数字货币和区块链
- 中心化数字货币

山东大学网络空间安全学院

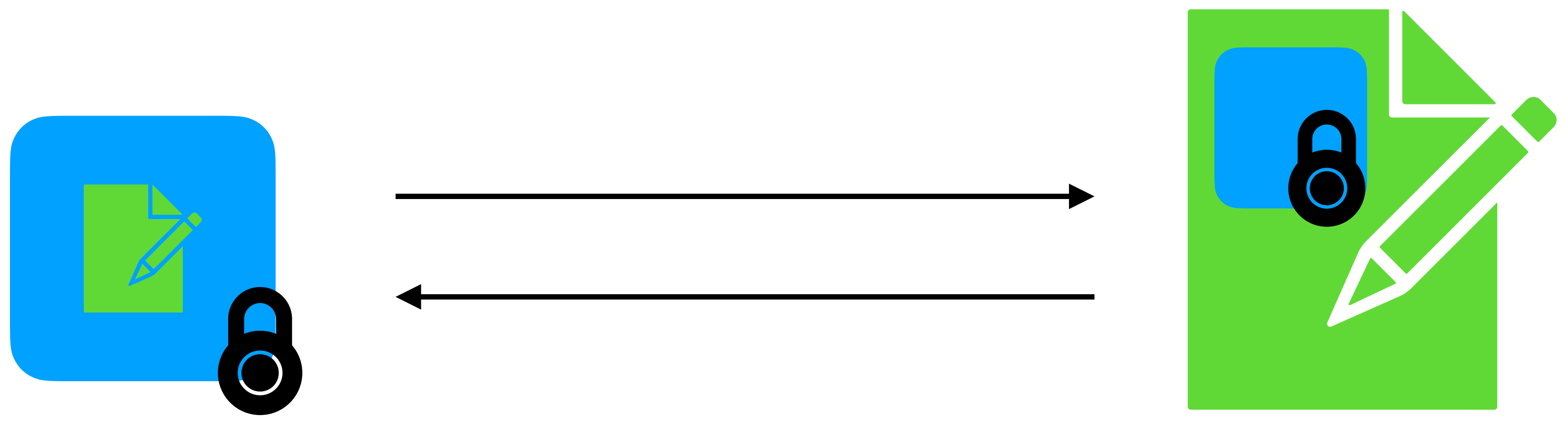
钱宸 2023年11月15日

数字货币

- 中心化数字货币定义与安全性
- 基于离散对数的数字货币
- 可传递数字货币定义
- 可传递数字货币构造

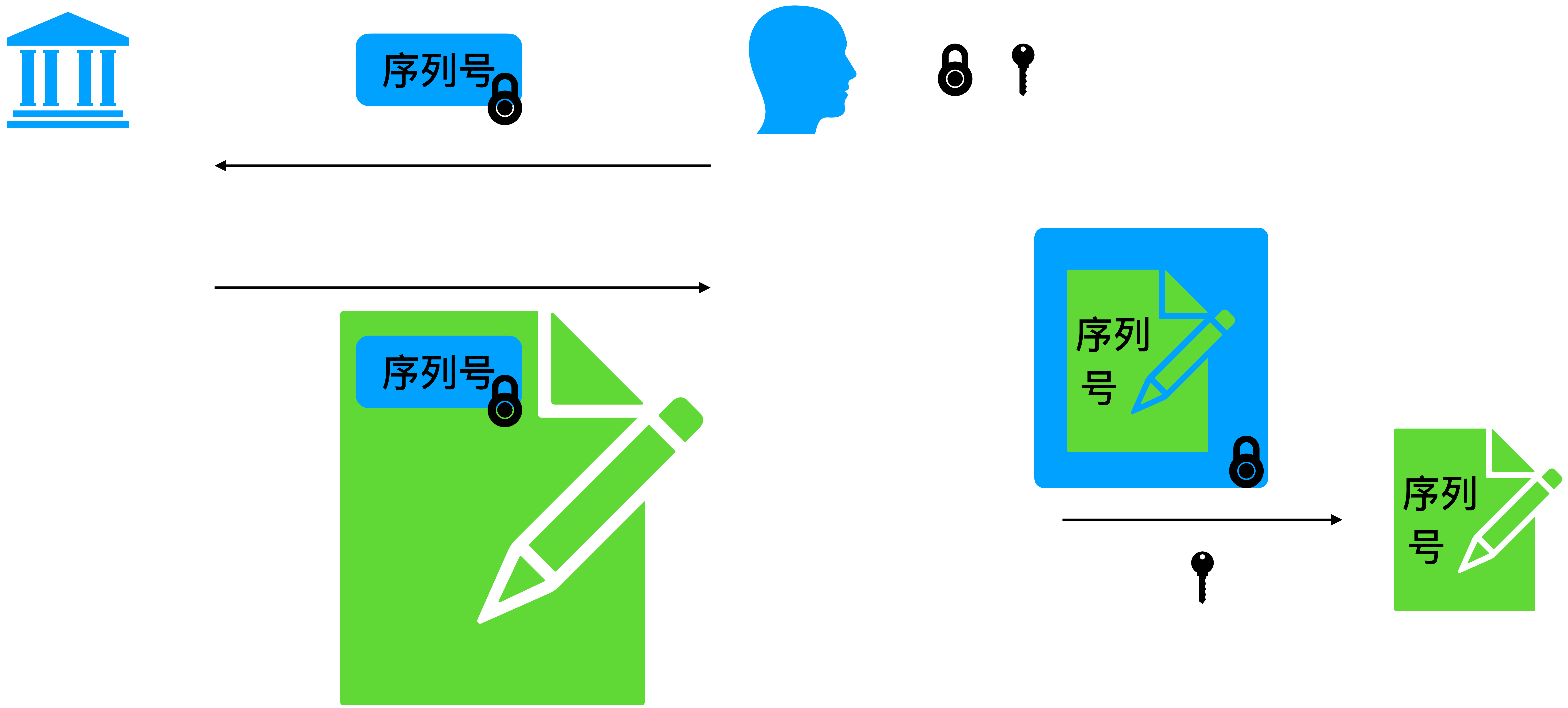
现金数字货币探索（二）

- 1988年David Chaum提出利用盲签名的方式来同时解决匿名性和双支付攻击
- 盲签名的重要特性：



现金数字货币探索（二）

- 如何利用盲签名的性质设计数字现金?



现金数字货币探索（二）

- Chuam电子货币的优缺点：
  - 匿名性：发送给银行的是加密信息，银行无法知道具体序列号✔
  - 中心服务器需要参与每一笔交易✘
  - 无法离线进行交易✘

现金数字货币探索（三）

- 1988年David Chaum, Amos Fiat & Moni Naor：离线双支付检测
- 不可思议!
  - 传统货币的不可复制性来源于特殊的纸张，油墨，水印的难复制特性
  - 数字货币是数字信息，可以实现完美复制（每个比特都相同）

现金数字货币探索（三）

- 解决方案?
  - 从信用货币中汲取灵感
  - 为了保证信用卡支付的安全性，每一笔信用卡支付实际需要经过联网认证
  - 那飞机上的信用卡如何支付?
    - 基于信用的支付方式
    - 支付结束后对双支付的检测

现金数字货币探索（三）

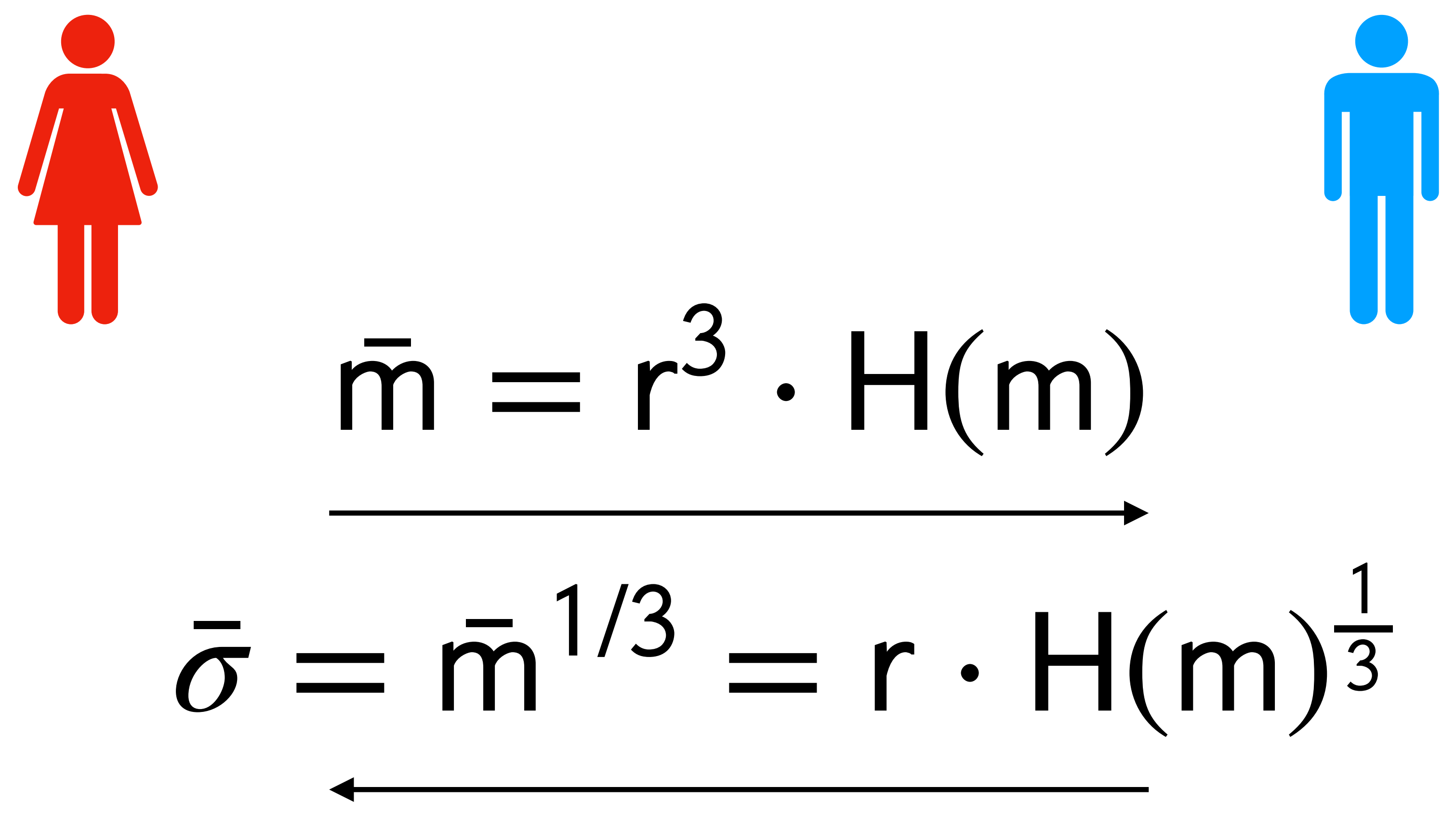
- David Chaum, Amos Fiat & Moni Naor 共同设计了一种加密算法
  - 电子货币中加密了身份信息
    - 即使银行也无法解密
  - 每次支付的时候，接受随机人让你解密一部分信息
  - 双支付发生了以后，两个不同的电子支付可以让银行追踪到个人信息

中心化的数字货币

- 盲签名系统
  - 解决数字货币的匿名性问题
- 抵御双支付攻击
  - 解决数字货币的伪造重用问题

数字货币 - Chaum-RSA-FDH

- 我们从盲签名算法开始：
  - RSA-FDH（RSA- Full Domain Hash）
    - $pk = N, 3, sk = 1/3 \mod \phi(n)$
    - 信息盲化：$\bar{m} = r^3 \cdot H(m)$
    - 盲签名：$\bar{\sigma} = r \cdot H(m)^{\frac{1}{3}}$



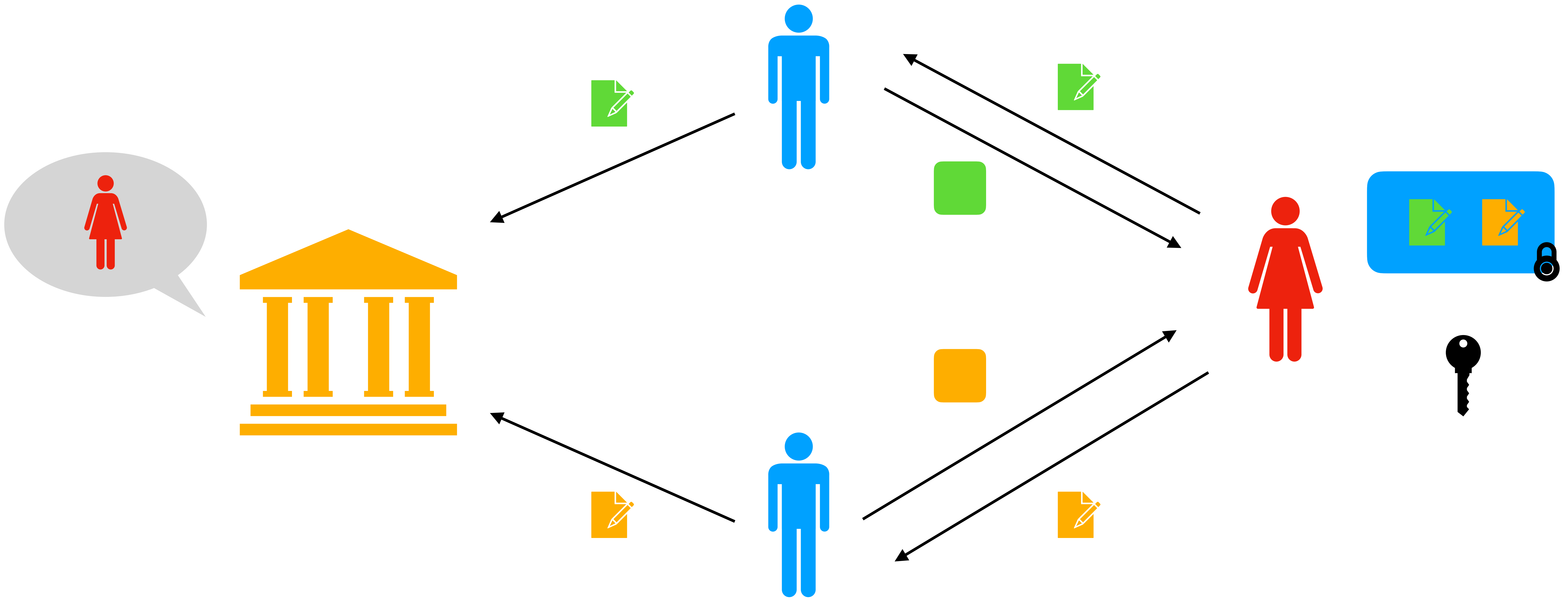
数字货币 - 抵御双支付攻击

- 如何生成可以抵抗双支付攻击的序列号?
- $f_{\mathsf{SN}}(id, n_s, n_r)$满足：（其中$n_s$是发送者的随机数，$n_r$是接收者的随机数）
  - 没有n_s的情况下算不出$f_{\mathsf{SN}}(id, n_s, n_r)$ - 货币安全性
  - 给出$f_{\mathsf{SN}}(id, n_s, n_r)$和$f_{\mathsf{SN}}(id, n_s, n_r')$且$n_r \neq n_r'$的时候，能算出$id$。
- $f_{\mathsf{SN}}(id, n_s, n_r) = pk_s^{n_r}\mathsf{H}(n_s)$
  - 货币安全性 → 哈希函数的随机性
  - 抗碰撞性 →$pk_s = (f_{\mathsf{SN}}(id, n_s, n_r) \cdot f_{\mathsf{SN}}(id, n_s, n_r)^{-1})^{1/(n_r - n_r')}$

中心化数字货币

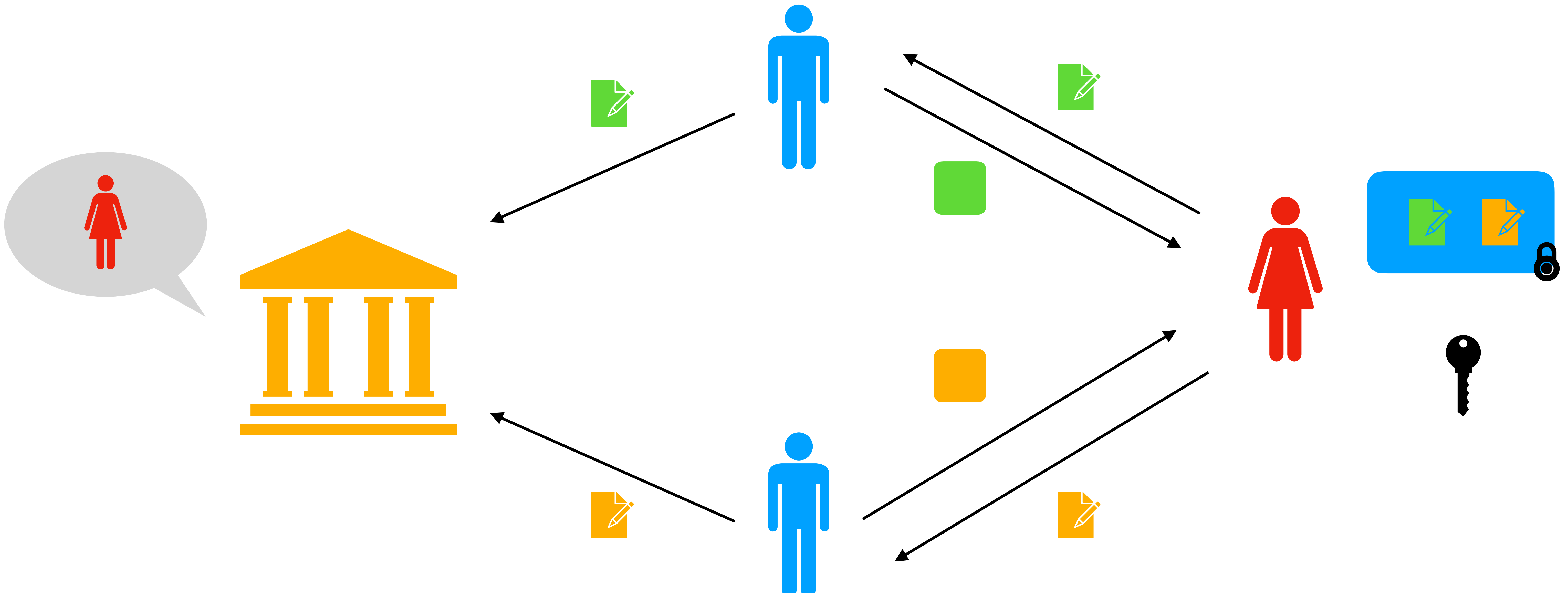
- 序列号：
  - $f_{SN}(pk_a, n_a, n_b) = pk_a^{n_b} H(n_a)$
  - 所需要的性质：
    - 抗双支付攻击：同样的货币发送给两个不同的人则身份揭露
    - 匿名性：序列号本身不泄漏身份信息

回顾



- 几个关键点：
  - Alice如果没有进行双支付的操作的情况下，银行不知道密钥。所以无法知道身份信息
  - 无法诬陷，Alice同一个人解密的相同的区域。

具体实现



- 几个关键点：
  - $pk_a^{n_b}H(n_a)$泄漏了部分pk_a的信息，但是在没有n_a的情况下无法计算pk_a
  - 如果Alice实施了双支付攻击，则根据$pk_a^{n_b}H(n_a)/pk_a^{n_b'}H(n_a)$可算出$pk_a$

中心化数字货币

- 抵御双支付攻击的初步想法：
  - $C_{a,b} = pk_a^{n_b} H(n_a)$
  - 然而，接收者Bob不知道n_a所以无法验证$C_{a,b}$的正确性。
- 仔细思考：
  - 其实是有问题的
    - 为了防止货币伪造，n_a不能泄漏
    - 为了抵御双支付，所有Alice产生的序列号又要满足使用同样的n_a

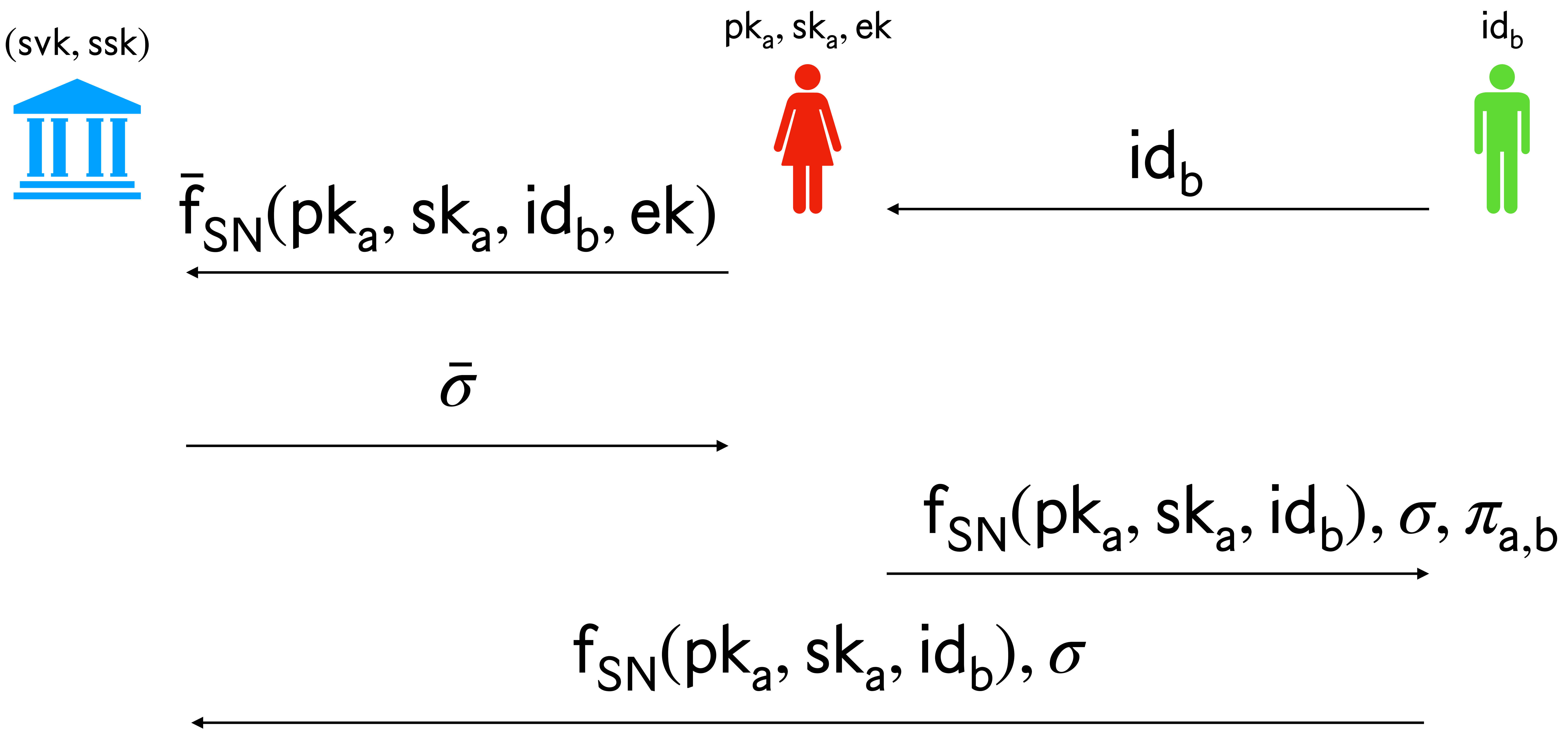
中心化数字货币

- 怎么办?
  - 这个过程中，需要Alice的身份参与 → 公钥加密系统，用sk_a当作n_a使用。
  - 使得Bob在能够验证的同时，不知道具体信息 → 零知识证明
- 解决方案：
  - $n_a \rightarrow sk_a$, $n_b \rightarrow id_b$
  - 给出零知识证明$\pi_{a,b}$证明：
    - 存在sk_a使得sk_a是pk_a的私钥，且$C_{a,b} = pk_a^{id_b} H(sk_a)$

中心化数字货币数字货币

- Alice通过与Bob的交互产生$f_{\mathrm{SN}}(pk_A, sk_a, id_b)$
- Alice向银行获取$f_{\mathrm{SN}}(pk_A, sk_a, id_b)$的盲签名$\sigma$
- Alice计算$\pi_{a,b}$证明f_{SN}是正确计算的
- 将$(f_{\mathrm{SN}}(pk_A, sk_a, id_b), \sigma, \pi_{a,b})$发送给Bob
- Bob验证签名安全性以及$\pi_{a,b}$的正确性，将钱存入银行

中心化数字货币



(svk, ssk)
pk_a, sk_a, ek
id_b
id_b
$\bar{f}_{SN}(pk_a, sk_a, id_b, ek)$
$\bar{\sigma}$
$f_{SN}(pk_a, sk_a, id_b), \sigma, \pi_{a,b}$
$f_{SN}(pk_a, sk_a, id_b), \sigma$

中心化数字货币安全性

- 正确性（Correctness）：正常支付的货币能够被验证
- 匿名性（Anonymity）：银行不知道支付者（Alice）的信息
    - 这里注意：
        - 接收货币的人（Bob）肯定是知道支付者（Alice）的信息的
        - 银行也是知道接收货币的人（Bob）的身份的
- 双支付攻击安全性（Double Spending）：支付者（Alice）如果进行了双支付，则身份信息会暴露给银行
- 不可抵赖（Undeniable）：货币无法被第三方复制

中心化数字货币安全性

- 需要的性质：
  - 数字货币的正确性：
    - 接收者需要验证：
      - $f_{SN}(pk_a, sk_a, pk_b), \pi_{a,b}$ → 序列号的生成以及零知识证明的正确性
      - $Ver(svk, f_{SN}(pk_a, sk_a, pk_b), \sigma) = 1$ → 盲签名正确性
    - 诚实的支付者：产生的数字货币可以被验证

中心化数字货币安全性

- 匿名性（Anonymity）：银行不知道支付者（Alice）的信息
    - 这里注意：
        - 接收货币的人（Bob）肯定是知道支付者（Alice）的信息的
            - $\pi_{a,b}$中包含$\mathrm{pk_a}$的信息，但是存入货币的时候（Bob）不会将$\pi_{a,b}$发给银行
        - 银行也是知道接收货币的人（Bob）的身份的
- 银行所知道的信息：
    - $\bar{f}_{\mathrm{SN}}$：盲化以后的序列号→ 盲签名的匿名性保证序列号的安全性
    - f_{SN}：序列号 →序列号的匿名性
    - σ：序列号的签名 →这里签名也不包含支付者（Alice）的信息

中心化数字货币安全性

- 双支付攻击安全性（Double Spending）：支付者（Alice）如果进行了双支付，则身份信息会暴露给银行
- 如果进行了双支付操作：
  - 则会产生$f_{\mathrm{SN}} = pk_a^{id_b}H(sk_a)$和$f'_{\mathrm{SN}} = pk_a^{id'_b}H(sk_a)$其中$id_b \neq id'_b$
  - 序列号的抗双支付攻击：
    - $pk_a = (f_{\mathrm{SN}} \cdot f_{\mathrm{SN}})^{1/(id_b - id'_b)}$

不可抵赖属性

- 不可抵赖（Undeniable）：货币无法被第三方复制
- 观察发送给银行的数字货币：
  - $f_{SN}(pk_a, sk_a, id_b), \sigma$
    - 如果σ验证通过，则$f_{SN}(pk_a, sk_a, id_b)$必须是经过银行发出的。
    - 而$f_{SN}(pk_a, sk_a, id_b)$在不知道sk_a的情况下无法生成：
      - $f_{SN}(pk_a, sk_a, id_b)$是由支付方（Alice）生成的