的示意，交易伪造除了需要篡改交易，可能还需要伪造与之对应的State Ledge(World state)，并解决相关篡改在整个区块链网络中的生效，基本原理类似

如何保障私密

– 确保交易仅仅向有限的全体可见，不对非授权的全体公开
– 简单来看，分别使用授权用户的“公钥”加密“数据”，只有授权用户能够用自己的“私钥”解密数据
– 实际实现，则通过“对称加密和公钥加密”相结合的方式



如何既保障交易私密，又可以实现监管

– “监管”是指无需交易方授权，监管者可以解密交易
– 但监管不能侵犯“不可抵赖性”，即，监管者不可以伪造别人的交易
– 采用PKI体系的“**双密钥对---签名密钥对和加密密钥对**”模式来实现：
  • 证书持有者有一对签名用途的密钥对
  • 证书持有者有一对加密用途的密钥对
  • CA签发证书时，对加密用途的密钥对进行备案，交由密钥管理中心存放
    – 特定的情况下，提取某用户的解密私钥，解密相关的交易数据
  • 签名用途的密钥对仍然在用户端产生，不做备案
    – 无私钥的情况下，无法伪造签名，因此无法伪造别人的交易
    – 从证书申请，到证书生成，常规意义的CA/RA体系都可以保障签名密钥的用户私密性

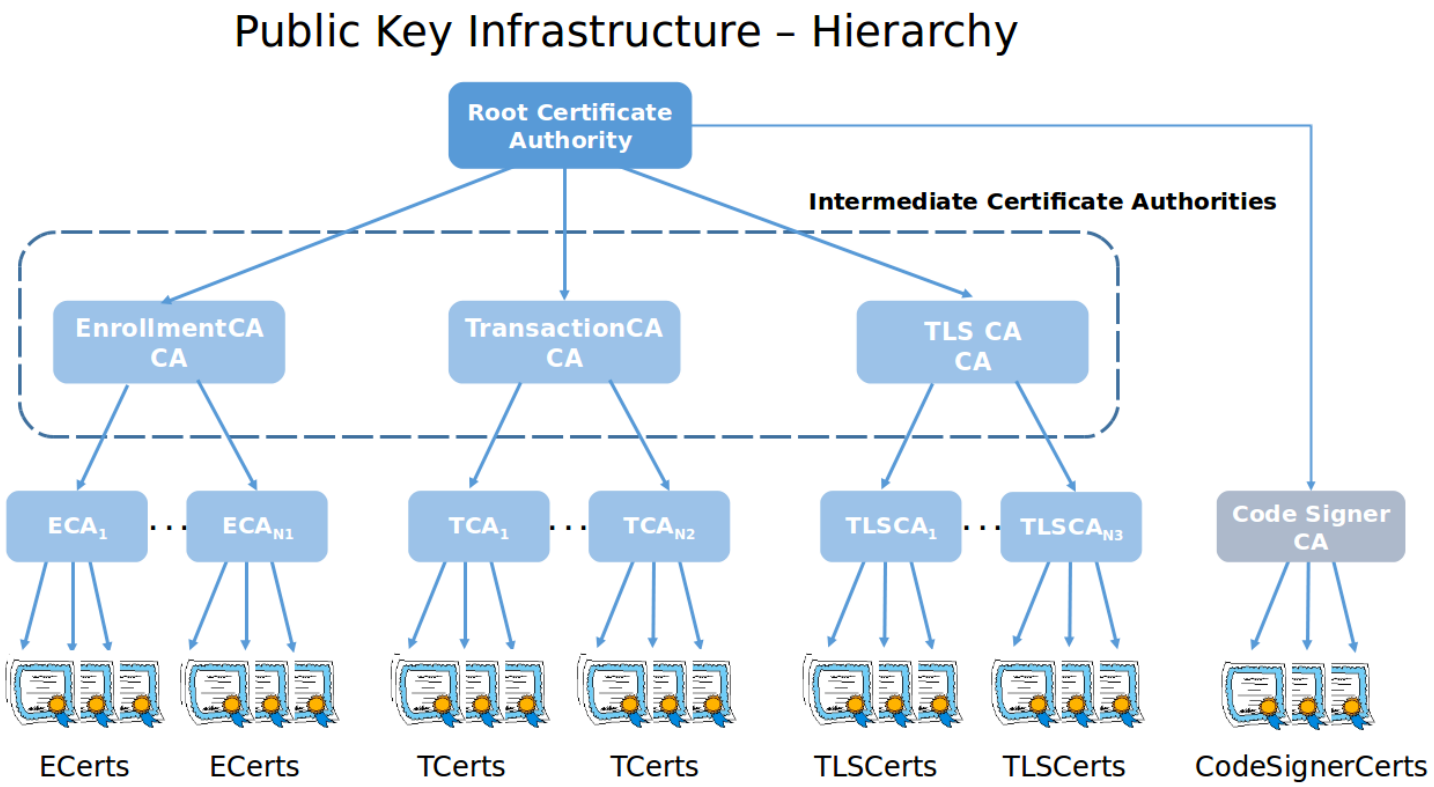
– 签名密钥对的管理
  • 签名密钥对由签名私钥和验证公钥组成；
  • 签名私钥是发送方身份的证明,具有日常生活中公章、私章的效力；
  • 签名私钥绝对不能够做备份和存档,丢失后只需重新生成新的密钥对；
  • 验证公钥需要存档，用于验证旧的数字签名。

– 加密密钥对的管理
  • 加密密钥对由加密公钥和解密私钥组成；
  • 为防止密钥丢失时数据无法恢复，解密私钥应该进行备份，同时还可能需要进行存档，以便能在任何时候解密历史密文数据；
  • 加密公钥则无需备份和存档,加密公钥丢失时，只需
    重新生成密钥对即可。

如何保护隐私？

– 确保从交易中无法追溯交易创建者的信息
– 问题
  • 由于交易中存在签名信息，而签名信息携带可以关联交易创建者证书的信息
  • 证书中包含交易创建者的识别信息
  • 如果不做实现特定的机制，交易中将可以追溯交易创建这的信息

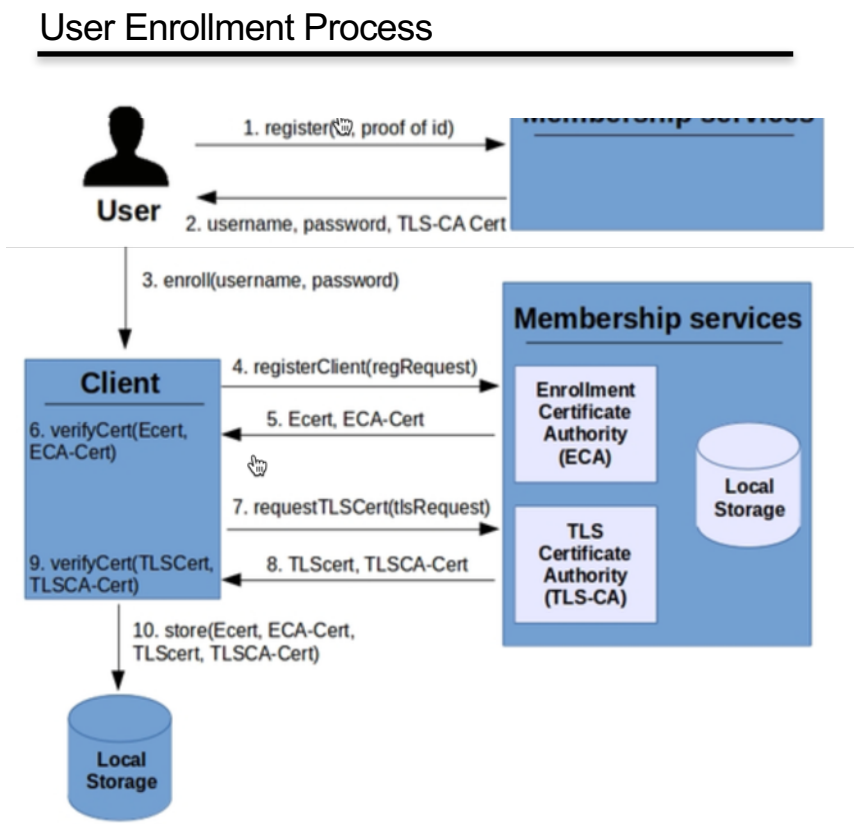
– 交易方持有多种类型的证书，交易不同环节将使用如下这些类型的证书：
  • E-Cert（Enrollment Cert）
    – 长期持有，携带或可以追溯使用者信息
    – 用于身份认证
  • T-Cert（Transaction Cert）
    – 每个交易时生成，用于交易的签名
  • TLS-Cert，长期持有，主要用于SSL/TLS通讯

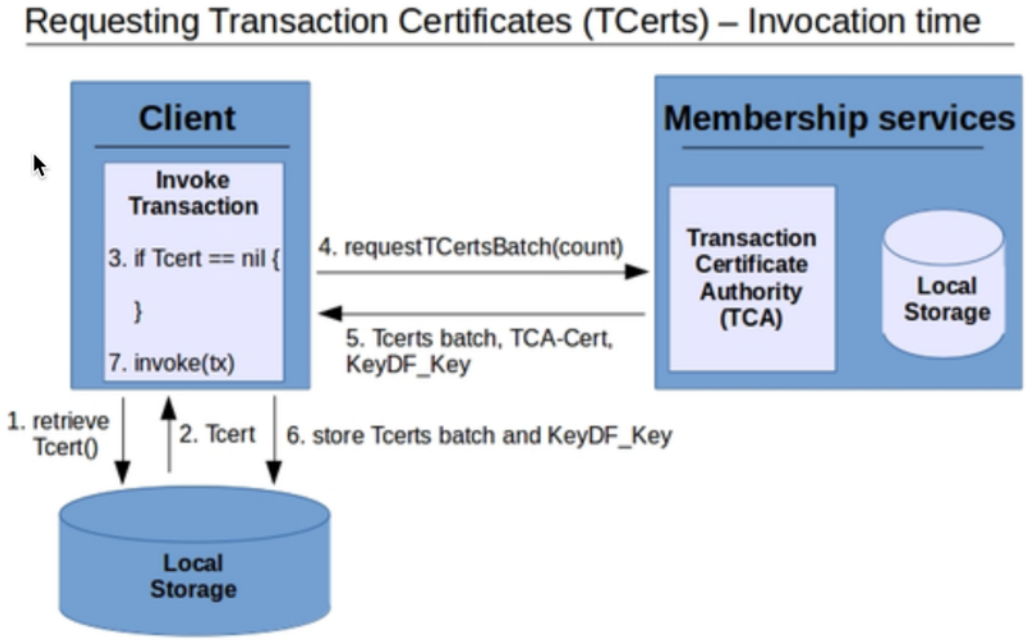


Reconciling transactional privacy with identity management

Open by design™

- Add certificates to transactions to implement a "permissioned" blockchain utilize
- A two-level system:
  - (Relatively) static enrollment certificates (ECerts), acquired via registration with an enrollment certificate authority (CA).
  - Transaction certificates (TCerts) that faithfully but pseudonymously represent enrolled users, acquired via a transaction CA.
- Offer mechanisms to conceal the content of transactions to unauthorized members of the system.





- 每个交易使用一个新的TCert
- TCert中不显式携带交易创建者的信息
- TCert和ECert的关系被隐秘保护

主要加密算法对比

非对称加密算法（公钥算法）	ECC/国密SM2	RSA
计算结构	基于椭圆曲线	基于特殊的可逆模幂运算
计算复杂度	完全指数级	亚指数级
相同的安全性能下所需的公钥位数	较少，（160位的ECC与1024位的RSA具有相同的安全等级）	较多
密钥生成速度	较RSA算法快百倍以上	慢
解密加密速度	较快	一般
安全性难度	基于离散对数问题ECDLP数学难题	基于分解大整数的难度

ECC（Elliptic Curves Cryptography）

AES（Advanced Encryption Standard）

对称加密算法	AES	国密SM4	3DES
计算结构	数据块长度和密钥长度都可变的分组加密RIJNDAEL算法	基本轮函数加迭代，含非线性变换	使用标准的算法和逻辑运算，先替换后置换，不含非线性变换
计算轮数	10/12/14轮	32轮	16*3轮
分组长度	128/192/256位	128位	64位
密钥长度/有效密钥长度	128位/112位	128位/128位	128位/112位
实现性能	软件、硬件实现都较快	软件、硬件实现都较快	软件慢、硬件快
安全性	较高	较高	较高

参考资料

- HyperLedge Fabric Protocol Specification #Security
  - http://openblockchain.readthedocs.io/en/latest/protocol-spec/#4-security_1
  - https://github.com/hyperledger/fabric/blob/master/docs/source/protocol-spec.rst

IBM

Open by design™

IBM开源技术微讲堂

区块链和HyperLedger系列

Q&A

扫码入群，与讲师互动

更多信息，请访问：http://ibm.biz/opentech-ma