



第十一章、密钥管理及其在区块链中的应用

何德彪

武汉大学
国家网络安全学院



目 录

- 11.1. 密钥管理的概述
- 11.2. 密钥分配与协商
- 11.3. 公钥密码的密钥分配
- 11.4. 移动互联网下的私钥安全分析
- 11.5. 针对ECDSA的协同签署方案
- 11.6. 针对SM2签名的协同签署方案
- 11.7. 针对SM9签名的协同签署方案
- 11.8. 密钥管理在区块链中的应用

目 录

- 11.1. 密钥管理的概述
- 11.2. 密钥分配与协商
- 11.3. 公钥密码的密钥分配
- 11.4. 移动互联网下的私钥安全分析
- 11.5. 针对ECDSA的协同签署方案
- 11.6. 针对SM2签名的协同签署方案
- 11.7. 针对SM9签名的协同签署方案
- 11.8. 密钥管理在区块链中的应用

11.1 密钥管理概述

1. 密钥管理的重要性

➤ 所有的密码技术都依赖于密钥

- ✓ 现代密码体制要求加密算法是可以公开评估的，整个密码系统的安全性并不取决于对密码算法的保密或者是对加密设备等的保护。
- ✓ 决定整个密码体制安全性的因素将是密钥的保密性（“一切秘密予于密钥之中！”）。
- ✓ 密码算法可以公开，密码设备可以丢失，但它们都不危及密码体制的安全性；但一旦密钥丢失，非法用户将会有可能窃取信息。

➤ 在考虑密码系统的应用设计时，特别是在商用系统的设计时，需要解决的核心问题是密钥管理问题，而不是密码算法问题。

- ✓ 例如商用系统可以使用公开了的、经过大量评估分析认为抗攻击能力比较强的算法。

➤ 密钥的管理本身是一个很复杂的课题，而且是保证安全性的关键点。

11.1 密钥管理概述

2. 密钥管理的概念

密钥管理是一门综合性的技术，涉及密钥的产生、检验、分发、传递、保管、使用、销毁的全部过程，还与密钥的行政管理制度以及人员的素质密切相关。

3. 密钥管理的目的

维持系统中各实体之间的密钥关系，以抗击各种可能的威胁：

- 密钥的泄露
- 密钥或公钥的身份的真实性丧失
- 未经授权使用

11.1 密钥管理概述

4. 密钥管理系统的要求

- 密钥难以被非法窃取；
- 在一定条件下获取了以前的密钥用处也很小；
- 密钥的分配和更换过程对用户是透明的。

5. 密钥的组织结构

适应于对密钥管理系统的要求，现有的计算机网络系统与数据库系统的密钥管理系统的设计大都采用了层次化的密钥结构。

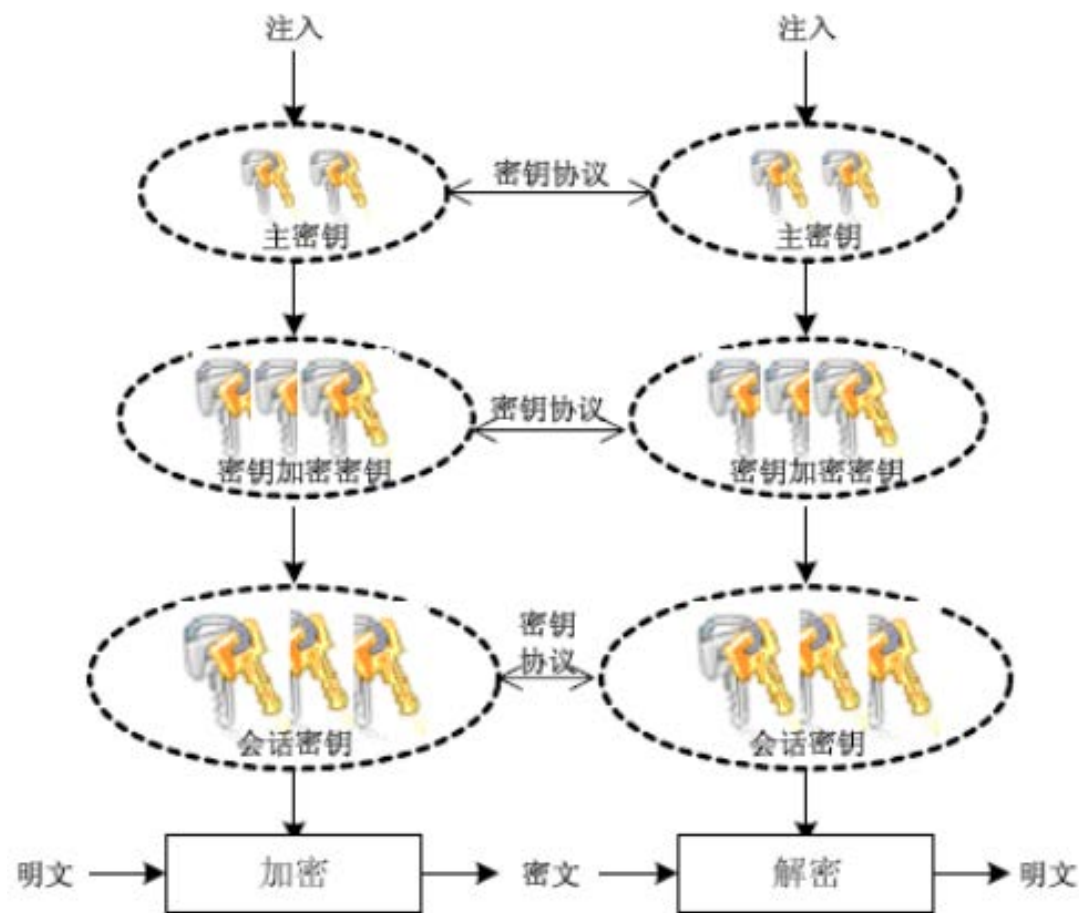


图11.1. 密钥组织结构示意图

11.1 密钥管理概述

6. 密钥管理的原则

(1) 区分密钥管理的策略和机制

- ✓ 策略是密钥管理系统的高级指导。策略着重原则指导，而不着重具体实现。
- ✓ 密钥管理机制是实现和执行策略的技术和方法。

(2) 全程安全原则

必须在密钥的产生、存储、备份、分发、组织、使用、更新、终止和销毁等的全过程中对密钥采取妥善的安全管理。

(3) 最小权利原则

应当只分发给用户进行某一事务处理所需的最小的密钥集合。

(4) 责任分离原则

一个密钥应当专职一种功能，不要让一个密钥兼任几种功能。

11.1 密钥管理概述

6. 密钥管理的原则

(5) 密钥分级原则

可减少受保护的密钥的数量，又可简化密钥的管理工作。一般可将密钥划分为三级：主密钥，二级密钥，初级密钥。

① 主密钥

主密钥对应于层次化密钥结构中的最高层次，它是对密钥加密密钥进行加密的密钥。

② 密钥加密密钥

密钥加密密钥一般是用来对传送的会话密钥或文件加密密钥进行加密时所采用的密钥，也称为二级密钥。

密钥加密密钥实际是用来保护通信或文件数据的会话密钥或文件加密密钥。在通信网中，一般在每个节点都分配有一个这类密钥，同时，为了安全，各节点的密钥加密密钥应互不相同。

11.1 密钥管理概述

6. 密钥管理的原则

③ 初级密钥

最底层的密钥，直接对数据进行加密和解密，分为初级文件密钥和会话密钥。

- ✓ 初级文件密钥：用于文件保密的初级密钥；
- ✓ 会话密钥：一般由系统自动产生，且对用户是不可见的。在一次通信或数据交换中，用户之间所使用的密钥。会话密钥可由通信用户之间进行协商得到。它一般是动态地、仅在需要进行会话数据加密时产生，并在使用完毕后立即清除掉。

11.1 密钥管理概述

6. 密钥管理的原则

④ 层次化的密钥结构的好处

✓ 安全性大大提高

下层的密钥被破译将不会影响到上层密钥的安全。在少量最初的处于最高层次的密钥注入系统之后，下面各层密钥的内容，可以按照某种协议不断地变化(例如可以通过使用安全算法以及高层密钥动态地产生低层密钥)，实现了“静止的密钥系统 → 动态的密钥系统”。

✓ 为密钥管理自动化带来了方便

开放式的网络应用环境不可能再进行人工密钥分配。层次化密钥结构中，除了一级密钥需要由人工装入以外，其他各层的密钥均可以由密钥管理系统按照某种协议进行自动地分配、更换、销毁等。

11.1 密钥管理概述

6. 密钥管理的原则

(6) 密钥更新原则

密钥必须按时更新。否则，即使是采用很强的密码算法，使用时间越长，敌手截获的密文越多，破译密码的可能性就越大。

(7) 密钥应当有足够的长度

密码安全的一个必要条件是密钥有足够的长度。密钥越长，密钥空间就越大，攻击就越困难，因而也就越安全。

(8) 密码体制不同，密钥管理也不相同

由于传统密码体制与公开密钥密码体制是性质不同的两种密码，因此它们在密钥管理方面而有很大的不同。

11.1 密钥管理概述

7. 密钥的生成

对于一个密码体制，如何产生好的密钥是非常关键，密钥选择的不当将会极大地影响密码体制的安全性。好的密钥应当具有良好的**随机性**和**密码特性**(例如避免弱密钥的出现等)。

密钥的生成一般都首先通过密钥产生器借助于某种噪声源产生具有较好统计分布特性的序列，然后再对这些序列进行各种随机性检验以确保其具有较好的密码特性。

不同层次的密钥产生的方式一般也不相同：

(1) 主密钥：虽然一般它的密钥量很小，但作为整个密码系统的核心，需要严格保证它的随机性，避免可预测性。因此，主密钥通常采用掷硬币、骰子或使用其它物理噪声发生器的方法来产生。

11.1 密钥管理概述

7. 密钥的生成

(2) 二级密钥: 可以采用伪随机数生成器、安全算法(例如, 可以在主机主密钥的控制下由 ANSI X9.17所给出的算法产生)或电子学噪声源产生。

(3) 会话密钥: 可以在密钥加密密钥的控制下通过安全算法动态地产生。

目 录

- 11. 1. 密钥管理的概述
- 11. 2. 密钥分配与协商
- 11. 3. 公钥密码的密钥分配
- 11. 4. 移动互联网下的私钥安全分析
- 11. 5. 针对ECDSA的协同签署方案
- 11. 6. 针对SM2签名的协同签署方案
- 11. 7. 针对SM9签名的协同签署方案
- 11. 8. 密钥管理在区块链中的应用

11.2 密钥分配与协商

1. 基本概念

密钥分配与密钥协商过程都是用以在保密通信双方之间建立通信所使用的密钥的协议(或机制)。在这种协议(或机制)运行结束时，参与协议运行的双方都将得到相同的密钥，同时，所得到的密钥对于其他任何方(除可能的可信管理机构外)都是不可知的。

(1) 密钥分配

密钥分配是这样一种机制，保密通信中的一方利用此机制生成并选择秘密密钥，然后将其安全传送给通信的另一方或通信的其他多方。典型协议：**Kerberos**密钥分配协议。

- **密钥预分配**：TA(可信管理机构 ,Trust Authority)预先分配密钥，用户之间的通信将使用预先分配好的密钥。
- **密钥在线分配**：在用户需要进行通信时TA才进行密钥分配，一般需要有一个在线TA。

11.2 密钥分配与协商

(2) 密钥协商

通常是一种协议，利用该协议，通信双方可以在一个公开的信道上通过相互传送一些消息来共同建立一个安全的共享秘密密钥。在密钥协商中，双方共同建立的秘密密钥通常是双方输入消息的一个函数。典型协议： Diffie-Hellman密钥交换协议

(3) 对密钥分配、密钥协商的安全威胁

- 被动攻击：窃听；
- 主动攻击：篡改、重放、冒充；

密钥分配协议和密钥协商协议的目的就是在协议结束时，通信双方具有一个相同的秘密密钥 k ，并且 k 不被其他人员知道。可以想象，在主动的对手存在的情况下，设计一个满足上述目的的协议是比较困难的。

11.2 密钥分配与协商

(4) 密钥分配基本方法

设A和B是通信双方，C是可信第三方，密钥分配主要有以下4种方法：

- ① 密钥由A选定，然后通过物理手段传给B。
- ② 密钥由第三方C选定，然后通过物理手段传给A和B。
- ③ 如果A和B事先已经有一个密钥，则一方可以使用原来的旧密钥去加密新密钥，并通过普通信道发送给另一方。
- ④ 如果A和B与第三方C分别有一个安全信道，则C为A和B选定密钥后，通过安全信道发送给A和B。

11.2 密钥分配与协商

	人工分法	需要第三方	特点
方法①	是	否	当 n 个用户时，密钥数为 C_n^2 ，因此当 n 很大时，人工分法密钥是不可行的。
方法②	是	是	
方法③	否	否	分配初始密钥代价大，数量为密钥数为 C_n^2 。攻击者一旦获得一个密钥，就可以获得以后所有的密钥。
方法④	否	是	虽然会话密钥数量为 C_n^2 ，但是需要人工分法的主密钥为 n 个。

11.2 密钥分配与协商

2.对称密码体制的密钥分配

(1) 无KDC的对称密钥分发(Key Distribution Center, KDC)

方法：用双方共享的共享主密钥加密会话密钥

① A向B发出建立会话密钥的请求和一个随机数/时戳 N_A 。

② B用与A共享的主密钥 K_m 对应答的消息加密，并发送给A。应答的加密消息中有B选取的会话密钥 k_s 、A、B的身份、 N_A 和另一个随机数/时戳 N_B 。

③ A使用新建立的会话密钥 k_s 对 N_B 加密后返回给B。

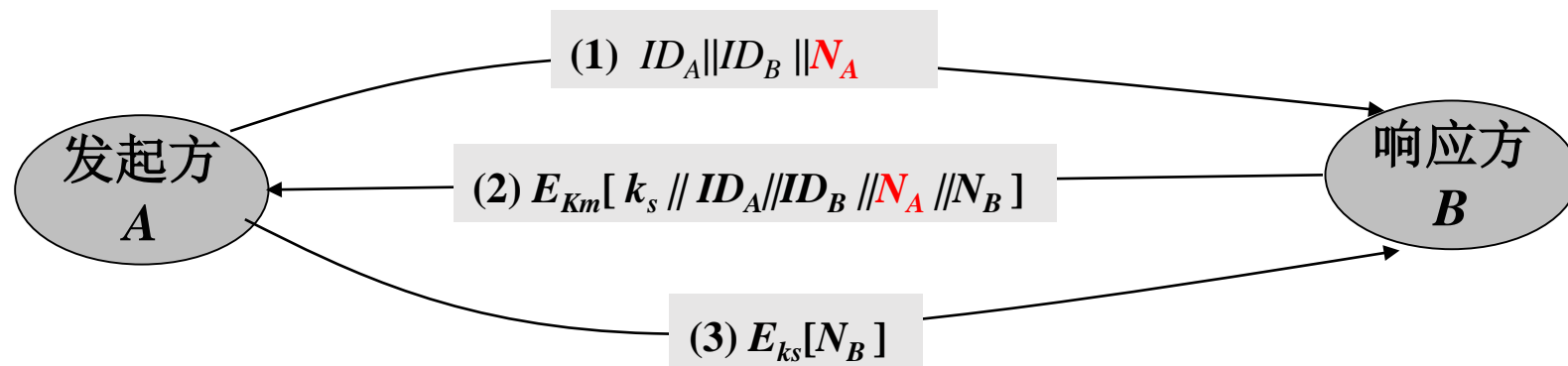
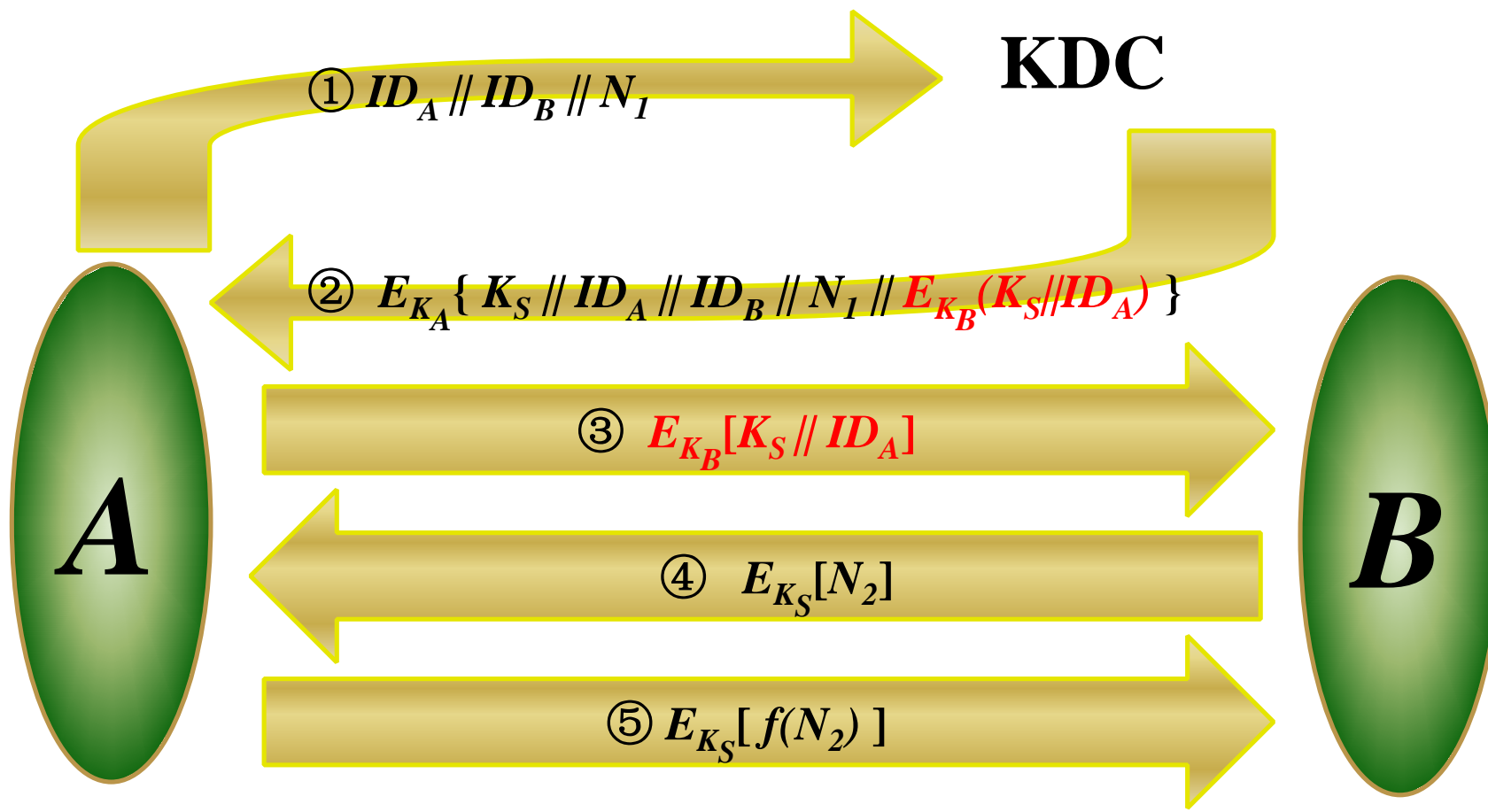


图11.2. 无KDC的对称密钥吩咐示意图

11.2 密钥分配与协商

(2) 有KDC的对称密钥分发

前提：两个用户A、B分别与密钥分配中心KDC有共享密钥 K_A ， K_B 。



11.2 密钥分配与协商

3. KDC的分层

➤ 如果网络中的用户数目非常多且地域分布非常广，可采用分层结构

- ✓ 在每个小范围(如一个LAN或一个建筑物)内建立一个本地KDC，负责该范围内的密钥分配；
- ✓ 如果两个不同范围的用户想获得共享密钥，需要通过各自的本地KDC，并由两个本地KDC经过一个全局KDC完成密钥分配。这样就建立了两层KDC结构。

➤ 层次化的优势

- ✓ 可减少主密钥的分布，因为大多数主密钥是在本地KDC和本地用户之间共享；
- ✓ 可将虚假KDC的危害限制到一个局部区域内。

目 录

- 11. 1. 密钥管理的概述
- 11. 2. 密钥分配与协商
- 11. 3. 公钥密码的密钥分配
- 11. 4. 移动互联网下的私钥安全分析
- 11. 5. 针对ECDSA的协同签署方案
- 11. 6. 针对SM2签名的协同签署方案
- 11. 7. 针对SM9签名的协同签署方案
- 11. 8. 密钥管理在区块链中的应用

11.3 公钥密码的密钥分配

1. 问题分析

和对称密码一样，公钥密码也需要进行密钥分配。但是，公钥密码的密钥分配与对称密码体制的密钥分配有着本质的差别。在密钥分配时必须做到：

- 公钥是公开的，因此不需确保秘密性。但必须确保公钥的真实性和完整性；
- 确保私钥的机密性、真实性和完整性。

如果公钥的真实性和完整性受到危害，则基于公钥的各种应用的安全将受到危害。

举例：C冒充A欺骗B的攻击方法

- ① 攻击者C在公钥数据库PKDB中用自己的公钥 PK_C 替换用户A的公钥 PK_A 。
- ② C用自己的私钥 SK_C 签名一个消息冒充A发给B。
- ③ B验证签名：因为此时PKDB中A的公钥已经替换为C的公开钥 PK_C ，故验证为真。

11.3 公钥密码的密钥分配

因验证签名为真，于是 B 认为攻击者 C 就是 A 。

- 若 B 要发送加密的消息给 A ，则 B 要用 A 的公钥进行加密，但 A 的公开钥已被换成 C 的公钥，因此 B 实际上是用 C 的公钥进行了加密。
- C 从网络上截获 B 发给 A 的密文。由于这密文实际上是用 C 的公钥加密的，所有 C 可以解密得到明文。 A 反而不能正确解密。

上述攻击成功的原因：

- ① 对存入 PKDB 的公钥没有采取保护措施，致使公开加密钥被替换而不能发现；
- ② 存入 PKDB 的公钥与用户的标识符之间没有绑定关系，致使 A 的公钥替换成 C 的公钥后不能发现公开钥与用户的标识符之间的对应关系被破坏。

怎么解决？

11.3 公钥密码的密钥分配

2. 公钥证书

日常生活中有许多使用证书的例子，例如汽车驾照。驾照由可信的公安机关签发，以标识驾驶员的驾驶资格。由于有公安机关的签章，**任何人都可以验证驾照的真实性**。又由于驾照上印有驾驶员的照片并盖了钢印，从而实现**驾驶员与驾照之间的严格绑定**。

数字世界也采用类似的技术解决上述问题。具体来说是采用数字签名技术可以克服上述两个缺点，确保公钥的安全分配。

- 经过可信实体签名的一组信息的集合被称为证书(Certificate)，而可信实体被称为签证机构 CA(Certification Authority)。
- 一般地讲，证书是一个数据结构，是一种由一个可信任的权威机构签署的信息集合。
- 在不同的应用中有不同的证书。例如公钥证书 PKC(Public Key Certificate)、良保密协议 (Pretty Good Privacy, PGP)证书、安全电子交易(Secure Electronic Transaction, SET)证书等。

11.3 公钥密码的密钥分配

公钥证书PKC是一种包含持证主体标识、持证主体公钥等信息，并由可信签证机构签署的信息集合。

- 公钥证书主要用于确保公钥的安全，**确保公钥与用户标识符之间绑定关系的安全**。这个公钥就是证书所标识的那个主体的合法的公钥。
- 公钥证书的持证**主体可以是人、设备、组织机构或其它主体**。
- 公钥证书一般以明文的形式进行存储和分配。
- 任何一个用户**只要知道签证机构的公钥，就能检查对证书签名的合法性**。如果检查正确，那么用户就可以相信那个证书所携带的公钥是真实的，而且这个公钥就是证书所标识的那个主体的合法的公钥。

11.3 公钥密码的密钥分配

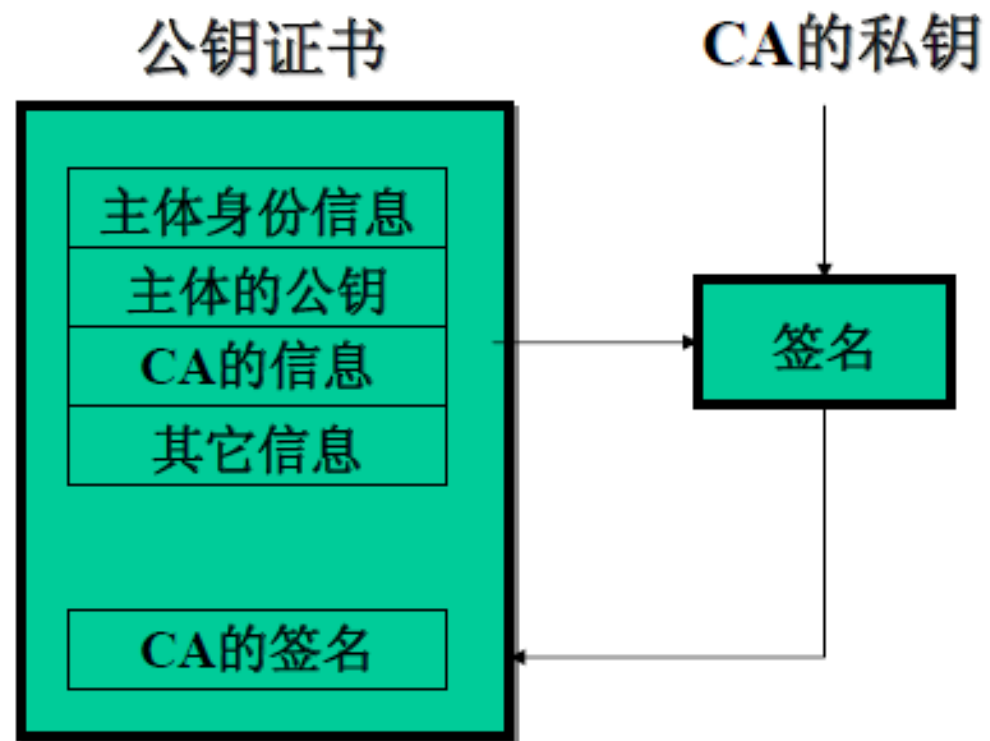


图11.3. 简单证书以意图

11.3 公钥密码的密钥分配

有了公钥证书系统后，如果某个用户需要任何其他已向CA注册的用户公钥，可向持证人(或证书机构)直接索取公钥证书。

- 用CA的公钥验证CA的签名，从而获得对公钥的信任。
- 由于公钥证书不需要保密，可以在网络上分发，从而实现公钥的安全网络分配。
- 又由于公钥证书有CA的签名，攻击者不能伪造合法的公钥证书。因此，只要CA是可信的，公钥证书就是可信的，其公钥就是可信的。

11.3 公钥密码的密钥分配

3. 使用公钥证书的主要好处

- ①用户只要获得用户的证书，就可以获得用户的公钥。
- ②用户只要获得 CA的公钥，就可验证证书的真伪，从而安全地获得用户的公钥。
- ③因此公钥证书为公钥的分发奠定了基础，成为公钥密码在大型网络系统中应用的关键技术。

这就是电子政务、电子商务等大型网络应用系统都采用公钥证书的原因。

11.3 公钥密码的密钥分配

4. X.509证书

- 目前应用最广泛的证书格式是国际电信联盟(International Telecommunication Union, ITU)提出的X.509版本3格式。
- X.509标准最早于1988年颁布。在此之后又于1993年和1995年进行过两次修改。
- INTERNET工程任务组(IETF)针对X.509在INTERNET环境的应用, 颁布了一个作为X.509子集的RFC2459。从而使X.509在INTERNET环境中得到广泛应用。

11.3 公钥密码的密钥分配



图11.4. 公钥证书结构示意图

11.3 公钥密码的密钥分配

5. 私钥安全存储

私钥保存形式主要有文件、密码设备和软件系统三种。

- 当用文件形式保存私钥时，私钥的安全性通常用口令保护，比如用口令加密存储私钥的文件。当系统用私钥签名或解密时，需要把私钥文件读入内存或者密码模块中进行密码运算。
- 当用密码设备(比如U盾，加密机等)形式保存私钥时，密码设备可提供安全机制保护私钥存储与访问的安全性。
- 当用软件系统方式保存密钥时，私钥完全由软件系统管理，其安全性完全依赖软件系统，不同系统下私钥存储形式和安全性可能完全不同。

目 录

- 11. 1. 密钥管理的概述
- 11. 2. 密钥分配与协商
- 11. 3. 公钥密码的密钥分配
- 11. 4. 移动互联网下的私钥安全分析
- 11. 5. 针对ECDSA的协同签署方案
- 11. 6. 针对SM2签名的协同签署方案
- 11. 7. 针对SM9签名的协同签署方案
- 11. 8. 密钥管理在区块链中的应用

11.4 移动互联网下的私钥安全分析

1. 移动互联网规模



当前，互联网已经由以笔记本、台式机为代表的个人电脑时代转向移动互联网。现在**超过九成**的网络访问都是通过**移动智能终端**来进行

手机网民规模
98.3%



互联网普及率
57.7%

数据来源：互联网络信息中心(CNNIC)发布第42次《中国互联网络发展状况统计报告》

11.4 移动互联网下的私钥安全分析

2. 移动互联网安全现状

密码技术是互联网安全的核心支撑，密码系统的安全性取决于密钥安全

要闻 每经网首页 > 要闻 > 正文

惊天数字货币被盗案再现！比特币有史以来最大的牛市信号能否抵御黑天鹅？

中证报 2019-07-16 13:10:31

www.remixpoint.co.jp **remixpoint**

2019年7月12日

各位

会社名	株式会社リミックスポイント
代表者名	代表取締役社長 CEO 小田 玄紀 (コード番号: 3825)
問合せ先	取締役 CFO 廣瀬 卓也 (TEL: 03-6303-0280)

当株式会社における仮想通貨の不正流出に関するお知らせとお詫び（第一報）

当株式会社で仮想通貨交換業を営む株式会社ビットポイントジャパン（本社：東京都港区、代表取締役小田玄紀、以下「BPJ」といいます。）の仮想通貨交換所における仮想通貨の不正な流出が判明いたしました。

現在、新規口座開設を含むBPJのサービスを全面的に停止するとともに、原因の究明、流出額の特定、被害の最小化等の対策を鋭意行っております。詳細が判明次第、速やかに公表する予定であります。なお、お客様からの預かり資産に被害が生じないように、BPJにおいて補償するなど、責任をもって対応する方針であります。

本件により、BPJのサービスをご利用のお客様、また、当社株主の皆様をはじめ関係者の皆様にご迷惑をおかけいたしますこと、深くお詫び申し上げます。

三星多个项目代码泄露：包括SmartThings源代码、证书和密钥

2019-05-09 10:56:29 来源：新智科技

5月9日早间消息，据美国科技媒体TechCrunch报道，一名信息安全研究员近期发现，三星工程师使用的一个开发平台泄露了多个内部项目，包括三星SmartThings敏感的源代码、证书和密钥。

三星数十个自主编码项目出现在旗下Vandev Lab的GitLab实例中。该实例被三星员工用于分享贡献各种应用、服务和项目的代码。由于这些项目被设置为“公开”，同时没有受到密码保护，因此任何人都可以查看项目，获取并下载源代码。

迪拜信息安全公司SpiderSilk的安全研究员莫撒布·胡赛因(Mossab Hussein)发现了这些敏感的文件。他表示，某个项目包含的证书允许访问正在使用的整个AWS帐号，包括100多个S3存储单元，其中保存了日志和分析数据。

VS

AWS Key Management Service 概览 功能 定价 入门 资源 常见问题

AWS Key Management Service (KMS)

轻松创建和控制用于加密数据的密钥

开始使用 AWS Key Management Service

云市场 开发者 支持 合作与生态 客户

密钥管理服务 KMS

安全、易用的密钥管理服务，轻松创建和管理加密数据的密钥

立即申请

Alibaba Cloud | Worldwide Cloud Services Partner

为何选择阿里云 产品 解决方案

Alibaba Cloud > 产品 > 密钥管理服务

密钥管理服务

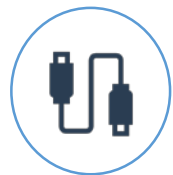
根据需要创建、删除和管理加密密钥

免费开通 联系销售

11.4 移动互联网下的私钥安全分析

3. 密钥存储的现状

利用内置密钥的USB硬件设备等，提供了较为完善的安全机制。

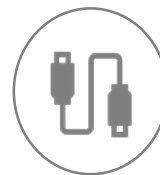


在密钥安全的情况下，确保了用户数据的安全性



已具备的安全能力

存在的问题



难以外接一个USB设备实现安全可控机制



移动终端安全防护能力差
攻击者通过权限提升获得或导出密钥

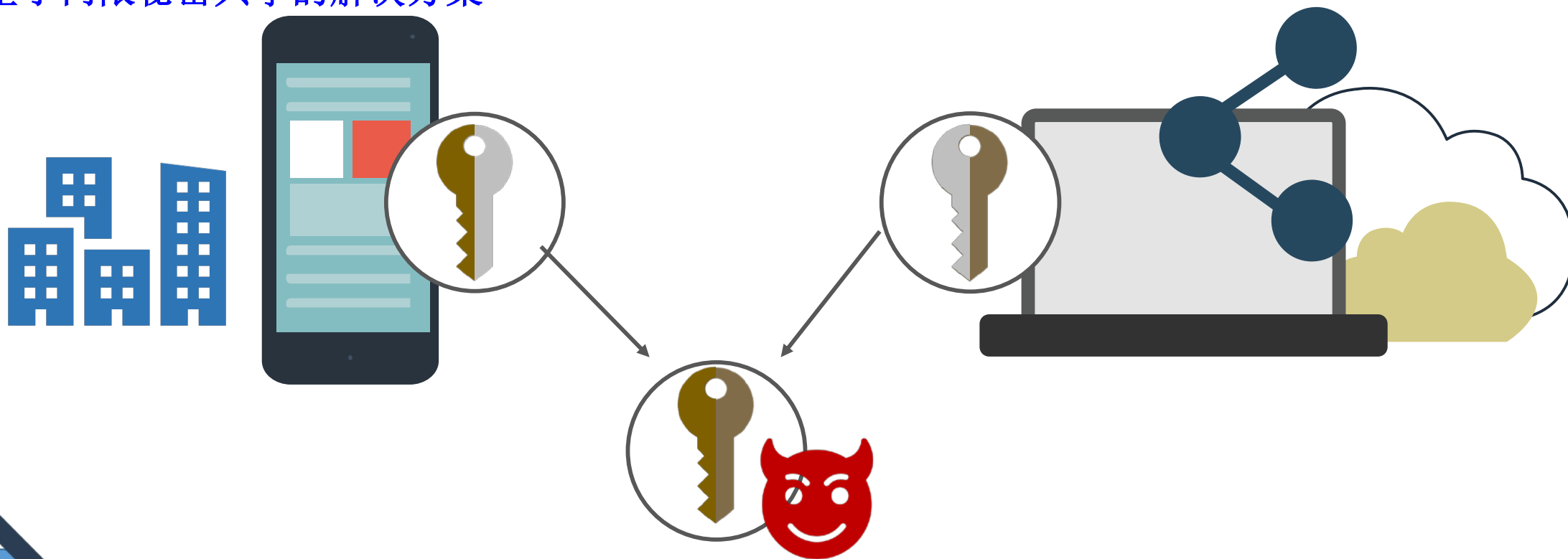
11.4 移动互联网下的私钥安全分析

4. 安全需求分析



11.4 移动互联网下的私钥安全分析

5. 基于门限秘密共享的解决方案



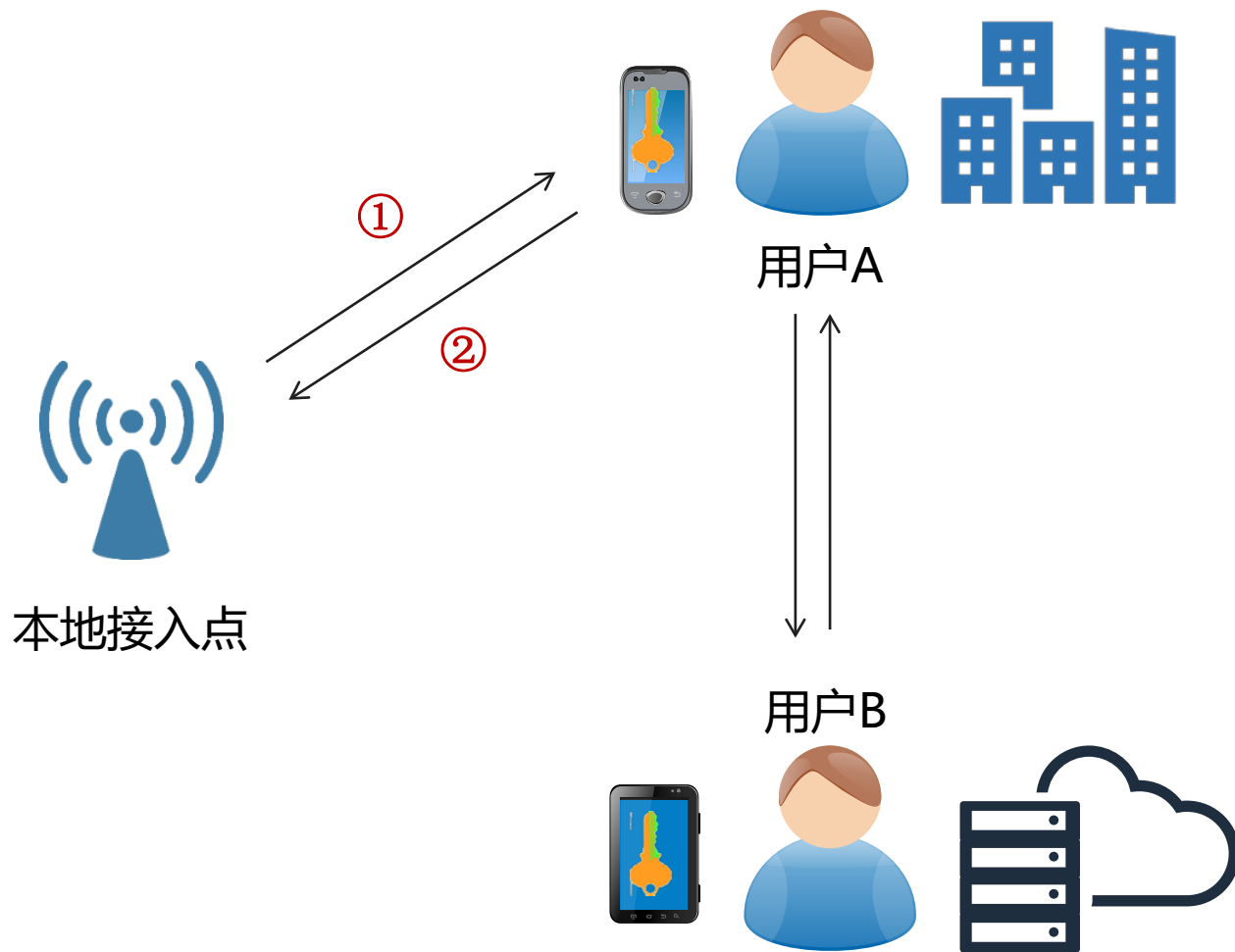
密钥通过 (t,n) 门限秘密共享，任意超过阈值 t 的参与者集合都能恢复完整密钥
拥有完整密钥的设备可能会成为攻击目标

11.4 移动互联网下的私钥安全分析

6. 基于安全多方计算的解决方案

基于MPC的私钥存储方式

- ① 密钥分离于多台机器，两方或多方联合计算且**不恢复完整密钥**
- ② 安全边界延伸到单个机器之外，可**跨越云、数据中心和不同节点**



目 录

- 11. 1. 密钥管理的概述
- 11. 2. 密钥分配与协商
- 11. 3. 公钥密码的密钥分配
- 11. 4. 移动互联网下的私钥安全分析
- 11. 5. 针对ECDSA的协同签署方案
- 11. 6. 针对SM2签名的协同签署方案
- 11. 7. 针对SM9签名的协同签署方案
- 11. 8. 密钥管理在区块链中的应用

11.5 针对ECDSA的协同签署方案

1. ECDSA算法回顾

➤ 密钥生成算法

从 \mathbb{Z}_q 中选取随机数作为私钥 x .

① 公钥 $Q = x \cdot G$, 为 \mathbb{G} 上的一个点.

➤ 签名算法

输入待签名消息 m , 签名者选取随机数 $k \in \mathbb{Z}_q$, 计算:

② $R = (r_x, r_y) = k \cdot G, r = r_x \bmod q, s = k^{-1}(H(m) + x \cdot r) \bmod q.$

最后输出关于 m 的签名 (r, s) .

➤ 验证算法

输入消息 m , 签名值 (r, s) 以及公钥 Q , 验证者检查 r, s 的取值范围是否为 \mathbb{Z}_q , 随后计算

$$(r'_x, r'_y) = R' = s^{-1}(H(m) \cdot G + r \cdot Q) \in \mathbb{G}$$

当且仅当 $r'_x = r$ 时, 验签通过, 否则验签失败.

11.5 针对ECDSA的协同签署方案

2. Paillier加密算法回顾

➤ 密钥生成

- ① 随机选取两个等长的素数 p 和 q 。
- ② 计算 $g = n + 1$, $\lambda = \phi(n)$ and $\mu = (\phi(n))^{-1} \bmod n$, 其中 $\phi(n) = (p - 1)(q - 1)$.
- ③ 公钥 $pk = (n, g)$, 私钥 $sk = (\lambda, \mu)$.

➤ 加密

- ① 从 Z_n^* 中随机选取一个元素 r .
- ② 计算密文 $c = Enc_{pk}(m) = g^m r^n \bmod n^2$, 其中 $0 \leq m < n$.

➤ 解密

- ① 解密密文 c , $m = Dec_{sk}(c) = L(c^\lambda \bmod n^2) \cdot \mu \bmod n$, 这里 $L(x) = \frac{x-1}{n}$.

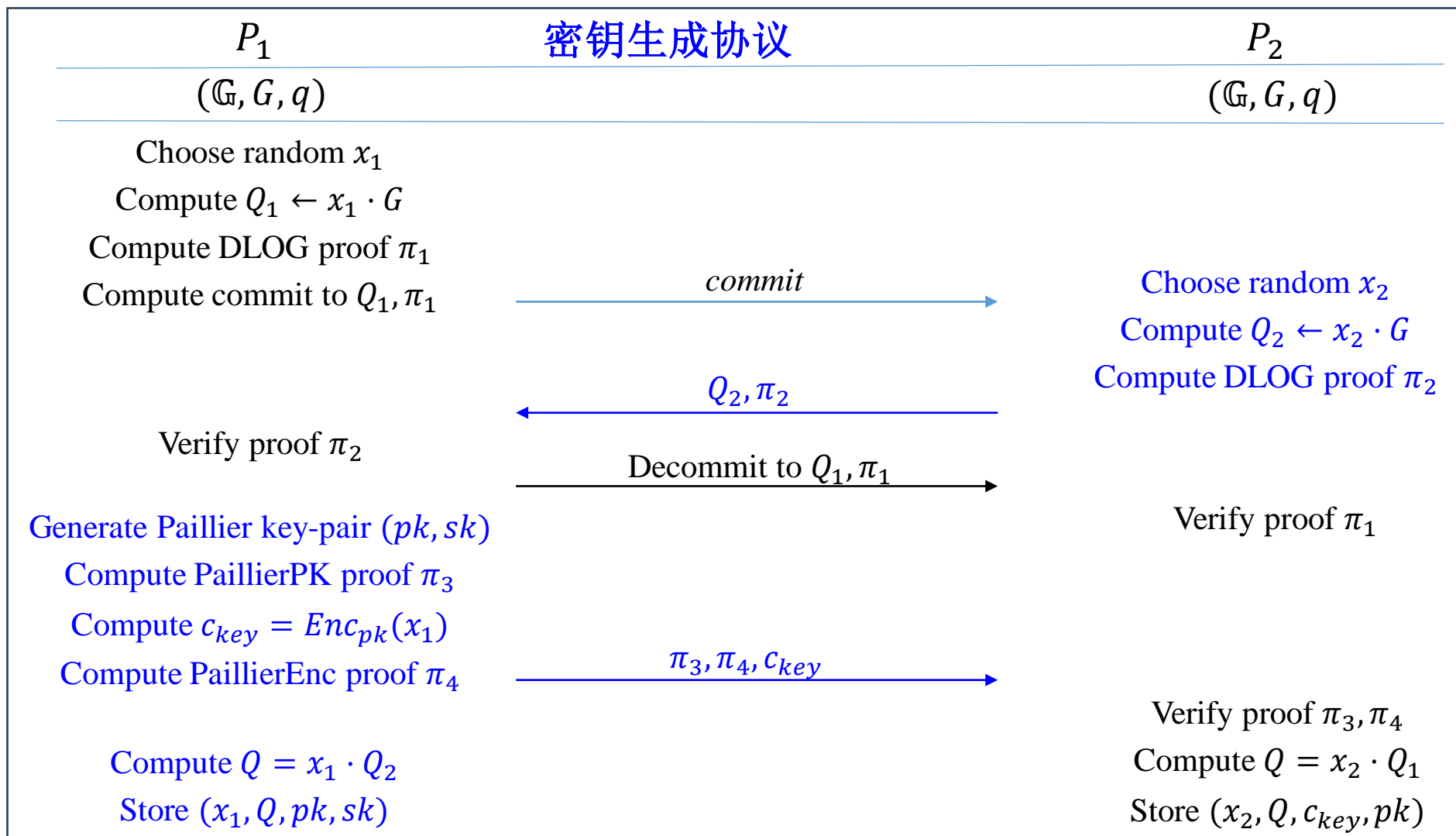
➤ 同态性

我们假设 $c_1 = Enc_{pk}(m_1)$, $c_2 = Enc_{pk}(m_2)$, 则有:

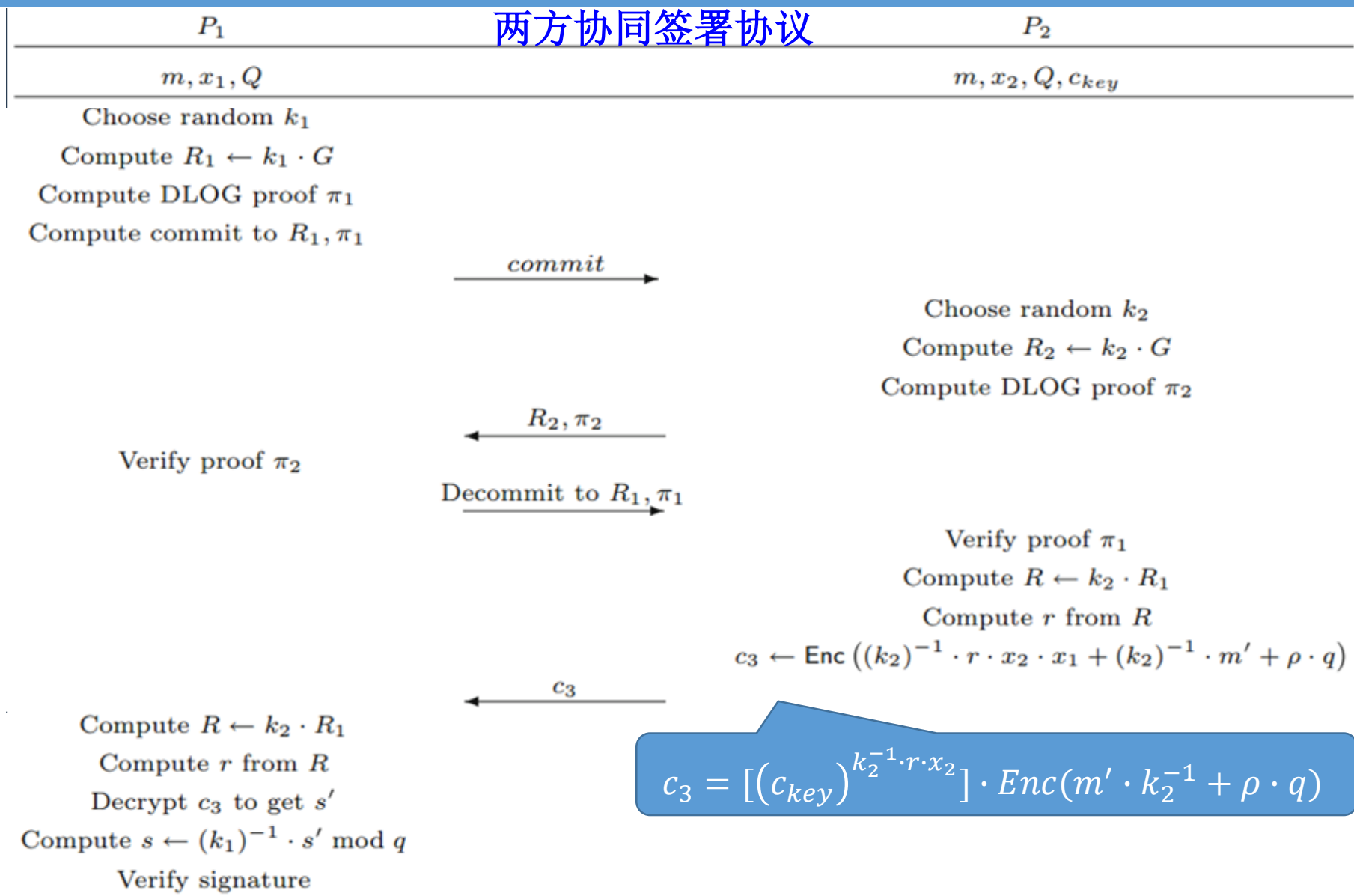
- ① $Dec_{sk}(Enc_{pk}(m_1) \cdot Enc_{pk}(m_2)) = m_1 + m_2$
- ② $Dec_{sk}(Enc_{pk}(m_1)^{m_2}) = m_1 m_2$

11.5 针对ECDSA的协同签署方案

3. 两方协同签署方案[Lindell'2017] Lindell Y. Fast secure two-party ECDSA signing, CRYPTO17', pp. 613-644, 2017.



11.5 针对ECDSA的协同签署方案



11.5 针对ECDSA的协同签署方案

► 正确性分析

公钥: $Q = x_1 \cdot x_2 \cdot G$

签名:

$$R = k_1 \cdot k_2 \cdot G$$

$$s = k_1^{-1} \cdot s' \pmod q$$

$$= k_1^{-1} \cdot Dec(c_3) \pmod q$$

$$= k_1^{-1} \cdot \left(k_2^{-1} \cdot r \cdot x_2 \cdot x_1 + k_2^{-1} \cdot H(m) \right) \pmod q$$

$$= (k_1 \cdot k_2)^{-1} \cdot (r \cdot x_1 \cdot x_2 + H(m)) \pmod q$$

公钥: $Q = x \cdot G$

签名:

$$R = k \cdot G$$

$$s = (k^{-1}) \cdot (r \cdot x + H(m)) \pmod q$$

□ 安全性分析

基于非标准的Paillier-ECDSA假设, 在game-based恶意模型下证明安全性

Lindell Y. Fast secure two-party ECDSA signing, CRYPTO'17, pp. 613-644, 2017.

11.5 针对ECDSA的协同签署方案

4. 两方协同签署方案[DKL'2018]

➤ 设计思路

公钥: $PK = sk_A \cdot sk_B \cdot G$

签名: $R = k_A \cdot k_B \cdot G$

$$sig = (k^{-1}) \cdot (H(m) + r \cdot x) \bmod q$$

$$= (k_A \cdot k_B)^{-1} \cdot (H(m) + r \cdot (sk_A \cdot sk_B))$$

$$= H(m) \cdot \left(\left(\phi + \frac{1}{k_A} \right) \cdot \frac{1}{k_B} - \frac{\phi}{k_B} \right) + r \cdot \left(\frac{sk_A}{k_A} \cdot \frac{sk_B}{k_B} \right)$$

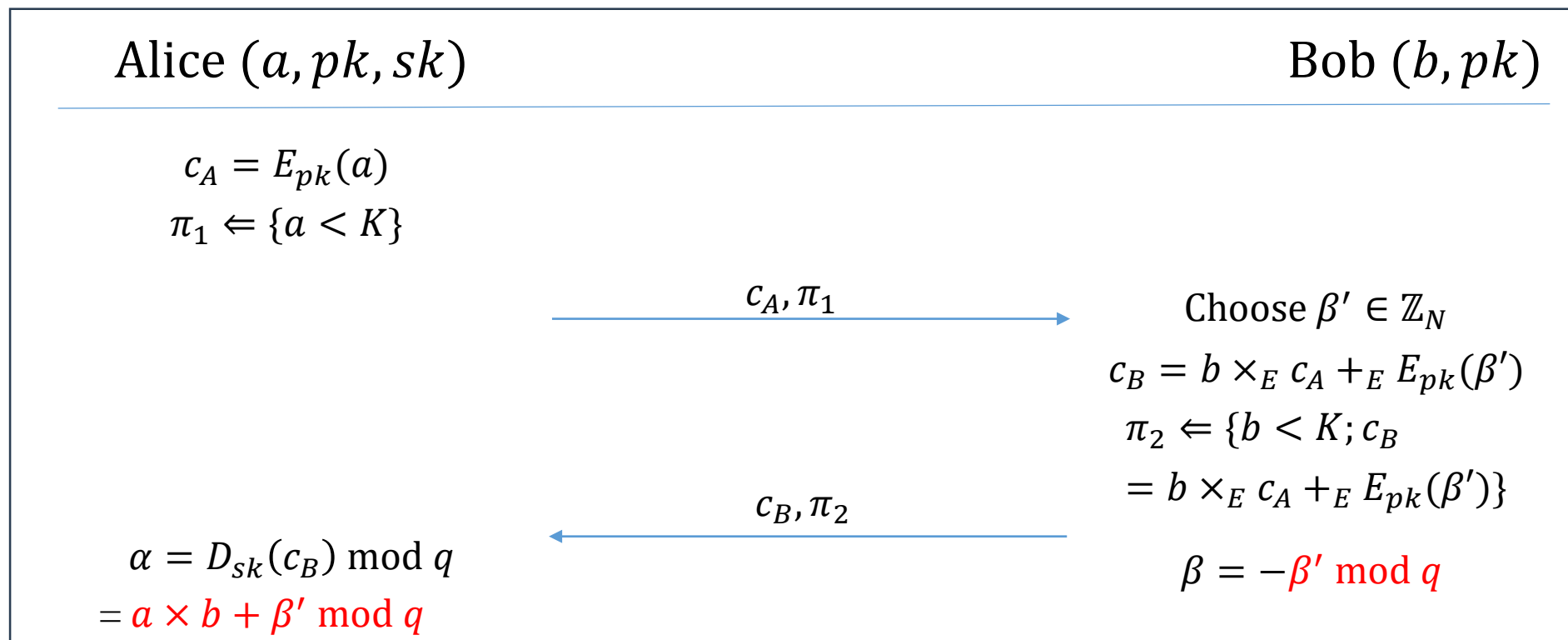
$$= H(m) \cdot \left(t_A^1 + t_B^1 - \frac{\phi}{k_B} \right) + r \cdot (t_A^2 + t_B^2)$$

$$= (H(m) \cdot t_A^1 + r \cdot t_A^2) + (H(m) \cdot (t_B^1 - \frac{\phi}{k_B}) + r \cdot t_B^2)$$

$$= sig_A + sig_B$$

11.5 针对ECDSA的协同签署方案

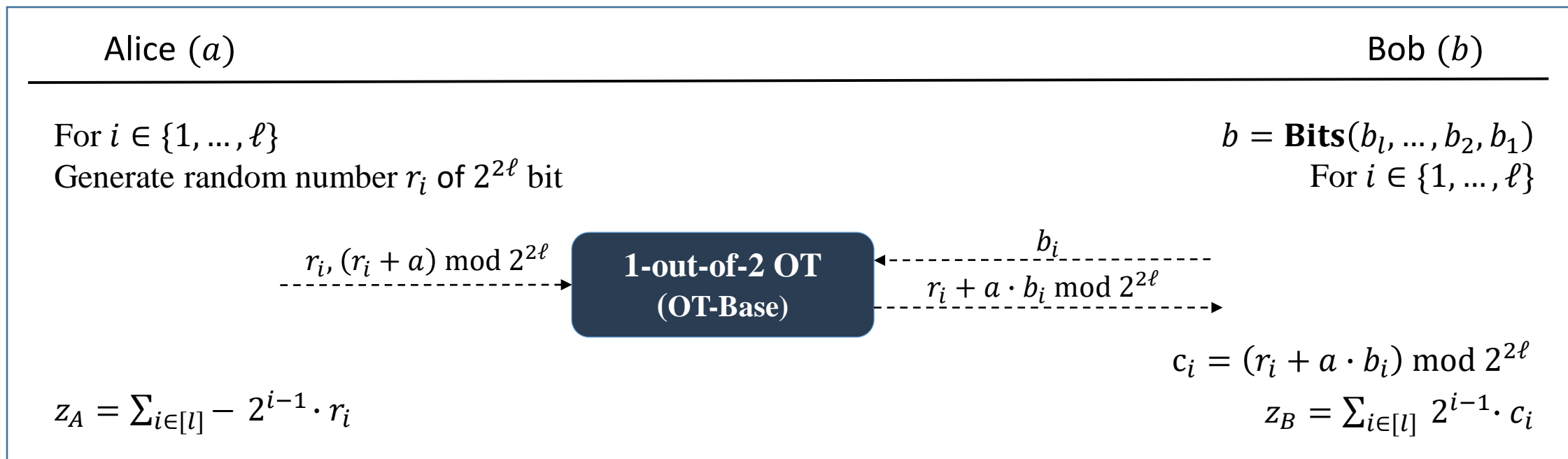
➤ MtA with check“MtAwc”



$$\alpha + \beta = a \times b \bmod q$$

11.5 针对ECDSA的协同签署方案

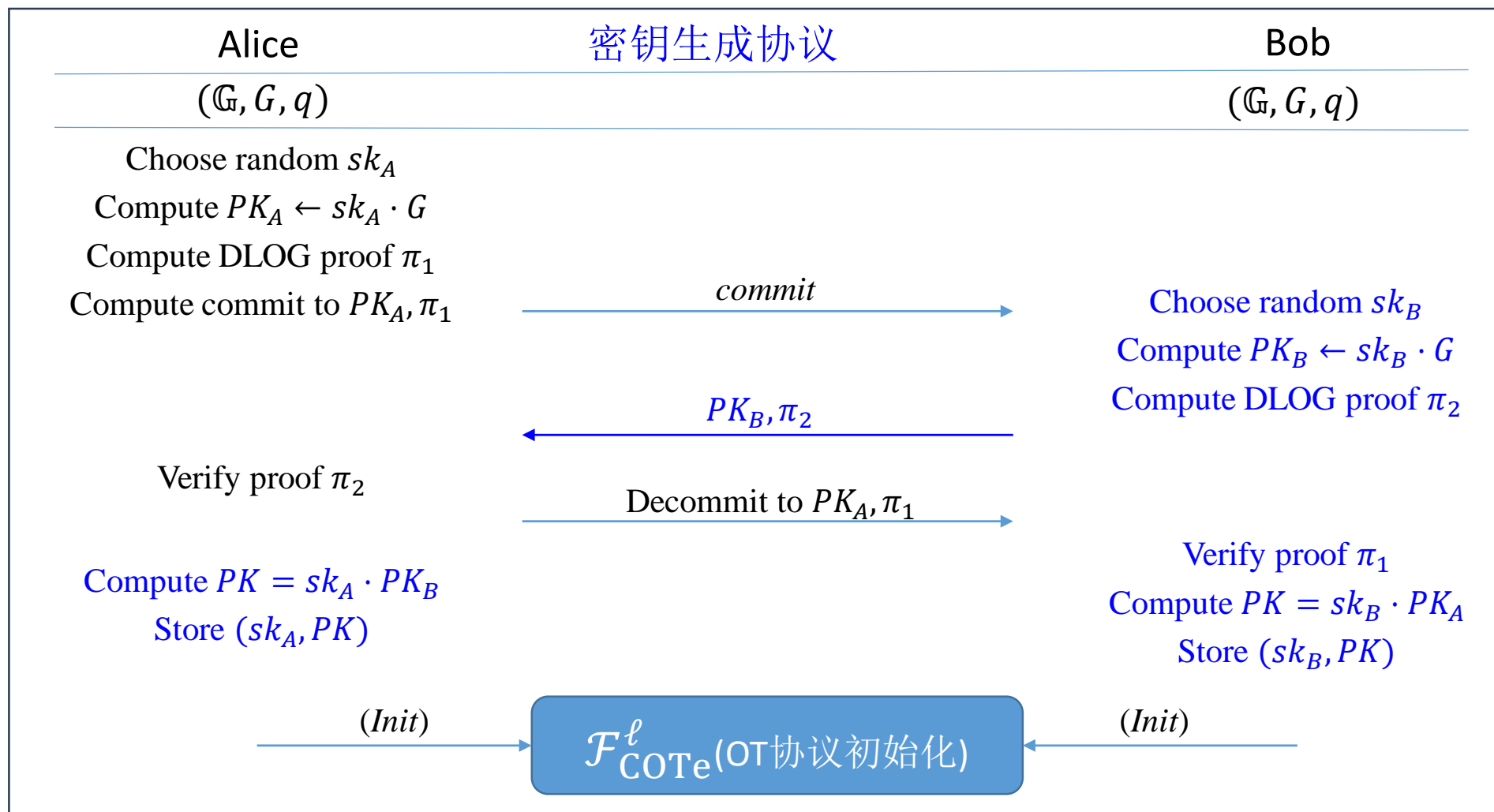
□ $\pi_{\text{mul}}^{\text{priv}}$ 的实现—OT-based



$$a \cdot b = z_A + z_B = \sum_{i \in [l]} 2^{i-1} \cdot r_i + \sum_{i \in [l]} 2^{i-1} \cdot (r_i + a \cdot b_i)$$

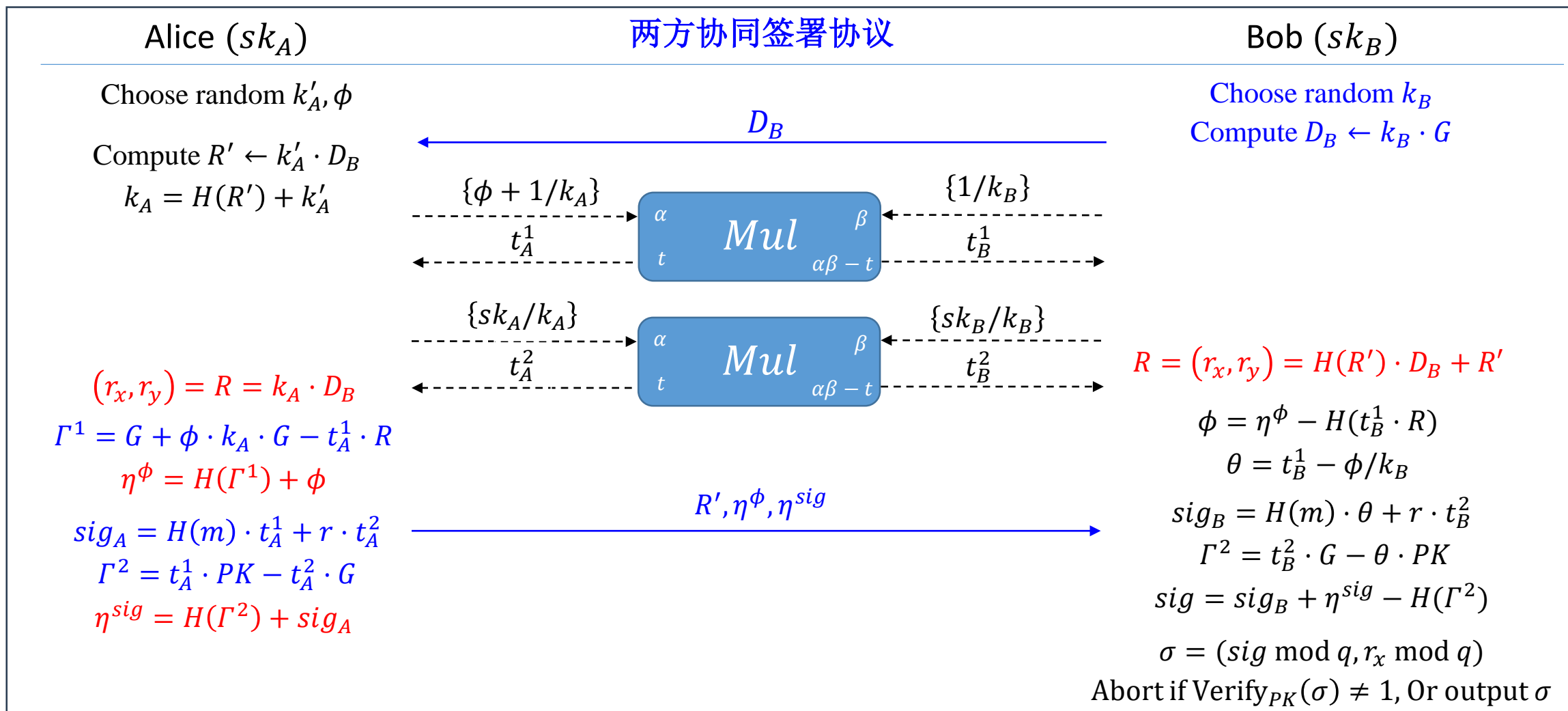
11.5 针对ECDSA的协同签署方案

4. 两方协同签署方案[DKL'2018]



11.5 针对ECDSA的协同签署方案

4. 两方协同签署方案[DKL'2018]



11.5 针对ECDSA的协同签署方案

➤ 正确性分析

公钥: $pk = sk_A \cdot sk_B \cdot G$

签名:

$$R = k_A \cdot k_B \cdot G$$

$$sig = sig_A + sig_B$$

$$= H(m) \cdot \left(t_A^1 + t_B^1 - \frac{\phi}{k_B} \right) + r \cdot (t_A^2 + t_B^2)$$

$$= H(m) \cdot \left(\left(\phi + \frac{1}{k_A} \right) \cdot \frac{1}{k_B} - \frac{\phi}{k_B} \right) + r \cdot \left(\frac{sk_A}{k_A} \cdot \frac{sk_B}{k_B} \right)$$

$$= (k_A \cdot k_B)^{-1} \cdot (H(m) + r \cdot (sk_A \cdot sk_B))$$

公钥: $Q = x \cdot G$

签名:

$$R = k \cdot G$$

$$s = (k^{-1}) \cdot (H(m) + r \cdot x) \bmod q$$

➤ 安全性分析

基于ECDSA安全假设, 在game-based恶意模型下的可证明安全性

Doerner J, et al. Secure two-party threshold ECDSA from ECDSA assumptions, S&P'18, pp. 980-997, 2018.

11.5 针对ECDSA的协同签署方案

5. 多方协同签署方案[LNR'2018]

多方协同签署方案?



$$x = x_1 \cdot x_2$$



$$x = x_1 \cdot x_2 \cdot x_3 \cdot \dots$$

$$x = x_1 + x_2 + x_3 + \dots$$

Lindell Y, Nof A, Ranellucci S, Fast secure multiparty ecDSA with practical distributed key generation and applications to cryptocurrency custody, CCS'18, pp. 1837-1854, 2018.

11.5 针对ECDSA的协同签署方案

思路分析

— 密钥生成:

$$x \leftarrow \mathbb{Z}_q, Q = x \cdot G$$

— 签名:

$$\textcircled{1} k \leftarrow \mathbb{Z}_q, (r_x, r_y) \leftarrow R = k \cdot G$$

$$\textcircled{2} r = r_x \bmod q$$

$$\textcircled{3} s = k^{-1} \cdot (H(m) + x \cdot r) \bmod q$$

$$\textcircled{4} \text{Outputs } \sigma = \{s, r\}$$

$$s = (k\rho)^{-1}(\rho(H(m) + r \cdot x)) \bmod q$$

假设有 n 个参与方, 需要每个参与方分别提供 x_i 和 k_i 最后达到 $x = \sum_{\ell=1}^n x_\ell, k = \sum_{\ell=1}^n k_\ell$ 的效果



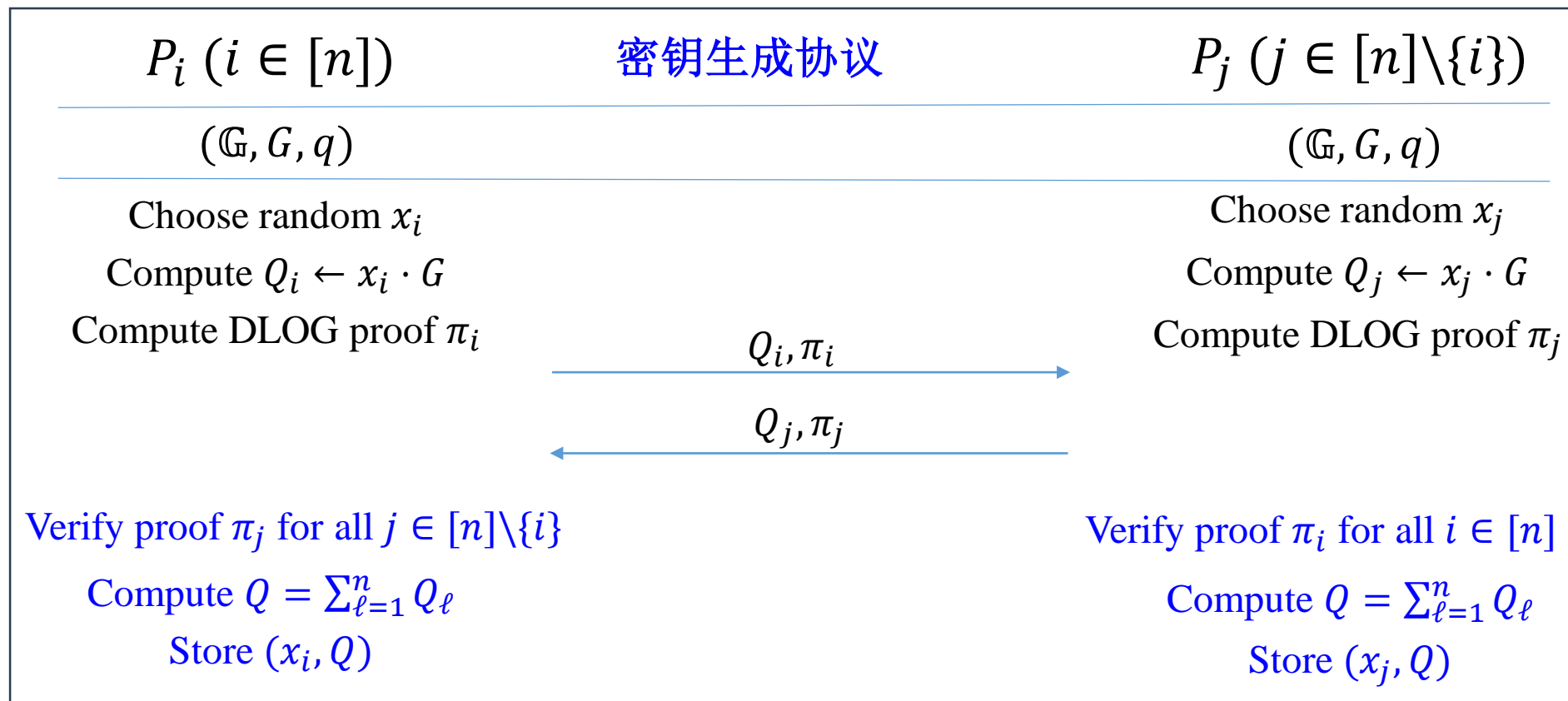
各参与方计算并发送 $c_1 + c_2 = a \cdot b \bmod q$
 $Q_i = x_i \cdot G$ 和 $R_i = k_i \cdot G$ 。
 当收到 $n-1$ 个 Q 和 R 。

$\tau = \sum_{i=1}^n k_i \cdot \sum_{j=1}^n \rho_j = \sum_{i=1}^n \sum_{j=1}^n k_i \cdot \rho_j$
 使用 $n(n-1)$ 个 π_{mul}^{priv} 协议, 乘法碎片转为加法碎片

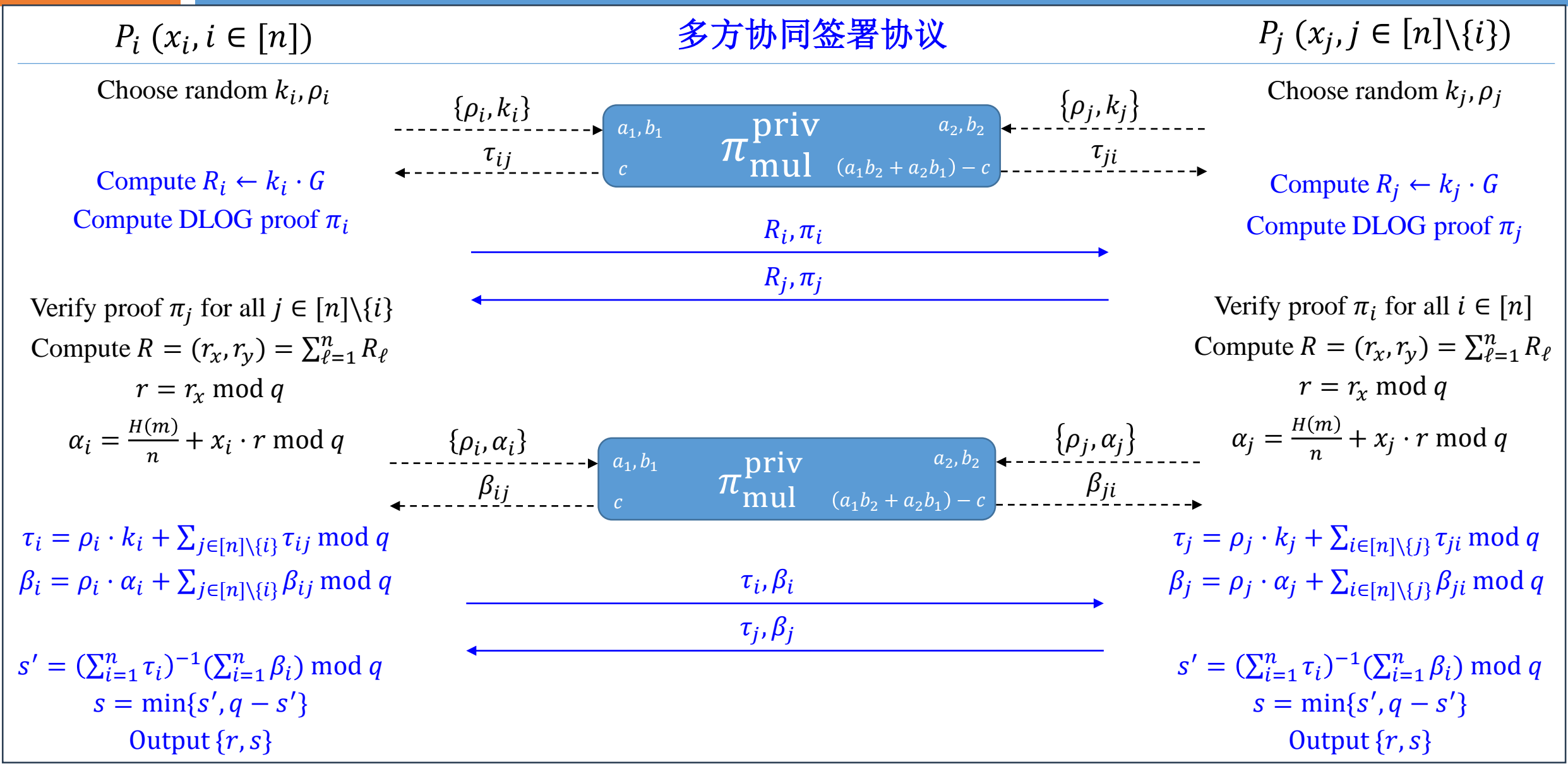
在线计算 $\tau = \sum_{i=1}^n k_i \cdot \sum_{j=1}^n \rho_j,$
 $\beta = \sum_{i=1}^n \rho_i \cdot \sum_{j=1}^n (H(m)/n + x_j \cdot r),$
 最后离线计算 $s = \tau^{-1} \beta \bmod q$

11.5 针对ECDSA的协同签署方案

5. 多方协同签署方案[LNR'2018]



11.5 针对ECDSA的协同签署方案



11.5 针对ECDSA的协同签署方案

➤ 正确性分析

公钥: $Q = (x_1 + \dots + x_n) \cdot G$

签名:

$$R = (k_1 + \dots + k_n) \cdot G$$

$$s = \alpha^{-1} \cdot \beta \pmod q$$

$$= \left(\sum_{i=1}^n \sum_{j=1}^n k_i \cdot \rho_j \right)^{-1} \cdot \left(\sum_{i=1}^n \sum_{j=1}^n \rho_i \cdot \left(\frac{H(m)}{n} + r \cdot x_j \right) \right) \pmod q$$

$$= \left(\sum_{i=1}^n k_i \right)^{-1} \cdot \left(\sum_{j=1}^n \rho_j \right)^{-1} \cdot \left(\sum_{i=1}^n \rho_i \right) \cdot \left(H(m) + r \cdot \sum_{j=1}^n x_j \right) \pmod q$$

$$= \left(\sum_{i=1}^n k_i \right)^{-1} \cdot \left(H(m) + r \cdot \sum_{j=1}^n x_j \right) \pmod q$$

公钥: $Q = x \cdot G$

签名:

$$R = k \cdot G$$

$$s = (k^{-1}) \cdot (H(m) + r \cdot x) \pmod q$$

➤ 安全性分析

根据乘法协议的不同, 安全假设不同, 使用ElGamal和零知识证明, 在game-based恶意模型下的可证明安全性

11.5 针对ECDSA的协同签署方案

6. 门限多方协同签署方案[GG'2018]

➤ 设计思路

公钥: $Q = (u_1 + \dots + u_n) \cdot G = (\lambda_1^S(0)x_1 + \dots + \lambda_t^S(0)x_t) \cdot G$

签名:

$$R = (\sum_{i \in S} k_i)^{-1} \cdot G$$

$$= \left((\sum_{i \in S} \gamma_i) (\sum_{i \in S} \gamma_i)^{-1} (\sum_{i \in S} k_i)^{-1} \right) \cdot G$$

$$s = (\sum_{i \in S} k_i) \cdot \left(h(m) + r \cdot (\sum_{i \in [n]} u_i) \right) \bmod q$$

$$= (\sum_{i \in S} k_i) \cdot \left(h(m) + r \cdot (\sum_{i \in S} \lambda_i^S(0) \cdot x_i) \right) \bmod q$$

$$= \sum_{i \in S} h(m) \cdot k_i + r \cdot (\sum_{i \in S} k_i) \cdot (\sum_{i \in S} w_i) \bmod q$$

$$= \sum_{i \in S} h(m) \cdot k_i + r \cdot \sigma_i \bmod q$$

公钥:

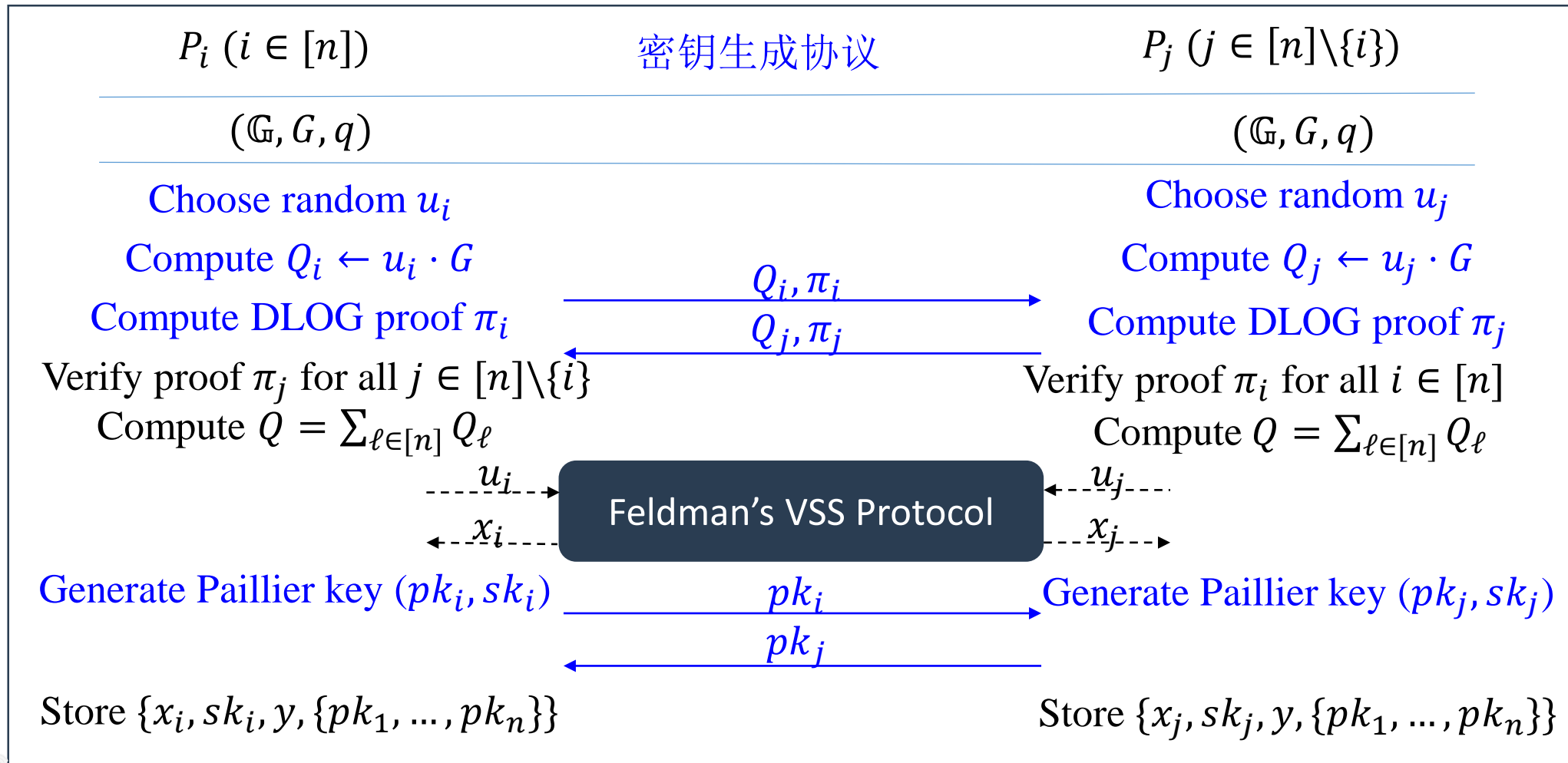
$$Q = g^x$$

签名:

$$R = k^{-1} \cdot G, r = H'(R)$$

$$s = k \cdot (h(m) + xr) \bmod q$$

11.5 针对ECDSA的协同签署方案



Feldman's VSS Protocol: $\sum_{i=1}^n u_i = x = \sum_{i \in S} \lambda_i^S(0) \cdot x_i, S \subseteq [n]$

11.5 针对ECDSA的协同签署方案

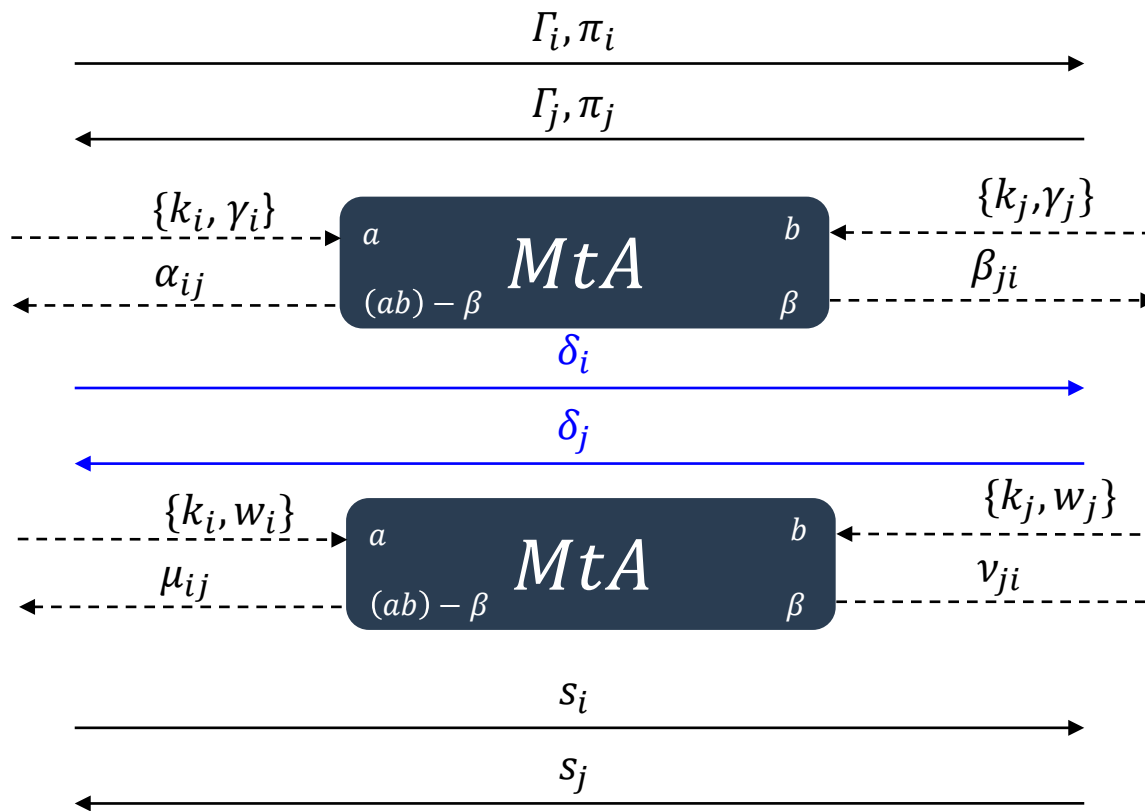
(n, t)门限签名协议

$P_i (x_i, i \in [n])$

$P_j (x_j, j \in [n])$

Choose random k_i, γ_i
 Compute $\Gamma_i \leftarrow \gamma_i \cdot G$
 Compute DLOG proof π_i
 Verify proof π_j for all $j \in [n]$

Choose random k_j, γ_j
 Compute $\Gamma_j \leftarrow \gamma_j \cdot G$
 Compute DLOG proof π_j
 Verify proof π_i for all $i \in [n]$



$$\delta_i = k_i \cdot \gamma_i + \sum_{j \neq i} \alpha_{ij}$$

$$w_i = (\lambda_i^S(0)) \cdot (x_i)$$

$$\delta_j = k_j \cdot \gamma_j + \sum_{i \neq j} \beta_{ji}$$

$$w_j = (\lambda_j^S(0)) \cdot (x_j)$$

$$\sigma_i = k_i \cdot w_i + \sum_{j \neq i} \mu_{ij}$$

$$\delta = \sum_{i \in S} \delta_i$$

$$R = \delta^{-1} \cdot G, r = H'(R)$$

$$s_i = h(m) \cdot k_i + r \cdot \sigma_i$$

$$\sigma_j = k_j \cdot w_j + \sum_{i \neq j} \nu_{ji}$$

$$\delta = \sum_{i \in S} \delta_i$$

$$R = \delta^{-1} \cdot G, r = H'(R)$$

$$s_j = h(m) \cdot k_j + r \cdot \sigma_j$$

$$s = \sum_{i \in S} s_i, \text{output } \{r, s\}$$

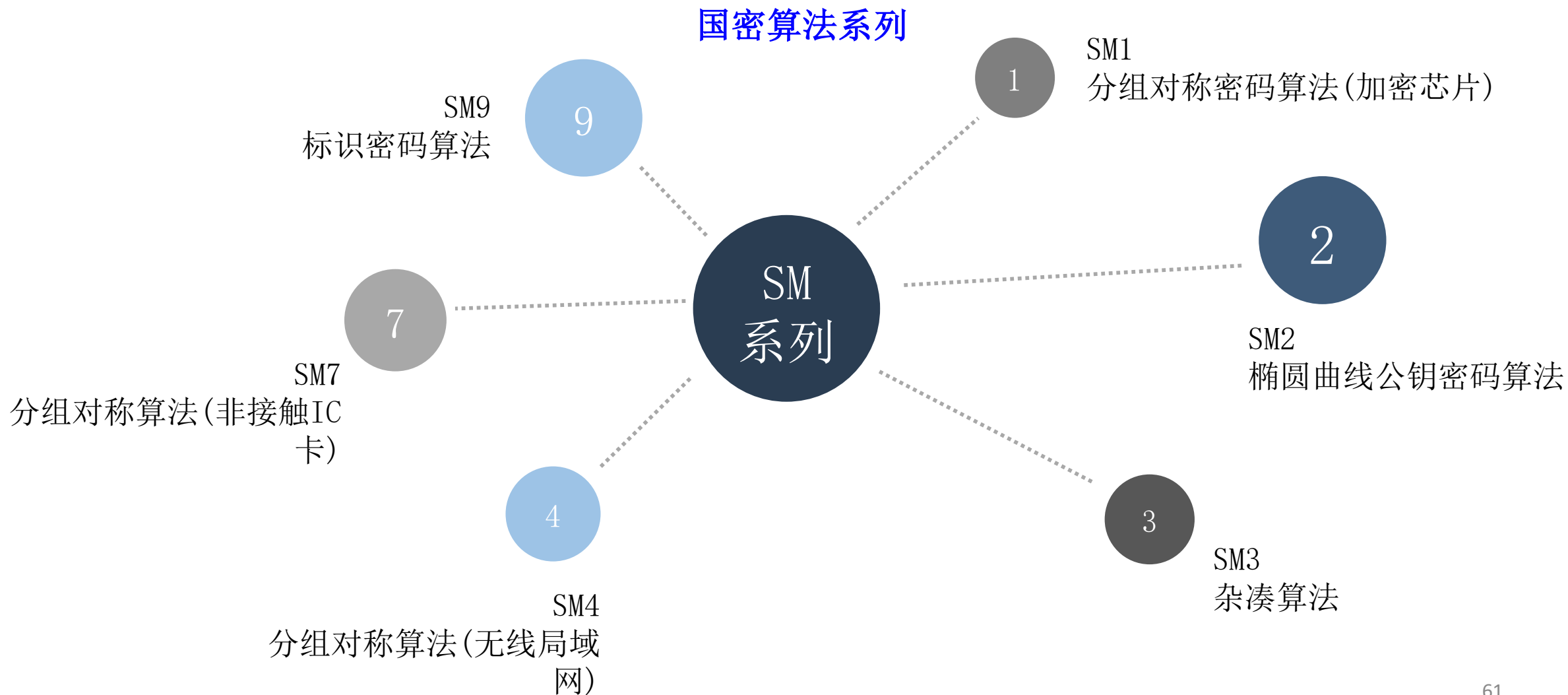
$$s = \sum_{i \in S} s_i, \text{output } \{r, s\}$$

目 录

- 11. 1. 密钥管理的概述
- 11. 2. 密钥分配与协商
- 11. 3. 公钥密码的密钥分配
- 11. 4. 移动互联网下的私钥安全分析
- 11. 5. 针对ECDSA的协同签署方案
- 11. 6. 针对SM2签名的协同签署方案
- 11. 7. 针对SM9签名的协同签署方案
- 11. 8. 密钥管理在区块链中的应用

11.6 针对SM2签名的协同签署方案

1. SM2签名算法回顾



11.6 针对SM2签名的协同签署方案

1. SM2签名算法回顾

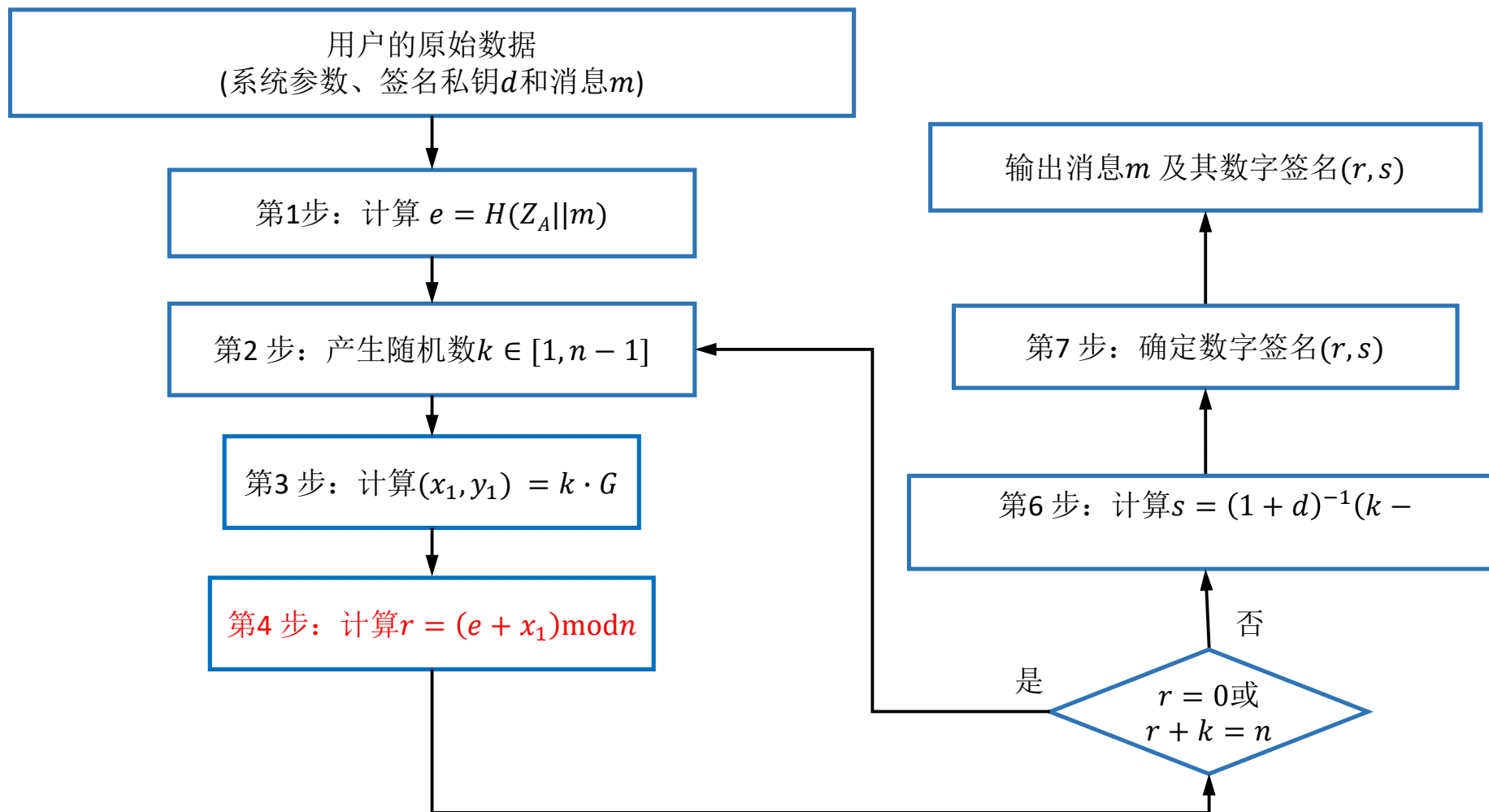
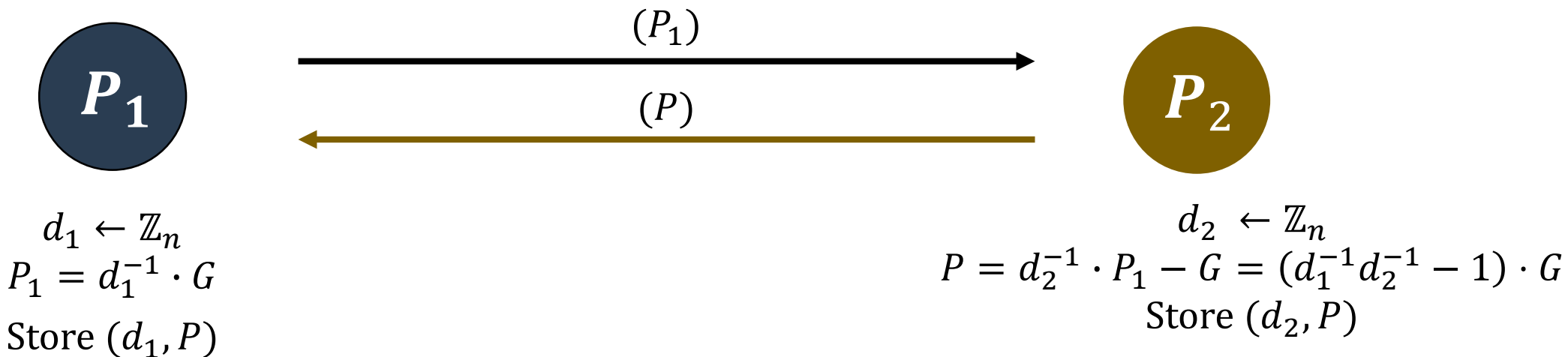


图6.14. SM2数字签名算法签名过程示意图

11.6 针对SM2签名的协同签署方案

2. 两方协同签署方案[林璟锵等'2014]

➤ 密钥生成协议



11.6 针对SM2签名的协同签署方案

2. 两方协同签署方案[林璟锵等'2014]

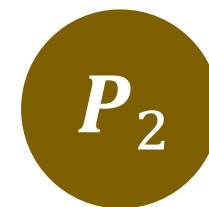
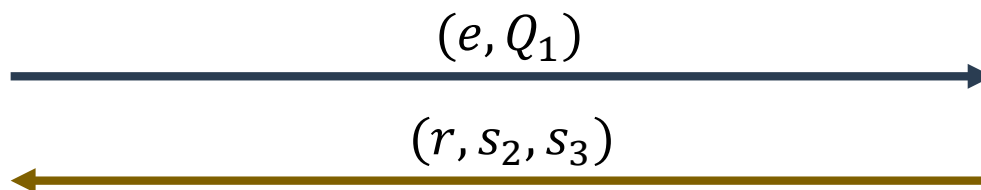
➤ 两方协同签名协议



$$e = H(Z||m)$$
$$k_1 \leftarrow \mathbb{Z}_n$$
$$Q_1 = k_1 \cdot G$$

$$s = (d_1 \cdot k_1) \cdot s_2 + d_1 \cdot s_3 - r \pmod n$$

Output $\{r, s\}$ after being verified



$$k_2, k_3 \leftarrow \mathbb{Z}_n$$
$$Q_2 = k_2 \cdot G$$
$$R = (r_x, r_y) = k_3 \cdot Q_1 + Q_2$$
$$= (k_1 \cdot k_3 + k_2) \cdot G$$
$$r = (r_x + e) \pmod n$$
$$s_2 = d_2 \cdot k_3 \pmod n$$
$$s_3 = d_2 \cdot (r + k_2) \pmod n$$

11.6 针对SM2签名的协同签署方案

2. 两方协同签署方案[林璟锵等'2014]

➤ 正确性分析

公钥: $Q = (d_1^{-1}d_2^{-1} - 1) \cdot G$

签名:

$$R = (k_1 \cdot k_3 + k_2) \cdot G = (r_x + r_y)$$

$$r = e + r_x \bmod n$$

$$s = (d_1 \cdot k_1) \cdot s_2 + d_1 \cdot s_3 - r \bmod n$$

$$= (d_1 \cdot k_1) \cdot d_2 \cdot k_3 + d_1 \cdot d_2 \cdot (r + k_2) - r \bmod n$$

$$= d_1 d_2 \cdot (k_1 k_3 + k_2 + r) - r \bmod n$$

公钥:

$$Q = (x^{-1} - 1) \cdot G$$

签名:

$$R = k \cdot G = (r_x + r_y)$$

$$r = e + r_x \bmod n$$

$$s = (x^{-1} - 1 + 1)^{-1} \cdot (k + r) - r \bmod n$$

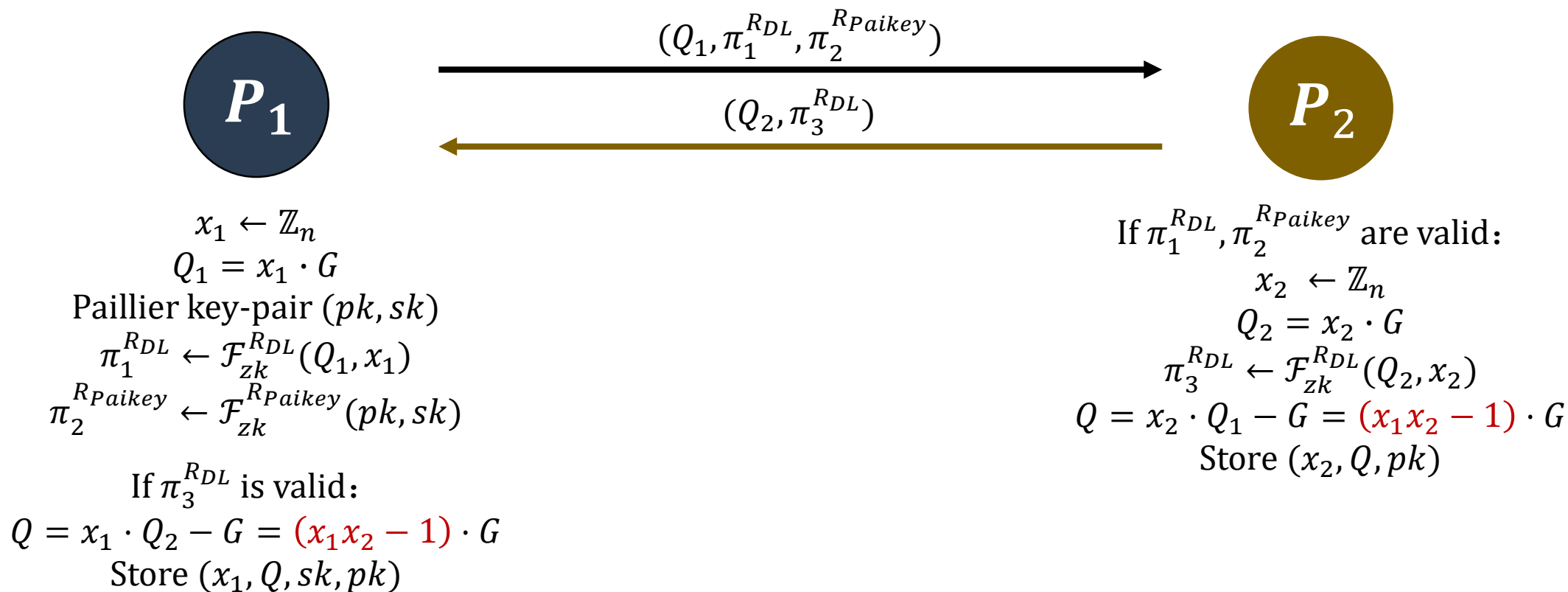
$$= x \cdot (k + r) - r \bmod n$$

林璟锵, 马原, 荆继武, 王琼霄, 雷灵光, 蔡权伟, 王雷. 适用于云计算的基于SM2算法的签名及解密方法和系统. 专利公开号: CN104243456B.

11.6 针对SM2签名的协同签署方案

3. 两方协同签署方案[何德彪等'2017]

➤ 密钥生成协议



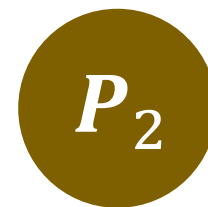
11.6 针对SM2签名的协同签署方案

3. 两方协同签署方案[何德彪等'2017] 两方协同签名协议



$$(R_1, C_k, \pi_4^{RDL}, \pi_5^{RPaiEnc})$$

$$(R_2, C_1, \pi_6^{RDL}, \pi_7^{RPaiHom})$$



$$\begin{aligned}
 k_1 &\leftarrow \mathbb{Z}_n \\
 R_1 &= k_1 \cdot G \\
 C_k &= Enc_{pk}(k_1) \\
 \pi_4^{RDL} &\leftarrow \mathcal{F}_{zk}^{RDL}(R_1, k_1) \\
 \pi_5^{RPaiEnc} &\leftarrow \mathcal{F}_{zk}^{RPaiEnc}(C_k, k_1, pk)
 \end{aligned}$$

If $\pi_6^{RDL}, \pi_7^{RPaiHom}$ are valid:

$$\begin{aligned}
 R &= (r_x, r_y) = k_1 \cdot R_2 \\
 e &= H(Z||m) \\
 r &= (r_x + e) \bmod n
 \end{aligned}$$

$$s' = Dec_{sk}(C_1) \bmod n$$

$$s = x_1^{-1} \cdot s' - r \bmod n$$

Output $\{r, s\}$ after being verified

If $\pi_4^{RDL}, \pi_5^{RPaiEnc}$ are valid:

$$\begin{aligned}
 k_2 &\leftarrow \mathbb{Z}_n \\
 R_2 &= k_2 \cdot G \\
 \pi_6^{RDL} &\leftarrow \mathcal{F}_{zk}^{RDL}(R_2, k_2) \\
 e &= H(Z||m) \\
 R &= (r_x, r_y) = k_2 \cdot R_1 \\
 r &= (r_x + e) \bmod n
 \end{aligned}$$

$$\begin{aligned}
 C_1 &= \left((C_k)^{k_2} \cdot Enc_{pk}(r + \rho n) \right)^{x_2^{-1}} \\
 &= Enc_{pk}(x_2^{-1}(k_1 k_2 + r + \rho n))
 \end{aligned}$$

$$\pi_7^{RPaiHom} \leftarrow \mathcal{F}_{zk}^{RPaiHom}(C_1, x_2, k_2, r, \rho, n)$$

11.6 针对SM2签名的协同签署方案

3. 两方协同签署方案[何德彪等'2017]

➤ 正确性分析

公钥:

$$Q = (x_1 x_2 - 1) \cdot G$$

签名:

$$R = (k_1 \cdot k_2) \cdot G = (r_x + r_y)$$

$$r = e + r_x \pmod n$$

$$s = x_1^{-1} \cdot s' - r \pmod n$$

$$= x_1^{-1} x_2^{-1} (k_1 k_2 + r) - r \pmod n$$

公钥:

$$Q = (x - 1) \cdot G$$

签名:

$$R = k \cdot G = (r_x + r_y)$$

$$r = e + r_x \pmod n$$

$$s = (x - 1 + 1)^{-1} \cdot (k + r) - r \pmod n$$

$$= x^{-1} \cdot (k + r) - r \pmod n$$

何德彪, 张语荻, 孙金龙. 一种SM2数字签名生成方法及系统. 申请公开号: CN107634836A, 2017

11.6 针对SM2签名的协同签署方案

4. 高效的两方协同签署方案[何德彪等'2018]

思路分析

✓ SM2密钥生成:

$$P_{pub} = (x - 1) \cdot G$$

• SM2签名:

- ① $(r_x, r_y) \leftarrow R = k \cdot G$
- ② $r = r_x + e \pmod n$
- ③ $s = x^{-1}(k + r) - r \pmod n$

$$s = (x\rho)^{-1}(\rho(k + r)) - r \pmod n$$

假设有2个参与方，每个参与方分别持有私钥 a 和 b ，后达到 x 的效果

π_{mul}^{priv}

$\sum_{i=1}^2 c_i$

$c_1 + c_2 = a \cdot b \pmod n$

各方计算并广播:

$$\alpha = \sum_{i=1}^2 x_i \cdot \sum_{i=1}^2 \rho_i = \sum_{i=1}^2 \sum_{j=1}^2 x_i \cdot \rho_j$$

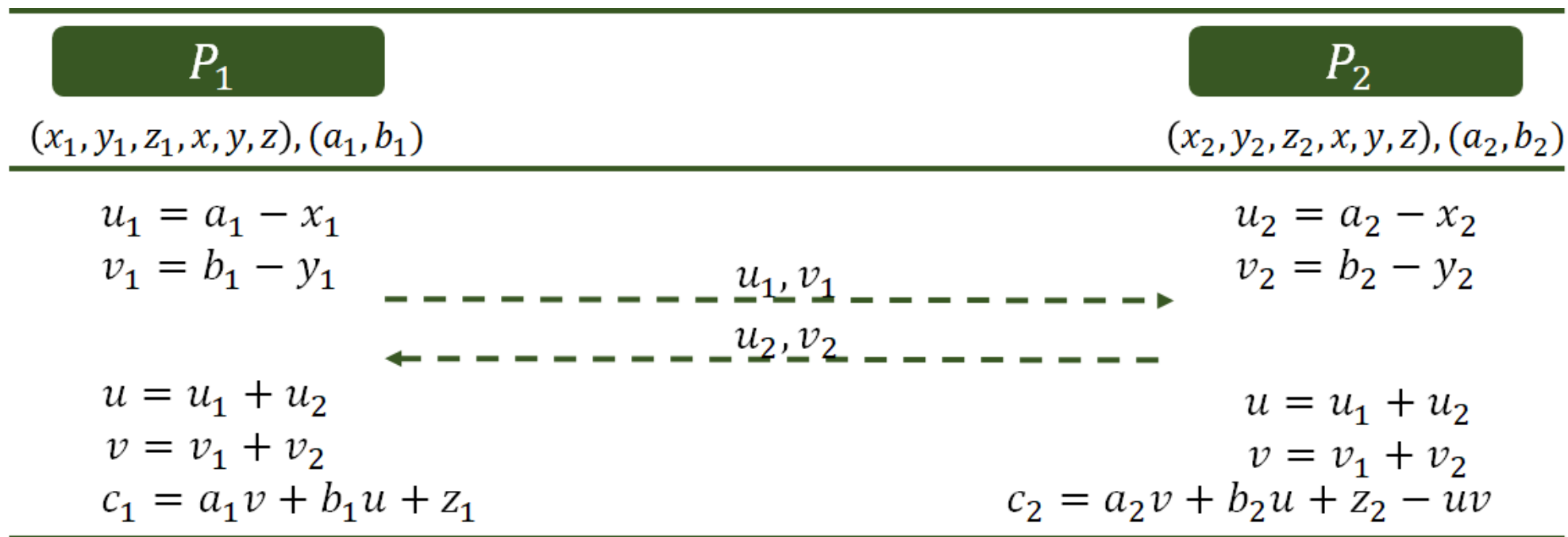
使用 $2(2-1)$ 个 π_{mul}^{priv} 协议，乘法碎片转为加法碎片

在线计算 $\alpha = \sum_{i=1}^2 x_i \cdot \sum_{i=1}^2 \rho_i$,
 $\beta = \sum_{i=1}^2 \rho_i \cdot \sum_{i=1}^2 (k_i + r/2)$,
最后离线计算 $s = \alpha^{-1}\beta - r \pmod n$

11.6 针对SM2签名的协同签署方案

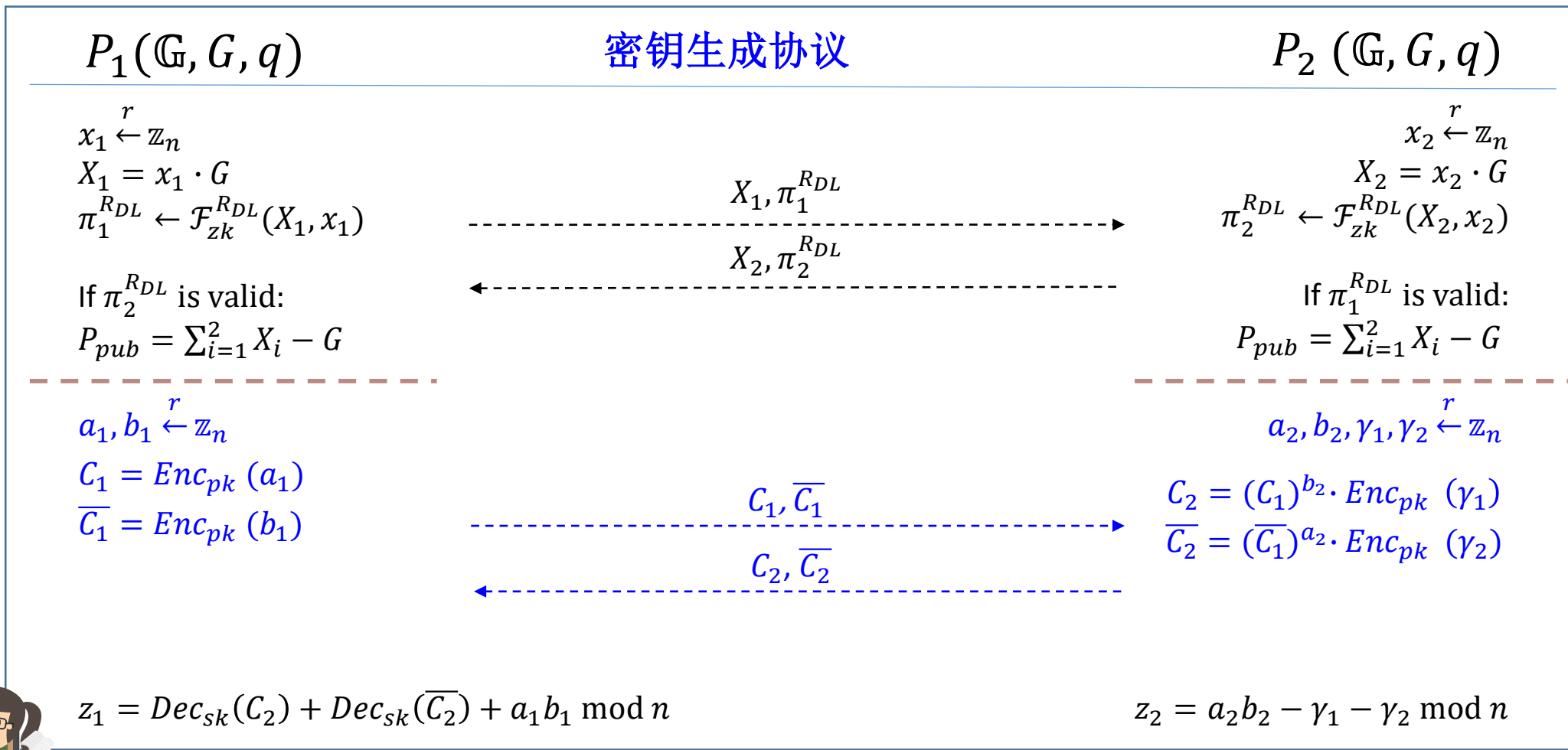
4. 高效的两方协同签署方案[何德彪等'2018]

➤ Beaver' triple—— $ab = ((a - x) + x)((b - y) + y) = (u + x)(v + y)$



备注：预计算阶段双方分别选择 $(x_1, y_1)(x_2, y_2)$ ，使用乘法器计算 $z_1 + z_2 = (x_1 + x_2)(y_1 + y_2)$
计算结果可以得到 $c_1 + c_2 = (a_1 + a_2)(b_1 + b_2)$

11.6 针对SM2签名的协同签署方案



$$z = z_1 + z_2 = a_1 b_2 + \gamma_1 + a_2 b_1 + \gamma_2 + a_1 b_1 + a_2 b_2 - \gamma_1 - \gamma_2 = a \cdot b \text{ mod } n$$



11.6 针对SM2签名的协同签署方案

两方协同签名协议

$P_1 (x_1, a_1, b_1, z_1)$

$$k_1, \rho_1 \leftarrow \mathbb{Z}_n$$

$$R_1 = k_1 \cdot G$$

$$\pi_3^{R_{DL}} \leftarrow \mathcal{F}_{zk}^{R_{DL}}(R_1, k_1)$$

If $\pi_4^{R_{DL}}$ is valid: $R = \sum_{i=1}^2 R_i$

$$r = e + r_x \bmod n$$

$$\delta_1 = k_1 + r/2 \bmod n$$

$$u_1 = x_1 - a_1 \bmod n$$

$$v_1 = \delta_1 - a_1 \bmod n$$

$$w_1 = \rho_1 - b_1 \bmod n$$

$$u = u_1 + u_2 \bmod n$$

$$v = v_1 + v_2 \bmod n$$

$$w = w_1 + w_2 \bmod n$$

$$\alpha_1 = x_1 w + \rho_1 u + z_1 - uw \bmod n$$

$$\beta_1 = \delta_1 w + \rho_1 v + z_1 - vw \bmod n$$

$$s' = (\alpha_1 + \alpha_2)^{-1}(\beta_1 + \beta_2) - r \bmod n$$

$$s = \min\{s', n - s'\}$$

$$a_1 \leftarrow k_1, b_1 \leftarrow \rho_1, z_1 \leftarrow \alpha_1$$

$P_2 (x_2, a_2, b_2, z_2)$

$$k_2, \rho_2 \leftarrow \mathbb{Z}_n$$

$$R_2 = k_2 \cdot G$$

$$\pi_4^{R_{DL}} \leftarrow \mathcal{F}_{zk}^{R_{DL}}(R_2, k_2)$$

If $\pi_3^{R_{DL}}$ is valid: $R = \sum_{i=1}^2 R_i$

$$r = e + r_x \bmod n$$

$$\delta_2 = k_2 + r/2 \bmod n$$

$$u_2 = x_2 - a_2 \bmod n$$

$$v_2 = \delta_2 - a_2 \bmod n$$

$$w_2 = \rho_2 - b_2 \bmod n$$

$$u = u_1 + u_2 \bmod n$$

$$v = v_1 + v_2 \bmod n$$

$$w = w_1 + w_2 \bmod n$$

$$\alpha_2 = x_2 w + \rho_2 u + z_2 \bmod n$$

$$\beta_2 = \delta_2 w + \rho_2 v + z_2 \bmod n$$

$$s' = (\alpha_1 + \alpha_2)^{-1}(\beta_1 + \beta_2) - r \bmod n$$

$$s = \min\{s', n - s'\}$$

$$a_2 \leftarrow k_2, b_2 \leftarrow \rho_2, z_2 \leftarrow \alpha_2$$

11.6 针对SM2签名的协同签署方案

4. 高效的两方协同签署方案[何德彪等'2018]

□ 正确性分析

公钥:

$$Q = (x_1 + x_2 - 1) \cdot G$$

签名:

$$R = (k_1 + k_2) \cdot G = (r_x + r_y)$$

$$r = e + r_x \pmod n$$

$$s = (\alpha_1 + \alpha_2)^{-1}(\beta_1 + \beta_2) - r \pmod n$$

$$= (x_1 + x_2)^{-1}(\rho_1 + \rho_2)^{-1}(k_1 + k_2 + r)(\rho_1 + \rho_2) - r \pmod n$$

$$= (x_1 + x_2)^{-1}(k_1 + k_2 + r) - r \pmod n$$

公钥:

$$Q = (x - 1) \cdot G$$

签名:

$$R = k \cdot G = (r_x + r_y)$$

$$r = e + r_x \pmod n$$

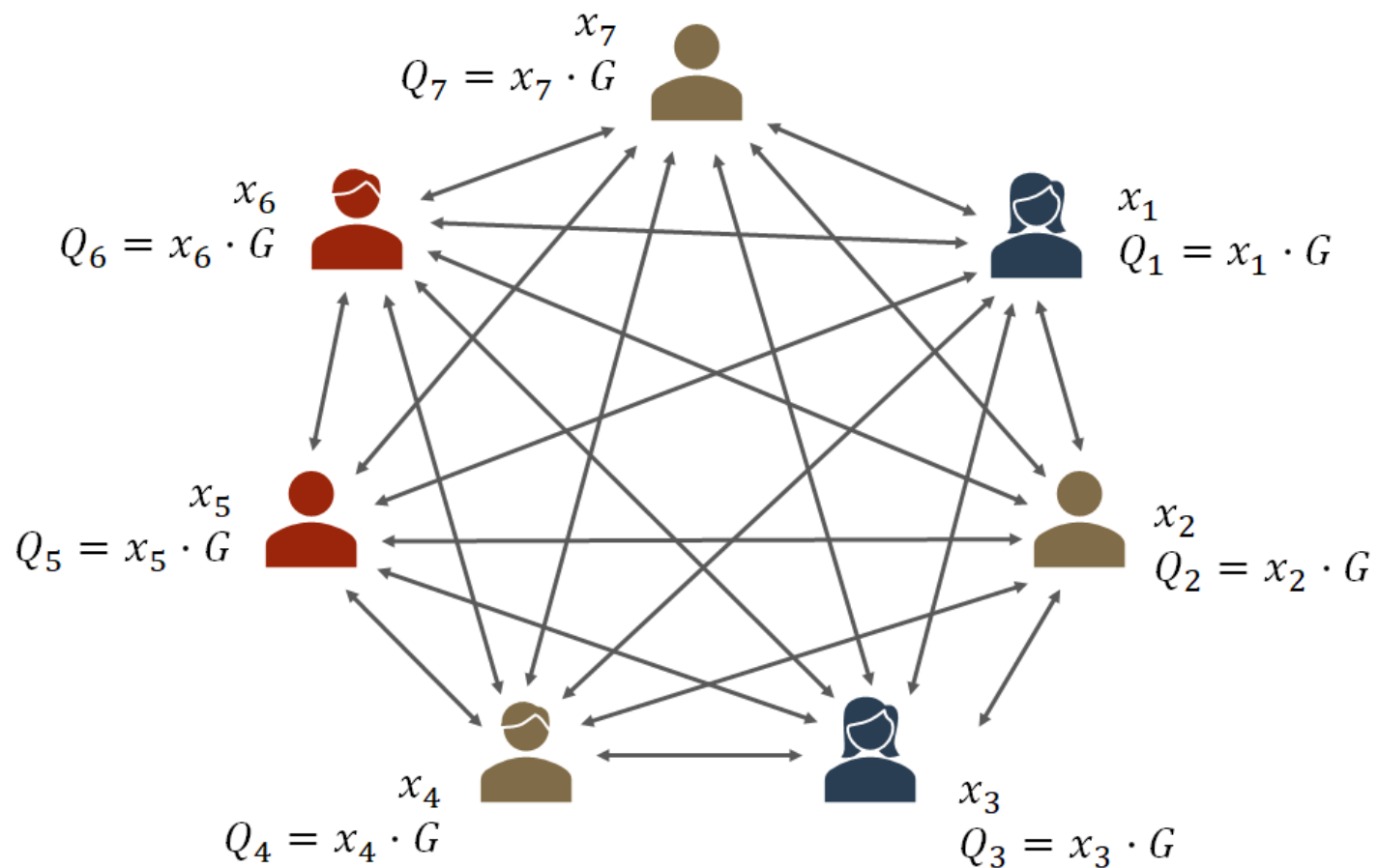
$$s = (x - 1 + 1)^{-1} \cdot (k + r) - r \pmod n$$

$$= x^{-1} \cdot (k + r) - r \pmod n$$

何德彪, 张佳妮, 冯琦, 王婧, 陈泌文. 一种轻量级SM2两方协同生成数字签名的方法. 申请公开号: CN110011803A, 2018

11.6 针对SM2签名的协同签署方案

5. 多方协同签署方案[何德彪等'2019]



11.6 针对SM2签名的协同签署方案

5. 多方协同签署方案[何德彪等'2019]

➤ 思路分析

- SM2密钥生成:

$$P_{pub} = (x - 1) \cdot G$$

- SM2签名:

$$\begin{aligned} \textcircled{1} & (r_x, r_y) \leftarrow R = k \cdot G \\ \textcircled{2} & r = r_x + e \text{ mod } n \\ \textcircled{3} & s = x^{-1}(k + r) - r \text{ mod } n \end{aligned}$$

$$s = (x\rho)^{-1}(\rho(k + r)) - r \text{ mod } n$$

假设有 τ 个参与方, 每个参与方分别持有私钥 k_i 和公钥 ρ_i , 通过 π_{mul}^{priv} 协议, 各方分别计算 $c_i = x_i \cdot \rho_i$ 后达到 x 的效果

$$c_1 + c_2 = a \cdot b \text{ mod } n$$

各方计算并广播:

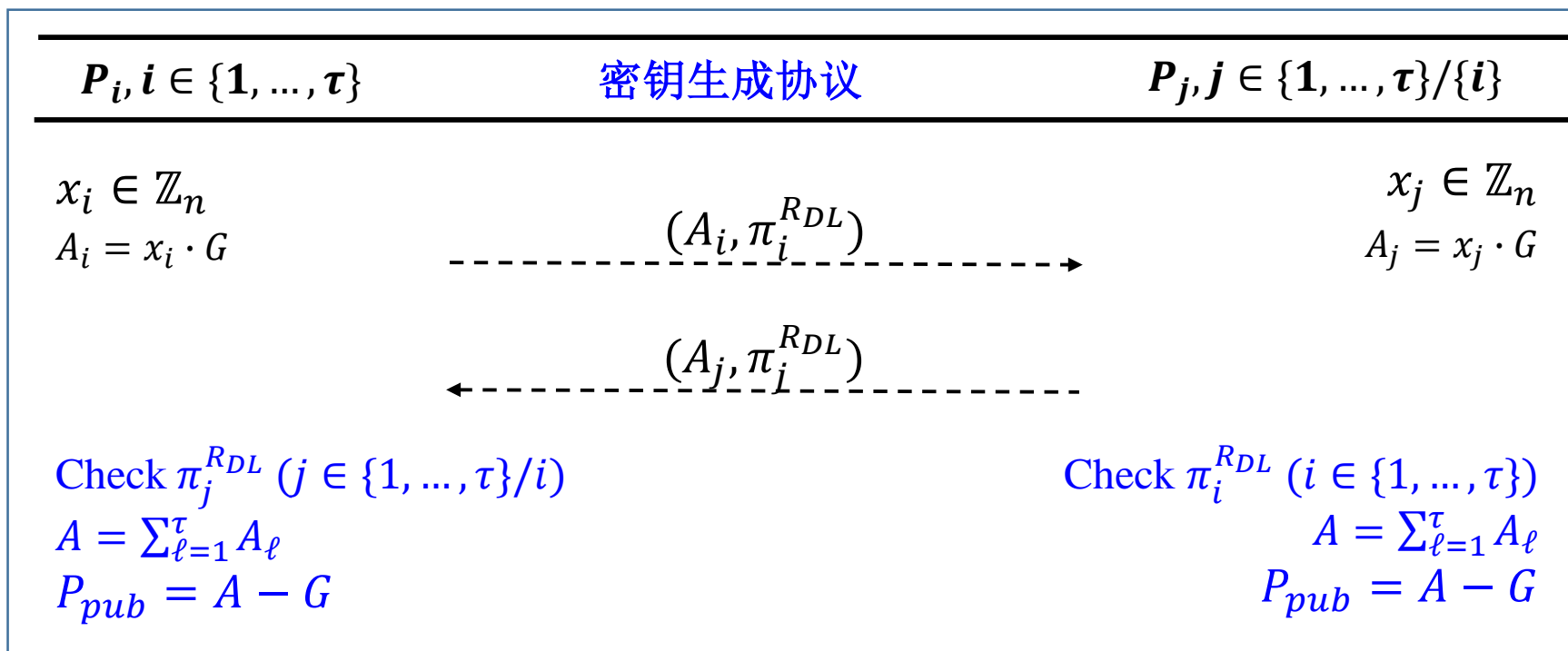
$$\alpha = \sum_{i=1}^{\tau} x_i \cdot \sum_{i=1}^{\tau} \rho_i = \sum_{i=1}^{\tau} \sum_{j=1}^{\tau} x_i \cdot \rho_j$$

使用 $n(n-1)$ 个 π_{mul}^{priv} 协议, 乘法碎片转为加法碎片

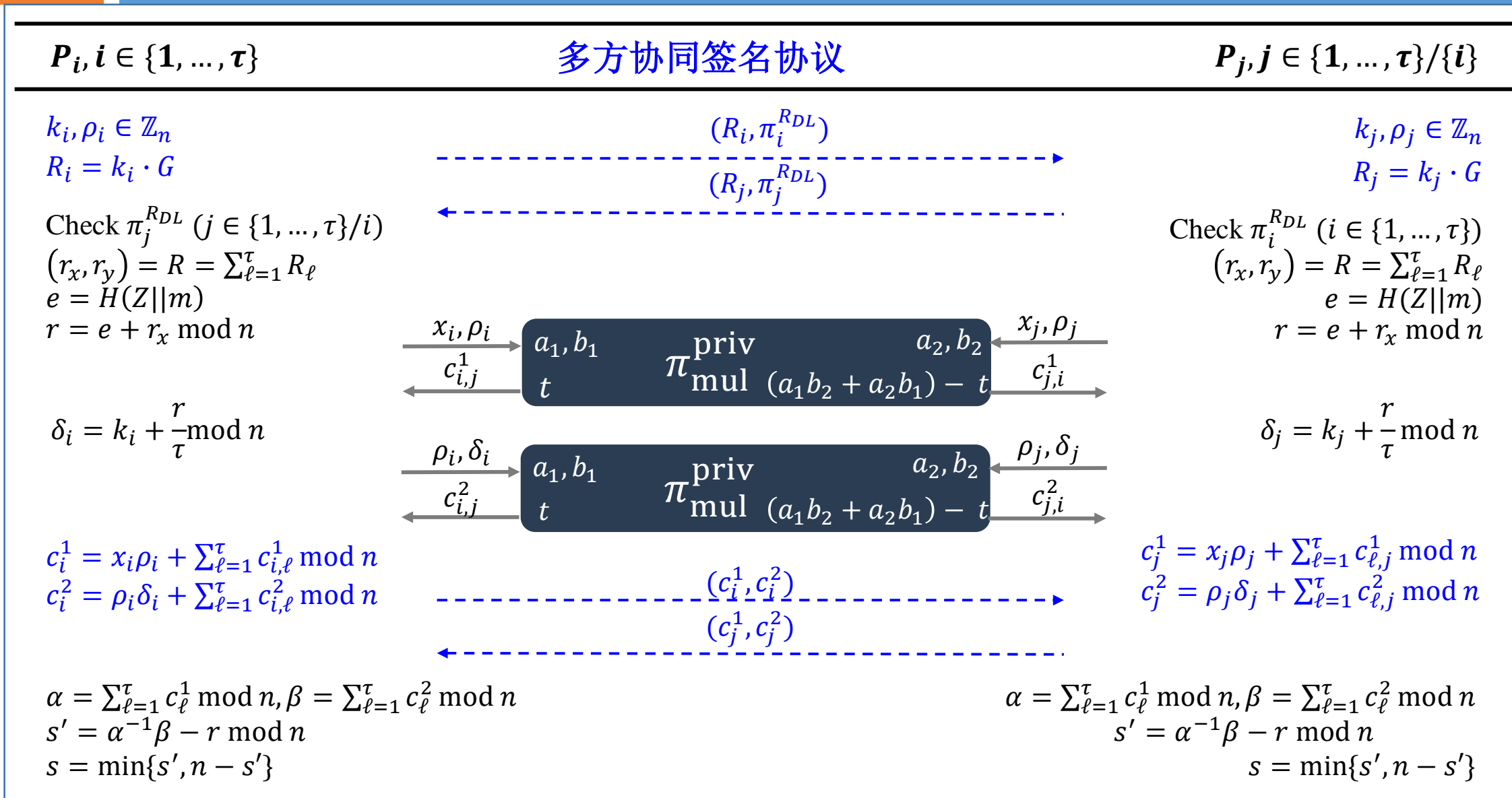
在线计算 $\alpha = \sum_{i=1}^{\tau} x_i \cdot \sum_{i=1}^{\tau} \rho_i$,
 $\beta = \sum_{i=1}^{\tau} \rho_i \cdot \sum_{i=1}^{\tau} (k_i + r/\tau)$,
最后离线计算 $s = \alpha^{-1}\beta - r \text{ mod } n$

11.6 针对SM2签名的协同签署方案

5. 多方协同签署方案[何德彪等'2019]



11.6 针对SM2签名的协同签署方案



11.6 针对SM2签名的协同签署方案

① SM2密钥生成: $x \leftarrow \mathbb{Z}_n$, $P_{pub} = (x^{-1} - 1) \cdot G$

② SM2签名: $(r_x, r_y) \leftarrow R = k \cdot G$, $r = r_x + e \bmod n$, $s = x(k + r) - r \bmod n$



① 分布式密钥生成: $x = \sum_{\ell=1}^{\tau} x_{\ell}$, $\rho = \sum_{\ell=1}^{\tau} \rho_{\ell}$, $P_{pub} = (x\rho)^{-1}(\rho \cdot G) - G$

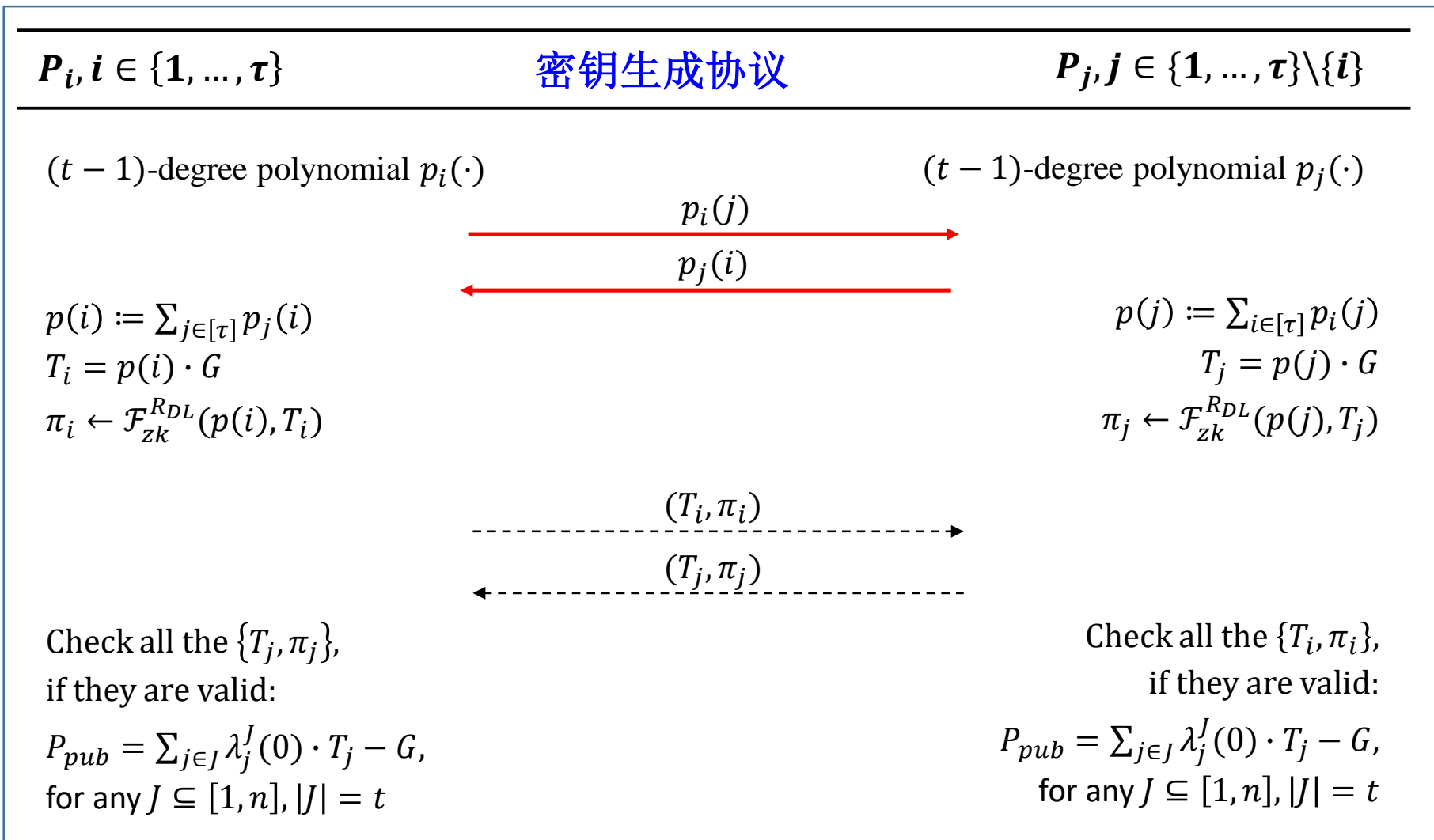
② 分布式签名: $k = \sum_{\ell=1}^{\tau} k_{\ell}$, $(r_x, r_y) \leftarrow R = k \cdot G$, $r = r_x + e \bmod n$, $s = x(k$

何德彪, 冯琦, 王婧, 林超, 张语荻, 张佳妮. 一种多方联合生成SM2数字签名的方法. 申请公开号: CN109547199A, 2019

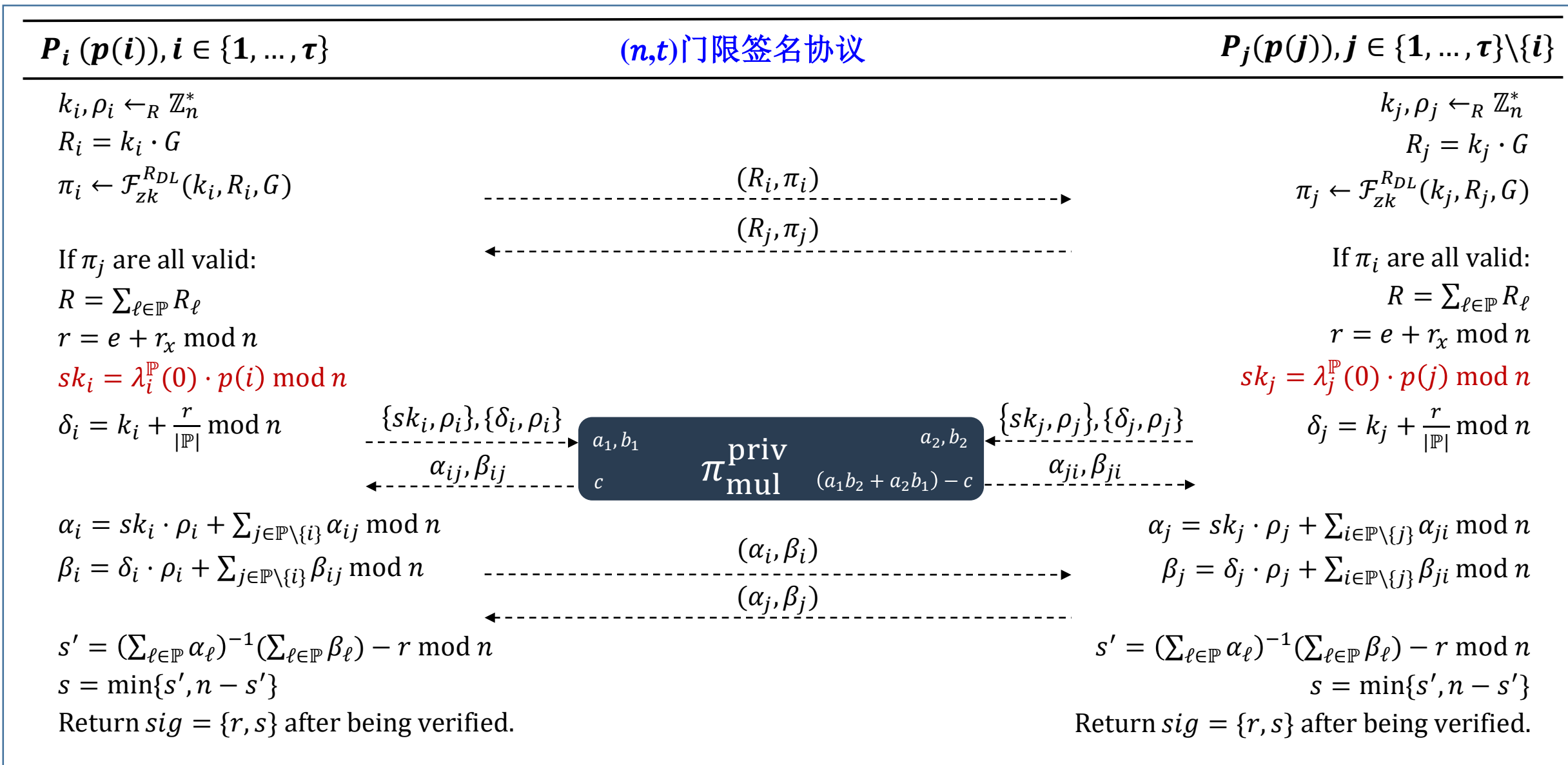
何德彪, 张语荻, 林超, 冯琦, 王婧, 张佳妮. 一种多方协同产生SM2数字签名的方法. 申请公开号: CN109474422A, 2019

11.6 针对SM2签名的协同签署方案

6. 门限协同签署方案[何德彪等'2019]



11.6 针对SM2签名的协同签署方案



目 录

- 11. 1. 密钥管理的概述
- 11. 2. 密钥分配与协商
- 11. 3. 公钥密码的密钥分配
- 11. 4. 移动互联网下的私钥安全分析
- 11. 5. 针对ECDSA的协同签署方案
- 11. 6. 针对SM2签名的协同签署方案
- 11. 7. 针对SM9签名的协同签署方案
- 11. 8. 密钥管理在区块链中的应用

11.6 针对SM9签名的协同签署方案

1. SM9签名算法回顾

➤ 系统参数生成

密钥生成中心(Key Generation Center, KGC)执行以下步骤生成系统参数和主私钥:

- ① KGC生成随机数 sk 做为主私钥, 这里 $0 < sk < q-1$;
- ② KGC计算系统公钥 $P_{pub}=sk \cdot P_2$;
- ③ KGC保存私钥 sk , 公布系统公钥.

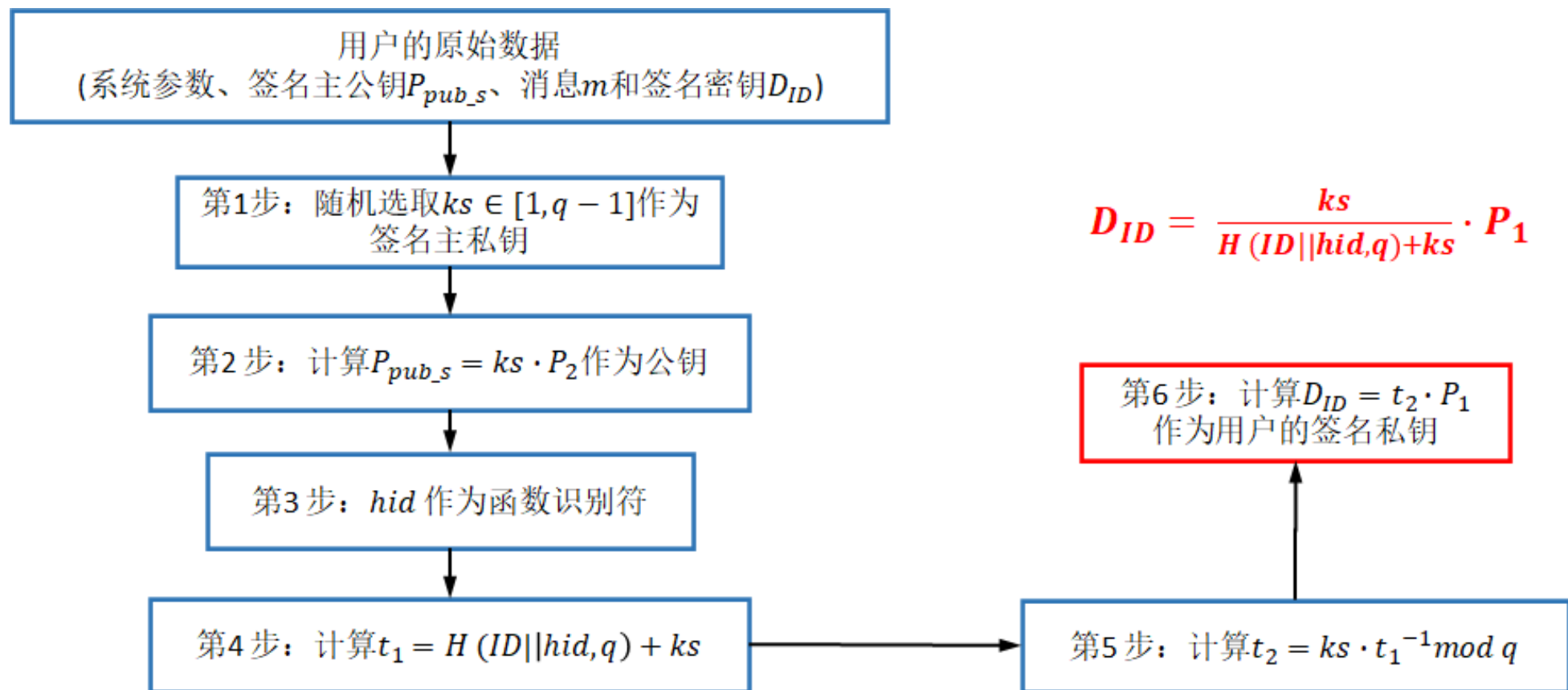
注意:

- ① SM9算使用BN曲线, G_1 和 G_2 分别是椭圆曲线 $E(F_p)$ 和 $E(F_{p^2})$ 的加法群, G_T 是乘法群 $F_{p^{12}}$, 群 G_2 中元素尺寸是群 G_1 中元素尺寸的2倍.
- ② 选择系统公钥为 G_2 中的元素, 那么就可以使得用户私钥和签名中一部分是 G_1 中元素, 降低了用户私钥和签名的尺寸.

11.6 针对SM9签名的协同签署方案

1. SM9签名算法回顾

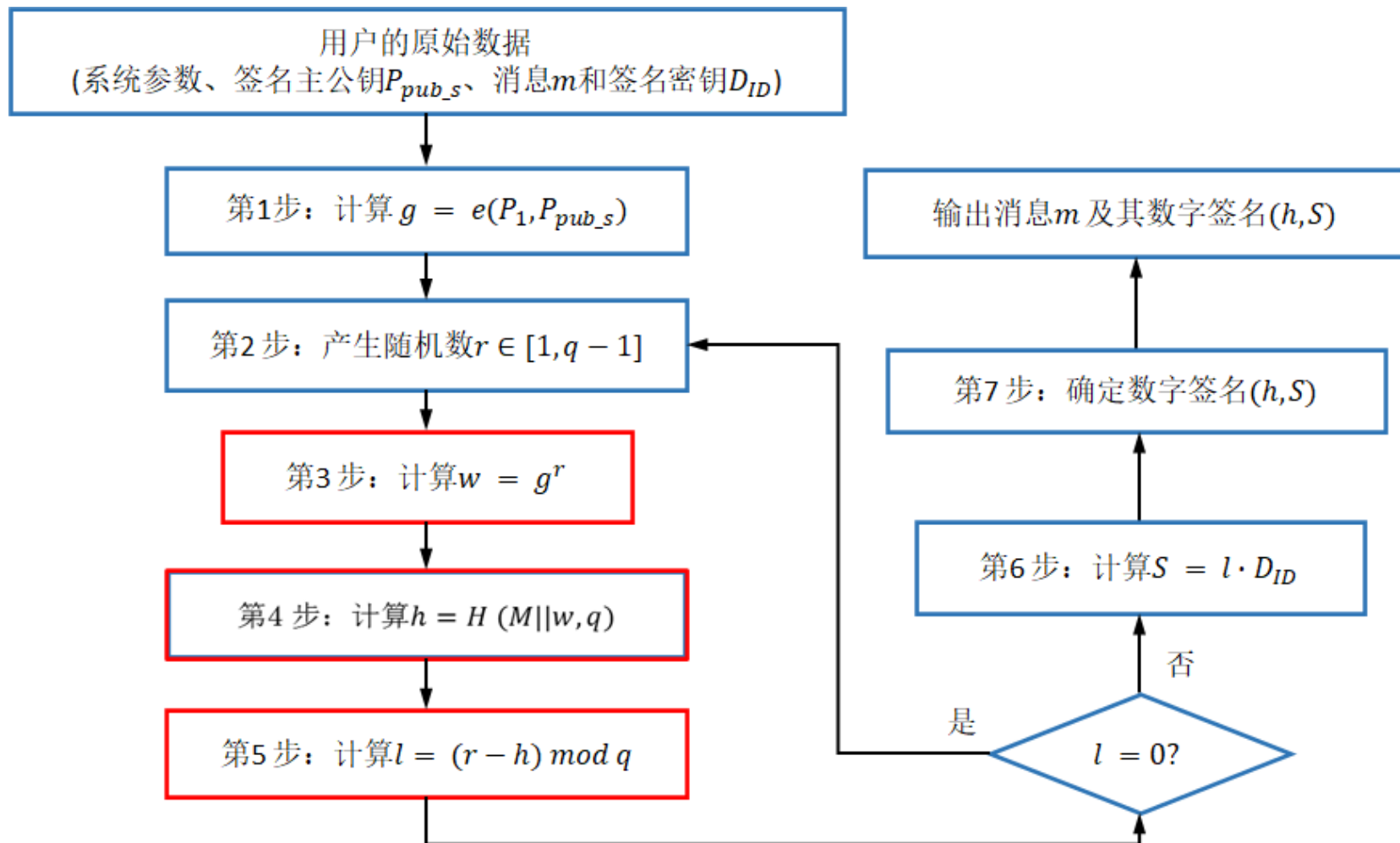
➤ 用户私钥生成



11.6 针对SM9签名的协同签署方案

1. SM9签名算法回顾

➤ 签名



11.6 针对SM9签名的协同签署方案

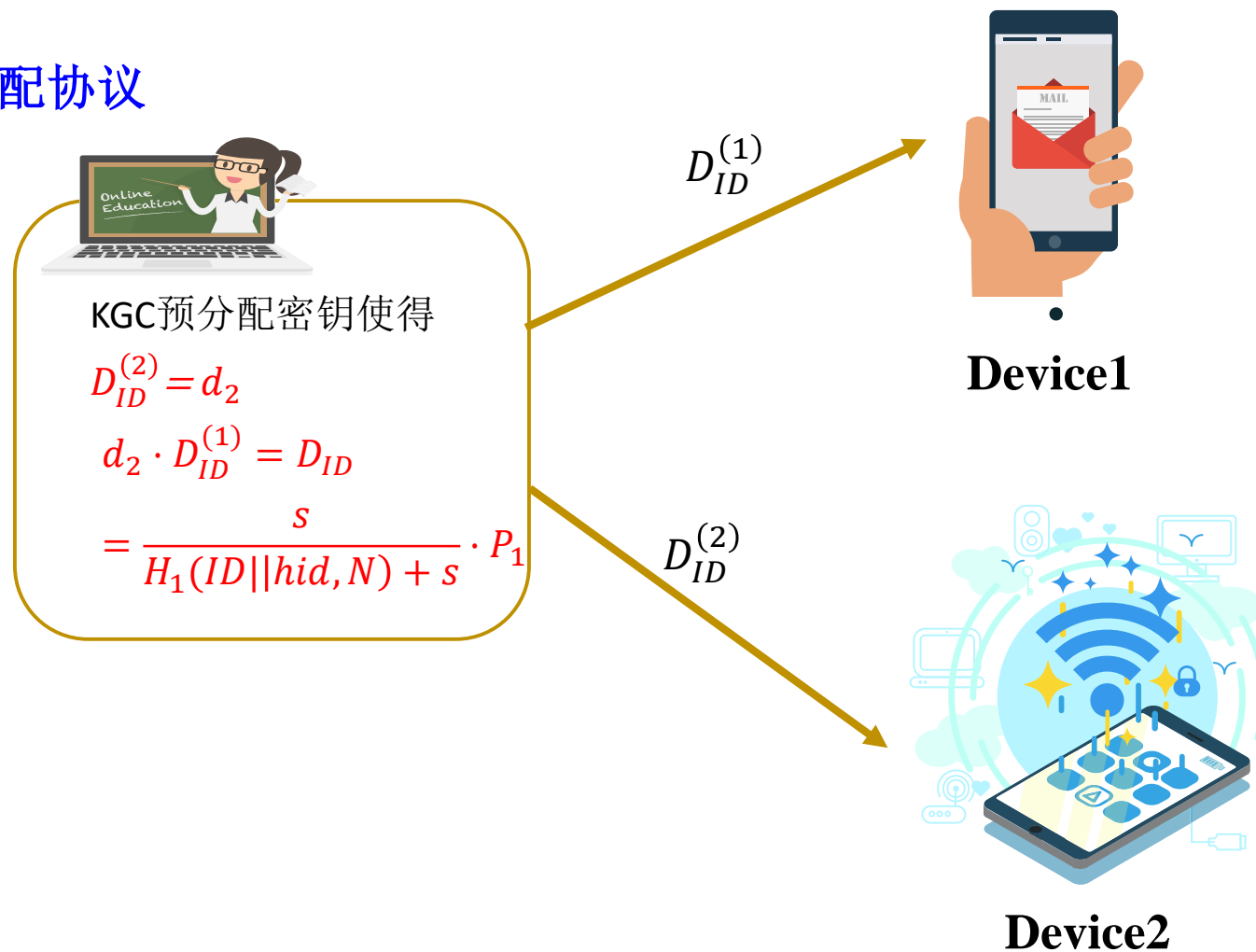
2. 两方协同签署方案[何德彪等'2017]



11.6 针对SM9签名的协同签署方案

2. 两方协同签署方案[何德彪等'2017]

➤ 密钥分配协议



11.6 针对SM9签名的协同签署方案

2. 两方协同签署方案[何德彪等'2017]

P1	两方协同签署协议	P2
compute $R_1 = g^{r_1}$ $C_1 = \text{Enc}(r_1)$	$\xrightarrow{R_1, C_1}$	Compute $R_2 = g^{r_2}$ $R = R_1^{r_2} = g^{r_1 r_2}$ compute $C_2 = (C_1^{r_2} \cdot \text{Enc}(\rho \cdot q - h))^{D_{ID}^{(2)}}$ $= \text{Enc}((r_1 r_2 - h + \rho \cdot q) \cdot d_2)$
$s = \text{Dec}(C_2) \bmod q$ $= (r_1 r_2 - h + \rho \cdot q) \cdot d_2 \bmod q$ $= (r_1 r_2 - h) \cdot d_2$ compute $S = s \cdot D_{ID}^{(1)}$ verify and output (h, S)	$\xleftarrow{R_2, C_2}$	

11.6 针对SM9签名的协同签署方案

2. 两方协同签署方案[何德彪等'2017]

➤ 正确性分析

密钥分发:

$$d_{ID}^{(2)} \cdot D_{ID}^{(1)} = D_{ID} = \frac{s}{H_1(ID||hid,N)+s} \cdot P_1$$

签名:

$$R = g^{r_1 r_2}$$

$$l = H_2(M||R, N) \bmod N$$

$$\begin{aligned} S &= \text{Dec} \left(\left(C_1^{r_2} \cdot \text{Enc}(\rho \cdot N - l) \right)^{d_{ID}^{(2)}} \right) \cdot D_{ID}^{(1)} \\ &= \text{Dec} \left(\text{Enc} \left((r_1 r_2 + \rho \cdot N - l) \cdot d_{ID}^{(2)} \right) \right) \cdot D_{ID}^{(1)} \\ &= (r_1 r_2 - l) \cdot (d_{ID}^{(2)} \cdot D_{ID}^{(1)}) = (r_1 r_2 - l) \cdot D_{ID} \end{aligned}$$

密钥分发:

$$D_{ID} = \frac{s}{H_1(ID||hid,N)+s} \cdot P_1$$

签名:

$$R = g^r$$

$$l = H_2(M||R, N) \bmod N$$

$$S = (r - l) \cdot D_{ID}$$

11.6 针对SM9签名的协同签署方案

2. 两方协同签署方案[何德彪等'2017]

➤ 安全性分析

定理： 如果Paillier加密方案是IND-CPA安全的，SM9签名在CMA下是存在不可伪造的，那么我们的两方分布式SM9签名协议也是安全的。

➤ 性能分析

	KeyGen	Sign	Verify
Mobile Device	851.71	1029.17	513.7
PC	174.55	324.87	152.6

Sign	Step1	Step2	Step3
Mobile Device	272.32	505.52	251.33
PC	89.69	155.4	79.78

备注：

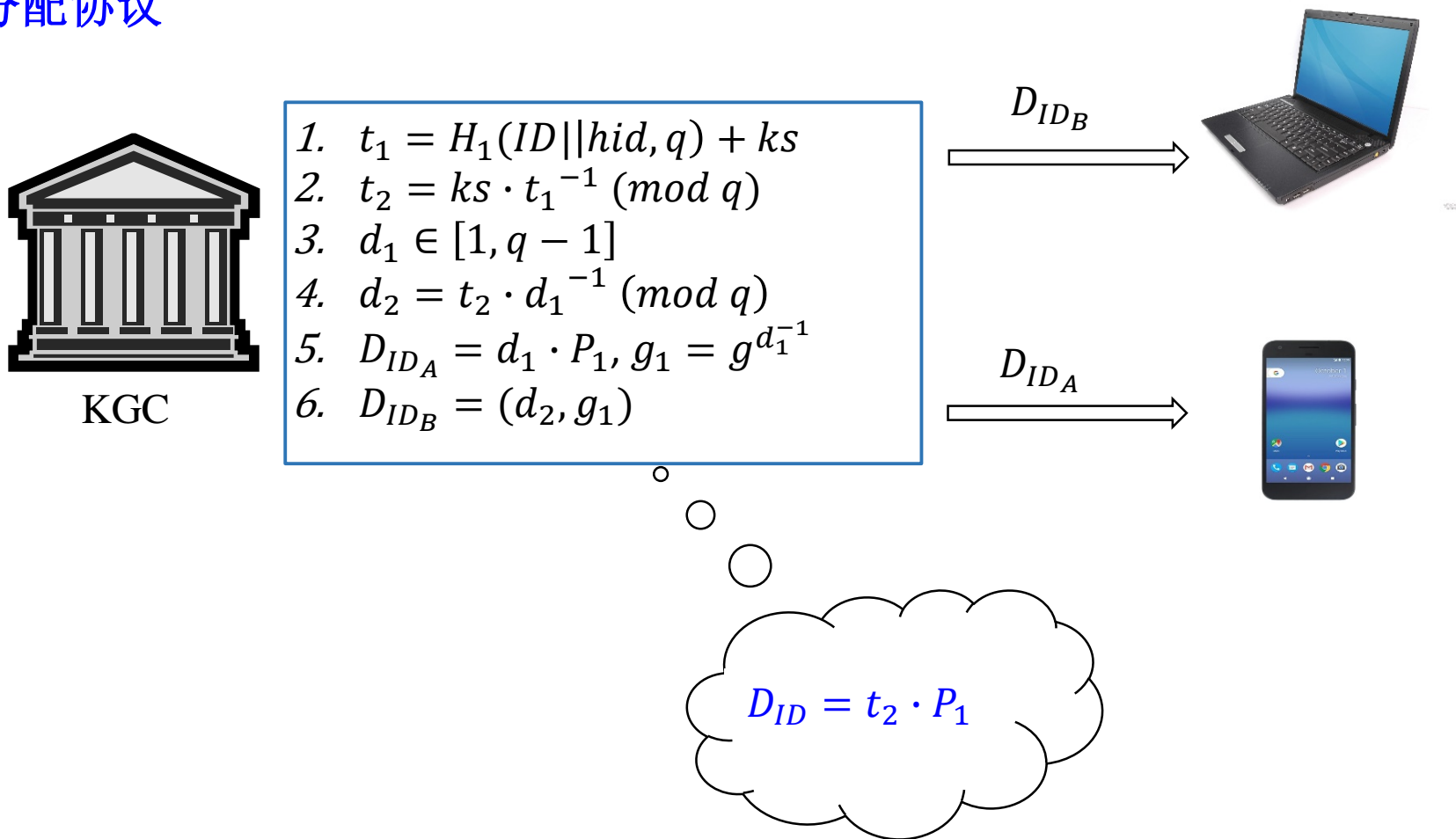
Mobile Device: 三星 Galaxy Nexus, 双核处理器, 2GB RAM, Android 4.4.2 10次(双线性对运算占用内存太大)

PC: 戴尔, Intel(R) Core(TM) i7-6700处理器, 3.40GHz, 8GB RAM Windows 10 专业版

11.6 针对SM9签名的协同签署方案

3. 轻量级两方协同签署方案[何德彪等'2018]

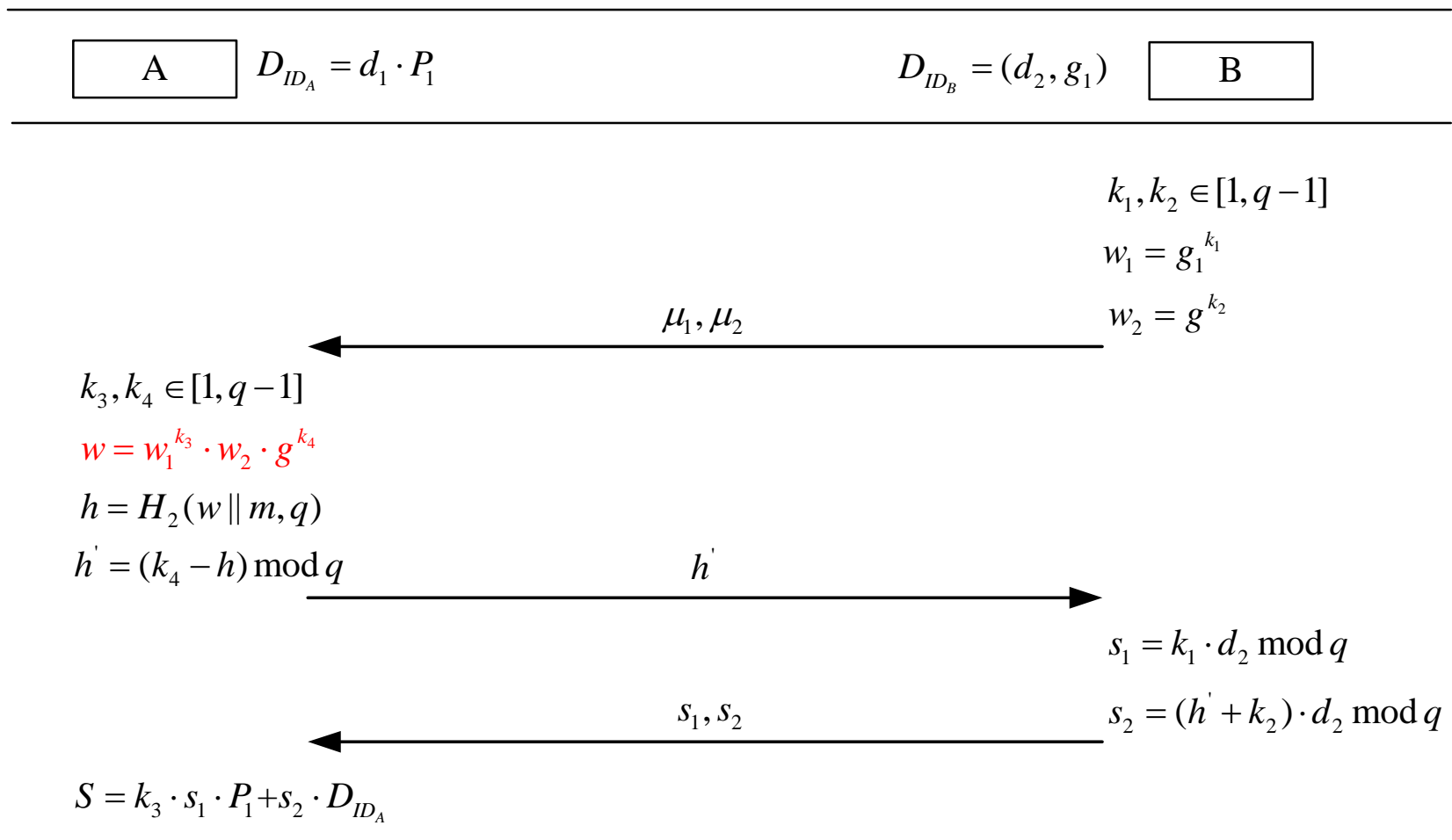
➤ 密钥分配协议



11.6 针对SM9签名的协同签署方案

3. 轻量级两方协同签署方案[何德彪等'2018]

➤ 协同签署方案

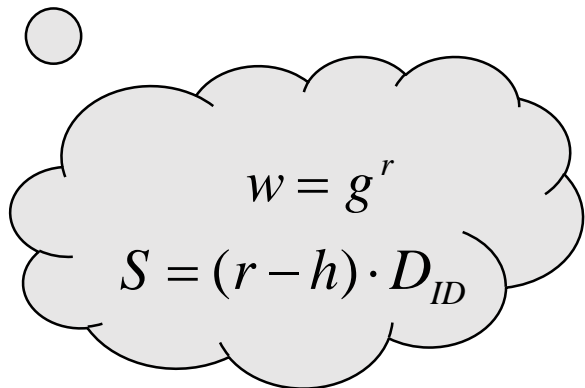


11.6 针对SM9签名的协同签署方案

3. 轻量级两方协同签署方案[何德彪等'2018]

➤ 方案正确性分析

$$\begin{aligned} W &= w_1^{k_3} \cdot w_2 \cdot g^{k_4} = g_1^{k_1 \cdot k_3} \cdot g^{k_2} \cdot g^{k_4} \\ &= g^{k_1 \cdot k_3 \cdot d_1^{-1}} \cdot g^{k_2} \cdot g^{k_4} \\ &= g^{k_1 \cdot k_3 \cdot d_1^{-1} + k_2 + k_4} \end{aligned}$$



$w = g^r$
 $S = (r - h) \cdot D_{ID}$

$$\begin{aligned} S &= k_3 \cdot s_1 \cdot P_1 + s_2 \cdot D_{ID_A} \\ &= k_3 \cdot k_1 \cdot d_2 \cdot P_1 + (h' + k_2) \cdot d_2 \cdot D_{ID_A} \\ &= k_1 \cdot k_3 \cdot d_2 \cdot P_1 + (k_4 - h + k_2) \cdot d_1 \cdot d_2 \cdot P_1 \\ &= (k_1 \cdot k_3 \cdot d_1^{-1} + k_2 + k_4 - h) \cdot (d_1 \cdot d_2 \cdot P_1) \\ &= (k_1 \cdot k_3 \cdot d_1^{-1} + k_2 + k_4 - h) \cdot \frac{ks}{H_1(ID || hid, q) + ks} \\ &= (k_1 \cdot k_3 \cdot d_1^{-1} + k_2 + k_4 - h) \cdot D_{ID} \end{aligned}$$

11.6 针对SM9签名的协同签署方案

3. 轻量级两方协同签署方案[何德彪等'2018]

➤ 方案安全性分析

定义. $k - CAA$ 难题: 给定 $e_1, e_2, \dots, e_k \in Z_q^*$, $P, sP, \frac{1}{s+e_1}P, \frac{1}{s+e_2}P, \dots, \frac{1}{s+e_k}P \in G$, 不存

在一个概率多项式算法能以不可忽略的概率计算出 $\frac{1}{s+e}P$, 其中 $e \in Z_q^*$

定义. 离散对数(DL)难题: 给定 $P, xP \in G$, 不存在一个概率多项式算法能以不可忽略的概率计算出 x , 其中 $x \in Z_q^*$ 。

定理. 如果 $k - CAA$ 难题和离散对数(DL)难题是困难的, 那么我们的两方分布式SM9签名协议也是安全的。

11.6 针对SM9签名的协同签署方案

3. 轻量级两方协同签署方案[何德彪等'2018]

➤ 性能分析

	KeyGen	Sign	Verify
Mobile Device	420.93	552.23	517.11
PC	126.38	193.23	160.7

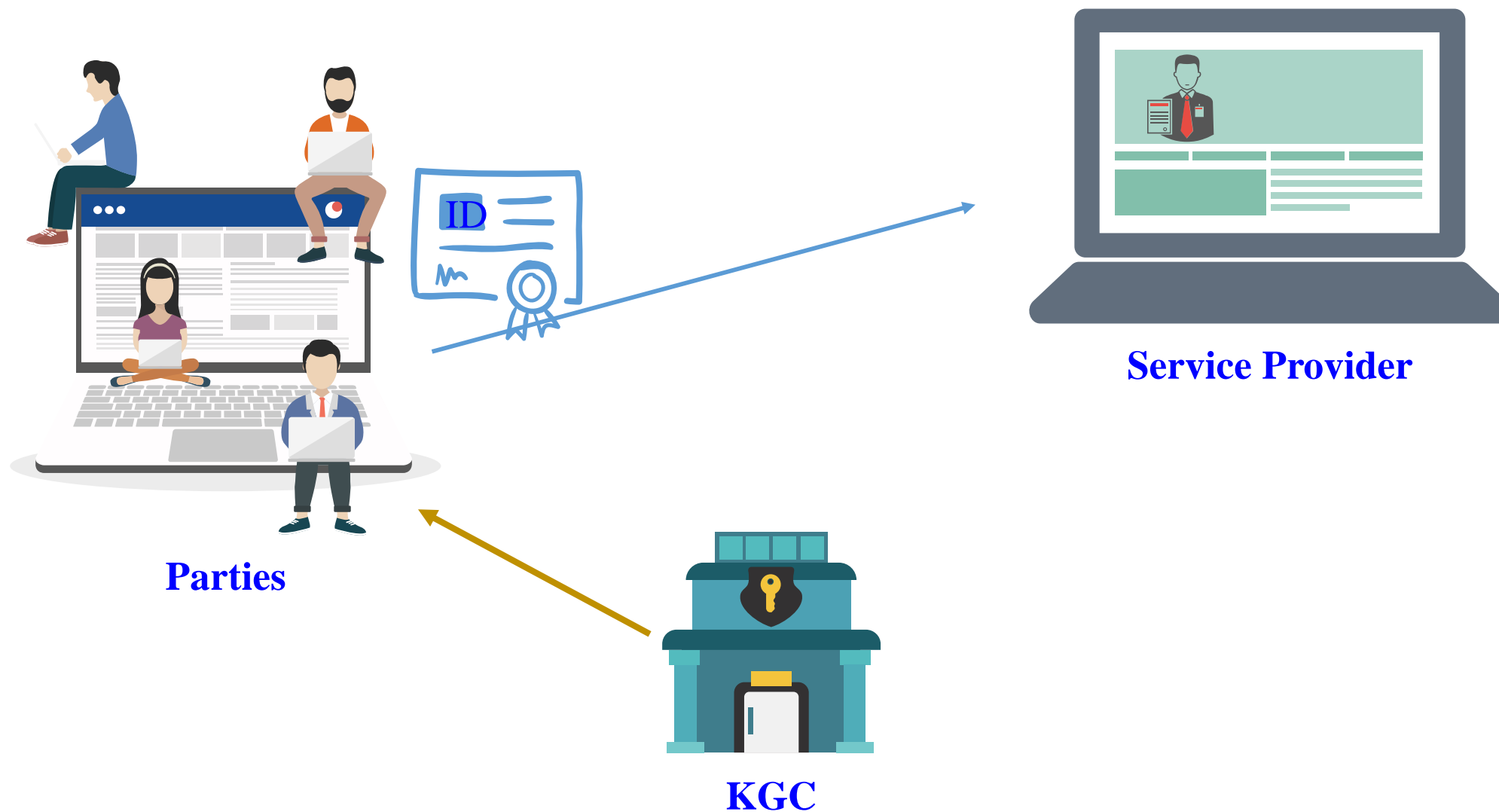
Sign	Step1	Step2	Step3	Step4
Mobile Device	264.86	255.34	0.08	31.95
PC	94.88	93.07	0.01	5.27

备注:

Mobile Device: 三星 Galaxy Nexus, 双核处理器, 2GB RAM, Android 4.4.2 10次(双线性对运算占用内存太大)

PC: 戴尔, Intel(R) Core(TM) i7-6700处理器, 3.40GHz, 8GB RAM Windows 10 专业版

11.6 针对SM9签名的协同签署方案



11.6 针对SM9签名的协同签署方案

4. 非对称多方协同签署方案[何德彪等'2019]

➤ 思路分析

密钥分发:

$$\left(\sum_{i=1}^{\tau} d_{ID}^{(i)}\right) \cdot D_{ID}^{(0)} = D_{ID} = \frac{s}{H_1(ID||hid,N)+s} \cdot P_1$$

签名:

$$R = g^{\sum_{i=1}^{\tau} r_i}$$

$$l = H_2(M||R, N) \bmod N$$

$$\begin{aligned} S &= \left(\sum_{i=1}^{\tau} r_i - l\right) \cdot \left(\sum_{i=1}^{\tau} d_{ID}^{(i)}\right) \cdot D_{ID}^{(0)} \\ &= \left(\sum_{i=1}^{\tau} \alpha_i\right) \cdot D_{ID}^{(0)} \end{aligned}$$

密钥分发:

$$D_{ID} = \frac{s}{H_1(ID||hid,N)+s} \cdot P_1$$

签名:

$$R = g^r$$

$$l = H_2(M||R, N) \bmod N$$

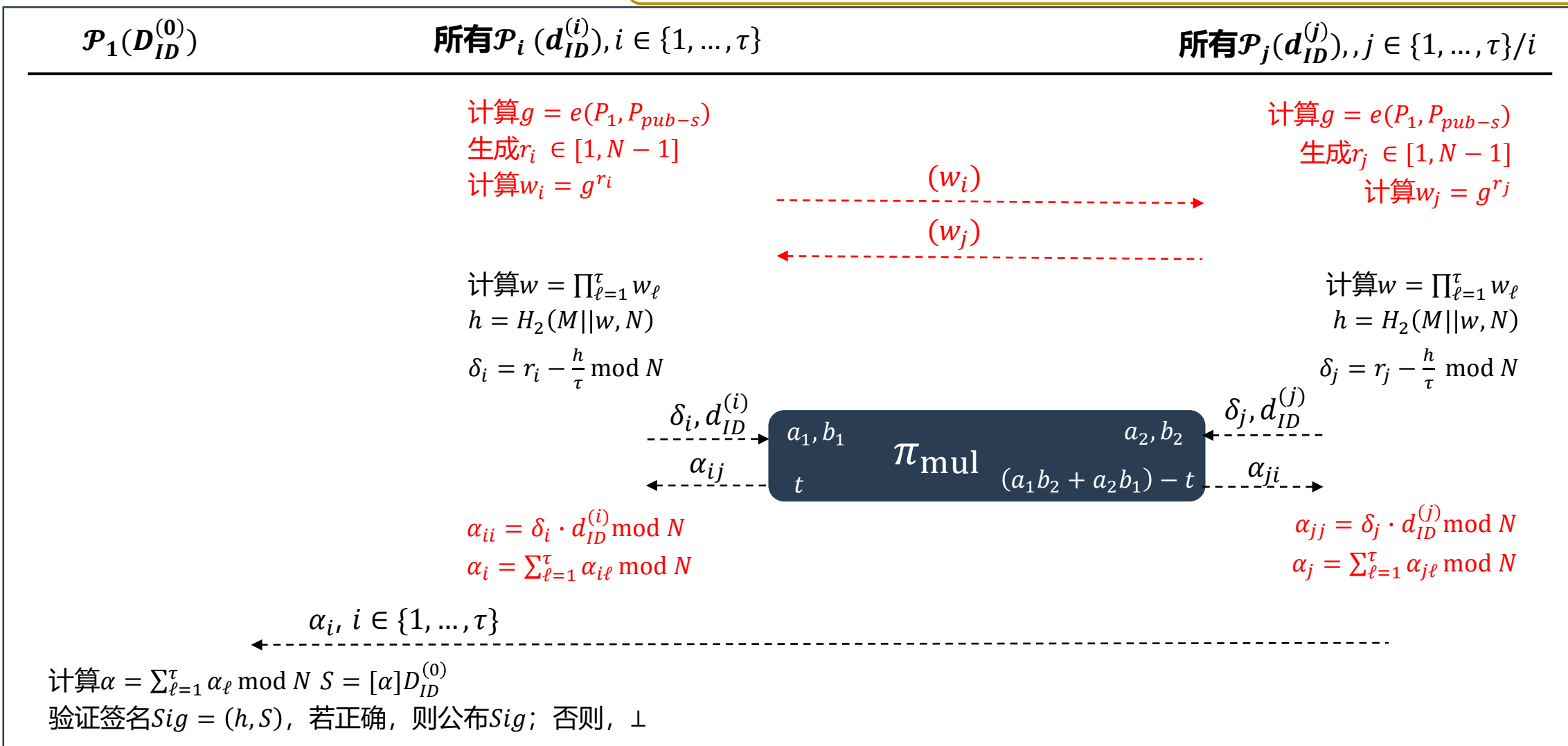
$$S = (r - l) \cdot D_{ID}$$

何德彪, 冯琦, 王婧, 周晓彤. 一种非对称环境下多方联合生成SM9数字签名的方法. 申请公开号: CN109194478A

11.6 针对SM9签名的协同签署方案



$$\left(\sum_{i=1}^{\tau} d_{ID}^{(i)}\right) \cdot D_{ID}^{(0)} = D_{ID}, P_1 \text{ 为主节点, 持有 } (d_{ID}^{(1)}, D_{ID}^{(0)})$$



11.6 针对SM9签名的协同签署方案

4. 非对称多方协同签署方案[何德彪等'2019]

➤ 正确性分析

密钥分发:

$$\left(\sum_{i=1}^{\tau} d_{ID}^{(i)}\right) \cdot D_{ID}^{(0)} = D_{ID} = \frac{s}{H_1(ID||hid,N)+s} \cdot P_1$$

签名:

$$R = g^{\sum_{i=1}^{\tau} r_i}$$

$$l = H_2(M||R, N) \bmod N$$

$$S = \left(\sum_{i=1}^{\tau} \alpha_i\right) \cdot D_{ID}^{(0)}$$

$$= \left(\sum_{i=1}^{\tau} \delta_i\right) \cdot \left(\sum_{i=1}^{\tau} d_{ID}^{(i)}\right) \cdot D_{ID}^{(0)}$$

$$= \left(\sum_{i=1}^{\tau} r_i - l\right) \cdot \left(\sum_{i=1}^{\tau} d_{ID}^{(i)}\right) \cdot D_{ID}^{(0)}$$

密钥分发:

$$D_{ID} = \frac{s}{H_1(ID||hid,N)+s} \cdot P_1$$

签名:

$$R = g^r$$

$$l = H_2(M||R, N) \bmod N$$

$$S = (r - l) \cdot D_{ID}$$

11.6 针对SM9签名的协同签署方案

4. 对称多方协同签署方案[何德彪等'2019]

➤ 思路分析

密钥分发:

$$\sum_{i=1}^{\tau} D_{ID}^{(i)} = D_{ID} = \frac{s}{H_1(ID||hid,N)+s} \cdot P_1$$

签名:

$$R = g^{\sum_{i=1}^{\tau} r_i}$$

$$l = H_2(M||R, N) \bmod N$$

$$S = (\sum_{i=1}^{\tau} r_i - l) \cdot (\sum_{i=1}^{\tau} D_{ID}^{(i)})$$

$\pi_{mul}^{\mathbb{G}_1}$ 乘法协议 —— $D_A + D_B = \delta_A \cdot D_{ID}^{(B)} + \delta_B \cdot D_{ID}^{(A)} \in \mathbb{G}_1$

密钥分发:

$$D_{ID} = \frac{s}{H_1(ID||hid,N)+s} \cdot P_1$$

签名:

$$R = g^r$$

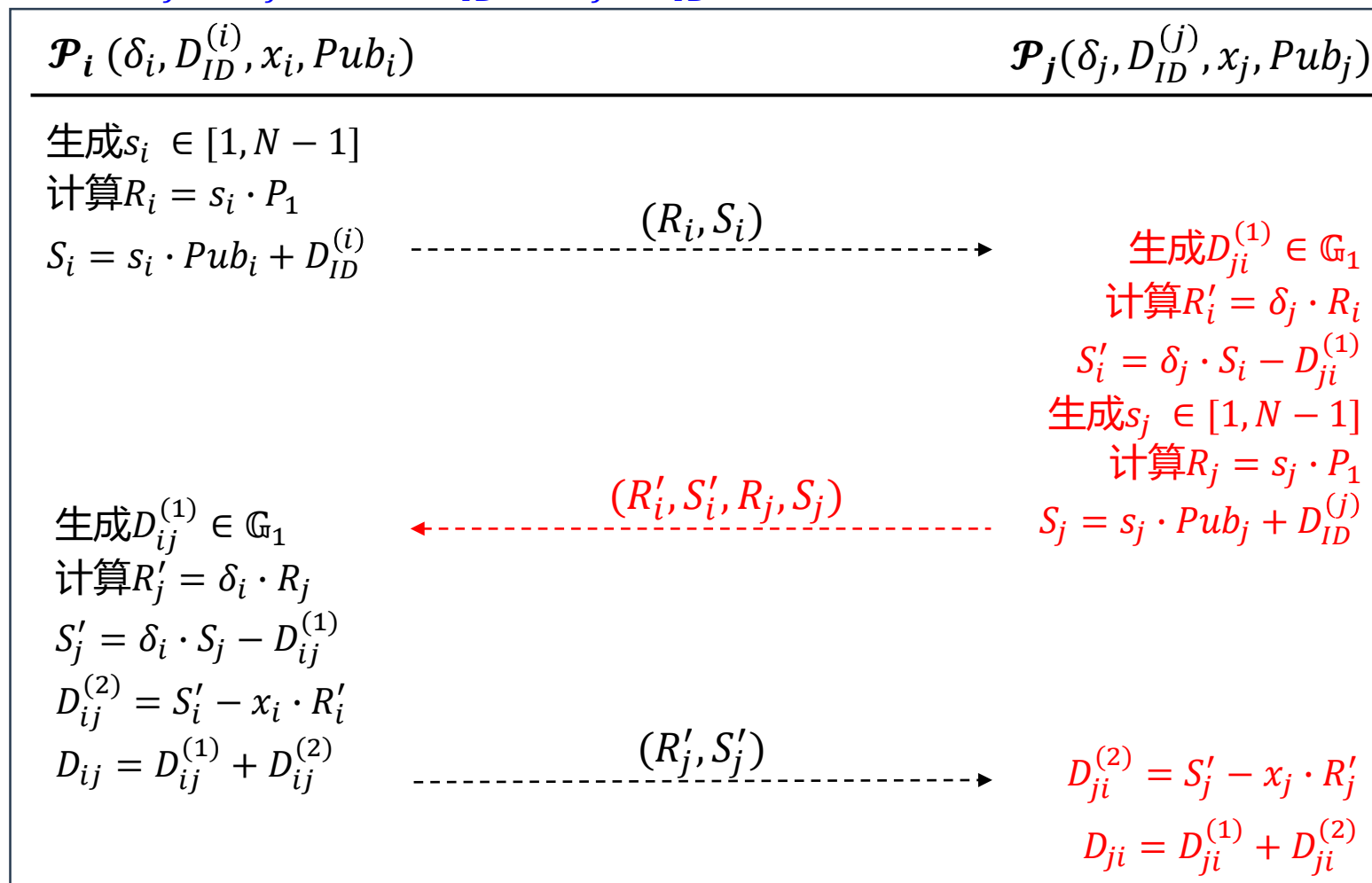
$$l = H_2(M||R, N) \bmod N$$

$$S = (r - l) \cdot D_{ID}$$

11.6 针对SM9签名的协同签署方案

4. 对称多方协同签署方案[何德彪等'2019]


➤ $\pi_{\text{mul}}^{\mathbb{G}_1}$ 乘法协议 —— $D_{ij} + D_{ji} = \delta_i \cdot D_{ID}^{(j)} + \delta_j \cdot D_{ID}^{(i)} \in \mathbb{G}_1$



11.6 针对SM9签名的协同签署方案

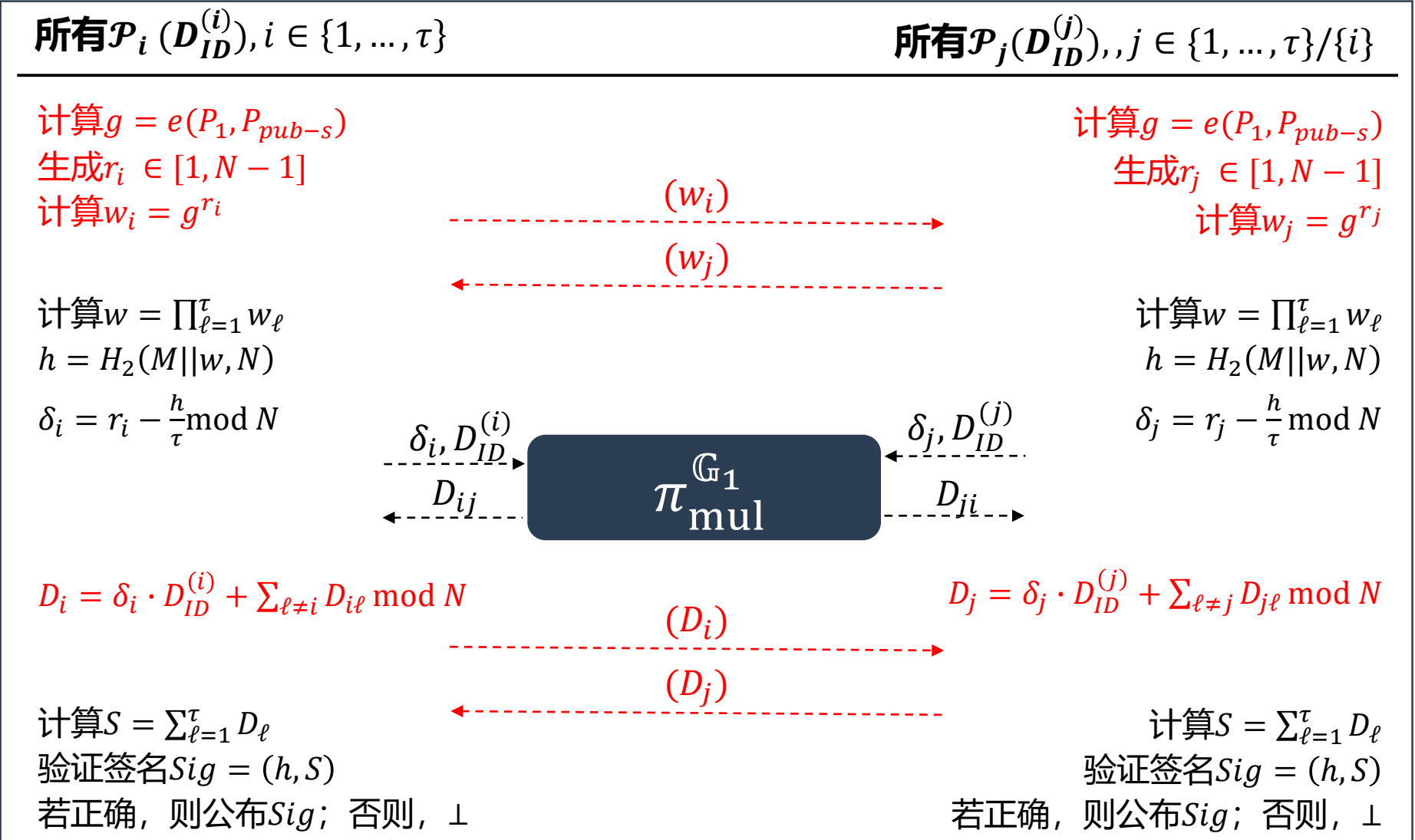
4. 对称多方协同签署方案[何德彪等'2019]

➤ 思路分析



$$- \sum_{i=1}^{\tau} D_{ID}^{(i)} = D_{ID}$$

$$- \pi_{mul}^{\mathbb{G}_1} \text{ 为 } \mathbb{G}_1 \text{ 群上的乘法器}$$



11.6 针对SM9签名的协同签署方案

4. 对称多方协同签署方案[何德彪等'2019]

➤ 正确性分析

密钥分发:

$$\sum_{i=1}^{\tau} D_{ID}^{(i)} = D_{ID} = \frac{s}{H_1(ID||hid,N)+s} \cdot P_1$$

签名:

$$R = g^{\sum_{i=1}^{\tau} r_i}$$

$$l = H_2(M||R, N) \bmod N$$

$$S = \sum_{i=1}^{\tau} D_i$$

$$= (\sum_{i=1}^{\tau} \delta_i) \cdot (\sum_{i=1}^{\tau} D_{ID}^{(i)})$$

$$= (\sum_{i=1}^{\tau} r_i - l) \cdot (\sum_{i=1}^{\tau} D_{ID}^{(i)})$$

密钥分发:

$$D_{ID} = \frac{s}{H_1(ID||hid,N)+s} \cdot P_1$$

签名:

$$R = g^r$$

$$l = H_2(M||R, N) \bmod N$$

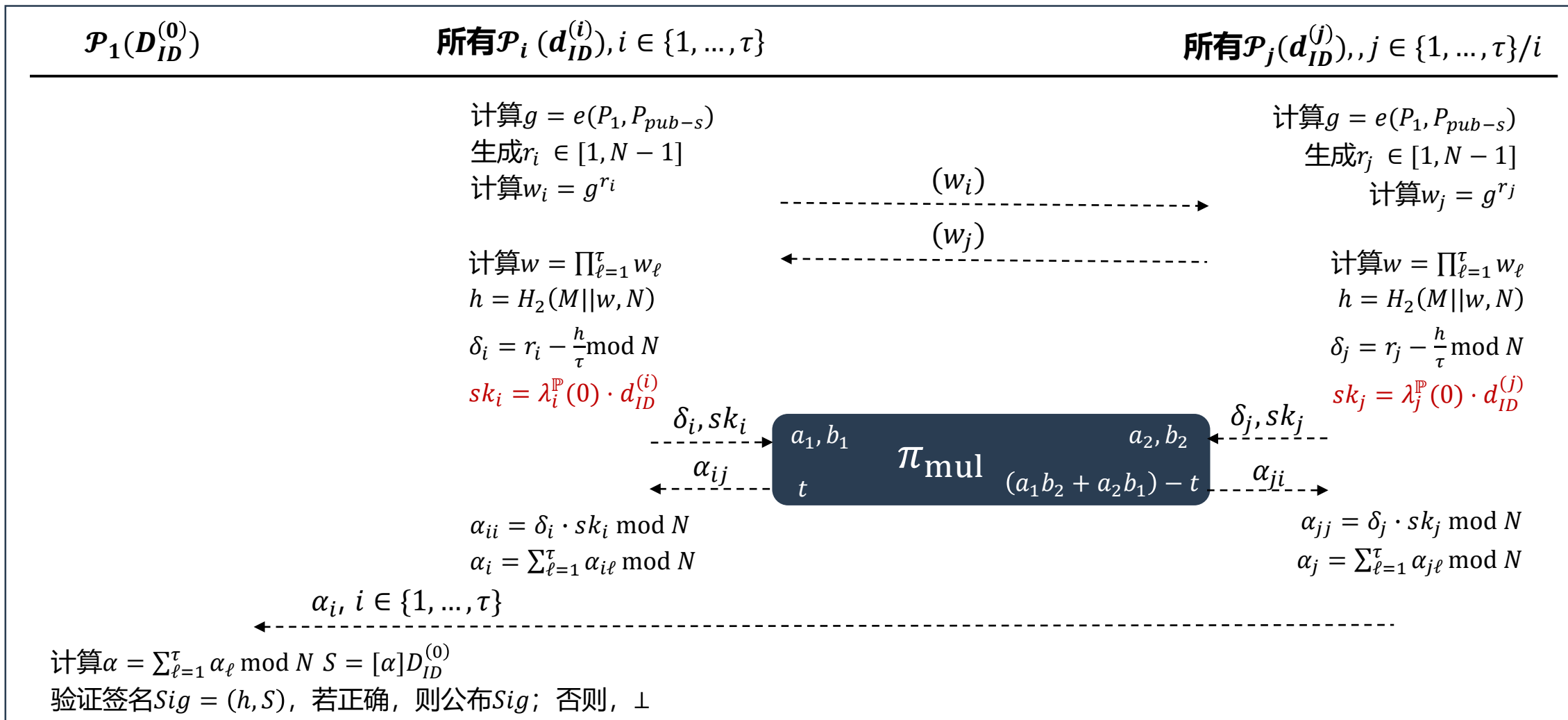
$$S = (r - l) \cdot D_{ID}$$

何德彪, 冯琦, 王婧, 林超, 张语荻. 一种对称环境下多方联合生成SM9数字签名的方法. 申请公开号: CN109660361A

11.6 针对SM9签名的协同签署方案



5. 非平衡 (n, t) 门限协同签署方案 $(\sum_{i=1}^t \lambda_i^{\mathbb{P}}(0) \cdot d_{ID}^{(i)}) \cdot D_{ID}^{(0)} = D_{ID}$, P_1 为主节点, 持有 $(d_{ID}^{(1)}, D_{ID}^{(0)})$

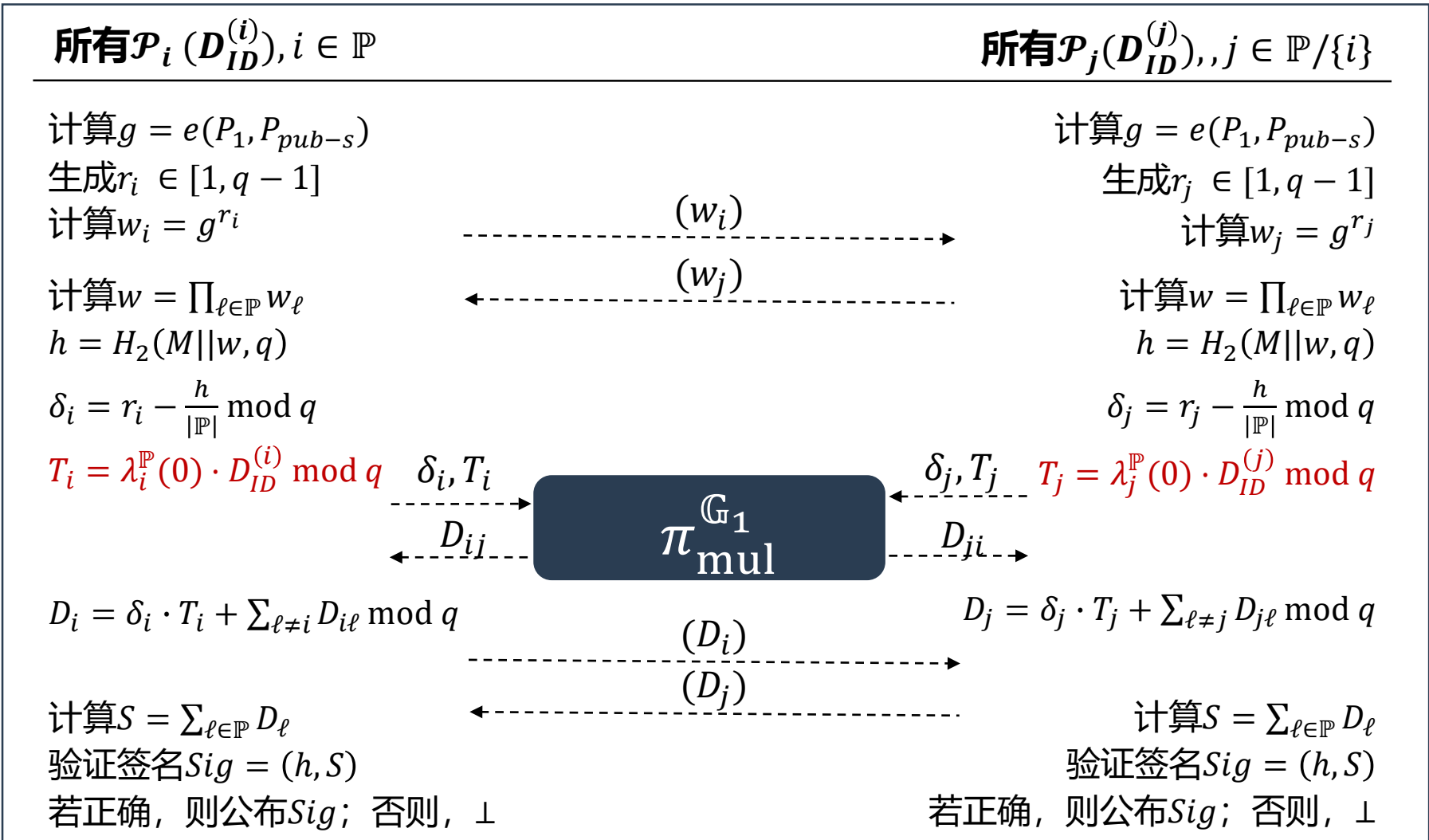


11.6 针对SM9签名的协同签署方案



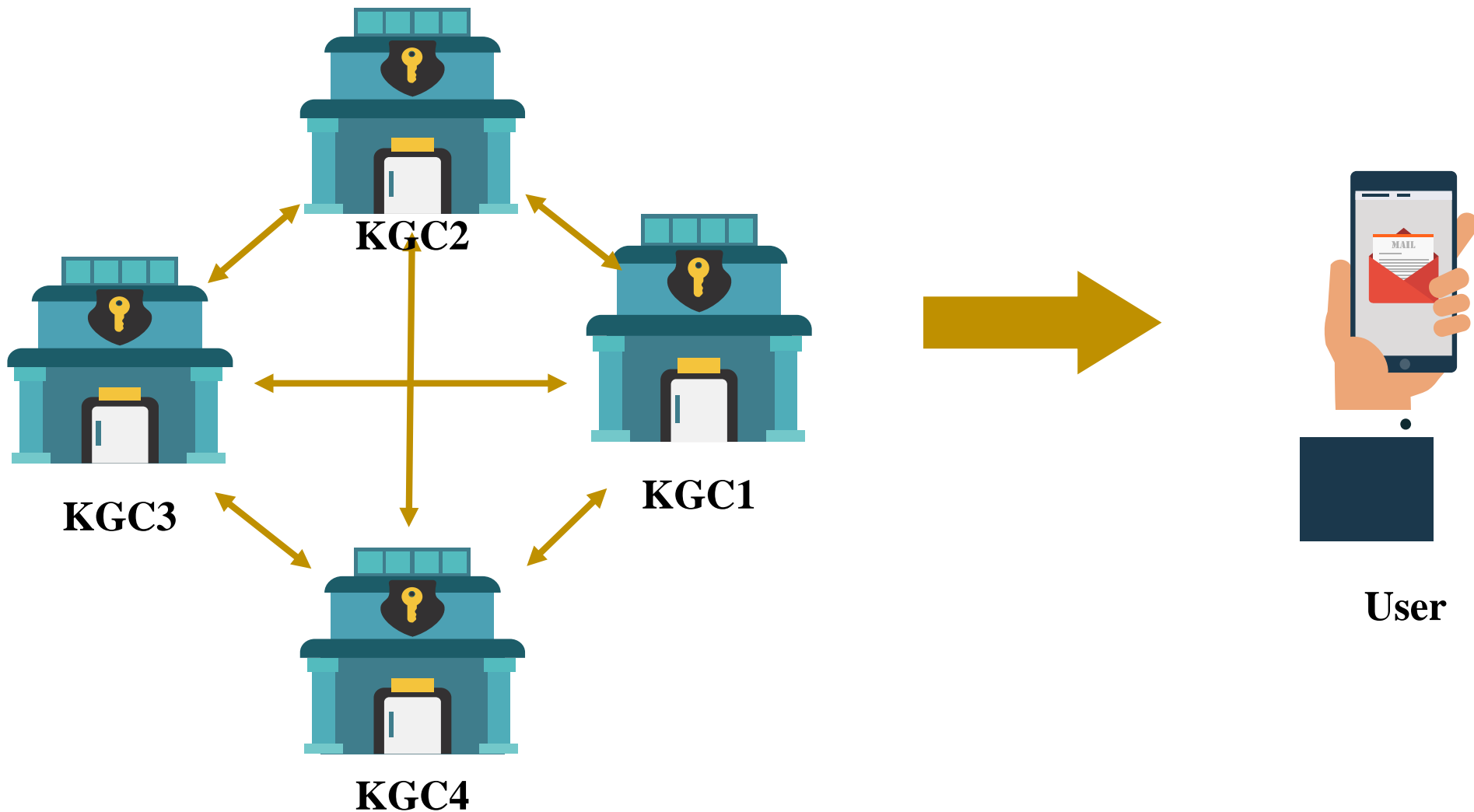
6. 平衡(n, t)门限协同签署方案

$$\sum_{i=1}^t \lambda_i^{\mathbb{P}}(0) \cdot D_{ID}^{(i)} = D_{ID}$$



11.6 针对SM9签名的协同签署方案

7. 多方协同密钥生成方案



11.6 针对SM9签名的协同签署方案

7. 多方协同密钥生成方案

➤ 思路分析

系统主密钥生成:

$$P_{pub-s} = (\sum_{i=1}^{\tau} s_i) \cdot P_2$$

用户密钥分发:

$$b = H_1(ID || hid, N) \bmod N$$

$$D_{ID} = (b + \sum_{i=1}^{\tau} s_i)^{-1} \cdot \sum_{i=1}^{\tau} s_i \cdot P_1$$

$$= ((b + \sum_{i=1}^{\tau} s_i) \cdot (\sum_{i=1}^{\tau} a_i))^{-1} (\sum_{i=1}^{\tau} a_i) (\sum_{i=1}^{\tau} s_i \cdot P_1)$$

$$= ((b + \sum_{i=1}^{\tau} s_i) \cdot (\sum_{i=1}^{\tau} a_i))^{-1} (\sum_{i=1}^{\tau} (a_i \cdot V))$$

预计算

系统主密钥生成:

$$P_{pub-s} = s \cdot P_2$$

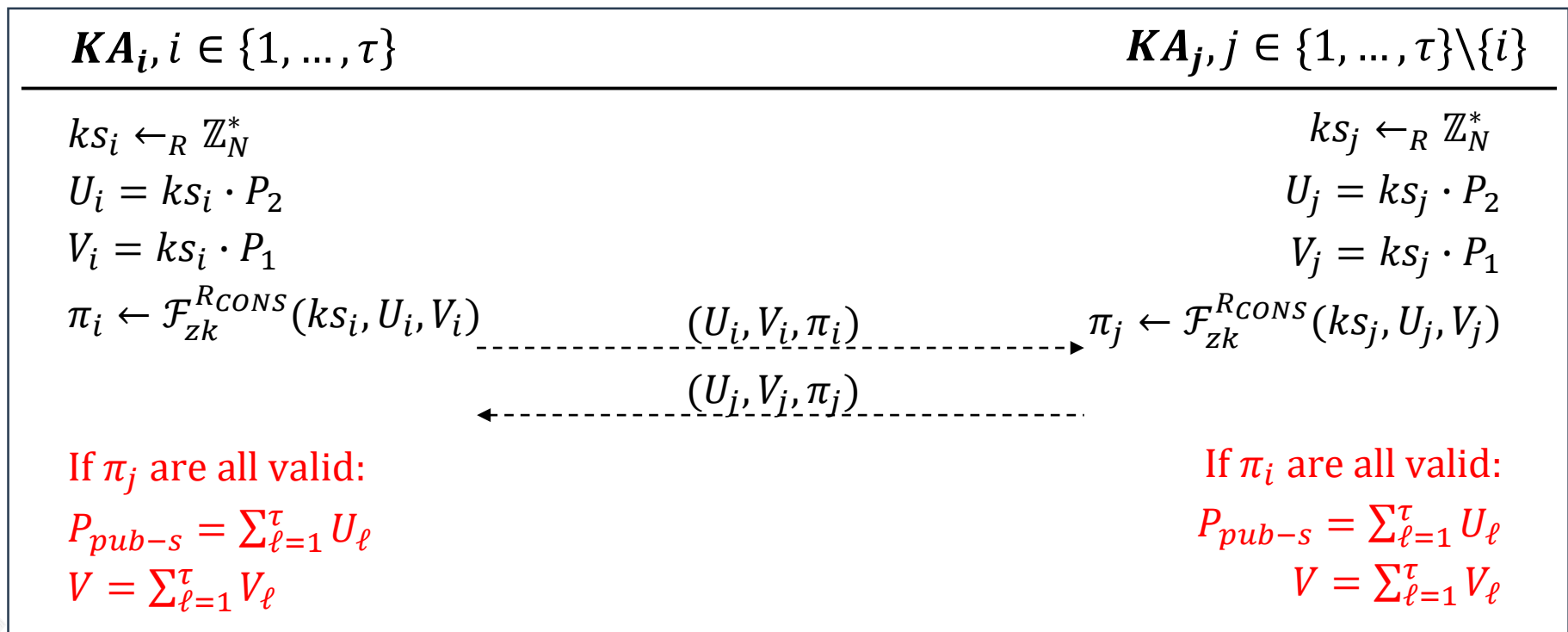
用户密钥分发:

$$D_{ID} = \frac{s}{H_1(ID || hid, N) + s} \cdot P_1$$

11.6 针对SM9签名的协同签署方案

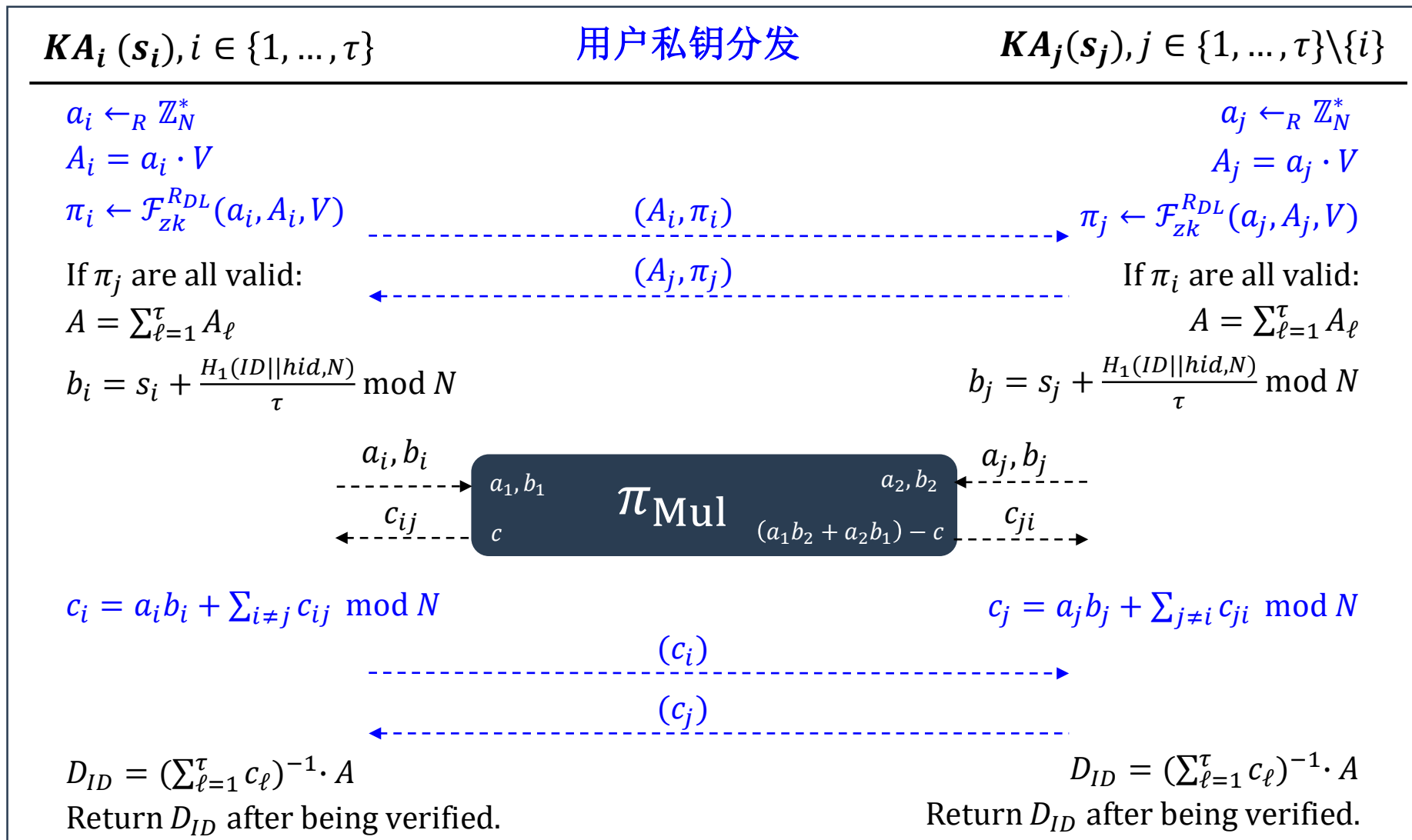
7. 多方协同密钥生成方案

➤ 系统主密钥生成协议



$$\pi \leftarrow \mathcal{F}_{zk}^{RCONS}(s, U, V), s.t., U = s \cdot P_2, V = s \cdot P_1$$

11.6 针对SM9签名的协同签署方案



11.6 针对SM9签名的协同签署方案

7. 多方协同密钥生成方案

► 正确性分析

系统主密钥生成:

$$P_{pub-s} = (\sum_{i=1}^{\tau} U_i) = (\sum_{i=1}^{\tau} ks_i) \cdot P_2$$
$$V = (\sum_{i=1}^{\tau} V_i) = (\sum_{i=1}^{\tau} ks_i) \cdot P_1$$

用户密钥分发:

$$b_i = ks_i + \frac{H_1(ID||hid,N)}{\tau} \bmod N$$
$$D_{ID} = (\sum_{\ell=1}^{\tau} c_{\ell})^{-1} \cdot A$$
$$= ((\sum_{i=1}^{\tau} b_i) \cdot (\sum_{i=1}^{\tau} a_i))^{-1} (\sum_{i=1}^{\tau} a_i \cdot V)$$
$$= (\sum_{i=1}^{\tau} ks_i + H_1(ID||hid,N))^{-1} \cdot V$$
$$= (\sum_{i=1}^{\tau} ks_i + H_1(ID||hid,N))^{-1} \cdot (\sum_{i=1}^{\tau} ks_i) \cdot P_1$$

系统主密钥生成:

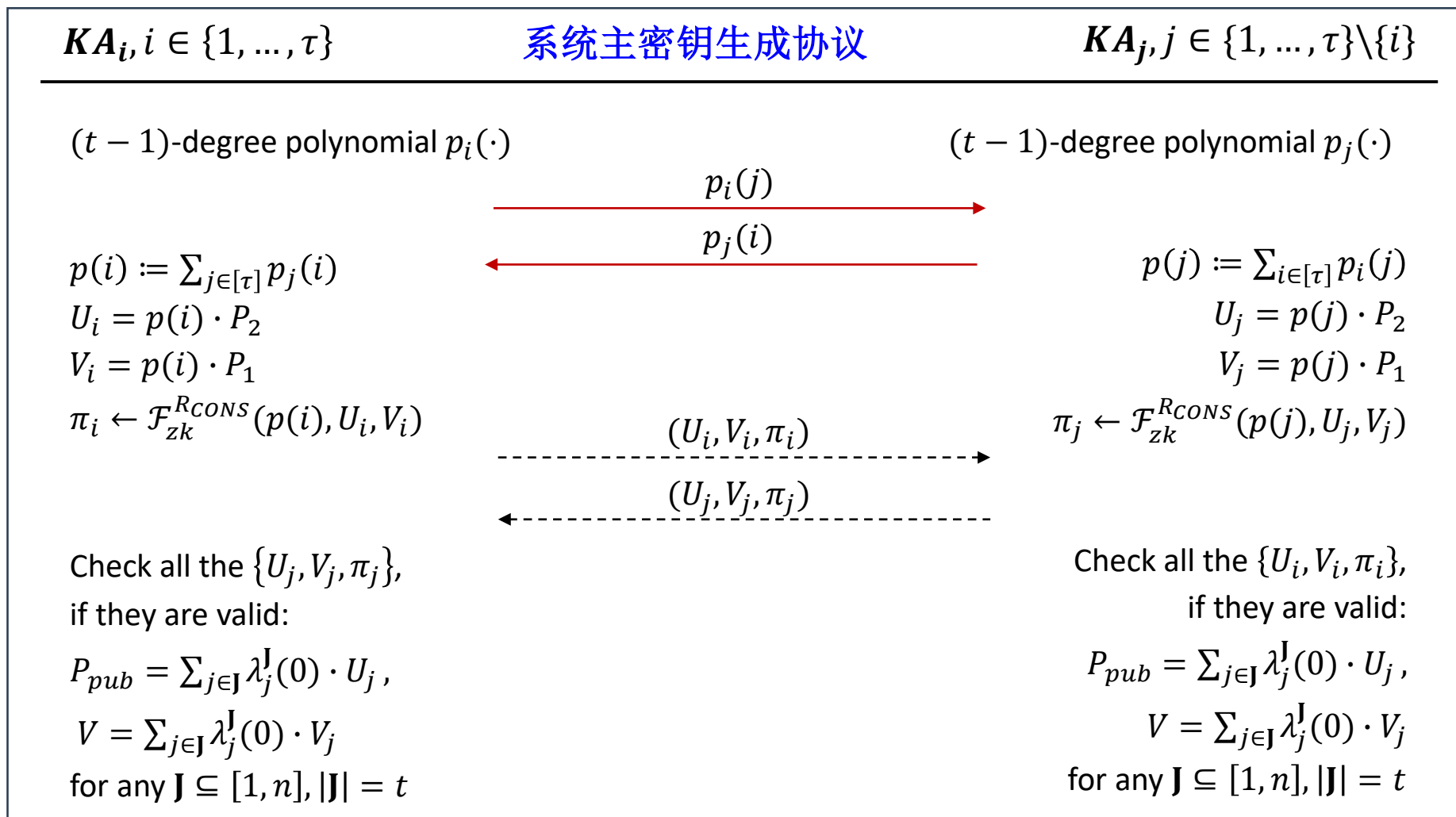
$$P_{pub-s} = s \cdot P_2$$

用户密钥分发:

$$D_{ID} = \frac{s}{H_1(ID||hid,N)+s} \cdot P_1$$

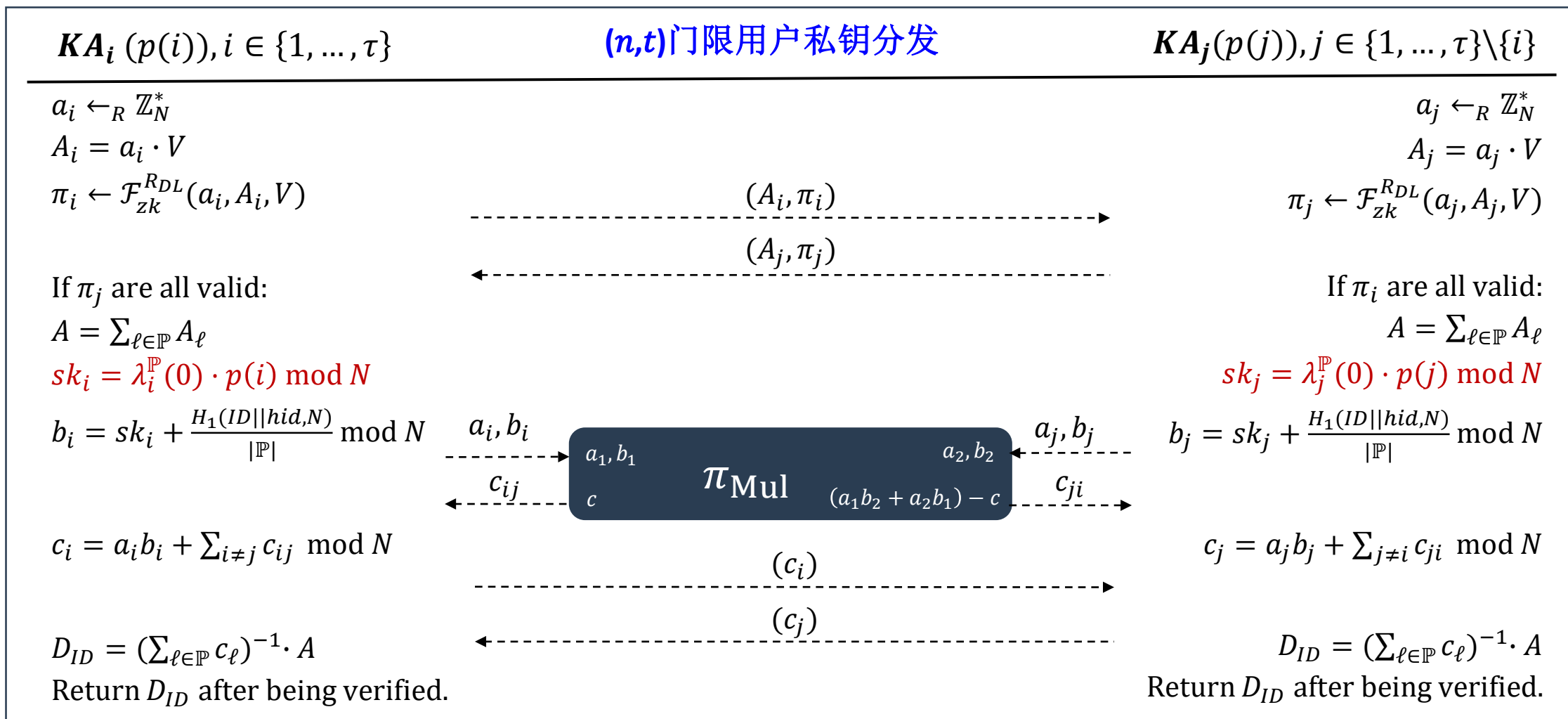
11.6 针对SM9签名的协同签署方案

8. (n, t) 门限多方协同密钥生成方案



11.6 针对SM9签名的协同签署方案

8. (n, t) 门限多方协同密钥生成方案



目 录

- 11. 1. 密钥管理的概述
- 11. 2. 密钥分配与协商
- 11. 3. 公钥密码的密钥分配
- 11. 4. 移动互联网下的私钥安全分析
- 11. 5. 针对ECDSA的协同签署方案
- 11. 6. 针对SM2签名的协同签署方案
- 11. 7. 针对SM9签名的协同签署方案
- 11. 8. 密钥管理在区块链中的应用

11.8. 密钥管理在区块链中的应用

1. 公钥的管理

➤ 联盟链和私链

- ✓ 用户随机生成私钥，并计算对应的公钥
- ✓ 利用PKI管理公钥

➤ 公链

- ✓ 用户随机生成私钥，并计算对应的公钥
- ✓ 映射成地址

11.8. 密钥管理在区块链中的应用

2. 私钥的管理

- 用户随机生成私钥
- 钱包存储密钥
 - ✓ 热钱包
 - ✓ 冷钱包
- 普通设备存储密钥
 - ✓ 协同签名
 - ✓ 白盒密码

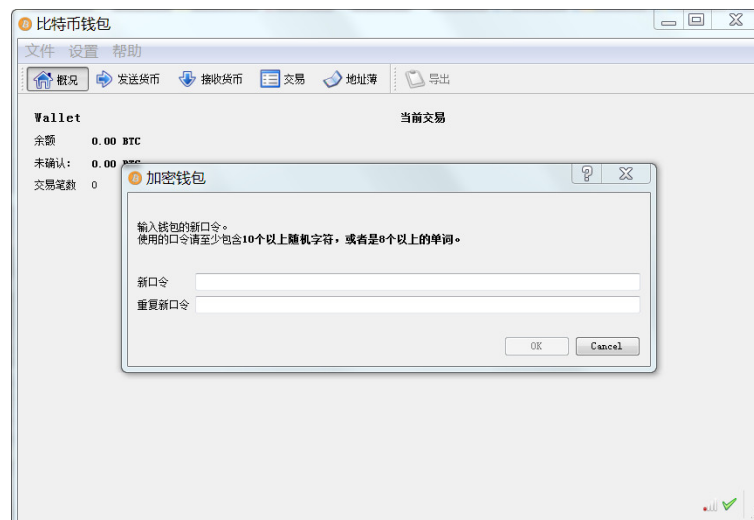


图11.6. 热钱包示意图



图11.7. 冷钱包示意图

11.8. 密钥管理在区块链中的应用



AWS Key Management Service

概览 功能 定价 入门 资源 常见问题

AWS Key Management Service (KMS)

轻松创建和控制用于加密数据的密钥

开始使用 AWS Key Management Service



云市场 开发者 支持 合作与生态 客户

密钥管理服务 KMS

安全、易用的密钥管理服务，轻松创建和管理加密数据的密钥

立即申请



Alibaba Cloud | Worldwide Cloud Services Partner

为何选择阿里云 产品 解决方案

Alibaba Cloud > 产品 > 密钥管理服务

密钥管理服务

根据需要创建、删除和管理加密密钥

免费开通 联系销售



谢谢!

