西安电子科技大学
XIDIAN UNIVERSITY

计算机科学与技术学院
School of Computer Science and Technology

第四章 大数据存储的安全与隐私

彭延国

ygpeng@xidian.edu.cn

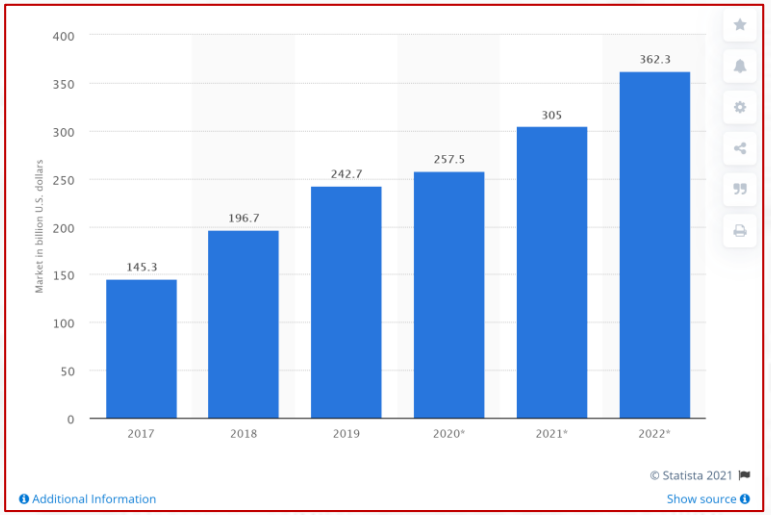
§4 大数据存储的安全与隐私 - 云计算

- 云计算(Cloud computing)
  - 是一种模式，能以泛在的、便利的、按需的方式通过网络访问可配置的计算资源(如网络、服务器、应用和服务)
  - 特征：
    - 按需获得的自助服务；
    - 广泛的网络接入方式；
    - 资源的规模池化；
    - 快捷的弹性伸缩；
    - 可计量的服务。

腾讯云

§4 大数据存储的安全与隐私 - 云计算发展历史

- 1983年，太阳电脑提出“网络是电脑”（“The Network is the computer”）。
- 1996年，Compaq公司在其公司的内部文件中，首次提及“云计算”这个词汇。
- 2006年3月，亚马逊推出弹性计算云服务。
- 2007年10月，Google与IBM开始在美国大学校园推广云计算。
- 2008年7月29日，雅虎、惠普和英特尔宣布推出云计算研究测试床，推进云计算。
- 2008年8月3日，戴尔申请“云计算”（Cloud Computing）商标，此举旨在加强对这一未来可能重塑技术架构的术语的控制权。



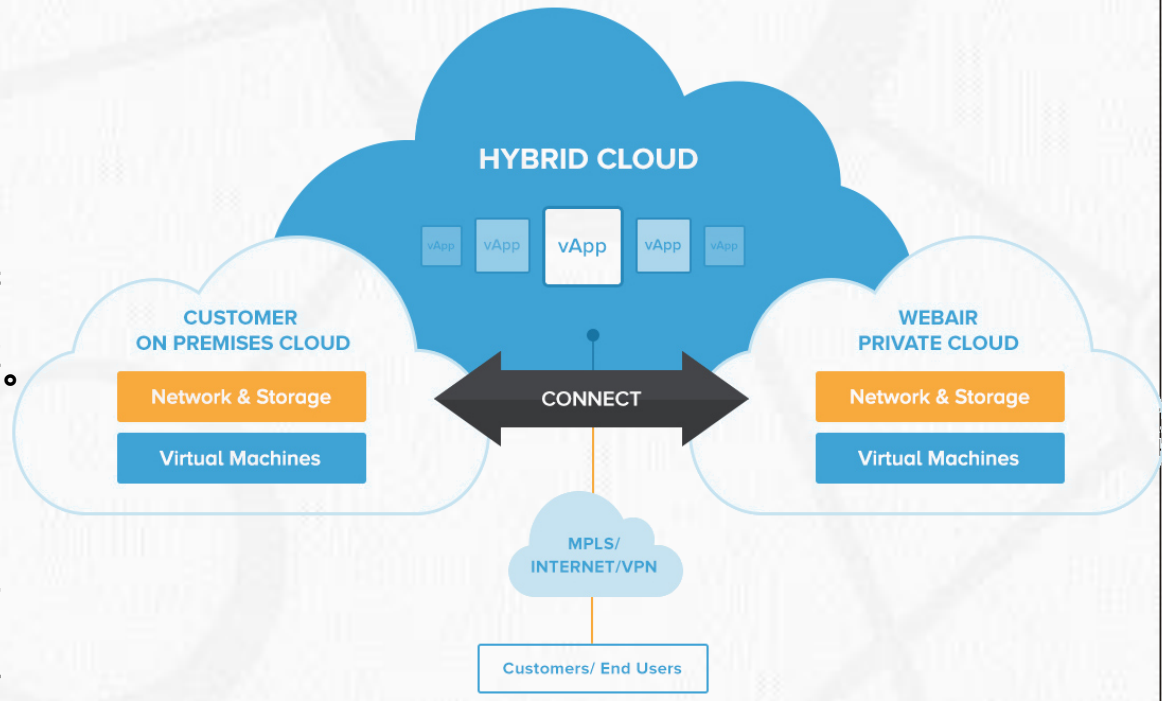
公有云全球范围消费趋势(from Statista)

	Total GDP spend USD Bn	Total IT spend[1] USD Bn	IT spend[1] as a % of total GDP spend (2018)	Total IT spend[1] USD Bn	Public Cloud spend as a % of total IT spend[1] (2018)
UK	2,622	137	5.2%	137	11.4%
USA	19,391	911	4.7%	911	11.4%
Canada	1,653	63	3.8%	63	11.3%
Australia	1,323	48	3.6%	48	7.7%
World	80,000	2,362	3.0%	2,362	7.9%
Germany	3,677	103	2.8%	103	6.9%
Brazil	2,056	40	1.9%	40	7.9%
India	2,597	42	1.6%	42	6.0%
China	12,238	172	1.4%	172	2.7%
Russia	1,578	15	1.0%	15	2.9%

2019年政府IT开支趋势(from NASSCOM)

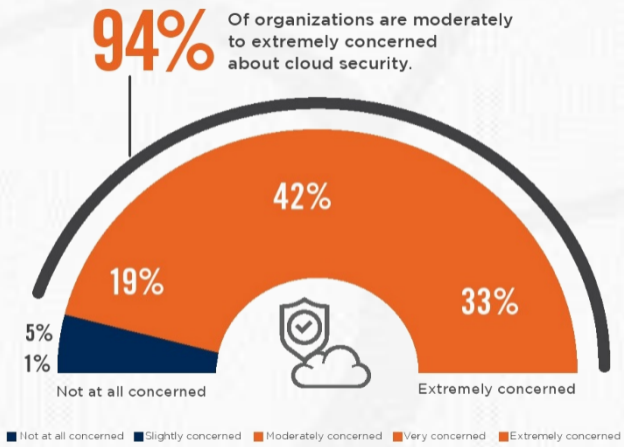
§4 大数据存储的安全与隐私 - 云计算部署方式

- 公有云(Public cloud)
  - 公用云服务可透过网络及第三方服务供应者，开放给客户使用。
- 私有云(Private cloud)
  - 私有云是由企业自建自用的云计算中心。
- 社区云(Community cloud)
  - 社区云由众多利益相仿的组织掌控及使用，例如特定安全要求、共同宗旨等。
- 混合云(Hybrid cloud)
  - 混合云的基础施舍是由上述两种或两种以上的云组成



§4 大数据存储的安全与隐私 - 云计算安全

- 云计算安全(Cloud computing security)
  - 指一套广泛的政策、技术与被布署的控制方法，以用来保护资料、应用程序与云计算的基础设施。
  - 94%的用户关注云计算安全；
  - 66%的用户对云计算安全信心不足；
  - 6/10的企业预计其云安全预算将在未来12个月内增加。
  - 企业将其安全预算的27%用于云安全；
  - 69%的企业将其团队的安全就绪水平评为中等或低于平均水平。
- 云计算安全将会得到持续、深刻的关注。
  - 存储安全；
  - 计算安全；
  - 共享安全等



2020年用户对云安全的关注
(from Cybersecurity Insider)

2020年用户对云安全的信心
(from Cybersecurity Insider)

西安电子科技大学
XIDIAN UNIVERSITY

贯穿大数据处理的整个过程

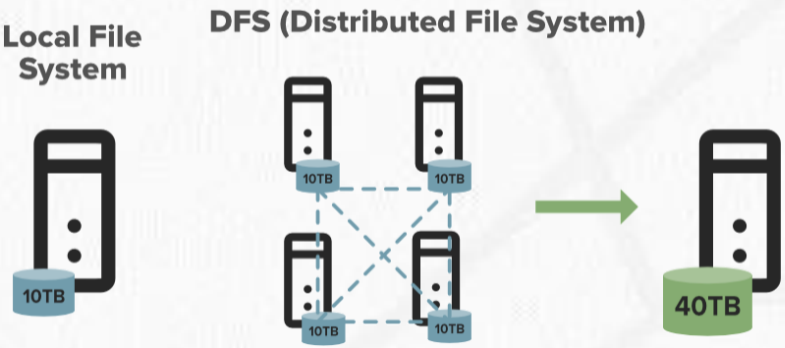
§4.1 大数据存储技术

§4.1 大数据存储技术 - 分布式文件系统

- 分布式文件系统(DFS: Distributed file system)：
  - 是指文件系统管理的物理存储资源不仅存储在本地节点上，还可以通过网络链接存储在非本地节点上；
  - 具有比本地系统更优异的数据备份、数据安全、规模可扩展等优点。

- 分布式文件系统的评价方式：
  - 数据的存储方式，即文件数据在各节点之间的分布策略；
  - 数据的读取速率，包括读写、网络传输等速率；
  - 数据的安全机制，包括冗余、备份、镜像、加解密等。

- 典型分布式文件系统：
  - GFS、HDFS、Lustre、Ceph等。

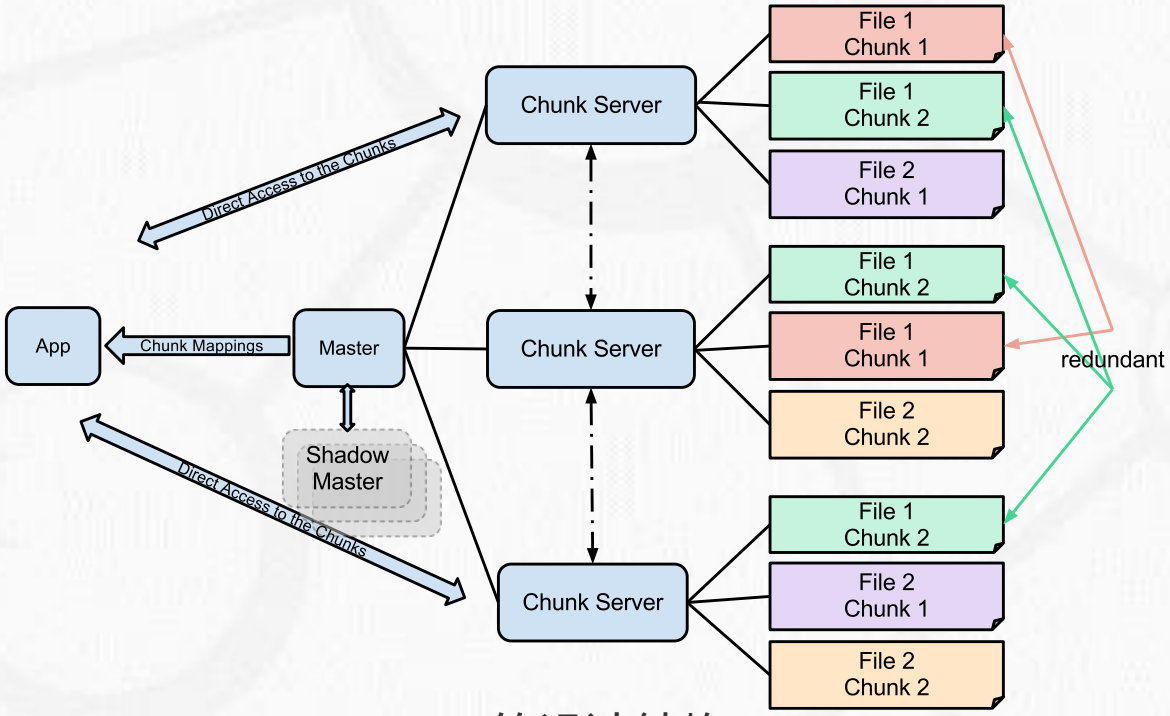


§4.1 大数据存储技术 - GFS文件系统

- Google文件系统(Google file system)
  - 一种由Google公司开发，运行于Linux平台上的专有分布式文件系统。
  - 设计理念：
    - 组件失效不再被认为是意外，而是被看做正常的现象
    - GFS的文件非常巨大
    - 对文件的操作具有特定的模式
    - 应用程序和文件系统API的协同设计提高了整个系统的灵活性
  - 一个GFS集群包含一个主服务器和多个块服务器，并被多个客户端访问。
  - 文件分成固定大小的“块”。每个块在创建时都由主服务器分配一个固定不变的64位句柄唯一标识。
  - 块服务器把块作为Linux文件存储在本地磁盘上，并根据指定的块句柄和字节范围对数据块进行读写操作。

§4.1 大数据存储技术 - GFS的设计架构

- 主服务器
  - 维护所有文件系统的元数据，包括名字空间、访问控制信息、文件到块的映射信息以及块当前的位置。
  - 控制其它系统级的活动，周期性地与块服务器通信，以下达指令和收集状态。
- GFS客户端
  - 代码被嵌入到每个应用中。
  - 它实现了文件系统API，实现主服务器与块服务器的通信从而代表应用实现读写操作。
  - 客户端与服务器交互从而实现元数据操作，但所有的数据操作都通过直接与块服务器交互而完成。



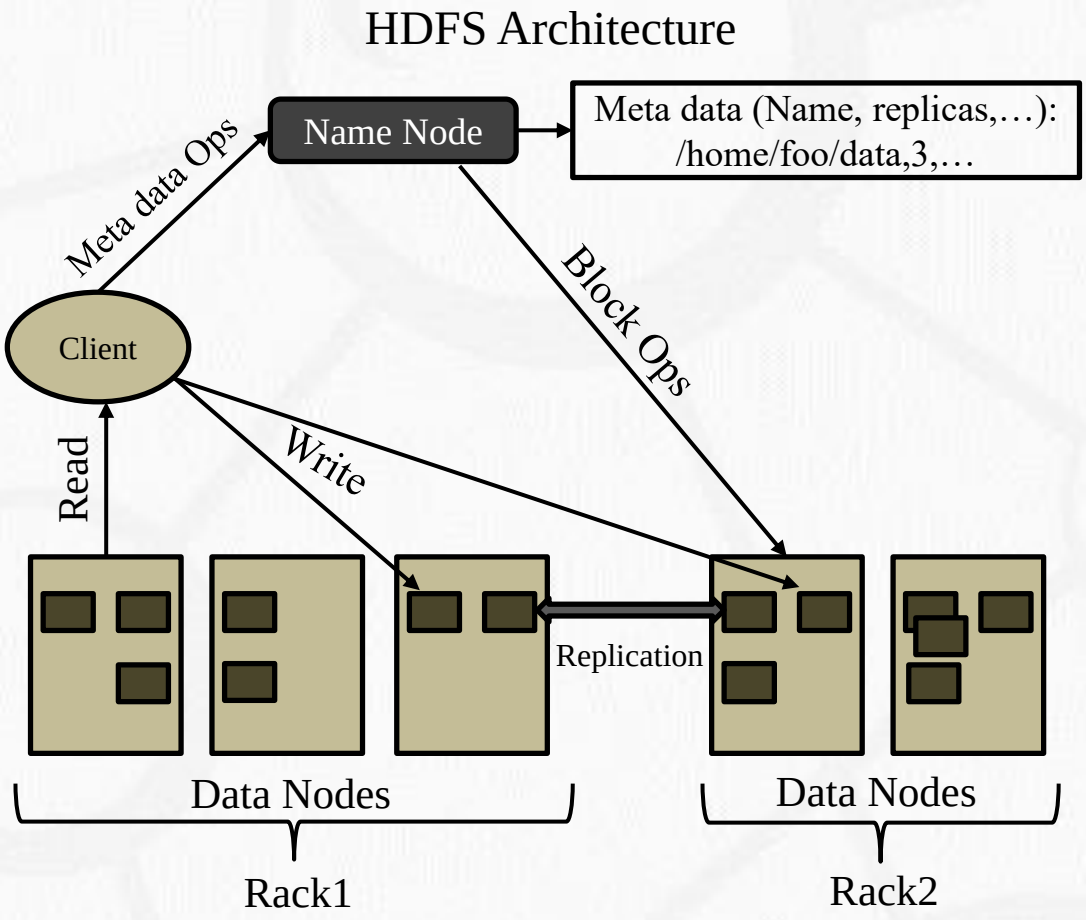
GFS的设计结构
(from Wikimedia)

§4.1 大数据存储技术 - HDFS文件系统

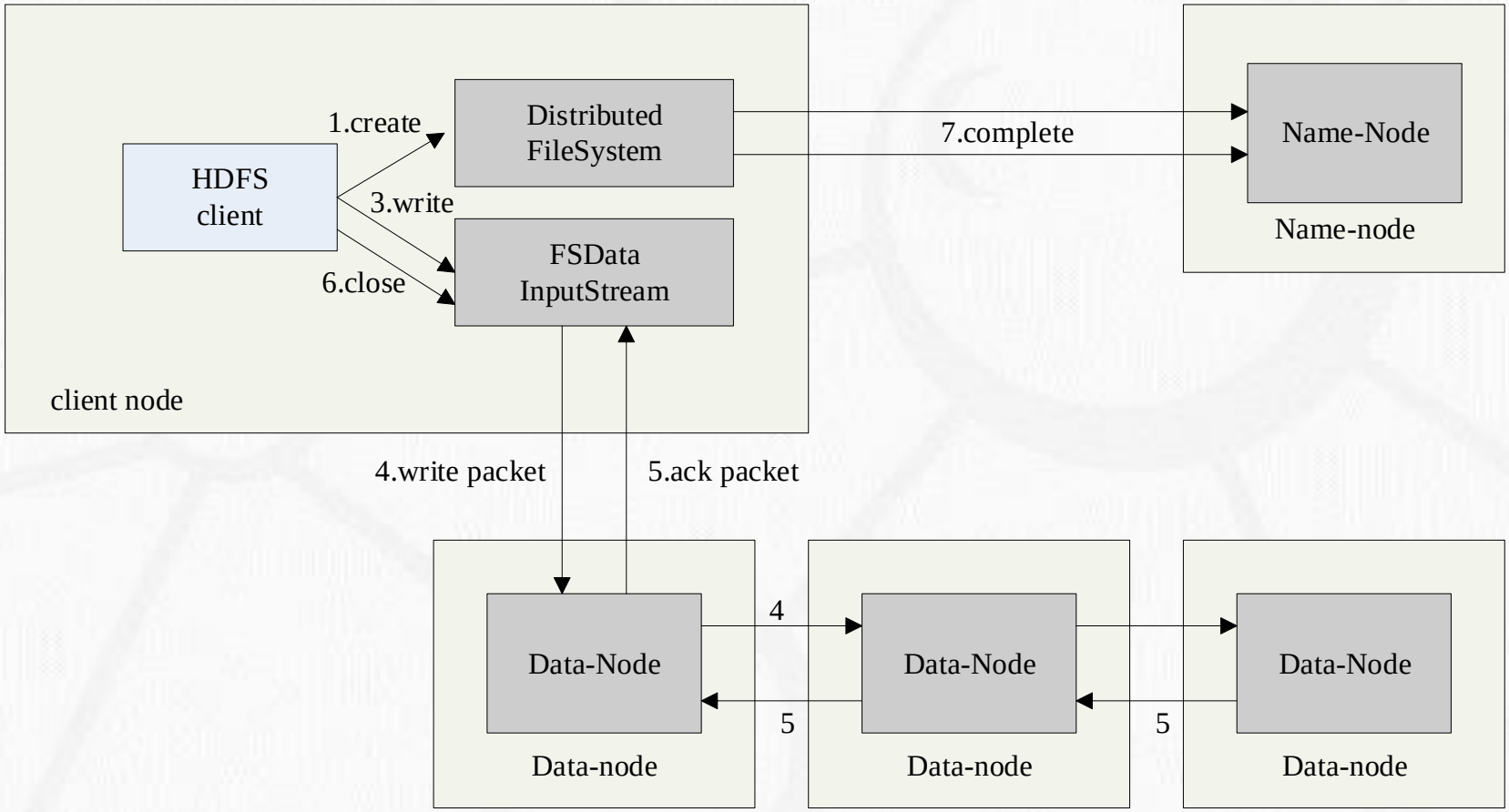
- HDFS(Hadoop distributed file system)
  - Apache Hadoop是一款支持数据密集型分布式应用程序并以Apache 2.0许可协议发布的开源软件框架。
  - HDFS是指被设计成适合运行在通用硬件(commodity hardware)上的分布式文件系统。
  - 是Apache Hadoop项目的一个子项目。
  - 具体特性：
    - “一次写入、多次读取(write-once-read many)”模型；
    - 将处理逻辑放置到数据附近，通常比将数据移向应用程序空间更好。

§4.1 大数据存储技术 - HDFS的体系结构

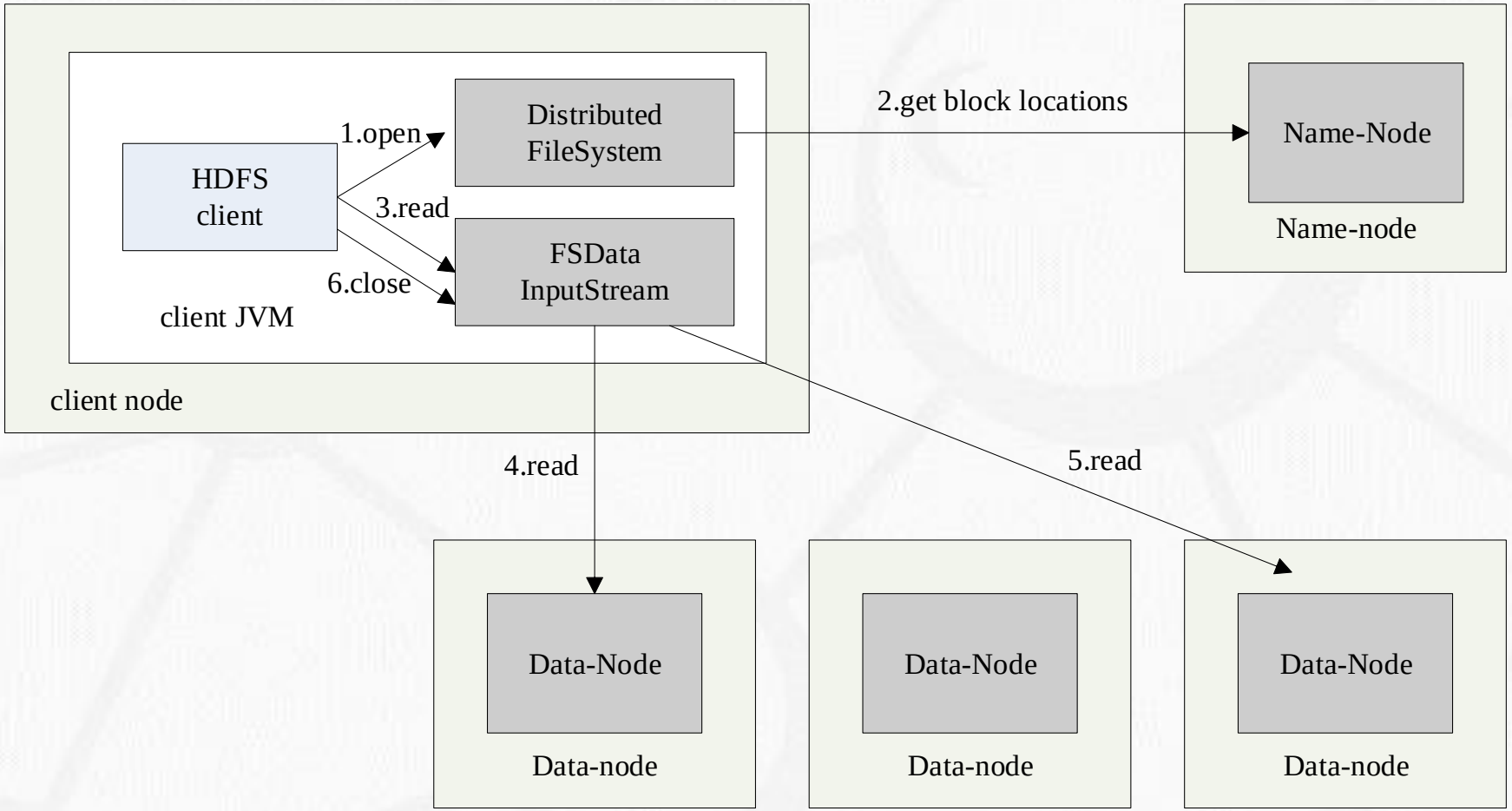
- HDFS的体系结构：
  - HDFS的高层设计包含命名节点(Name node)与数据节点(Data node)；
  - 设计原则：
    - 元数据与数据分离
    - 主从结构
    - 一次写入多次读取
    - 移动计算比移动数据更划算



§4.1 大数据存储技术 - HDFS的写操作

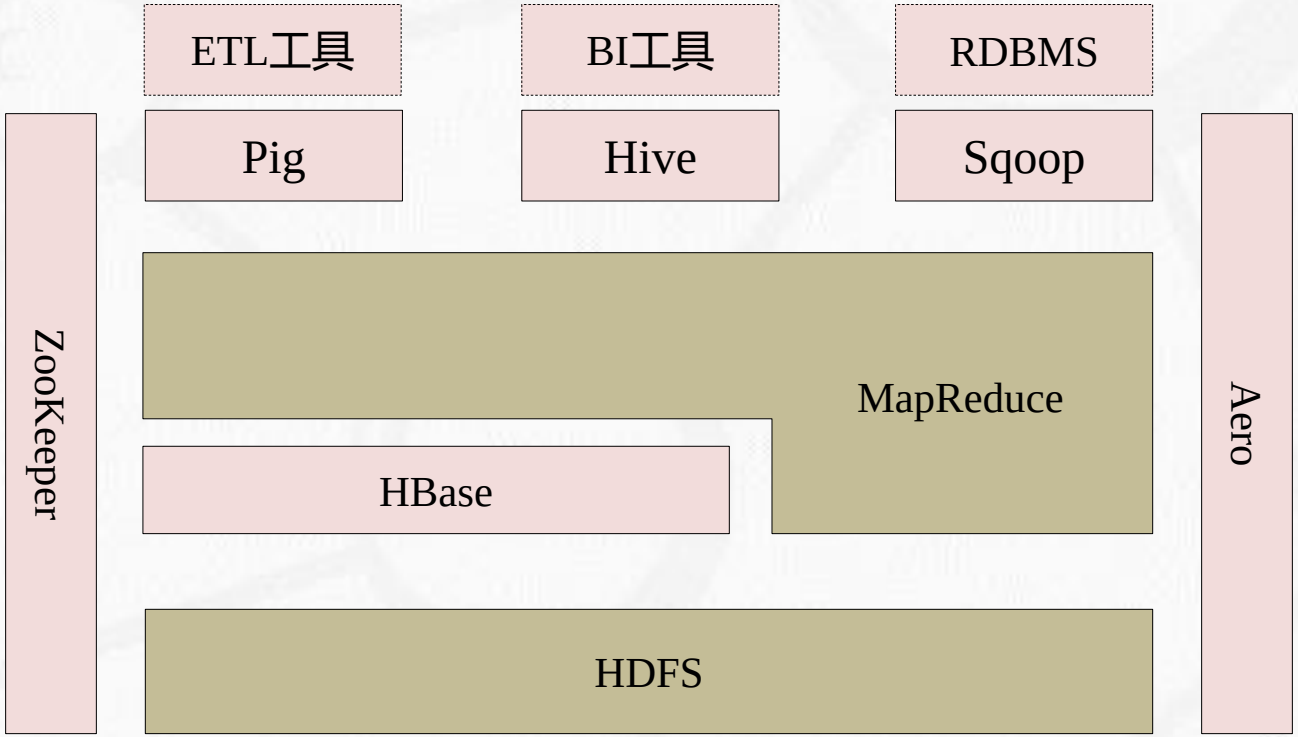


§4.1 大数据存储技术 - HDFS的读操作



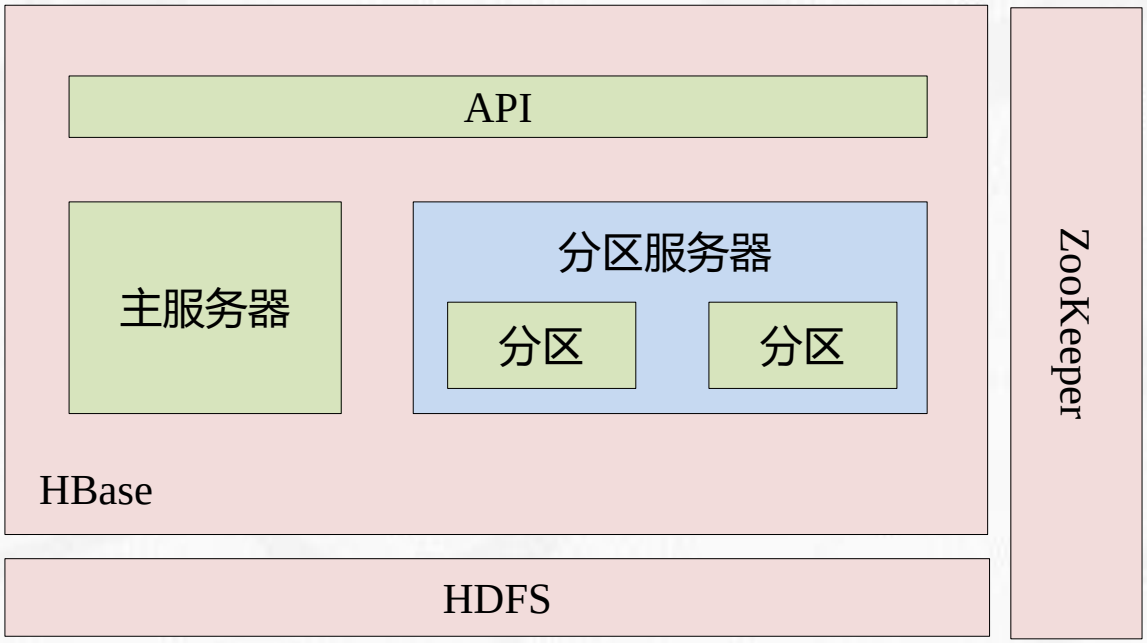
§4.1 大数据存储技术 - HBase

- HBase
  - 是一个高可靠、高性能、面向列、可伸缩的分布式数据库；
  - 主要用来存储非结构化和半结构化数据；
  - 是Hadoop的组成部分，Google BigTable的开源实现。

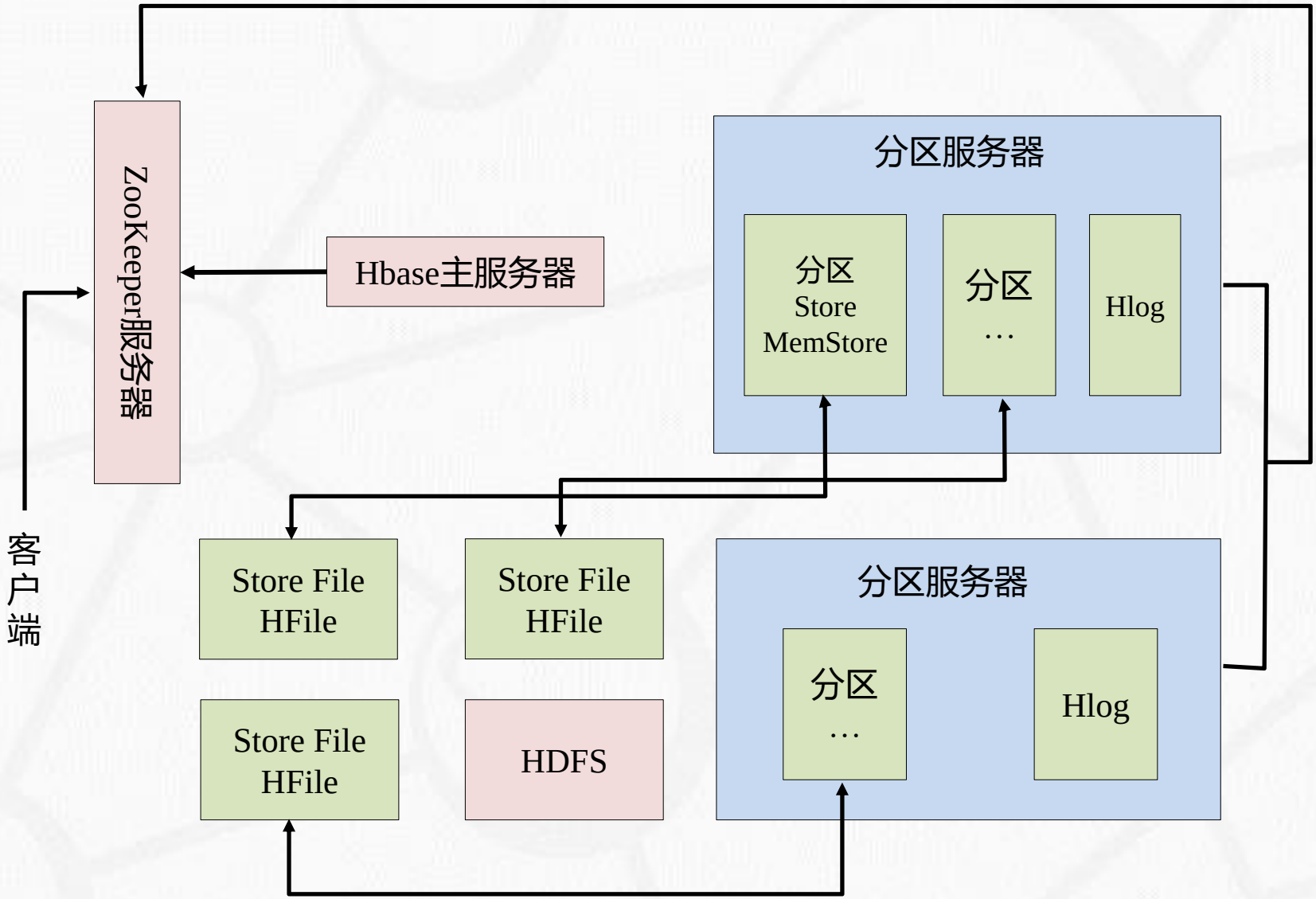


§4.1 大数据存储技术 - HBase的实现模块

- HBase的实现包括三个主要功能模块：
  - 链接到每个客户端的库函数；
  - 一个主服务器：管理和维护HBase的分区信息；
  - 多个分区服务器：存储和维护分配给自己的分区，处理来自客户端的读写请求。

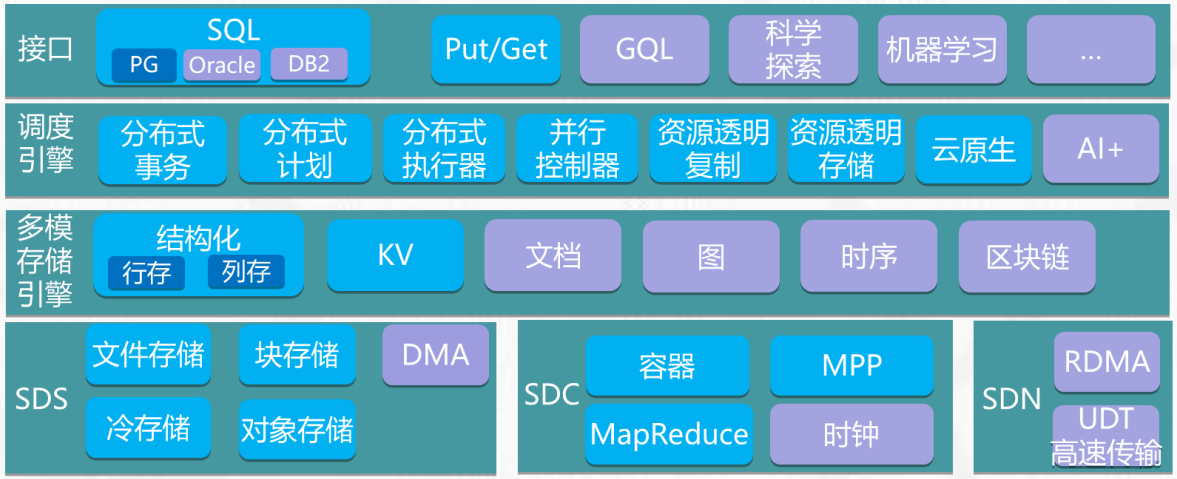


§4.1 大数据存储技术 - HBase的工作原理



§4.1 大数据存储技术 - 云原生数据库(1)

- 云原生数据库(Cloud native database)
  - 云原生数据库是一种运行在云计算平台上，提供按需访问服务的数据库。
- 云原生概念的发展
  - 2017年，Pivotal的高级产品经理Matt Stine将云原生归纳为模块化、可观察、可部署、可测试、可替换、可处理等六大特质。
  - 云原生的代表技术包括容器、服务网格、微服务、不可变基础设施和声明式API。
- 云原生数据库的特征
  - 普遍可访问、高可用性
  - 高扩展性、可迁移性
  - 演进式设计、快速迭代



云原生数据库

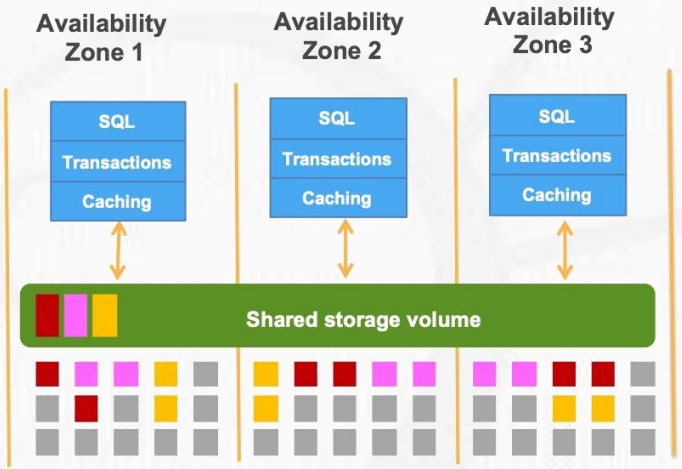
§4.1 大数据存储技术 - 云原生数据库(2)

- 云原生数据库的优势
  - 部署快捷：用户可在几分钟内完成云原生数据库的部署；
  - 可靠性高：具有故障自动单点切换、数据库自动备份、容灾备份等功能；
  - 成本低廉：支付的费用远远低于自建数据库所需的成本。

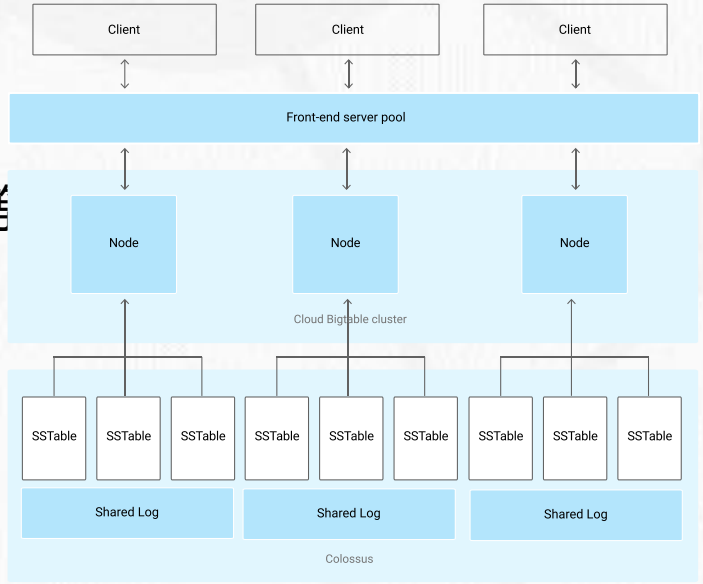
- 云数据库与云存储的区别
  - 对应的层面不同：云存储位于资源层(IaaS)，提供的是存储资源能力；云数据库位于平台层(PaaS)，提供的是中间件服务能力。
  - 提供的服务不同：云存储提供的主要是非结构化数据的存储能力；云原生数据库提供的是基础数据库和数据对象管理能力。
  - 两者的关系：对于前者，可以认为是将文件系统封装并提供文件系统API；对于后者，是在文件系统基础上封装为数据库，不提供文件系统API直接调用。

§4.1 大数据存储技术 - 典型的云原生数据库

- Amazon
  - S3、RDS、Aurora
- Google
  - Cloud BigTable、Cloud Storage
- 阿里巴巴
  - PolarDB-X、AnalyticDB、数据湖分析(DLA) 等
- 其它
  - 浪潮云溪NewSQL数据库、华为高斯数据库等



Amazon S3存储架构



Cloud BigTable架构

§4.1 大数据存储技术 - 将数据部署到云端

- 大数据特性：
  - Volume、Velocity、Variety、Veracity、Value
- 带来的困境
  - 用户端难以存储和处理
- 解决途径
  - 将数据加密后存储在云端

flickr

foursquare

阿里云邮 qiye.aliyun.com

NETFLIX

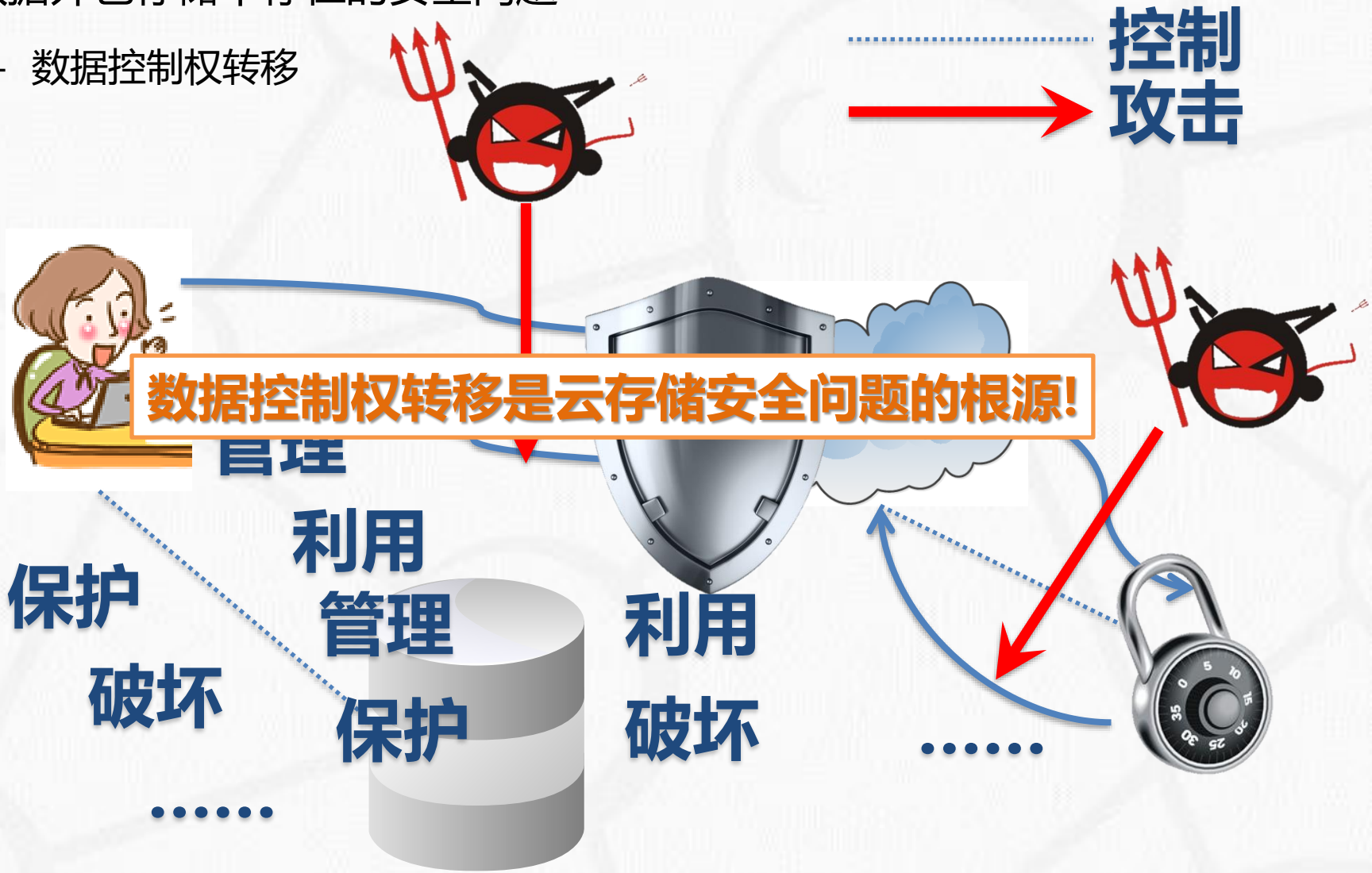
EasyChair The conference system

Expedia

https://aws.amazon.com/cn/solutions/case-studies/

§4.1 大数据存储技术 - 数据控制权的转移

- 数据外包存储中存在的安全问题
  - 数据控制权转移



§4.1 大数据存储技术 - 云端大数据存储的安全问题

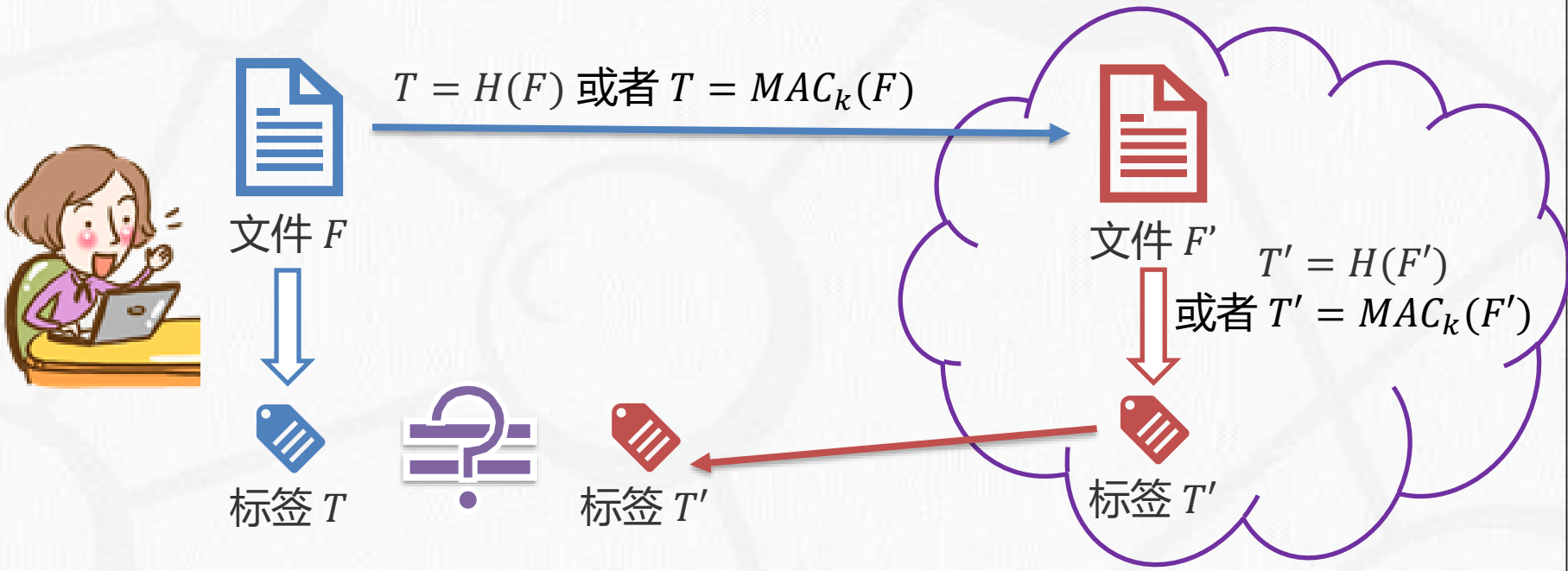
- 云端大数据存储的典型安全问题
  - 数据拥有证明(Provable data possession)：证明数据存在且完整；
  - 大数据访问控制(Access control)：不同用户角色访问不同数据；
  - 数据去重(Deduplication)：去除重复数据；
  - 大数据安全存储系统。
- 其它安全问题
  - 确定性数据删除(Data assured deletion)：确保数据可删除；
  - 数据定位(Data location)：确定数据正处于的位置；
  - 数据脱敏技术(Data Masking)：敏感数据消除；
  - ……

证明数据存在且完整

§4.2 大数据完整性校验

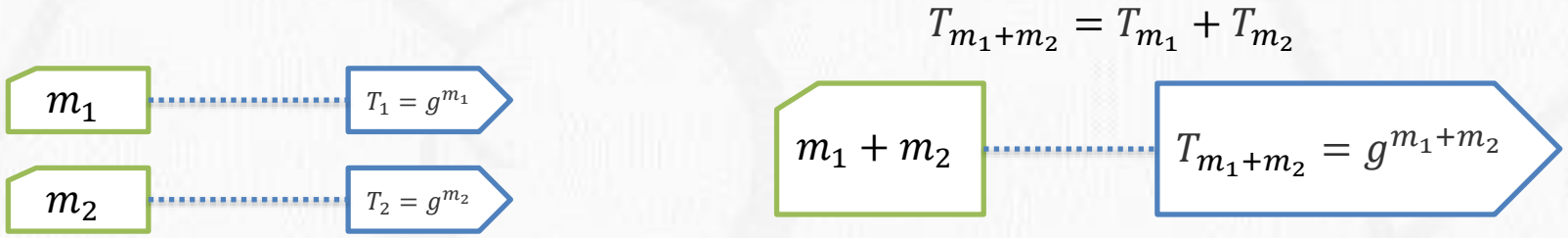
§4.2 大数据完整性校验 - 数据持有证明

- 数据持有证明(PDP: Provable data possession)
  - 是一种加密技术，允许用户将他们的数据存储在不受信任的服务器(公有云)上，并且有概率的保证服务器持有原始数据；
  - 客户端需要存储的仅仅是私钥，且不必取回文件；
  - 服务器不需要访问整个存储的文件；
  - 也称为数据可恢复性证明(POR: Proof of data retrivability)。



§4.2 大数据完整性校验 - HVT

- 同态可验证标签(HVT: Homomorphic Verifiable Tags)
  - HVT是一个对值$(T_{i,m}, W_i)$，与消息m一起存储在服务器中；
    - 给定一个消息m，T_m就是HVT；
    - W_i是一个随机值，i是索引值。
  - 特性：
    - 无锁验证(Blockless verification)；
    - 同态计算：
      - 一个HVT值$T_{m_i+m_j}$对应的就是消息$m_i + m_j$的HVT值。



基于离散对数构造同态可验证标签

§4.2 大数据完整性校验 - Giuseppe's PDP (概览)

配置

$KeyGen(1^{\kappa}) \rightarrow (\boldsymbol{pk}, \boldsymbol{sk})$

$TagBlock(pk, sk, m) \rightarrow T_m$

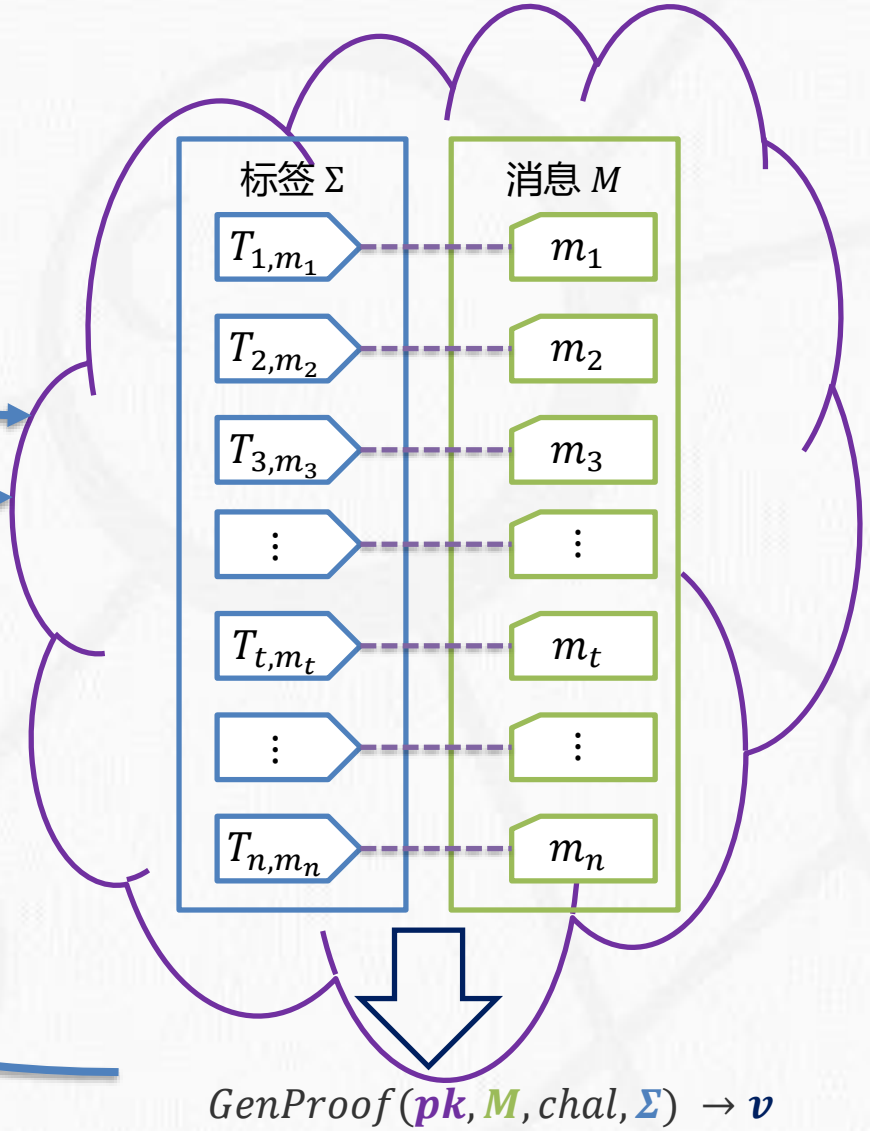
1. $< T_{i,m_i}, m_i >, i \in \{1, n\}$

2. 挑战$chal$

挑战

3. 证据$\boldsymbol{v}$

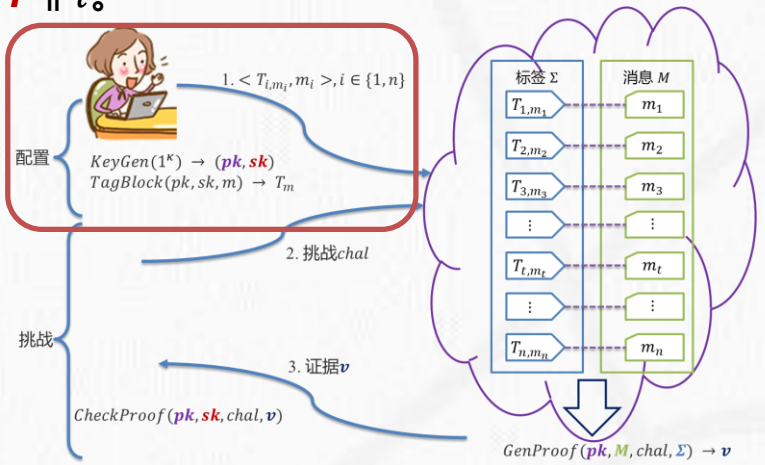
$CheckProof(\boldsymbol{pk}, \boldsymbol{sk}, chal, \boldsymbol{v})$



[1] Giuseppe A, Randal B, Reza C, et al. Provable Data Possession at Untrusted Stores[C]. ACM CCS. 2007: 598-610.

§4.2 大数据完整性校验 - Giuseppe's PDP (配置)

- 配置阶段：
  - $KeyGen(1^{\kappa}) \rightarrow (\boldsymbol{pk}, \boldsymbol{sk})$
    - 选择三个素数：$p = 2p' + 1$， $q = 2q' + 1$和e；
    - $\boldsymbol{pk} = (\boldsymbol{N}, \boldsymbol{g})$，$N = pq$是RSA参数， g是QR_N的生成元；
    - $\boldsymbol{sk} = (\boldsymbol{e}, \boldsymbol{d}, \boldsymbol{r})$，其中$ed \equiv 1 \bmod p'q'$，并且$r \xleftarrow{R} \{0,1\}^{\kappa}$；
  - $\forall i \in \{1, n\}$， $(\boldsymbol{T_{i,m_i}}, W_i) \leftarrow TagBlock(pk, (d, r), m_i, i)$：
    - $\boldsymbol{T_{i,m_i}} = (h(W_i) \cdot g^{m_i})^d \bmod N$，其中$W_i = \boldsymbol{r} \| i$。
  - 外包数据：
    - $\boldsymbol{pk}, M, \Sigma = \{T_{i,m_i}\}_{i=1}^{n}$

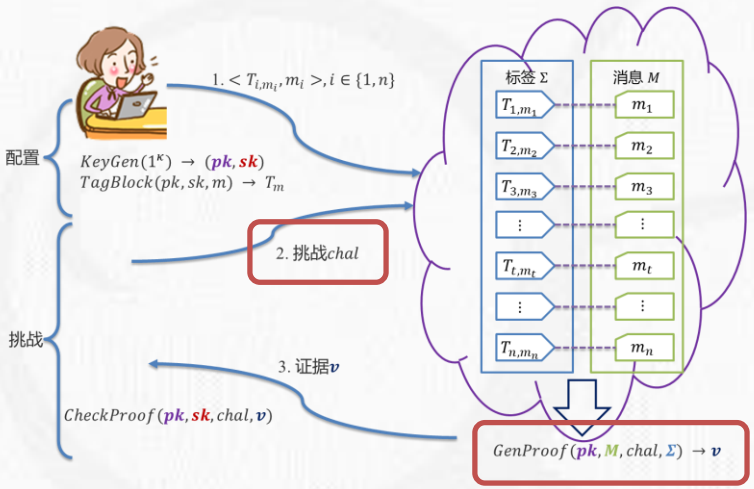


§4.2 大数据完整性校验 - Giuseppe's PDP (证据)

- 挑战阶段：验证块号为1~c的消息。
  - 生成挑战(C→S)：$chal = (c, k_1, k_2, g_s)$
    - $k_1 \xleftarrow{R} \{0,1\}^\kappa$，$k_2 \xleftarrow{R} \{0,1\}^\kappa$；
    - $g_s = g^s \bmod N$，$s \leftarrow \mathbb{Z}_N^*$；
    - c：待挑战块的序号。

用于混淆的密钥

秘密值



  - 计算证据(S→C)：v
    - 对于$1 \le i \le c$，$i_j = \pi_{k_1}(j)$，$a_j = f_{k_2}(j)$
      - $T = T_{i_1,m_{i_1}}^{a_1} \cdot T_{i_2,m_{i_2}}^{a_2} \cdots T_{i_c,m_{i_c}}^{a_c}$

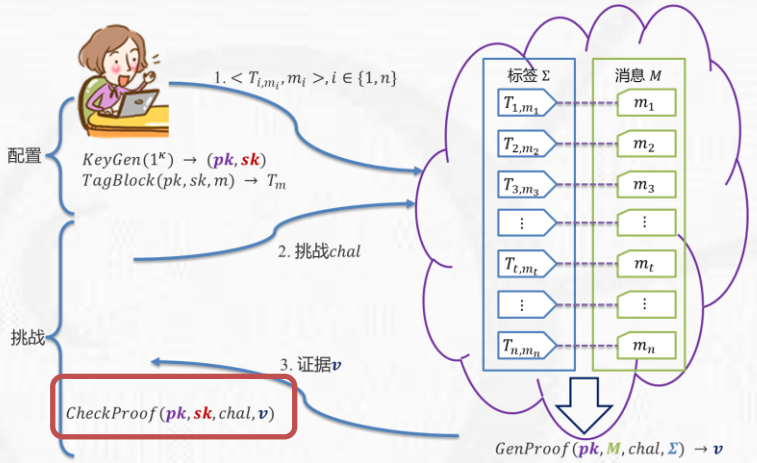
$$= \left(h(W_{i_1})^{a_1} \cdots h(W_{i_c})^{a_c} \cdot g^{a_1 m_1} \cdots g^{a_c m_c}\right)^{d} \bmod N$$

      - $\rho = H(g_s^{a_1 m_{i_1} + \cdots + a_c m_{i_c}} \bmod N)$
    - 证据：$v = (T, \rho)$

$T_{i,m_i} = (h(W_i) \cdot g^{m_i})^d \bmod N$，其中$W_i = r \| i$。

§4.2 大数据完整性校验 - Giuseppe's PDP (验证)

- 验证阶段：
  - 先验知识：
    - $\boldsymbol{sk} = (\boldsymbol{e}, \boldsymbol{d}, \boldsymbol{r})$；
    - $chal = (c, k_1, k_2, g_s)$；
    - $\tau = T^{\boldsymbol{e}}$；
    - $v = (T, \rho)$。



$$T = \left(h(W_{i_1})^{a_1} \cdots h(W_{i_c})^{a_c} \cdot g^{a_1 m_1} \cdots g^{a_c m_c}\right)^{\boldsymbol{d}} \bmod N$$

  - 验证过程：
    - 对于$1 \le i \le c$，$i_j = \pi_{\boldsymbol{k_1}}(j)$，$a_j = f_{\boldsymbol{k_2}}(j)$，$W_{i_j} = \boldsymbol{r} \| i_j$；
    - $\bar{\tau} = \dfrac{\tau}{h(W_{i_j})^{a_j}} \bmod N = g^{a_1 m_{i_1} + \cdots + a_c m_{i_c}} \bmod N$；

$$\rho = H(g_s^{a_1 m_{i_1} + \cdots + a_c m_{i_c}} \bmod N)$$

    - 如果$H(\bar{\tau}^{\boldsymbol{s}} \bmod N) = \rho$，验证成功；否则验证失败。

§4.2 大数据完整性校验 - Giuseppe's PDP (正确性)

- 正确性分析
  - 秘密值s的引入：证据v的随机性。

$$g_s = g^s \ mod \ N$$

$$\rho = H(g_s^{a_1 m_{i_1} + \cdots + a_c m_{i_c}} \ mod \ N)$$

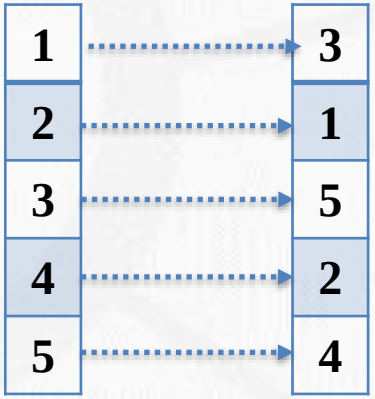
  - 用于混淆的密钥k_1和k_2：增加下标的随机性。
  - 混淆的可重复性。

计算证据

$$1 \le i \le c, \ i_j = \pi_{k_1}(j), \ a_j = f_{k_2}(j)$$

验证证据

$$1 \le i \le c, \ i_j = \pi_{k_1}(j), \ a_j = f_{k_2}(j)$$



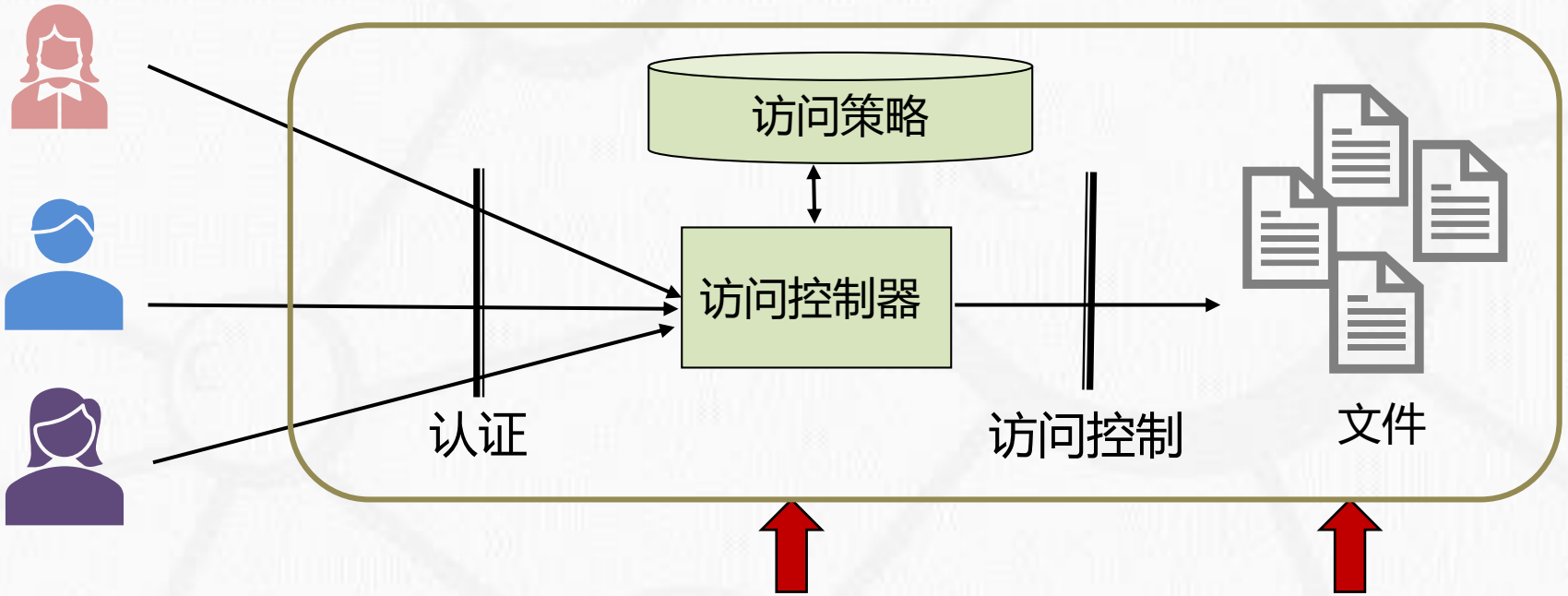
  - 正确性：$H(\tau^s \ mod \ N) = \rho$

- 安全性分析：
  - 在不知道s、 d 、 k_1和k_2的情况无法生成外包数据：$T_{i,m_i} = (h(W_i) \cdot g^{m_i})^d \ mod \ N$
  - 无法破解s的情况，云端无法伪造证据。

不同的用户对不同的数据进行访问

§4.3 属性基访问控制

§4.3 属性基访问控制 - 传统访问控制模型

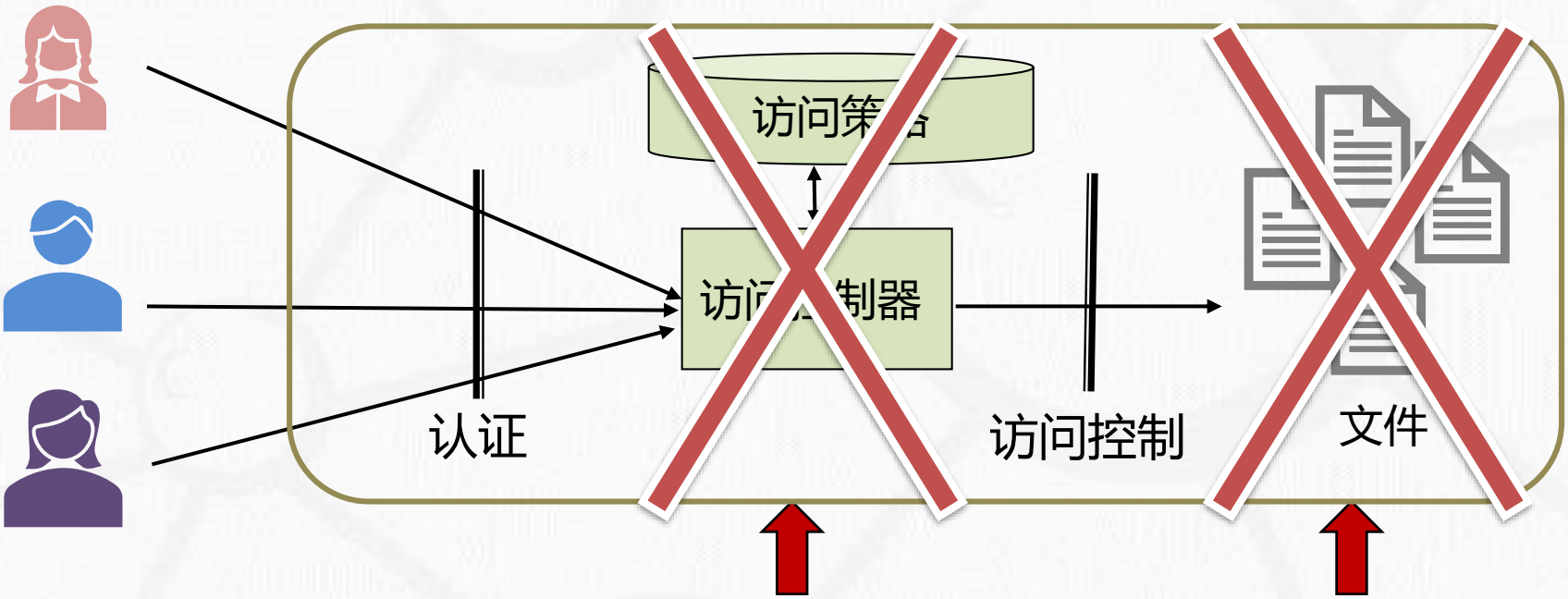


- 优势：灵活、可扩展；
- 缺陷：数据存储脆弱。

§4.3 属性基访问控制 - 数据损失案例

- 纽约门罗银行数据泄露(2008年2月)
  - 造成一百余万的人员机密信息泄露，被盗金额数据不详；
- 哈特兰支付系统数据泄露(2009年)
  - 黑客成功攻入后台数据库，获得了超过一千万信用卡交易记录。公司用于进行赔付的金额达到4000余万美元；
- 中国交通银行数据泄露(2021年1月)
  - 有黑客在国外某论坛上发帖，以8.8BTC的总价(折合人民币200余万元)售卖中国交通银行1679万笔数据。

§4.3 属性基访问控制 - 数据损失原因



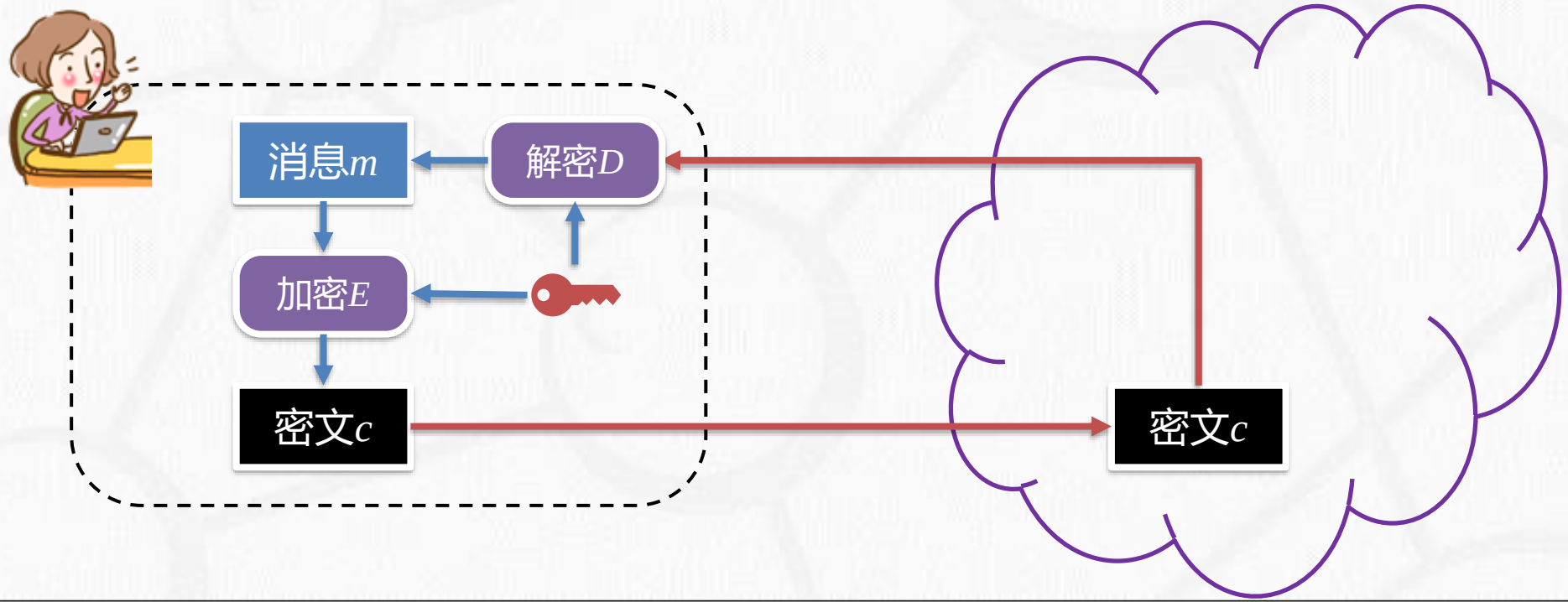
信任访问控制器

文件的保密存放

- **访问控制器本身就不可信!**

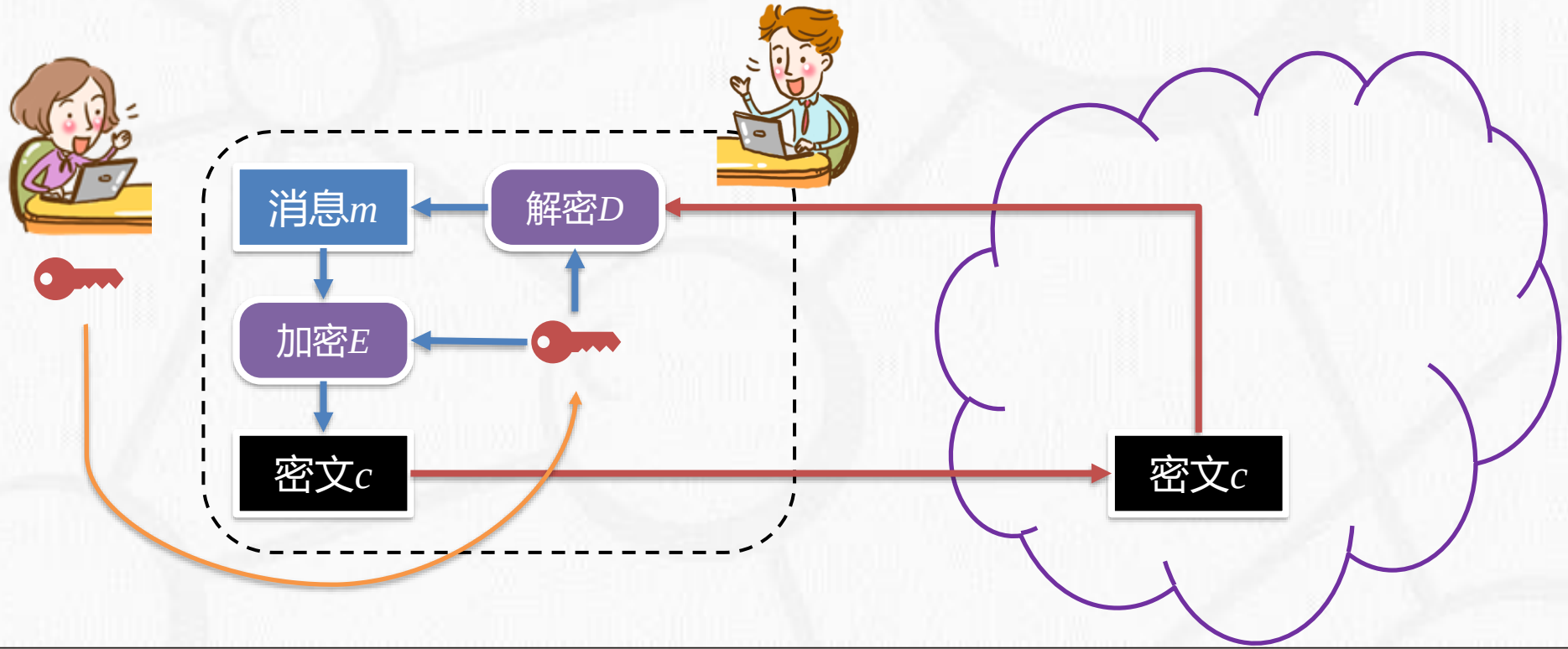
§4.3 属性基访问控制 - 基于加密的访问控制

- 基于加密的访问控制：
  - 核心思想：将数据用用户私钥进行加密，持有私钥的用户可访问数据。
  - 特点：
    - 加密数据存储在云端；
    - 每一个用户可以解密并访问自己的数据。



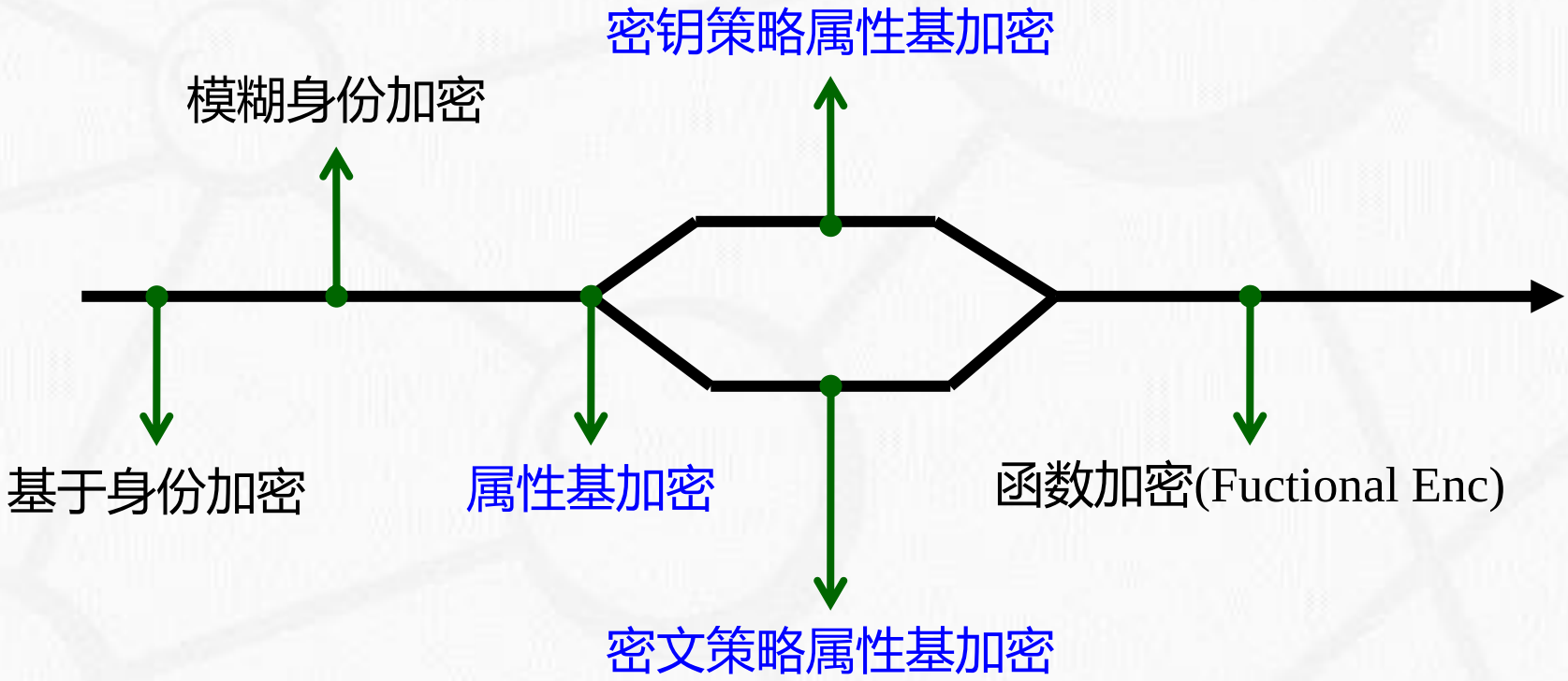
§4.3 属性基访问控制 - 与其他用户分享数据

- 对称密码解决方案
  - 在线密钥分发
- 公钥密码解决方案
  - 公钥证书管理开销大



§4.3 属性基访问控制 - 属性基加密的由来

- 属性基加密的初衷
  - 密钥管理可延展性强，不要求用户在线；
  - 不需要可信第三方的协助就能完成访问控制；
  - 支持可表达、可扩展的访问控制策略。



§4.3 属性基访问控制 - 属性基加密

- 属性基加密的目标：
  - 采用特定的属性集进行数据加密
  - “1对多”的公钥加密系统
  - 内嵌的访问控制机制

所有的“教授，CS PhD”可以访问

Alice 教授 ✓

Bob CS PhD ✓

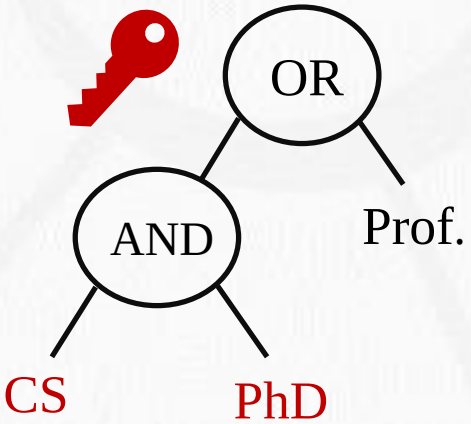
Charlie EE PhD ✗

  - 这是一个典型的密钥策略属性基加密

§4.3 属性基访问控制 - 密钥策略属性基加密

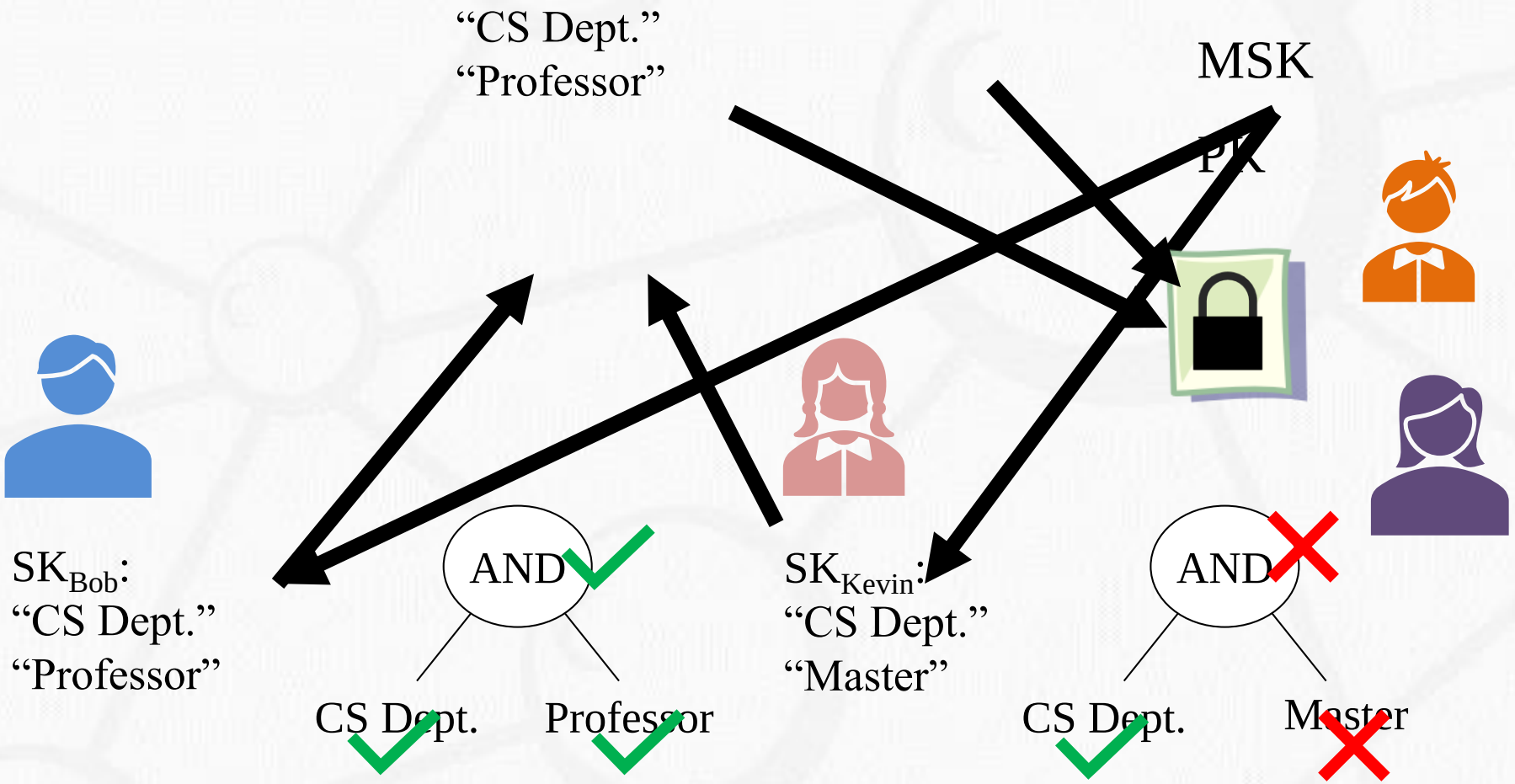
- 密钥策略属性基加密(KP-ABE: Key-Policy Attribute-Based Encryption)
  - 密文拥有一组属性;
  - 密钥对应一组访问控制结构;
  - 当且仅当满足密钥策略才能进行解密。

“All professors, CS PhD”



[1] Sahai A, Waters B. Fuzzy identity-based encryption[C]. EUROCRYPT. 2005: 457–473.

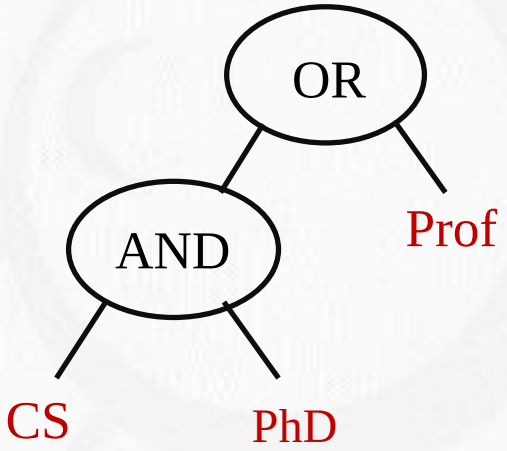
§4.3 属性基访问控制 - KP-ABE例子



§4.3 属性基访问控制 - 密文策略属性基加密(1)

- 密文策略属性基加密(CP-ABE: Ciphertext-Policy Attribute-Based Encryption)
  - 密文与一组访问策略关联

(CS AND PhD) OR Prof



  - 密钥与一些属性关联
    - 属性通过数学的方式内嵌到密钥之中。

SK

Alice

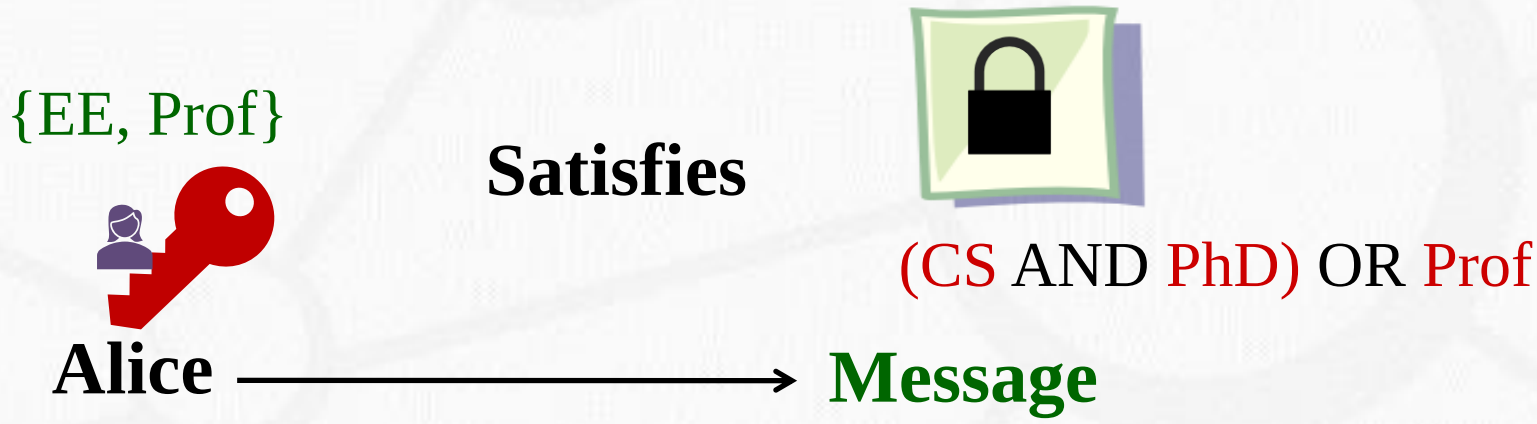
{EE, Prof}

Bob

{CS, PhD}

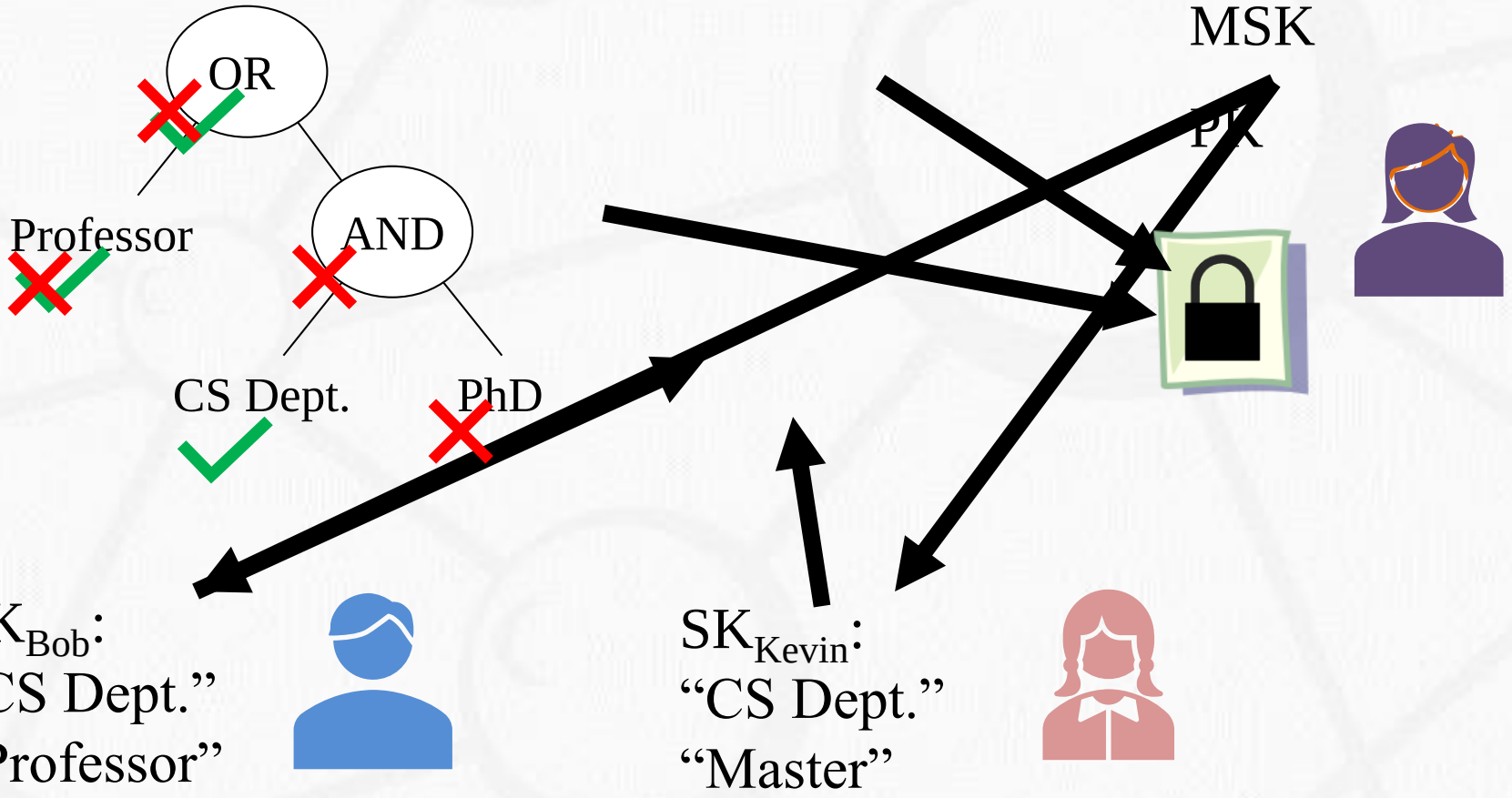
§4.3 属性基访问控制 - 密文策略属性基加密(2)

- 当且仅当密钥中的属性满足密文中的访问策略才能进行解密



- 在解密过程中没有可信第三方进行策略检测并作出决定；
- 策略检测在揭秘过程中完成。

§4.3 属性基访问控制 - CP-ABE例子



§4.3 属性基访问控制 - ABE的优势

- 属性基加密的优势：
  - 用户自定义访问策略；
  - 访问策略在揭秘过程中强制执行
    - 不需要其他方(特别是可信第三方)参与；
  - 针对每一个明文，仅生成一个密文。

- CP-ABE相比KP-ABE更加适合于云计算场景：
  - 用户可以根据用户属性为每一个明文指定访问策略；
  - 用户仅仅保存一个密钥；
  - 用户可以在不更换公私钥的情况下更新密文策略。

§4.3 属性基访问控制 - 椭圆曲线

- 离散对数问题(DHP: Discrete Log Problem)：
  - 给定一个阶为q乘法循环群G，其单位元为1。假设B是G的一个元，并且g是G的一个生成元。求解x，令$g^x = B$。

$$m^e \bmod p \xrightarrow{\text{Easy!}} ? = c$$

$$?^e \bmod p \xleftarrow{\text{Hard!}} c$$

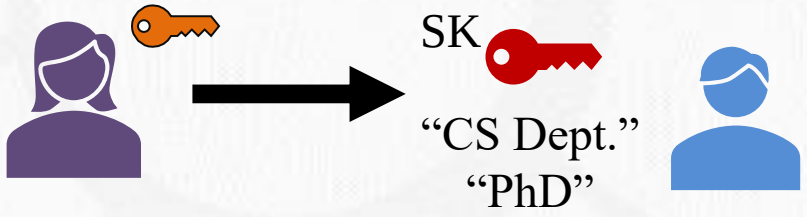
单向函数(离散对数求解问题)

- 双线性对：
  - G_1和G_2是两个循环群，其中G_1是加法群，G_2是乘法群。它们的阶为大素数q。构造出如下双线性对：$\hat{e}: G_1 \times G_1 \to G_2$。其中，$G_1$和$G_2$中的离散对数问题是难以求解的，并且该双线性对满足以下性质：
    - 双线性：$\forall P, Q \in G_1, \forall a, b \in Z, \exists\ \hat{e}(aP, bQ) = \hat{e}(bP, aQ) = \hat{e}(P, Q)^{ab}$;
    - 非退化性：$\exists P, Q \in G_1, st\ \hat{e}(P, Q) \neq 1 \in G_2$;
    - 可计算性：$\forall P, Q \in G_1$，能够在多项式时间内计算出$\hat{e}(P, Q)$的值。

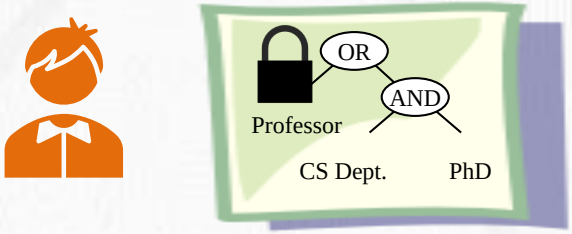
§4.3 属性基访问控制 - CP-ABE算法概览

Setup(λ) -> MSK, PK

KeyGen(MSK, Attrs.) -> SK



Encrypt(PK ,M, Access policy) -> CT



Decrypt(SK, CT) -> M

§4.3 属性基访问控制 - CP-ABE的Setup

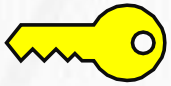
Authority

$a, b \in_R Z_P$

MSK

MSK = a

Public Key



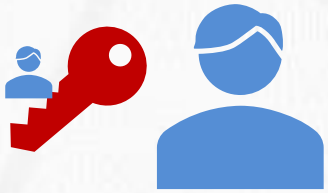
PK = (g, g^b, $e(g, g)^a$, H: $\{0,1\}^* \rightarrow G$)

§4.3 属性基访问控制 - CP-ABE的KeyGen(1)

Authority

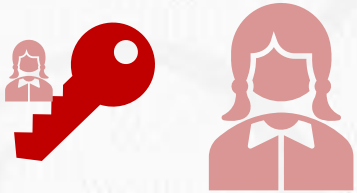
Authority issues secret keys for users who have attributes

Bob



"CS Dept."
"Professor"

Kevin



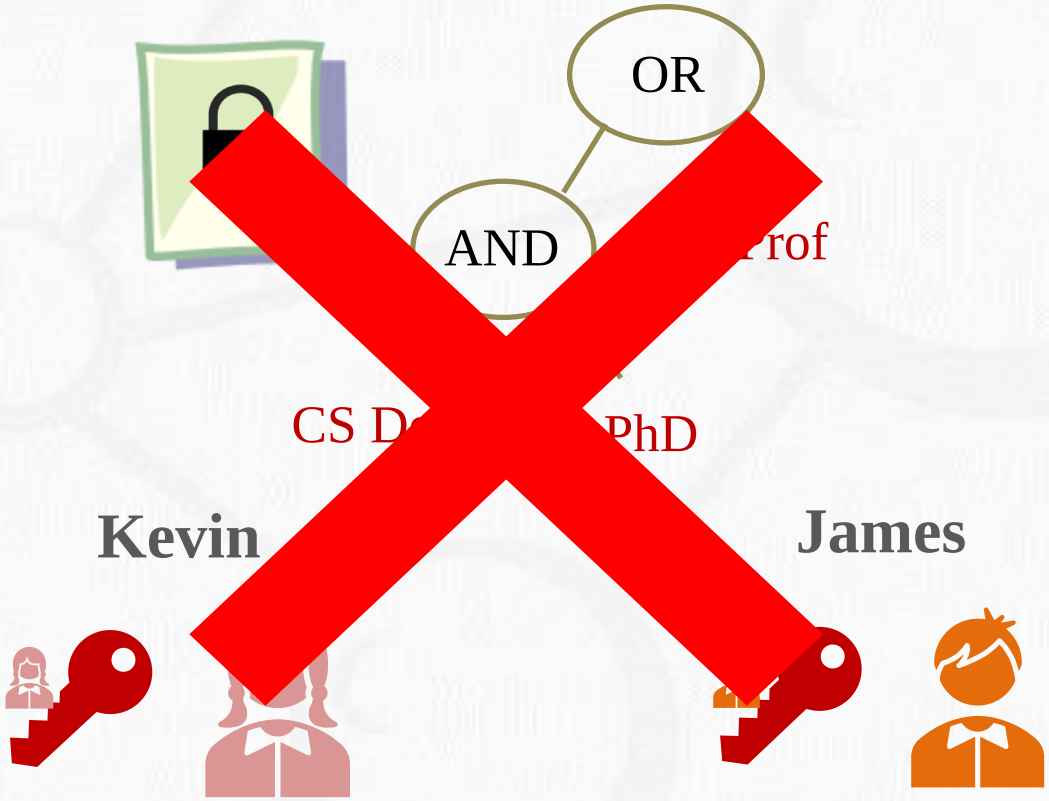
"CS Dept."
"Master"

James

"EE Dept."
"PhD"

§4.3 属性基访问控制 - KeyGen的核心关注

- 用户必须防止用户的共谋
  - 多个用户将自身的属性进行组合，已获得合法授权。



§4.3 属性基访问控制 - CP-ABE的KeyGen(2)

Authority

MSK = a

Bob has attributes:

{“PhD”, “CS Dept.”, “TA”}

$$SK = (g^{a+bt}, g^{t}, H(\text{“PhD”})^{t}, H(\text{“CS Dept.”})^{t}, H(\text{“TA”})^{t})$$

‘t’: random number in Z_p

‘t’ ties components together

Personalization!

Collusion Resistance

§4.3 属性基访问控制 - CP-ABE的密钥个性化

Kevin:
"CS Dept."
…

SK 

g^{a+bt}, g^{t}, H("CS Dept.")^{t},

James:
"PhD"
…

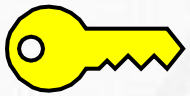
$g^{a+bt'}$, $g^{t'}$, H("PhD")$^{t'}$

Components are incompatible

(Formal security proofs in papers)

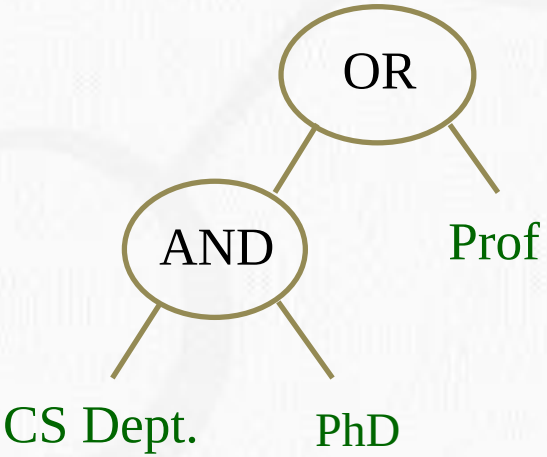
§4.3 属性基访问控制 - CP-ABE的加密(1)

Data Owner



$PK = (g,\ g^b,\ e(g, g)^a , H: \{0,1\}^* \rightarrow G)$

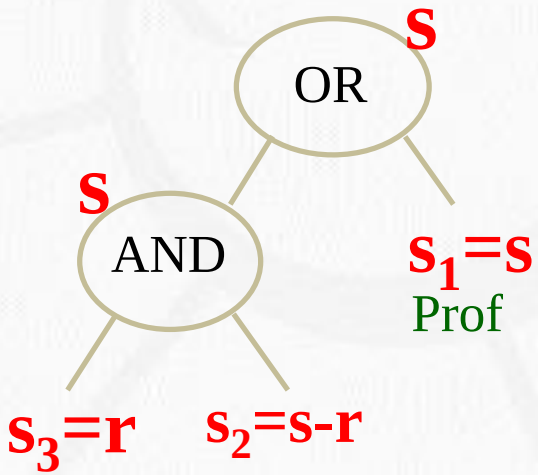
Given a file M and an access policy, data owner will perform the following



§4.3 属性基访问控制 - CP-ABE的加密(2)

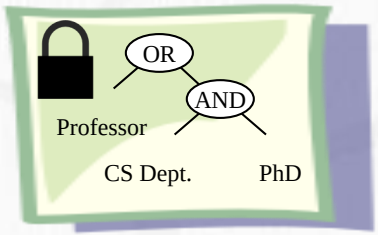
Data Owner

Data Owner generates random s, then computes



Ciphertext:

$$CT = (M \bullet e(g,g)^{as}, g^{s},$$



$$C_1 = (g^{bs_1} H(\text{“Prof”})^{r1}, g^{r1}),\ C_2 = (g^{bs_2} H(\text{“PhD”})^{r2}, g^{r2}),$$

$$C_3 = (g^{bs_3} H(\text{“CS Dept.”})^{r3}, g^{r3}))$$

§4.3 属性基访问控制 - CP-ABE的解密

Ciphertext CT

$CT = (M \cdot e(g,g)^{as}, g^s, C_1 = (g^{bs_1} H(\text{“Prof”})^{r1}, g^{r1}),$

$C_2 = (g^{bs2} H(\text{“PhD”})^{r2}, g^{r2}), C_3 = (g^{bs3} H(\text{“CS Dept.”})^{r3}, g^{r3}))$

Secret Key SK

$SK = (g^{a+bt}, g^t, H(\text{“Prof”})^t, H(\text{“PhD”})^t, H(\text{“CS Dept.”})^t)$

$$e(g^{a+bt}, g^s) = e(g,g)^{as} e(g,g)^{bts}$$

“Prof”	OR	“PhD” AND “CS Dept.”

$$e(g,g)^{bts} = \frac{e(g^{bs_1} H(\text{“Prof”})^{r1}, g^t)}{e(g^{r1}, H(\text{“Prof”})^t)}$$

$$\frac{e(g^{bs2} H(\text{“PhD”})^{r2}, g^t)}{e(g^{r2}, H(\text{“PhD”})^t)} \bullet \frac{e(g^{bs3} H(\text{“CS Dept.”})^{r3}, g^t)}{e(g^{r3}, H(\text{“CS Dept.”})^t)}$$

$$= e(g,g)^{bts2} e(g,g)^{bts3}$$

$$= e(g,g)^{bts}$$

§4.1 & §4.2 & §4.3 小结

- 内容回顾
  - 大数据存储技术
  - 数据持有证明
  - 属性基加密

- 掌握
  - 数据持有证明的工作原理
  - KP-ABE和CP-ABE的区别和联系
  - CP-ABE的工作原理

Thanks!
Questions & Advices!