第六章：数字签名

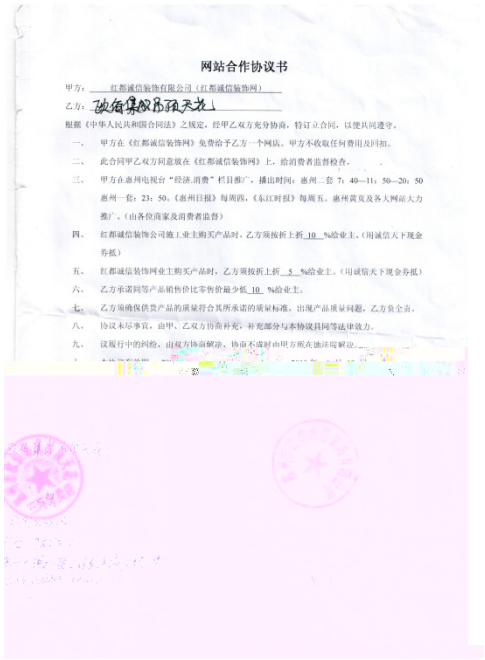
赵臻

zzhen@xidian.edu.cn

2023.05.10

问题的提出

手写签名：传统的确认方式，如书信、签约、支付、批复等



网站合作协议书

甲方：红都诚信装饰有限公司（红都诚信装饰网）

乙方：

根据《中华人民共和国合同法》之规定，经甲乙双方充分协商，特订立合同，以便共同遵守。

一、甲方在《红都诚信装饰网》免费给予乙方一个网店。甲方不收取任何费用及回扣。

二、此合同甲乙双方同意放在《红都诚信装饰网》上，给消费者监督检查。

三、甲方在惠周电视台“经济.消费”栏目推广，播出时间：惠州二套 7：40—11：50—20：50 惠州一套：23：50，《惠州日报》每周四，《东江时报》每周五。惠州黄页及各大网站大力推广。（由各位商家及消费者监督）

四、红都诚信装饰公司施工业主购买产品时，乙方须按折上折 10 %给业主。（用诚信天下现金券抵）

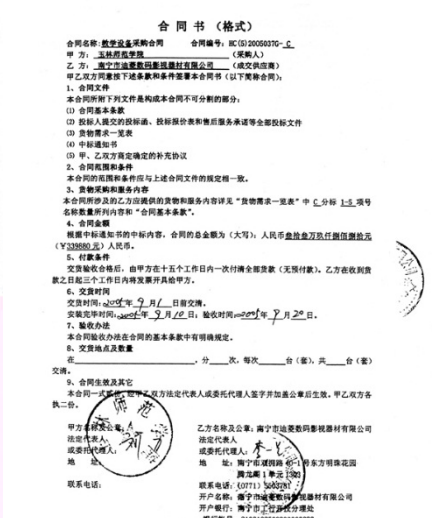
五、红都诚信装饰网业主购买产品时，乙方须按折上折 5 %给业主。（用诚信天下现金券抵）

六、乙方承诺同等产品销售价比零售价最少低 10 %给业主。

七、乙方须确保供货产品的质量符合其所承诺的质量标准，出现产品质量问题，乙方负全责。

八、协议未尽事宜，由甲、乙双方协商补充，补充部分与本协议具同等法律效力。

九、议履行中的纠纷，由双方协商解决，协商不成时由甲方所在地法院解决。



合 同 书 （格式）

合同名称：教学设备采购合同　　合同编号：HC(S)20050370-C

甲 方：玉林师范学院　　（采购人）

乙 方：南宁市迪爱数码影视器材有限公司　　（成交供应商）

甲乙双方同意按下述条款和条件签署本合同书（以下简称合同）：

1、合同文件

本合同所附下列文件是构成本合同不可分割的部分：

(1) 合同基本条款

(2) 投标人提交的投标函、投标报价表和售后服务承诺等全部投标文件

(3) 货物需求一览表

(4) 中标通知书

(5) 甲、乙双方商定确定的补充协议

2、合同范围和条件

本合同的范围和条件应与上述合同文件的规定相一致。

3、货物采购和服务内容

本合同所涉及的乙方应提供的货物和服务内容详见“货物需求一览表”中 C 分标 1-5 项号名称数量所列内容和“合同基本条款”。

4、合同金额

根据中标通知书的中标内容，合同的总金额为（大写），人民币叁拾叁万玖仟捌佰捌拾元（￥339880.元）人民币。

5、付款条件

交货验收合格后，由甲方在十五个工作日内一次付清全部货款（无预付款）。乙方在收到货款之日起三个工作日内将发票开具给甲方。

6、交货时间

交货时间：2005年 9月 1 日前交清。

安装完毕时间：2005年 9月 10 日；验收时间2005年 9月 20 日。

7、验收办法

本合同验收办法在合同的基本条款中有明确规定。

8、交货地点及数量

在_____，分____次，每次_____台（套），共____台（套）交清。

9、合同生效及其它

本合同一式叁份，经甲乙双方法定代表人或委托代理人签字并加盖公章后生效。甲乙双方各执二份。

甲方名称及公章：

法定代表人

或委托代理人：

地　　址：

联系电话：

乙方名称及公章：南宁市迪爱数码影视器材有限公司

法定代表人

或委托代理人：

地　　址：南宁市建园路 49-1 号东方明珠花园 [illegible] 单元 [illegible]

联系电话：(0771) 5603287

开户名称：南宁市迪爱数码影视器材有限公司

开户银行：南宁市工行 [illegible] 分理处

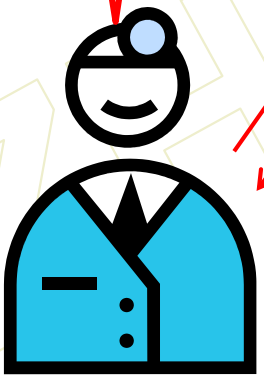
银行账号：2102108509000003088

数字签名

从我账户给Bob转入100万

1. 这么大一笔资金，这消息是Alice发的吗？
2. 要是过后Alice不承认这笔转账怎么办？

Alice，把你要转账的消息签名之后再发给我



Alice

Bob

- **数字签名**是手写签名数字化的产物，但又有着显著的区别
  - **每个消息的签名都不同**，否则签名就会被获取并复制到另外的文件中
- 数字签名的基础是公钥密码学

ZHAO ZHEN

数字签名的基本概念

数字签名应该具有的性质

完整性

一个被签了名的消息，无法分割成为若干个被签了名的子消息。这一性质保证了被签名的消息不能被断章取义。

（这个性质的本质是：）当一个签名消息被分割成多个子消息发送出去，则签名已经被破坏了，收到消息的人会辨认出签名是无效的（不合法的）。

数字签名的基本概念

身份唯一性（不可伪造性）

被Alice签名的消息只能由Alice生成。

（这个性质的本质是：） Bob在收到一个“被Alice签名的消息”时，他有办法检验，该签名是否真的被Alice签名的消息。

或许攻击者Eve截获了大量的被Alice签名的消息，但他仍然不能伪造出一个新的，别人认可的“被Alice签名的消息”。

如果无论Eve截获多少被Alice签名的消息，他伪造新的“被Alice签名的消息”的成功概率仍然没有丝毫提高，则称该签名算法是零知识的。

数字签名的基本概念

不可否认性（公开可验证性）

被Alice签名的消息，在未来不能被Alice否认。

（这个性质的本质是：）Bob在收到一个被Alice签名的消息时，他有办法向第三方证明该签名是真的被Alice签名的消息。

如果一个数字签名具有**不可伪造性**，则Bob能够自行验证签名消息的真伪；而如果一个数字签名具有**公开可验证性**，则Bob能够向他人证明签名消息的真伪。

数字签名的基本概念

公钥密码的签名思想

既然要求数字签名具有这么多的性质，怎样来构造数字签名？

答案是：以公钥密码为基础的数字签名算法才能具有如此强大的信息安全功能。

回忆公钥密码：

- 明文m，密文c。
- Alice的加密密钥（公钥）是z，解密密钥（私钥）是k。
- 加密方程$c=E(m, z)$，解密方程$m=D(c, k)$。

数字签名的基本概念

公钥密码的签名方案（一）

Alice欲发消息m给Bob。

（1） Alice用自己的私钥k对消息m“解密”$s=D(m, k)$，s就是对消息m的签名值，$(m，s)$就是一个签名消息。

（2） Alice将$(m，s)$发送给Bob。

（3） Bob收到$(m，s)$后，用Alice的公钥z，将消息m与签名s做如下的检验：是否$m=E(s, z)$。若是则$(m，s)$是Alice发送的签名消息。

数字签名的基本概念

在上述方案中，

- “密文”变成了消息m，“解密方程”变成了签名方程$s=D(m, k)$。
- “明文”变成了签名值s，“加密方程”变成了验证方程$m=E(s, z)$。
- 任何人只要拥有Alice的公钥z，都可以对签名消息$(m，s)$进行验证。
- 是否只有Alice自己才能生成自己的签名消息呢？

数字签名的基本概念

签名方案（一）的安全性分析：

（1）设Eve知道Alice的公钥z，选定消息m，对签名值s进行伪造。

- 要想让伪造的签名值s通过检验， Eve必须选择s满足验证方程$m=E(s, z)$。然而在验证方程中是解不出s的，必须得到Alice的私钥k，用签名方程得到s：$s=D(m, k)$。
- 这就是说，对事先设定的消息m来说，签名消息(m, s)具有身份唯一性和不可伪造性。

数字签名的基本概念

（2）设Eve知道Alice的公钥z，设定签名值s，反过来对消息m进行伪造。此时消息m的内容就不能是选定的。

- Eve选择一个“签名值”s，用验证方程计算“消息”$m=E(s, z)$。
- Eve冒充Alice将(m，s)发送给Bob。
- Bob用验证方程检验得$m=E(s, z)$。于是Bob认为(m，s)就是Alice发送的签名消息。攻击成功。

为了抵抗这种攻击，合法签名消息(m，s)中的消息m必须是有意义的课文，而不是乱码。

数字签名的基本概念

公钥密码的签名方案（二）

额外使用一个公开的杂凑函数H。 设Alice欲发消息m给Bob。

（1）Alice用H将消息m进行处理，得$h=H(m)$。

（2）Alice用自己的私钥k对h“解密”$s=D(h, k)$，s就是对消息m的签名值，$(m，s)$就是一个签名消息。

（3） Alice将$(m，s)$发送给Bob。

（4） Bob收到$(m，s)$后，用Alice的公钥z，将消息m与签名s做如下的检验：是否$H(m)=E(s, z)$。若是则$(m，s)$是Alice发送的签名消息。

数字签名的基本概念

在上述方案中，

- 签名方程是$s=D(H(m), k)$。
- 验证方程是$H(m)=E(s, z)$。
- 任何人只要拥有Alice的公钥z，都可以对签名消息$(m，s)$进行验证。
- 设攻击者Eve知道Alice的公钥z，他试图伪造一个$(m，s)$，让Bob相信$(m，s)$是Alice的签名消息。伪造的$(m，s)$必须满足验证方程$H(m)=E(s, z)$。

数字签名的基本概念

Eve面临着两难问题：

- 如果选定消息m，再匹配签名值s，则在验证方程$H(m)=E(s, z)$中无法解出s。（这是公钥密码的基本安全性）
- 如果选定签名值s，再匹配消息m，则在验证方程$H(m)=E(s, z)$中能够解出$H(m)$，却无法得到m。（这是杂凑函数的性质）

如此看来，签名方案（二）似乎具有身份唯一性和不可伪造性。

数字签名的基本概念

签名方案（二）存在的问题

（1）如果给Eve更加宽松的条件，设他不但知道Alice的公钥z，而且已经截获了许多Alice的签名消息

$(m^{(1)}, s^{(1)}), (m^{(2)}, s^{(2)}), \ldots, (m^{(N)}, s^{(N)})$。

Eve伪造新的Alice的签名消息(m, s)是否更加容易？

如果允许重复发送，则Eve的伪造是轻而易举的，他只需要发送$(m^{(n)}, s^{(n)})$即可。此称为重放攻击。

数字签名的基本概念

为了使签名方案（二）抵抗重放攻击，通常使用两种方法：

- Alice已经发送过的签名消息必须存储备案，不得再次发送。一旦发现有再次发送，则肯定是重放攻击。但Eve可以根据公钥密码的结构缺陷来伪造。比如$(m, s)=(m^{(1)}m^{(2)}, s^{(1)}s^{(2)})$。第一种方法没有抵抗这种伪造的能力。
- Alice的签名消息中必须含有时戳。一旦发现发送时间过于久远，则肯定是重放攻击。但“时间过于久远”的标准很模糊。

数字签名的基本概念

（2）注意到Alice先得到$h=H(m)$，再计算出$s=D(h, k)$。这就要求：

公钥密码对课文h能够正确进行解密运算。

换句话说，对课文h进行解密运算的结果，再进行加密就得到h。

事实上，许多公钥密码具有如下的性质：

当某个明文加密得到h，则对h解密再加密就得到h；

当h是随意选择的，则对h解密再加密很难得到h。

比如，NTRU就是这样的公钥密码。因此NTRU无法用签名方案（二）来构造数字签名。

数字签名的目的和要求

- **数字签名的目的：**保证信息的完整性和真实性，即消息没有被篡改，而且签名也没有被篡改，消息只能始发于所声称的一方
- 一个完善的签名方案应满足以下三个条件：
  1. **不可否认性：**签名者事后不能否认或抵赖自己的签名
  2. **不可伪造性：**其他任何人均不能伪造签名，也不能对接收或发送的信息进行篡改、伪造和冒充
  3. **公正的仲裁：**若当事双方对签名真伪发生争执时，能通过公正的仲裁者验证签名来确定其真伪

数字签名的定义

- **ISO对数字签名的定义：**

  是指附加在数据单元上的一些数据，或是对数据单元所做的密码变换，这种数据或变换允许数据单元的接收者用以确认**数据单元来源**和数据单元的**完整性**，并保护数据，防止被人伪造

ZHAOJIEN

签名方案的组成

- 签名方案是满足下列条件的五元组(P, A, K, S, V)：
  1. $\boldsymbol{P}$:所有可能消息组成的有限集
  2. $\boldsymbol{A}$:所有可能签名组成的有限集
  3. $\boldsymbol{K}$:所有可能密钥组成的有限集
  4. 对每一个$k \in K$，有一个签名算法$S_k \in S$和一个相应的验证算法$V_k \in V$。对消息$x \in P$和相应签名$y \in A$，签名算法S_k：$P \to A$和验证算法V_k：$P \times A \to \{0, 1\}$都满足：
     - 当$y = S_k(x)$时，$V_k(x, y) = 1$
     - 否则$V_k(x, y) = 0$

ZHAOCHEN

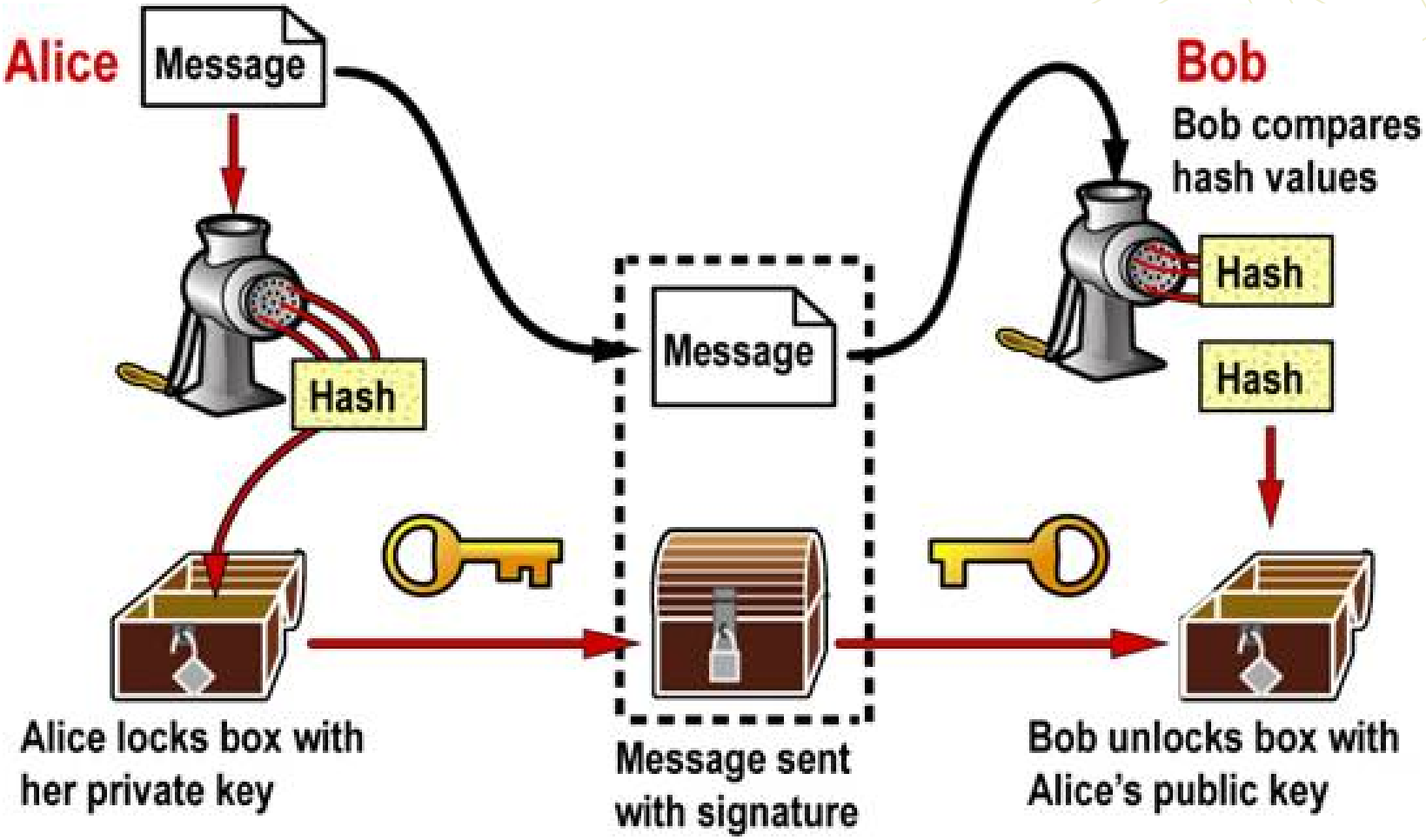
数字签名的过程

- **数字签名方案一般包括三个过程：**

1. **系统初始化过程**：产生数字签名方案中的所有系统和用户参数(**公开的+秘密的**)

2. **签名过程**：用户利用给定的**签名算法**对消息签名，签名过程可以公开也可以不公开，但一定包含仅签名者才拥有的秘密信息（签名密钥）

3. **验证过程**：验证者利用公开的**验证方法**对给定消息的签名进行验证


Alice
Message
Hash
Alice locks box with her private key
Message
Message sent with signature
Bob
Bob compares hash values
Hash
Hash
Bob unlocks box with Alice's public key

RSA签名

- 基于RSA公钥体制的签名方案通常称为**RSA数字签名方案**。
  RSA签名体制的基本算法如下：

1. **密钥的生成**（与加密系统一样）：

   公钥$Pk = \{e, n\}$;私钥$Sk = \{d\}$

2. **签名过程** (d, n)：

   用户A对消息$M \in Z_n$进行签名，计算

   $$S = Sig(H(M)) = H(M)^d \bmod n;$$

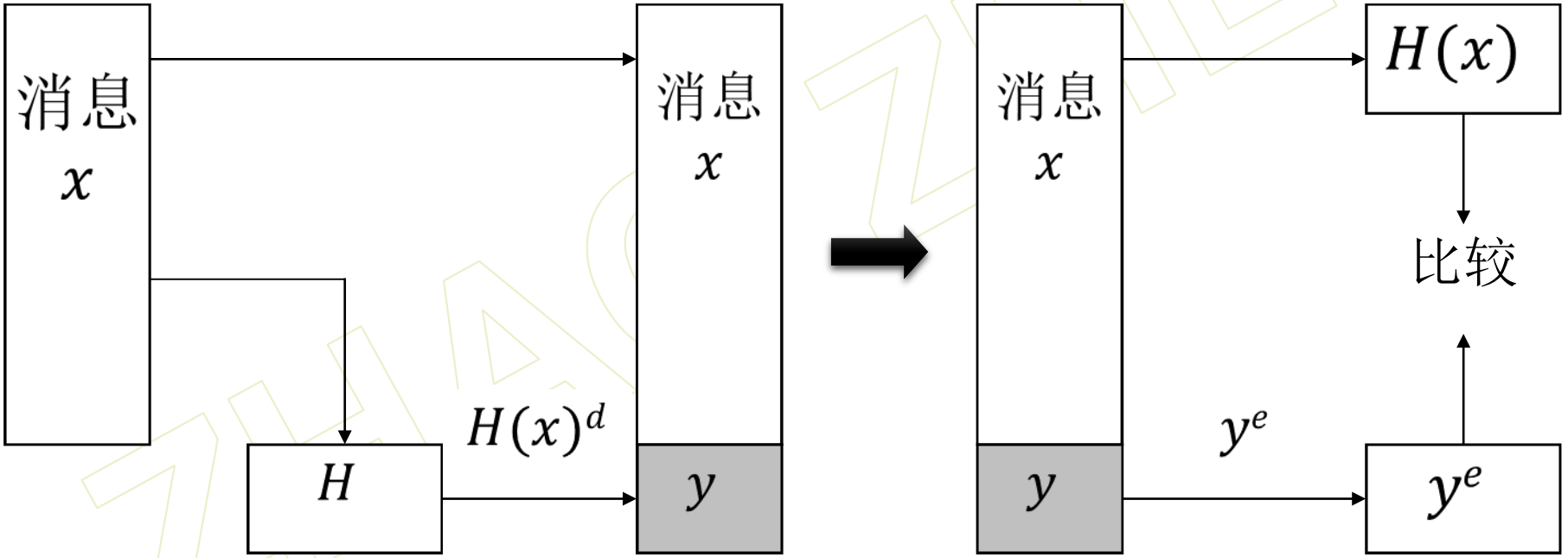
   并将S附在消息M后

3. **验证过程** (e, n)：

   给定(M, S)，$Ver(M, S)$为真$\Leftrightarrow$

   $$H(M) = S^e \pmod n \text{成立}$$

RSA签名方案图



Problem: 为什么对消息的*Hash*函数值签名而不是直接对消息签名？

对RSA签名的攻击

- 假设RSA直接对消息进行签名
- 一般攻击：攻击者任选一个数据Y，用A的公钥计算$X=Y^e \bmod n$，于是便可以用***Y*伪造*A*对消息*X*的签名**，因为

$$Y = X^d \bmod n$$

  - 实际意义不大：伪造的消息X具有实际意义的概率很小
  - Hash函数可以抵御这种攻击

ZHAO ALIEN

对RSA签名的攻击

- 利用已有签名进行攻击：如果消息M_1、M_2的签名分别是S_1和S_2，则任何知道M_1，S_1，M_2，S_2的人可以伪造对消息M_1M_2的签名S_1S_2，因为

$$Sig(M_1M_2) = Sig(M_1)Sig(M_2)$$

- Remark：用户不要轻易对其他用户提供的随机数据进行签名
- 更有效的方法：对数据的Hash值签名

ZHAO ZHEN

对RSA签名的攻击

- 利用签名获得明文：
  - 攻击者截获密文$C=M^e \bmod n$，选择随机数r，并计算
    $$x=r^e \bmod n;\quad y=x\cdot C \bmod n;$$
    然后攻击者设法让发送者对y签名，获得：
    $$S=y^d \bmod n$$
    攻击者计算：
    $$r^{-1}\cdot S \bmod n = r^{-1}y^d \bmod n$$
    $$= r^{-1}x^d\, C^d \bmod n = C^d \bmod n = M,$$
  - Remark：用户不要轻易对其他用户提供的随机数据进行签名
  - 更有效的方法：对数据的Hash值签名

H(M)的重要性

- $H(M)$的另一个作用—加快签名速度
  - 对整个消息签名，由于公钥体制速度比较慢，当消息比较长时，签名与验证过程都会相当慢
  - 对消息的Hash值签名，则无论消息多长，签名都只与Hash值的长度有关

ZHAO ZHEN

ElGamal数字签名

ElGamal数字签名

- 1985年，ElGamal提出了一个基于离散对数问题的签名方案，后来称为ElGamal数字签名方案。
- 1991年该数字签名方案的变形被NIST确定为数字签名标准(DSS)。

ElGamal数字签名

ElGamal数字签名

ElGamal的密钥生成：选择一个大的素数p。选择g为域$GF(p)$的本原元素。选择正整数x。计算$y=g^x(\mathrm{mod}p)$。

Alice的公钥是（p，g，y），私钥是x。

签名方案是上述的签名方案(二)，额外使用一个公开的杂凑函数H。设Alice欲发消息m给Bob。

ElGamal数字签名

（1） Alice用H将消息m进行处理，得$h=H(m)$。

（2） Alice选择秘密随机数k，满足

$$0<k<p-1，且(k, p-1)=1，$$

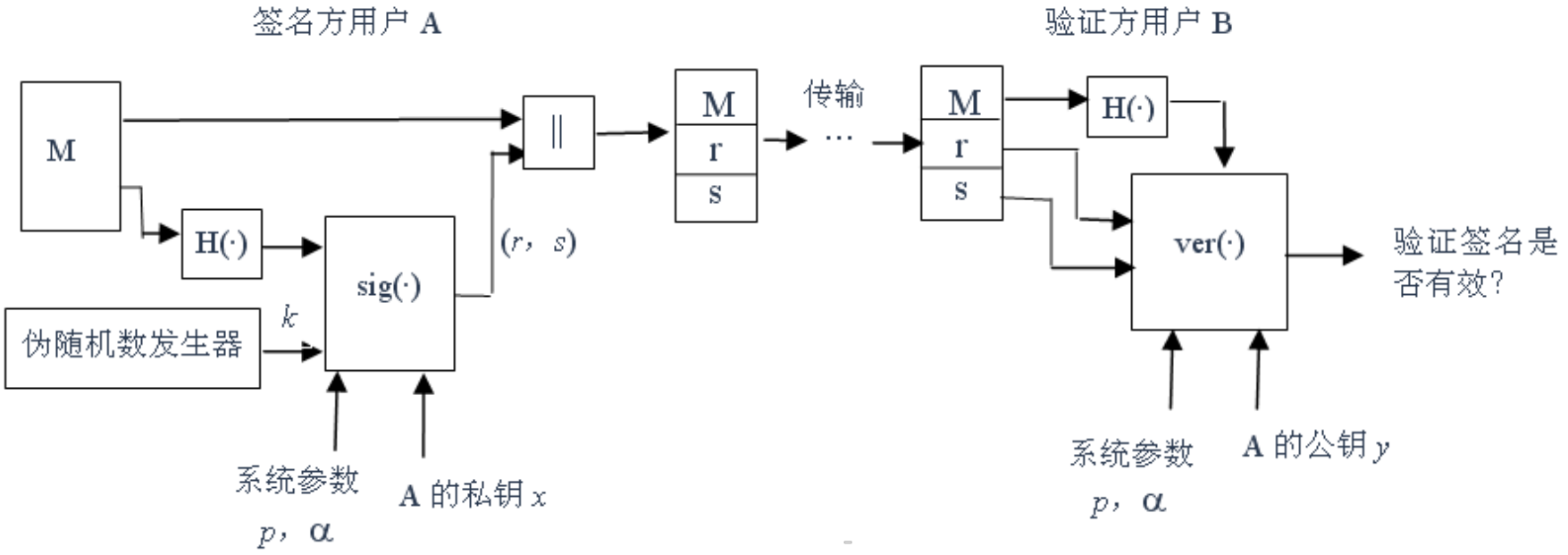
计算

$$r=g^k(\mathrm{mod}p);$$

$$s=(h-xr)k^{-1}(\mathrm{mod}\ (p-1))。$$

（3） Alice将$(m，r，s)$发送给Bob。

ZHAOZHEN



- 3. **验证签名过程**:接收方收到M与其签名(r,s)后:

① 计算消息M的Hash值$H(M)$;

② 验证公式

$$y^r r^s = g^{H(M)} \bmod p$$

成立则确认(r,s)为有效签名，否则认为签名是伪造的

- 随机化数字签名
  - ElGamal数字签名在生成签名时会选择一个整数k，同一个消息签名两次得到的结果通常是不同的，所以是一个随机化数字签名。
- 如果在两次签名时选择相同的随机数则是不安全的。
  - 分析：$s_1 \equiv [H(m_1) - xr]k^{-1} \bmod (p-1)$，$s_2 \equiv [H(m_2) - xr]k^{-1} \bmod (p-1)$

将两式相减，得$s_1 - s_2 \equiv [h(m_1) - h(m_2)]k^{-1} \bmod (p-1)$

进而得$k = \frac{h(m_1) - h(m_2)}{s_1 - s_2} \bmod (p-1)$

由$s_1 \equiv [h(m_1) - xr]k^{-1} \bmod (p-1)$求出私钥

$$x = \frac{h(m_1) - s_1 k}{r} \bmod (p-1)$$

- 例子
  - 密钥生成：设$p=29$, $g=2$, $x=7$,计算$y\equiv 2^7 \bmod 29=12$。则公钥$y=12$，私钥$x=7$。
  - 签名：假设消息m的Hash值为8，即$h(m)=8$，随机选取整数$k=5$，计算

    $$r\equiv 2^5 \bmod 29\equiv 3,\ s\equiv (8-7\times 3)\times 5^{-1} \bmod 28\equiv 3$$

    签名为$(r, s)=(3, 3)$。
  - 验证：收到消息m和签名(r, s)时，先计算$h(m)=8$，然后计算

    $$12^3\times 3^3\equiv 24 \bmod 29,\ 2^8\equiv 24 \bmod 29$$

    等式成立，签名是合法的。

Schnorr数字签名

Alice拥有3个正整数(p，q，g)，向自己的通信伙伴公开。其中：

- p是模数（即将要进行(mod$p$)模数运算），它是一个素数，值的范围在2^{511}到2^{512}之间（即p是一个长度为512的比特串）。
- q也是模数（即将要进行(mod$q$)模数运算），它是一个素数，$2^{159} < q$（即q是一个长度不小于160的比特串），并且q是p-1的一个因子。
- g是域$GF(p)$的元素，且$g^q=1(\mathrm{mod}p)$。

Schnorr数字签名

Alice选择x，其中$1<x<q$。

Alice计算$y=g^x(\mathrm{mod}p)$。

Alice的公钥是(p，q，g，y)， Alice的私钥是x。

签名方案是上述的签名方案(二)，额外使用一个公开的杂凑函数H， H的输出长度是160比特。设Alice欲发消息m给Bob。

Schnorr数字签名

（1）Alice选择秘密随机数k，满足$0<k<q$，计算

$$r=g^k(\mathrm{mod}p);$$

$$e=H(r, m);$$

$$s=k+xe(\mathrm{mod}q)。$$

（3） Alice将(m，e，s)发送给Bob。

（4） Bob用Alice的公钥，计算$r'=g^s y^{-e}(\mathrm{mod}\ p)$。检验是否

$$e=H(r', m)。$$

若是则(m，e，s)是Alice发送的签名消息。

Schnorr数字签名

Schnorr签名与ElGamal签名的不同点：

- 在ElGamal体制中，g为域$GF(p)$的本原元素；而在Schnorr体制中，g只是域$GF(p)$的阶为q的元素，而非本原元素。因此，虽然两者都是基于离散对数的困难性，然而ElGamal的离散对数阶为p-1，Schnorr的离散对数阶为$q<p$-1。从这个角度上说，ElGamal的安全性似乎高于Schnorr 。

- 签名长度比较： Schnorr比ElGamal签名长度短。

ElGamal：(m，r，s)，其中r的长度为$|p|$， s的长度为$|p-1|$。

Schnorr： (m，e，s)，其中e的长度为$|q|$， s的长度为$|q|$。

- 在Schnorr签名中， $r=g^k(\mathrm{mod}p)$可以预先计算，k与m无关，因而签名只需一次mod$q$乘法及减法。所需计算量少，速度快，适用于灵巧卡采用。

数据签名标准DSS

- 1991年，1991年，数字签名标准**DSS**（Digital Signature Standard）

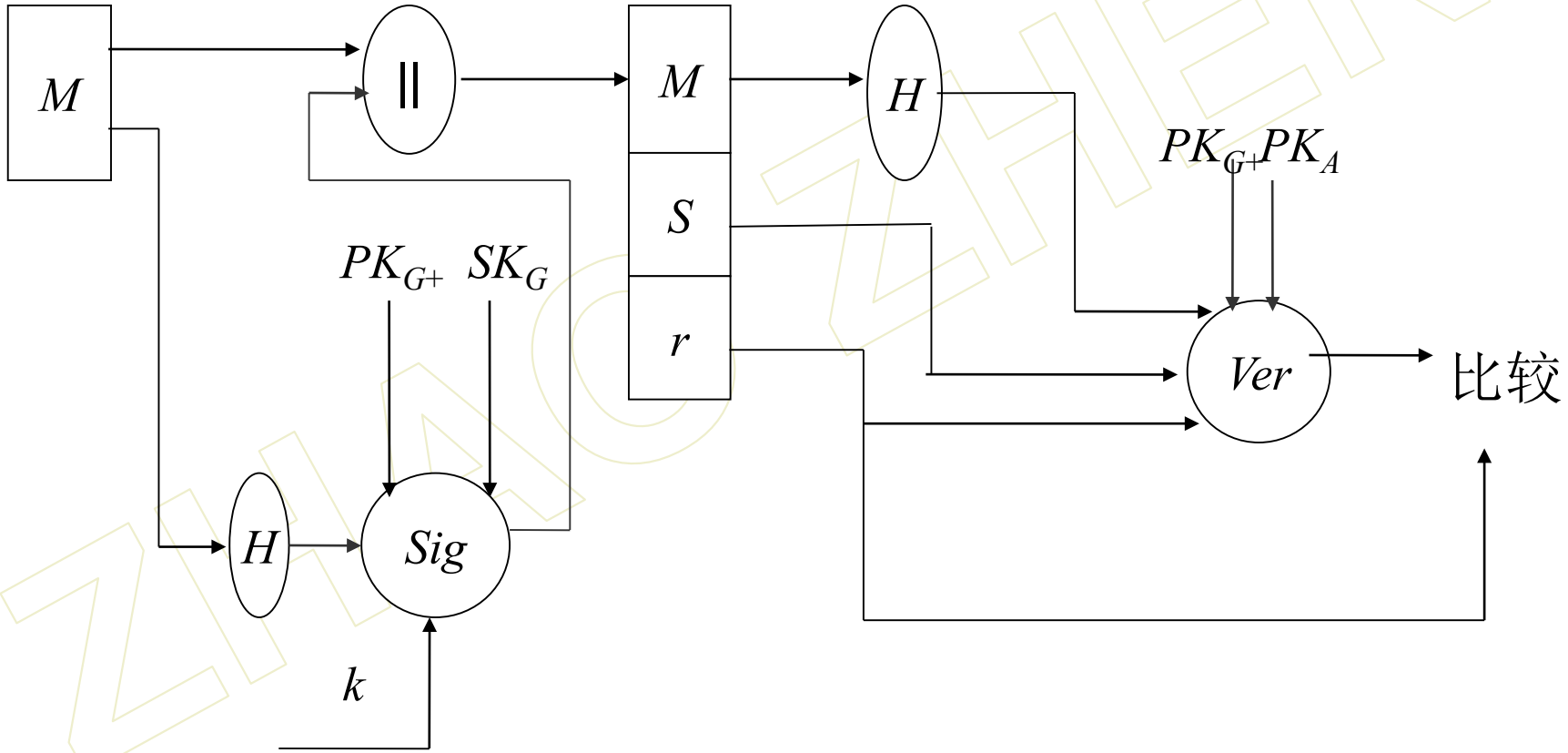
在ElGamal数字签名和Schnorr数字签名的基础上发展而来，被NIST确定为数字签名标准。

  - 数字签名标准DSS中的算法称为DSA(digital signature algorithm)，其**安全性是基于离散对数问题的困难性**。
  - 和DES一样，DSS也引起了激烈的争论。反对者认为：**密钥太短**、**效率不如RSA高**、**不能实现数据加密**并怀疑NIST在DSS中**留有后门**
  - 与RSA不同的是，DSS只能用于签名，不能用于加密。
  - 随后，美国政府对其做了一些改进
- 目前DSS的应用已经十分广泛，并被一些国际标准化组织采纳为国际标准
- 2000年，美国政府将RSA和椭圆曲线密码引入到数字签名标准中，进一步丰富了DSS算法

DSS算法描述

- **DSS签名：**消息的Hash值连同随机数k一起作为签名函数的输入，签名函数还需使用发送方的私钥SK_A和一些公开参数PK_G+，签名函数的两个输出构成了消息的签名(s,r)

- 接收方收到消息后再产生消息的Hash值，将Hash值与签名一起输入验证函数，验证函数还需输入PK_G+和发送方的公钥PK_A。验证函数的输出如果与收到的签名成分r相等，则签名有效


M
||
M
S
r
H
PK_{G+} PK_A
PK_{G+} SK_G
Ver
比较
H
Sig
k

数字签名算法DSA

- **DSS的签名算法称为DSA**，使用以下参数：
  1. p为素数,$2^{L-1}<p<2^{L}$,其中$512 \le L \le 1024$且L为64的倍数，即
     $$L=512+64j \quad j=0,1,2,\ldots,8$$
  2. q为素数,是$p-1$的素因子，$2^{159}<q<2^{160}$
  3. $g=h^{(p-1)/q} \bmod p$，$1<h<p-1$，且满足$g>1$
  4. x为随机数，$0<x<q$，***计算$y=g^x \bmod p$***

- 参数p、q、g可以公开，且可为一组用户公用。***x和y分别为用户的私钥和公钥***

DSA签名生成过程

- 选择一个随机数k，$0<k<g$，进行如下计算：

$$r=(g^k \bmod p) \bmod q,$$
$$s=k^{-1}(SHA(M)+xr) \bmod q$$

  - 检验r和s是否为零，若$r=0$或$s=0$，则重新产生k，并重新计算产生签名r和s
  - 每一签名使用不同的k

- 把签名r和s附在M后面发给接收者 $(M \,||\, r \,||\, s)$

DSA签名验证过程

1. 首先检验是否有$0<r<q$，$0<s<q$，若其中之一不成立，则签名为假
2. 计算：

$$w=s^{-1} \bmod q,$$
$$u_1=w \cdot SHA(M) \bmod q,$$
$$u_2=rw \bmod q,$$
$$v=(g^{u_1} y^{u_2} \bmod p) \bmod q$$

3. 若$v=r$，则签名为真，否则签名为假或数据被篡改

ZHAOZHIEN

DSA签名生成过程

- **DSS的模数*p***
  - **最初DSS的模数*p*固定在512位。**
  - **后来NIST对标准进行了修改，允许使用不同大小的模数。**
  - **以目前的密码分析能力，2048位的*p*是比较安全的。**

安全等级 (分组密码的密钥长度)	Hash函数 (Hash值长度)	*q*	*p*	签名长度
80	160	160	1024	320
112	224	224	2048	448
128	256	256	3072	512

DSS的标准参数

ZHAO ZHEN

DSA签名生成过程

- 例子
  - 密钥生成：设q=101, p=78×101+1=7879, h=3, $g\equiv3^{78} \bmod 7879\equiv170$。设用户私钥为$x$=75，用户公钥为$y\equiv170^{75} \bmod 7879\equiv4567$。
  - 签名：设消息m的Hash值为1234，$h(m)$=1234，随机选取整数k=50，计算:

    $r\equiv(170^{50} \bmod 7879) \bmod 101\equiv94$, $s\equiv50^{-1}\times(1234+75\times94) \bmod 101\equiv97$

    消息m的签名为(r, s)=(94, 97)。
  - 验证：收到消息m和签名(r, s)时，先计算$h(m)$=1234，然后计算:

    $w\equiv97^{-1} \bmod 101\equiv25$, $u_1\equiv1234\times25 \bmod 101\equiv45$

    $u_2\equiv94\times25 \bmod 101\equiv27$, $v\equiv(170^{45}\times4567^{27} \bmod 7879) \bmod 101\equiv94$

    $v=r$成立，签名是合法的。

ECDSA签名算法

签名过程：消息m，全局参数$D=(q,a,b,G,n,h)$，签名者的公私钥对(Q,d)，$(Q=dG)$

1. 选择一个随机数k，计算$kG=(x_1,y_1)$
2. $r=x_1 \bmod n$；如果$r=0$，则回到步骤1
3. $e=SHA1(m)$
4. $s=k^{-1}(e+dr) \bmod n$，如果$s=0$，则回到步骤1；
5. 对消息的签名为(r,s)，签名者把消息m和签名(r,s)发送给接收者

- **验证签名：**收到消息m和签名(r,s)之后
  1. 检验r、s，要求r、$s \in [1,n-1]$
  2. 计算$e=SHA1(m)$
  3. 计算$w=s^{-1} \bmod n$
  4. 计算$u_1=ew \bmod n$；$u_2=rw \bmod n$
  5. 计算$X=u_1G+u_2Q$。如果$X=0$，表示签名无效；否则，$X=(x_1,y_1)$，计算$v=x_1 \bmod n$
  6. 如果$v=r$，表示签名有效；否则表示签名无效

ZHAO ZHEN

基于离散对数问题的数字签名

- 基于离散对数问题的数字签名
  - **ElGamal签名方案、DSS签名方案、Schnorr签名方案都是基于离散对数困难问题的签名方案。**
  - 密钥产生方式具有共同点。
  - 签名方式类似性。

基于离散对数问题的数字签名

- 离散对数签名方案
  - ElGamal签名方案、DSA签名方案、Schnorr签名方案都可以归结为离散对数签名方案的特例。
  - 参数与密钥生成
    - p: 大素数。
    - q: $p-1$或$p-1$的大素因子。
    - g: $g \in_R Z_p^*$，且$g^q \equiv 1 \bmod p$
    - x: 签名者的私钥，$1 \le x \le q-1$。
    - y: 签名者的公钥，$y \equiv g^x \bmod p$。

ZHAO ZHEN

基于离散对数问题的数字签名

- 签名
  - 对待签消息m，计算m的Hash值$h(m)$。
  - 随机选择整数k，$1 \le k \le q-1$，计算$r \equiv g^k \bmod p$。
  - 从签名方程

$$ak \equiv (b+cx) \bmod q$$

中解出s，其中方程的系数a、b、c有多种选择，对m的签名为(r, s)。

- 验证
  - 验证者在收到消息m和签名(r, s)后，验证下面等式是否成立：

$$r^a \equiv g^b y^c \bmod p$$

基于离散对数问题的数字签名

- 参数a，b，c可能的选择

$\pm r'$	$\pm s$	$h(m)$
$\pm r'h(m)$	$\pm s$	1
$\pm r'h(m)$	$\pm h(m)s$	1
$\pm h(m)r'$	$\pm r's$	1
$\pm h(m)s$	$\pm r's$	1

  - 表中$r' \equiv r \bmod q$，每一行的三个值都可以对a、b、c进行不同的组合，每一行都有**24种**不同的组合方式，总共可以得到**120种**基于离散对数问题的数字签名方案。
  - 但不是每一种方案都是安全的。

基于离散对数问题的数字签名

- $\{a,b,c\}=\{r',s,h(m)\}$时的签名和验证方程

	签名方程	验证方程
①	$r'k\equiv s+h(m)x \bmod q$	$r^{r'}\equiv g^{s}y^{h(m)} \bmod p$
②	$r'k\equiv h(m)+sx \bmod q$	$r^{r'}\equiv g^{h(m)}y^{s} \bmod p$
③	$sk\equiv r'+h(m)x \bmod q$	$r^{s}\equiv g^{r'}y^{h(m)} \bmod p$
④	$sk\equiv h(m)+r'x \bmod q$	$r^{s}\equiv g^{h(m)}y^{r'} \bmod p$
⑤	$mk\equiv s+r'x \bmod q$	$r^{m}\equiv g^{s}y^{r'} \bmod p$
⑥	$mk\equiv r'+sx \bmod q$	$r^{m}\equiv g^{r'}y^{s} \bmod p$

- 表中④就是数字签名算法**DSA**的情况，这时**$a=s$, $b=h(m)$, $c=r'$.**

基于身份的签名方案

- **Shamir的基于身份的数字签名方案：**
  1. **初始化**：KGC选择n为两个大素数p、q的乘积，e是与$\varphi(n)$互素的大素数，f是一个单向函数，$<n, e, f>$为公开的参数
  2. **密钥提取**：公开用户身份值i，与i值相对应的私钥为g，满足：$g^e = i \pmod n$
  3. **签名算法**：对消息m签名，首先选择一个随机数r，计算$t = r^e \pmod n$，$s = g \cdot r^{f(t \cdot m)}$，则将签名$(s, t)$发送给验证者
  4. **验证算法**：验证者收到签名(s, t)之后，如果

$$s^e \stackrel{?}{\Leftrightarrow} i \cdot t^{f(t \cdot m)}$$

成立，则接受该签名为有效签名；否则，拒绝该签名

Cha-Cheon的基于身份的数字签名方案

- 1. 设置：生成素数q，两个q阶群$\boldsymbol{G}_1$和$\boldsymbol{G}_2$，一个双线性映射$\hat{e}: \boldsymbol{G}_1 \times \boldsymbol{G}_1 \to \boldsymbol{G}_2$。生成元$P \in \boldsymbol{G}_1$。随机数$s \in Z_p^*$，令$P_{pub}=sP$。Hash函数$H_1: \{0,1\}^* \to \boldsymbol{G}_1^*$；$H_2: \{0,1\}^* \times \boldsymbol{G}_1^* \to Z_p^*$。系统参数$par=<q, \boldsymbol{G}_1, \boldsymbol{G}_2, \hat{e}, P, P_{pub}, H_1, H_2>$，主密钥为$s$
  2. 析出：给定$ID \in \{0,1\}^*$，KGC 计算$Q_{ID}=H_1(ID)$，私钥$d_{ID}=sQ_{ID}$
  3. 签名：给定消息m，选取随机数$r \in Z_p^*$，计算$U=rQ_{ID}$，$h=H_2(m,U)$和$V=(r+h)d_{ID}$，签名$\sigma=(U,V)$
  4. 验证：验证$\hat{e}(P,V)=\hat{e}(P_{pub}, U+hQ_{ID})$是否成立
- Cha和Cheon在**RO模型下证明了方案的安全性**，他们的思想被普遍应用于基于身份的签名方案的安全性证明中

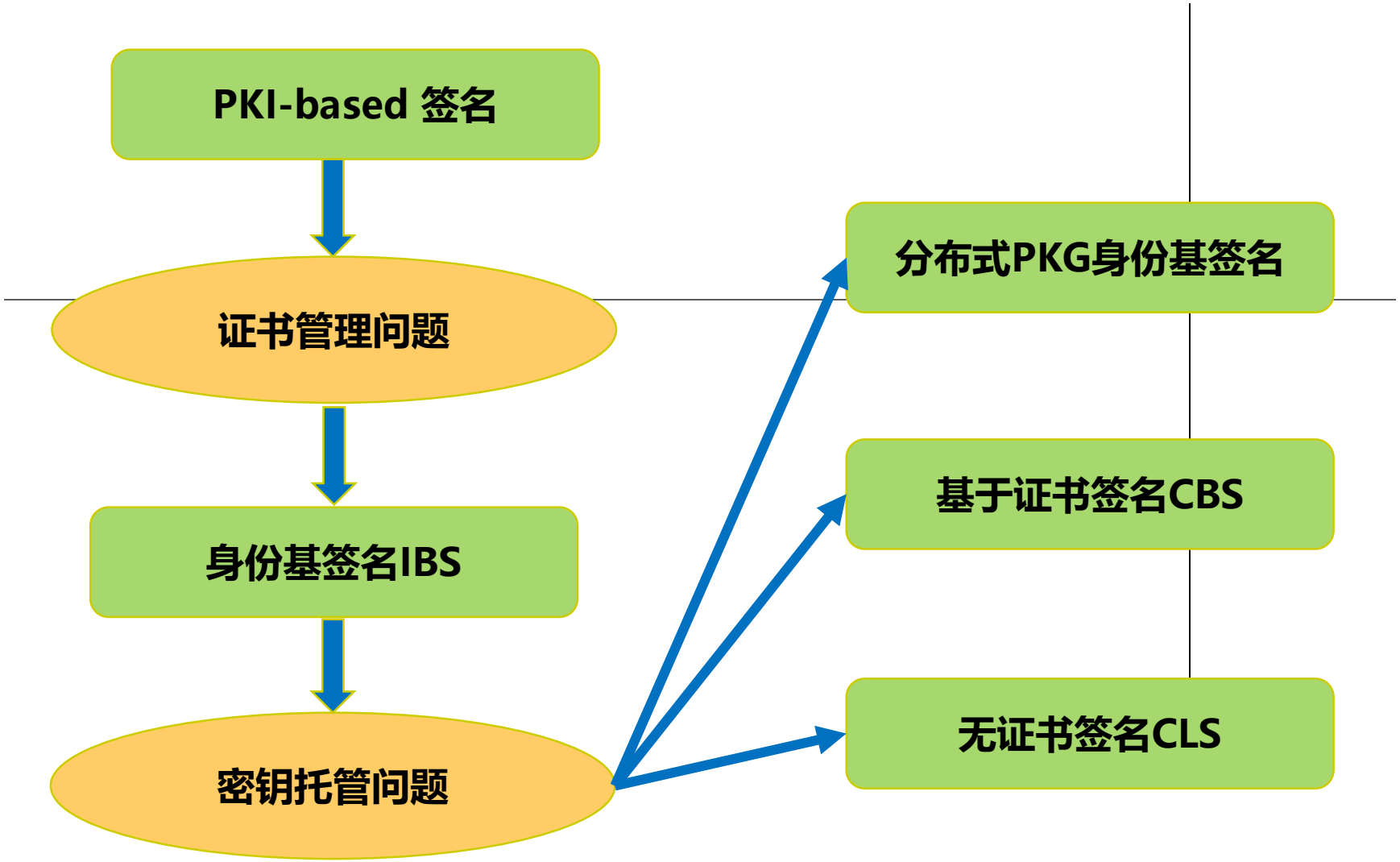
其它签名算法

GOST签名标准，为俄国采用的数字签名标准，自1995启用，正式称为GOST R34.10-94。算法与Schnorr模式下的ElGamal签名及NIST的DSA很相似。算法中也有一个类似于SHA的杂凑函数$H(x)$，其标准号为GOST R34. 11-94。

ESIGN签名体制。日本NTT的T. Okamoto等设计的签名方案。宣称在密钥签名长度相同条件下，至少和RSA，DSA一样安全，且比它们都快。

OSS签名体制，Ong，Schnorr和Shamir[1984]提出的一种利用$mod\ n$下多项式的签名算法。方案基于二次多项式。

拓展签名概念


PKI-based 签名
证书管理问题
身份基签名IBS
密钥托管问题
分布式PKG身份基签名
基于证书签名CBS
无证书签名CLS

从签名者角度出发：

- 传统PKI-based 数字签名中，签名者“能”
  - 知道即将发布的消息内容M
  - 能自己一个人完成签名计算
  - 能用其私钥对任意消息进行签名
  - 能用其私钥进行无限制的签名
- 传统PKI-based 数字签名中，签名者“不能”
  - 不能控制签名的验证（任何人都能验证）
  - 不能在不给私钥的前提下让其他人完成签名

ZHAO ZHEN

从验证者角度出发：

- 传统PKI-based 数字签名中，验证者“能”
  - 能自己验证签名的正确性
  - 能获得签名者的签名
  - 能知道签名者是谁
  - 能知道被签的消息是M
  - 能将该签名转发给别人且别人可验证该签名的合法性
- 传统PKI-based 数字签名中，验证者“不能”
  - 不能对签名者签过名的消息内容进行处理

ZHAO ZHEN

- 传统PKI-based 数字签名中，签名者“能”
  - 知道即将发布的消息内容M

转换为

  - 不知道即将发布的消息内容M

盲签名，Blind Signature (1982, David Chaum)

ZHAO ZHEN

- 盲签名, **Blind Signature**：签名者不知道即将发布的消息内容M
  - 在协议完成之后，签名计算者只知道他对某一个消息完成了签名，签名接收者成功获得一枚对消息M的数字签名，而且消息M是由签名接收者决定的
  - 应用场景：消息不重要，重要的是签名一个签名。对应一百块，所有人可以花钱从signer处买签名。Signer对未知消息M签名发送给不同的接收者，接收者可以将该签名用于不同商品的购物，商家可以用该签名从signer处换取金钱，而由于signer并不能从消息和签名得出其与接收者的联系，因此达到了消费者匿名。

盲签名

- 在选举投票和数字货币协议中将会碰到这类要求。设B是一位仲裁人，A要B签署一个文件，但不想让他知道所签的是什么，而B也并不关心所签的内容，他只是要确保在需要时可以对此进行仲裁。可通过下述协议实现。

ZHAO ZHEN

盲签名

(a) A取一文件并以一随机值乘之。称此随机值为**盲因子**，称用**盲因子**乘后的文件为盲文件。

(b) A将此盲文件送给B。

(c) B对盲文件签名。

(d) A以盲因子除之，得到B对原文件的签名。

（若签名函数和乘法函数是可换的，则上述作法成立。否则要采用其它方法(而不是乘法)修改原文件。）

盲签名

安全性讨论

B可以欺诈吗？是否可以获取有关文件的信息？

若盲因子完全随机，则可保证B不能由(b)中所看到的盲文件得出原文件的信息。即使B将(c)中所签盲文件复制，他也不能(对任何人)证明在此协议中所签的真正文件，而只是知道其签名是合法的，并可向他人证实其签名合法。即使他签了100万个文件，也无从得到所签文件的信息。

盲签名算法

D. Chaum曾提出第一个实现盲签名的算法，他采用了RSA算法。令B的公钥为e，秘密钥为d，模为n。

(a) A需要B对消息m进行盲签名，选$1<k<n$，作$t=mk^e(\mathrm{mod}n)\rightarrow$B。

(b) B对t签名，$t^d=(mk^e)^d(\mathrm{mod}n)\rightarrow$A。

(c) A计算 $s=t^d/k(\mathrm{mod}n)$ 得

$$s=(mk^e)^d/k(\mathrm{mod}n)=m^d(\mathrm{mod}n),$$

(m, s)就是B对m按RSA体制的合法签名，任何知道公钥e的人都能验证$s^e=m(\mathrm{mod}n)$。

➢ 如何定义数字签名PKS的算法？

$Setup(\lambda) \rightarrow SP$
$KeyGen(SP) \rightarrow (pk, sk)$
$SignPro(Signer: SP, sk; Verifier: pk, M) \rightarrow \sigma$
$Verify(SP, pk, M, \sigma) \rightarrow \{0, 1\}$

数字签名的形式化算法定义

➢ 数字签名安全模型：抗选择消息攻击的存在不可伪造性（EU-CMA）

(Existential inforgeability against chosen message attacks)

挑战者C

方案

敌手A

1. 系统初始化, 运行**Setup**
2. 运行**KeyGen**, 生成Signer公私钥对(pk, sk)，保留sk

SP, pk

Signature Query

3. 想要询问消息M_1,...M_n对应的签名

SignPro

4. 运行**SignPro**

$(M_1, \ldots, M_n, M_{n+1}, \sigma_1, \ldots, \sigma_n, \sigma_{n+1})$

5. 攻击

n+1

如果：$\sigma_1, \ldots, \sigma_n, \sigma_{n+1}$是$M_1, \ldots, M_n, M_{n+1}$的合法签名;

则敌手攻击成功。令 Pr[A successfully attack the PKS scheme] ≤ ε， ε 是可忽略的。

- 传统PKI-based 数字签名中，签名者“能”
  - 能自己一个人完成签名计算

转换为

  - 不能自己一个人完成签名计算

门限签名，Threshold Signature

ZHAO ZHEN

- 门限签名，**Threshold Signature**：签名者不能自己一个人完成签名计算
  - **(n,t)-门限签名，一共有n个成员，至少t位成员签字，才能够最终聚合成一个合法签名，即要求参与的签名者数量满足最低的门限值**
  - 优点
    a. 虽然是t个用户签名，但最终会聚合成一个签名
    b. 且只需要用一个系统公钥去验证即可
    c. 验证者不知道签名者是哪些人
  - 缺点
    a. 用户更新时引起的密钥更新问题
  - 应用场景：董事会通过某项决议

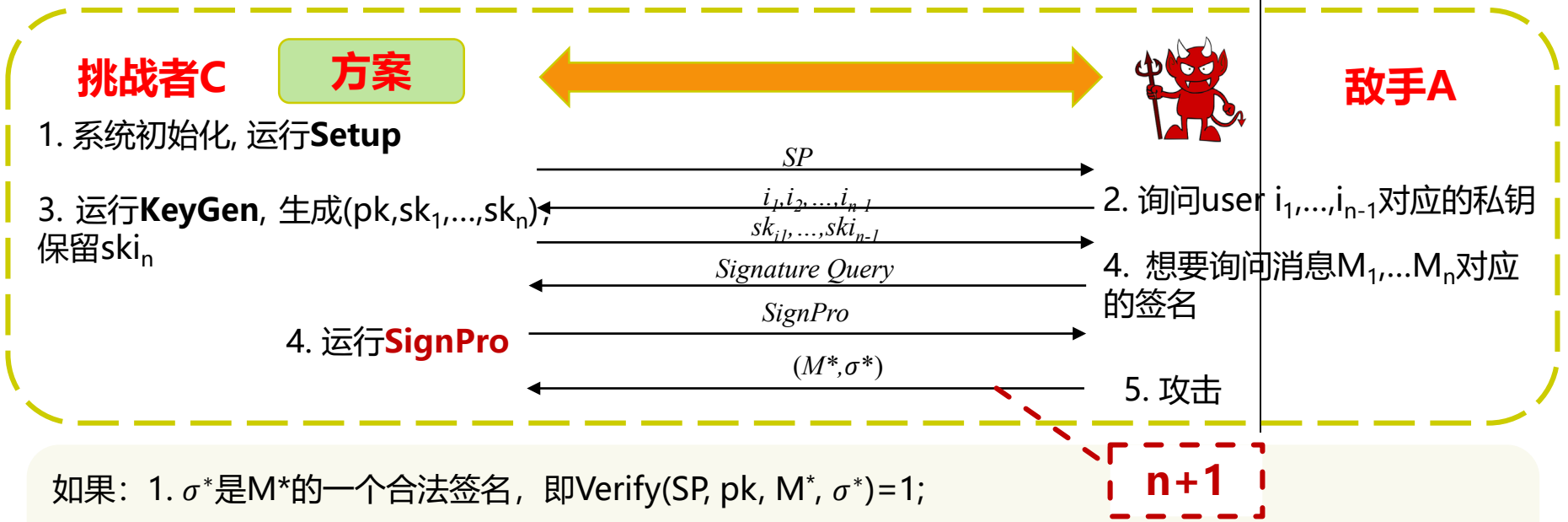
➢ 如何定义门限签名的算法？

$Setup(\lambda) \rightarrow SP$
$KeyGen(SP) \rightarrow (pk, sk_1, \ldots, skn)$
$SignPro(t\ Signer\ i: SP, ski) \rightarrow \sigma$
$Verify(SP, pk, M, \sigma) \rightarrow \{0, 1\}$

数字签名的形式化算法定义

➢ 门限签名安全模型：抗选择消息攻击的存在不可伪造性（EU-CMA）

(Existential inforgeability against chosen message attacks)



如果：1. σ^*是M*的一个合法签名，即Verify(SP, pk, M^*, σ^*)=1;

2. M*是一个新消息，即M*没有被敌手询问过；

则敌手攻击成功。令 Pr[A successfully attack the PKS scheme] ≤ ε， ε 是可忽略的。

- 传统PKI-based 数字签名中，签名者“能”
  - 能用其私钥对任意消息进行签名

转换为

  - 不能用其私钥对任意消息进行签名

防止双重认证签名

Double Authentication Preventing Signature

ZHAO ZHEN

- 防止双重认证签名：签名者不能用其私钥对任意消息进行签名
  - 针对一个消息，Signer如果进行两次签名，则verifier可以从这两个签名中得到signer的私钥
  - 应用场景：例如电子货币，一个电子货币只能消费一次，如果商家拿到两个电子货币对应的签名，可以从中得到不法消费者的私钥；例如其他要求只能对同一个消息只能进行一次签名的场景

ZHAO ZHEN

➢ 如何定义防止双重认证签名的算法？

$Setup(\lambda) \rightarrow SP$

$KeyGen(SP) \rightarrow (pk, sk)$

$Sign(SP, sk, M) \rightarrow \sigma$

$Verify(SP, pk, M, \sigma) \rightarrow \{0, 1\}$

Ext(SP, $Sign(SP, sk, M1) \rightarrow \sigma 1, Sign(SP, sk, M2) \rightarrow \sigma 2$))→sk

数字签名的形式化算法定义

➢ 防止双重认证签名安全模型：抗选择消息攻击的存在不可伪造性（EU-CMA）

挑战者C

方案

敌手A

1. 系统初始化, 运行**Setup**
2. 运行**KeyGen**, 生成Signer公私钥对(pk, sk)，保留sk

SP, pk →

← *Signature Query: M*

3. 询问消息M对应的签名

← *Sign Results:*σ

4. 运行**Sign**, 生成M的签名σ

← (M^*, σ^*)

5. 攻击

如果：1. σ^*是M*的一个合法签名，即Verify(SP, pk, M^*, σ^*)=1;

2. M*是一个新消息，即M*没有被敌手询问过;

则敌手攻击成功。令 Pr[A successfully attack the PKS scheme] ≤ ε， ε 是可忽略的。

- 传统PKI-based 数字签名中，签名者“能”
  - 能用其私钥进行无限制的签名

转换为

  - 不能用其私钥进行无限制的签名

使用次数有限签名

ZHAO ZHEN

- 使用次数有限签名：签名者不能用其私钥进行无限制的签名
  - 签名者使用其私钥只能进行有上限次数的签名
  - 应用场景：例如董事会的有限次签名授权
  - 构造：可基于防止双重认证+有限消息空间

ZHAO ZHEN

- 传统PKI-based 数字签名中，签名者“不能”

  - 不能控制签名的验证（任何人都能验证）

  转换为

  - 能控制签名的验证（不是任何人都能验证）

不可否认签名 Undeniable Signature

ZHAO ZHEN

- 不可否认签名：能控制签名的验证（不是任何人都能验证，只有部分人可以验证）
  - 验证者需要签名者的帮助才能够进行验证
  - 应用场景：会员制度的签名验证，只有缴费会员才能够得到服务器的支持去验证（消息，签名）的合法性

ZHAO ZHEN

➢ 如何定义防止双重认证签名的算法？

$Setup(\lambda) \rightarrow SP$

$KeyGen(SP) \rightarrow (pk, sk)$

$Sign(SP, sk, M) \rightarrow \sigma$

$VerifyPro(Signer: SP, sk, Verifier: SP, pk, M, \sigma) \rightarrow \{0,1\}$

数字签名的形式化算法定义

➢ 不可否认签名安全模型：抗选择消息攻击的存在不可伪造性（EU-CMA）

挑战者C

方案

敌手A

1. 系统初始化, 运行**Setup**
2. 运行**KeyGen**, 生成Signer公私钥对(pk, sk)，保留sk

SP, pk →

← *Signature Query: M*

3. 询问消息M对应的签名

← *Sign Results:*σ

4. 运行**Sign**, 生成M的签名σ

← (M^*, σ^*)

5. 攻击

如果：1. σ^*是M*的一个合法签名，即Verify(SP, pk, M^*, σ^*)=1;

2. M*是一个新消息，即M*没有被敌手询问过;

则敌手攻击成功。令 Pr[A successfully attack the PKS scheme] ≤ ε， ε 是可忽略的。

- 传统PKI-based 数字签名中，签名者“不能”
  - 不能在不给私钥的前提下让其他人完成签名

转换为

  - 能在不给私钥的前提下让其他人完成签名

代理重签名 Proxy Re-Signature

ZHAO ZHEN

- 代理重签名：能在不给私钥的前提下让其他人完成签名
  - 签名者A和B使用其私钥为代理生成一个代理密钥，代理可以基于此代理密钥将别的签名者A的签名转换成签名者B的签名
  - 应用场景：第三方机构作为代理监管下的签名转换

ZHAO ZHEN

➢ 如何定义代理重签名的算法？

$Setup(\lambda) \rightarrow SP$

$KeyGen(SP) \rightarrow (pk, sk)$

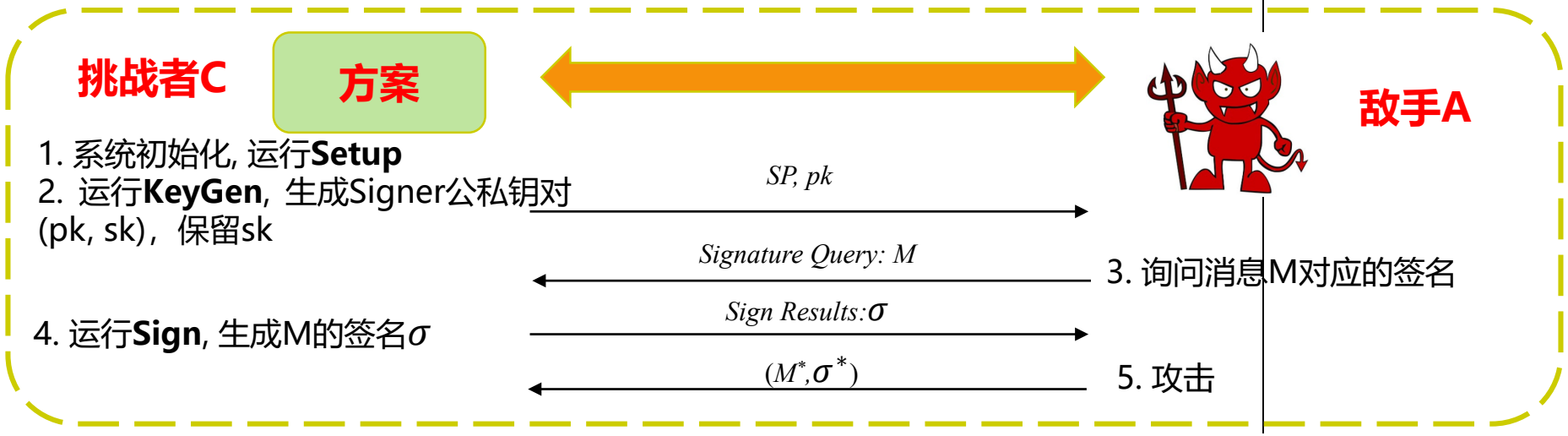
$ProxyKeyGen(SP, skA, skB) \rightarrow (sk_{A \rightarrow B})$

$Sign(SP, sk, M) \rightarrow \sigma$

$ReSig(SP, sk_{A \rightarrow B}, Sign(SP, sk_A, M) \rightarrow \sigma_A) \rightarrow \sigma_B \subseteq Sign(SP, skB, M)$

数字签名的形式化算法定义

➢ 代理重签名安全模型：抗选择消息攻击的存在不可伪造性（EU-CMA）



如果：1. σ^*是M*的一个合法签名，即Verify(SP, pk, M*, σ^*)=1;

2. M*是一个新消息，即M*没有被敌手询问过；

则敌手攻击成功。令 Pr[A successfully attack the PKS scheme] ≤ ε， ε 是可忽略的。

从验证者角度出发：

- 传统PKI-based 数字签名中，验证者“能”
  - 能自己验证签名的正确性
  - 能获得签名者的签名
  - 能知道签名者是谁
  - 能知道被签的消息是M
  - 能将该签名转发给别人且别人可验证该签名的合法性
- 传统PKI-based 数字签名中，验证者“不能”
  - 不能对签名者签过名的消息内容进行处理

ZHAO ZHEN

- 传统PKI-based 数字签名中，验证者“能”
  - 能自己验证签名的正确性

转换为

  - 不能自己验证签名的正确性

门限签名验证 Threshold Signature Verification

ZHAO ZHEN

- 传统PKI-based 数字签名中，验证者“能”

  - 能获得签名者的签名

  转换为

  - 不能获得签名者的签名

可验证加密签名

ZHAO ZHEN

- 传统PKI-based 数字签名中，验证者“能”
  - 能知道被签的消息是M

转换为

  - 不能知道签名者是谁

群签名、环签名、属性基签名

Group、Ring、Attribute-based Signature

ZHAO ZHEN

特殊用途的数字签名

群签名

群签名(Group Signature)是面向群体密码学中的一个课题，1991年由Chaum和van Heyst提出。它有下述几个特点：

①只有群中成员能代表群体签名；

② 接收到签名的人可以用公钥验证群签名，但不可能知道由群体中那个成员所签；

③ 发生争议时可由群体中的成员或可信赖机构识别群签名的签名者。

这类签名可用于投标中，以防止作弊。

特殊用途的数字签名

- 群签名组成
  - (1)建立：一个用以产生群公钥和私钥的多项式概率算法。
  - (2)加入：一个用户和群管理员之间的交互式协议。
  - (3)签名：一个概率算法，当输入一个消息、一个群成员的私钥和一个群公钥后，输出对该消息的签名。
  - (4)验证：给定一个消息的签名和一个群公钥后，判断该签名相对于该群公钥是否有效。
  - (5)打开：给定一个签名、群公钥和群私钥的条件下确定签名者的身份。

特殊用途的数字签名

- 群签名方案满足的性质
  - (1)正确性：由群成员使用群签名算法产生的群签名，必须为验证算法所接受。
  - (2)不可伪造性：只有群成员才能代表群体进行签名。
  - (3)匿名性：给定一个合法的群签名，除了群管理员外，任何人要识别出签名者的身份在计算上是困难的。
  - (4)不可关联性：除了群管理员外，任何人要确定两个不同的群签名是否为同一群成员所签在计算上是困难的。
  - (5)可跟踪性：群管理员总能打开一个群签名以识别出签名者的身份。
  - (6)可开脱性：群管理员和群成员都不能代表另一个群成员产生出有效的群签名，群成员不必为他人产生的群签名承担责任。
  - (7)抗联合攻击：即使一些群成员串通在一起也不能产生一个合法的群签名，使得群管理员不能跟踪该签名。

- 传统PKI-based 数字签名中，验证者“能”

  - 能知道签名者是谁

  转换为

  - 不能知道被签的消息是M

功能签名 Functional Signature

ZHAO ZHEN

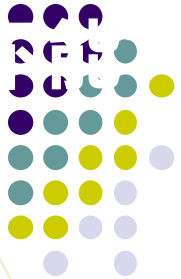
- 传统PKI-based 数字签名中，验证者“能”
  - 能将该签名转发给别人且别人可验证该签名的合法性

转换为

  - 不能将该签名转发给别人且别人可验证该签名的合法性

指定验证者签名 Designated Verifier Signature

ZHAO ZHEN



- 传统PKI-based 数字签名中，验证者“不能”

  - 不能对签名者签过名的消息内容进行处理

  转换为

  - 能对签名者签过名的消息内容进行处理

同态签名 Homomorphic Signature

ZHAO ZHEN

- **Motivation:** 在传统**PKI-based PKS** 以及身份基签名　**IBS**中，一个签名有且只有一个相关的签名者，是否可以放松条件？

属性基签名，**Attribute-based signature**

ZHAO ZHEN

➢ 如何定义属性基签名ABS的算法？

PKG

$Setup(\lambda) \rightarrow SP$

$KeyGen(SP, att) \rightarrow (d_{att})$

$Sign(SP, d_{att}, M) \rightarrow \sigma$

$Verify(SP, att, M, \sigma) \rightarrow \{0, 1\}$

数字签名的形式化算法定义

➢ ABS安全模型：抗选择消息攻击的存在不可伪造性（EU-att-CMA）

(Existential inforgeability against chosen message and adaptively chosen attributes attacks)

挑战者C **方案** **敌手A**

1. 系统初始化, 运行**Setup**

SP

att

2. 询问属性att对应的私钥

d_{att}

3. 运行**KeyGen**, 生成att对应的私钥d_{att}

Signature Query: M

4. 询问消息M对应的签名

Sign Results:σ

4. 运行**Sign**, 生成M的签名σ

(M^*, σ^*, att^*)

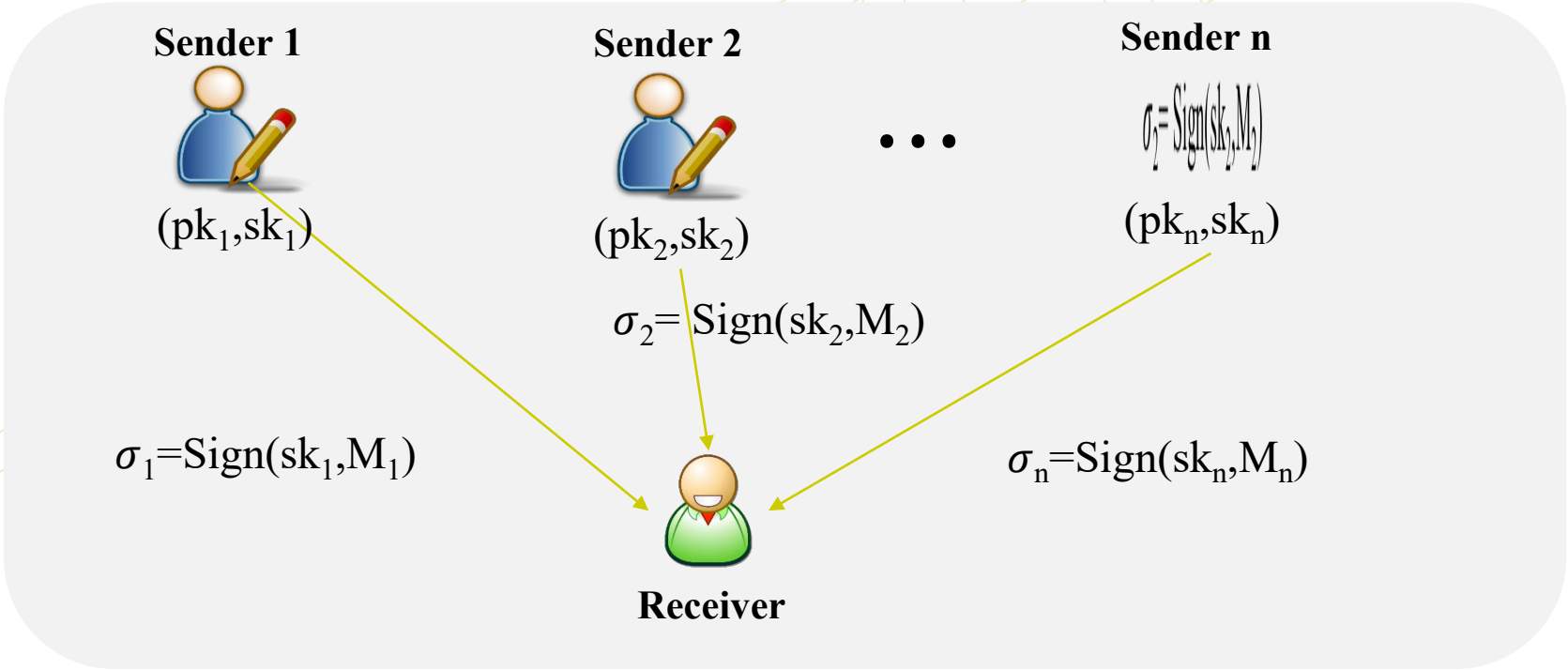
5. 攻击(att*未被询问过私钥)

如果：1. σ^*是M*的一个合法签名，即Verify(SP, att, M^*, σ^*)=1;

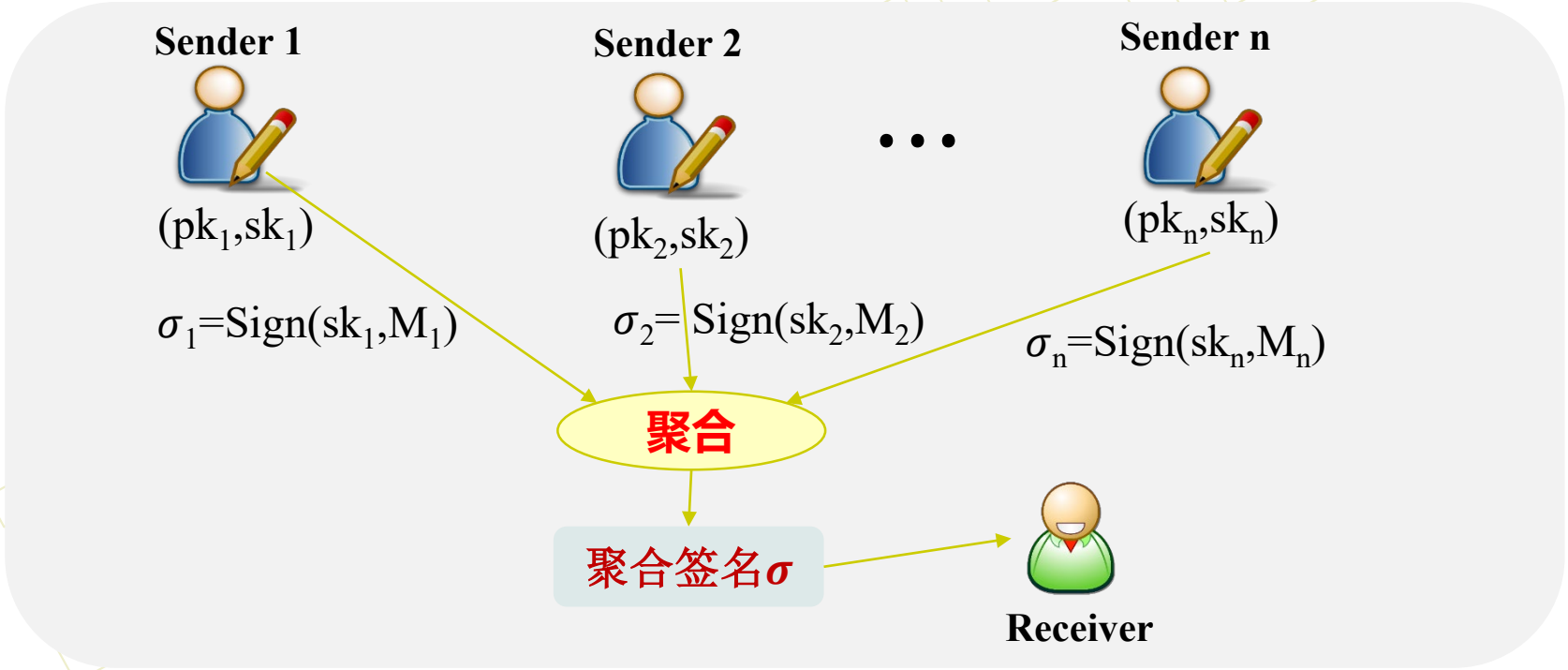
2. M*是一个新消息，即M*没有被敌手询问过；

则敌手攻击成功。令 Pr[A successfully attack the PKS scheme] ≤ ε， ε 是可忽略的。

- **Motivation: 在PKI-based PKS中，对于下面的场景，如何提高效率？**



- **Motivation: 在PKI-based PKS中，对于下面的场景，如何提高效率？Aggregate signature: 聚合**



- Dan Boneh, Aggregate and Verifiably Encrypted Signatures from Bilinear Maps, Eurocrypt 2003

➢ 如何定义聚合签名AS的算法？

$Setup(\lambda) \rightarrow SP$
$KeyGen(SP) \rightarrow (pk_i, ski)$
$Sign(SP, sk_i, Mi) \rightarrow \sigma_i$
$Verify(SP, pki, Mi, \sigma_i) \rightarrow \{0,1\}$
$Agg(SP, \sigma_1, \dots, \sigma_n) \rightarrow \sigma$
$AggVer(SP, pk_1, \dots, pkn, M_1, \dots, Mn, \sigma_1, \dots, \sigma_n) \rightarrow \{0,1\}$

数字签名的形式化算法定义

➢ AS安全模型：抗选择消息攻击的存在不可伪造性（EU-CMA）

挑战者C

方案

敌手A

1. 系统初始化, 运行**Setup**
2. 运行**KeyGen**, 生成Signer公私钥对 (**pk_1, sk_1**)，保留sk_1

SP, pk_1 →

← *Signature Query: M*

Sign Results:σ →

← (pk_2,…,pk_n, M_1,…,M_n,σ)

4. 运行**Sign**, 生成M的签名σ

5. 攻击

如果：1.σ是一个合法签名，即$AggVer(SP, pk_1, \dots, pkn, M_1, \dots, Mn, \sigma_1, \dots, \sigma_n)$=1;

2. M_1是一个新消息，即M_1没有被敌手询问过；

则敌手攻击成功。令 Pr[A successfully attack the PKS scheme] ≤ ε， ε 是可忽略的。

The scheme comprises five algorithms: *KeyGen*, *Sign*, *Verify*, *Aggregate*, and *AggregateVerify*. The first three are as in ordinary signature schemes; the last two provide the aggregation capability. The scheme employs a full-domain hash function $h : \{0,1\}^* \to G_2$, viewed as a random oracle.

Key Generation. For a particular user, pick random $x \stackrel{R}{\leftarrow} \mathbb{Z}_p$, and compute $v \leftarrow g_1^x$. The user's public key is $v \in G_1$. The user's secret key is $x \in \mathbb{Z}_p$.

Signing. For a particular user, given the secret key x and a message $M \in \{0,1\}^*$, compute $h \leftarrow h(M)$, where $h \in G_2$, and $\sigma \leftarrow h^x$. The signature is $\sigma \in G_2$.

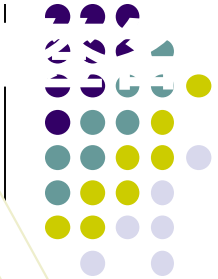
Verification. Given user's public key v, a message M, and a signature σ, compute $h \leftarrow h(M)$; accept if $e(g_1, \sigma) = e(v, h)$ holds.

Aggregation. For the aggregating subset of users $U \subseteq \mathbb{U}$, assign to each user an index i, ranging from 1 to $k = |U|$. Each user $u_i \in U$ provides a signature $\sigma_i \in G_2$ on a message $M_i \in \{0,1\}^*$ of his choice. The messages M_i must all be distinct. Compute $\sigma \leftarrow \prod_{i=1}^{k} \sigma_i$. The aggregate signature is $\sigma \in G_2$.

Aggregate Verification. We are given an aggregate signature $\sigma \in G_2$ for an aggregating subset of users U, indexed as before, and are given the original messages $M_i \in \{0,1\}^*$ and public keys $v_i \in G_1$ for all users $u_i \in U$. To verify the aggregate signature σ,

1. ensure that the messages M_i are all distinct, and reject otherwise; and
2. compute $h_i \leftarrow h(M_i)$ for $1 \le i \le k = |U|$, and accept if $e(g_1, \sigma) = \prod_{i=1}^{k} e(v_i, h_i)$ holds.

- Dan Boneh, Aggregate and Verifiably Encrypted Signatures from Bilinear Maps, Eurocrypt 2003



- 聚合签名：

Motivation: 将n个签名聚合成一个签名σ，然后使用n个消息M1, M2, …, Mn以及该聚合签名σ对这n个签名进行验证，即，聚合签名“imply”批量验证

- 批量验证签名：

Motivation: 同时验证 n 个签名，支持批量验证的签名方案不一定可以聚合签名

ZHAO ZHEN

- BLS签名介绍
  - 2001年Asaiacrypt,简称 BLS签名方案

Short Signatures from the Weil Pairing*

Dan Boneh† dabo@cs.stanford.edu
Ben Lynn blynn@cs.stanford.edu
Hovav Shacham hovav@cs.stanford.edu

  - 标准安全参数、相同的安全级别下，BLS的签名的尺寸是DSA签名的一半。

“We introduce a short signature scheme based on the Computational Diffie-Hellman assumption on certain elliptic and hyper-elliptic curves. For standard security parameters, the signature length is about half that of a DSA signature with a similar level of security.”

➢ BLS签名方案

- 概念回顾：数字签名算法定义

$Setup(\lambda) \rightarrow SP$

$KeyGen(SP) \rightarrow (pk, sk)$

$Sign(SP, sk, M) \rightarrow \sigma$

$Verify(SP, pk, M, \sigma) \rightarrow \{0,1\}$

数字签名的形式化算法定义

正确性：对于任意的$SP \leftarrow Setup(\lambda), (pk, sk) \leftarrow KeyGen(SP), M \in$ 消息空间$, \sigma \leftarrow Sign(SP, sk, M)$, 有$1 \leftarrow Verify(SP, pk, M, \sigma)$.

- BLS签名算法构造

SysGen: The system parameter generation algorithm takes as input a security parameter λ. It chooses a pairing group $\mathbb{PG} = (\mathbb{G}, \mathbb{G}_T, g, p, e)$ and a cryptographic hash function $H : \{0,1\}^* \rightarrow \mathbb{G}$. The algorithm returns the system parameter $SP = (\mathbb{PG}, H)$.

KeyGen: The key generation algorithm takes as input SP. It randomly chooses $\alpha \in \mathbb{Z}_p$ and computes $h = g^\alpha$. The algorithm returns a public/secret key pair (pk, sk) as follows.

$$pk = h, \quad sk = \alpha.$$

Sign: The signing algorithm takes as input a message $m \in \{0,1\}^*$, the secret key sk and SP. It returns the signature σ_m on m as

$$\sigma_m = H(m)^\alpha.$$

Verify: The verification algorithm takes as input a message-signature pair (m, σ_m), the public key pk and SP. It accepts the signature if

$$e(\sigma_m, g) = e(H(m), h).$$

➢ BLS签名方案

- BLS签名算法构造

$Setup(\boldsymbol{\lambda}) \rightarrow \boldsymbol{SP = (PG, H)}$

$KeyGen(\boldsymbol{SP}) \rightarrow (\boldsymbol{pk = h = g^{\alpha}, sk = \alpha})$

$Sign(\boldsymbol{SP, sk, M}) \rightarrow \boldsymbol{\sigma = H(m)^{\alpha}}$

$Verify(\boldsymbol{SP, pk, M, \sigma}) \rightarrow \boldsymbol{\{0, 1\}}$

SysGen: The system parameter generation algorithm takes as input a security parameter λ. It chooses a pairing group $\mathbb{PG} = (\mathbb{G}, \mathbb{G}_T, g, p, e)$ and a cryptographic hash function $H : \{0,1\}^* \rightarrow \mathbb{G}$. The algorithm returns the system parameter $SP = (\mathbb{PG}, H)$.

KeyGen: The key generation algorithm takes as input SP. It randomly chooses $\alpha \in \mathbb{Z}_p$ and computes $h = g^{\alpha}$. The algorithm returns a public/secret key pair (pk, sk) as follows.

$$pk = h, \quad sk = \alpha.$$

Sign: The signing algorithm takes as input a message $m \in \{0,1\}^*$, the secret key sk and SP. It returns the signature σ_m on m as

$$\sigma_m = H(m)^{\alpha}.$$

Verify: The verification algorithm takes as input a message-signature pair (m, σ_m), the public key pk and SP. It accepts the signature if

$$e(\sigma_m, g) = e(H(m), h).$$

- 正确性验证：

对于任意的$SP = (PG, H) \leftarrow Setup(\lambda), (pk = h = g^{\alpha}, sk = \alpha) \leftarrow KeyGen(SP), M \in \{0,1\}^*, \sigma = H(m)^{\alpha} \leftarrow Sign(SP, sk, M)$，有

$$e(\sigma, g) = e(H(m)^{\alpha}, g) = e(H(m), g^{\alpha}) = e(H(m), h).$$

故$1 \leftarrow Verify(SP, pk, M, \sigma)$，BLS签名方案的正确性验证完毕。

➢ BLS签名方案

- BLS的EU-CMA安全模型回顾

挑战者C　方案　敌手A

1. 系统初始化, 运行**Setup**

2. 运行**KeyGen**, 生成Signer公私钥对(pk, sk)，保留sk

4. 运行**Sign**, 生成M的签名σ

SP, pk →

$Signature\ Query: M$ ←

$Sign\ Results: \sigma$ →

(M^*, σ^*) ←

5. 攻击

如果：1. σ^*是M*的一个合法签名，即Verify(SP, pk, M^*, σ^*)=1;

2. M*是一个新消息，即M*没有被敌手询问过；

则敌手攻击成功。

➢ BLS签名方案

- BLS签名方案在EU-CMA安全模型下的安全定义

  如果任何PPT敌手在EU-CMA游戏中胜利的优势为$\epsilon(\lambda)$，且$\epsilon(\lambda)$是可忽略的，则**BLS方案在EU-CMA安全模型下是安全的**。

如果：1. σ^*是M*的一个合法签名，即Verify(SP, pk, M^*, σ^*)=1;

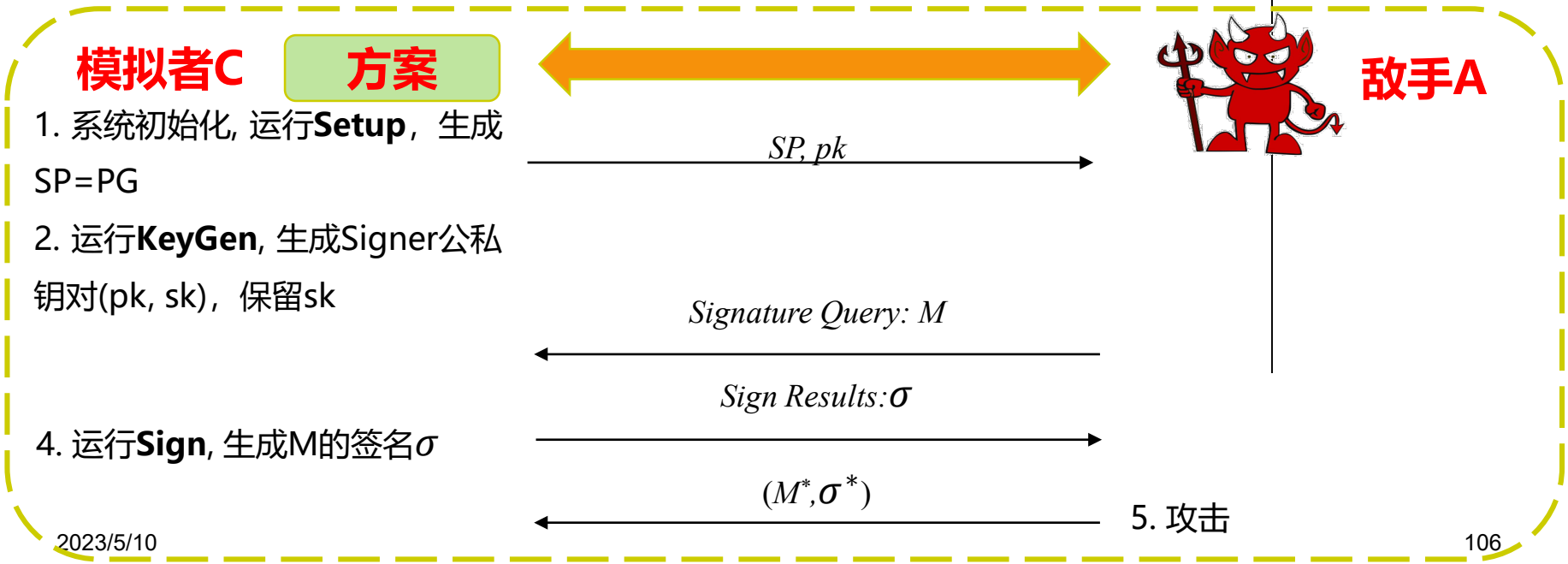
2. M*是一个新消息，即M*没有被敌手询问过；则敌手攻击成功。

➢ BLS签名方案安全归约证明前置知识

- **随机谕言机：**通常用来表示理想的哈希函数H。哈希函数与随机谕言机在安全归约中的区别如下：
  - 计算方法：安全归约证明中，给定输入x，敌手自行计算哈希函数H(x)；但若 H 被设置为随机谕言机，敌手需向随机谕言机询问H(x)。
  - 输入：两者有相同的输入空间。哈希函数的输入次数允许是指数级别，但随机谕言机只允许多项式次数的输入。
  - 输出：两者有相同的输出空间。哈希函数的输出由给定输入x和哈希函数H决定，但随机谕言机的输出由它的控制者--模拟者决定。
- **随机谕言机模型：**在随机谕言机模型下的安全证明，即方案中至少有一个哈希函数被视为随机谕言机。
- **标准模型：**不使用随机谕言机模型的安全证明，称为标准模型下的安全证明。

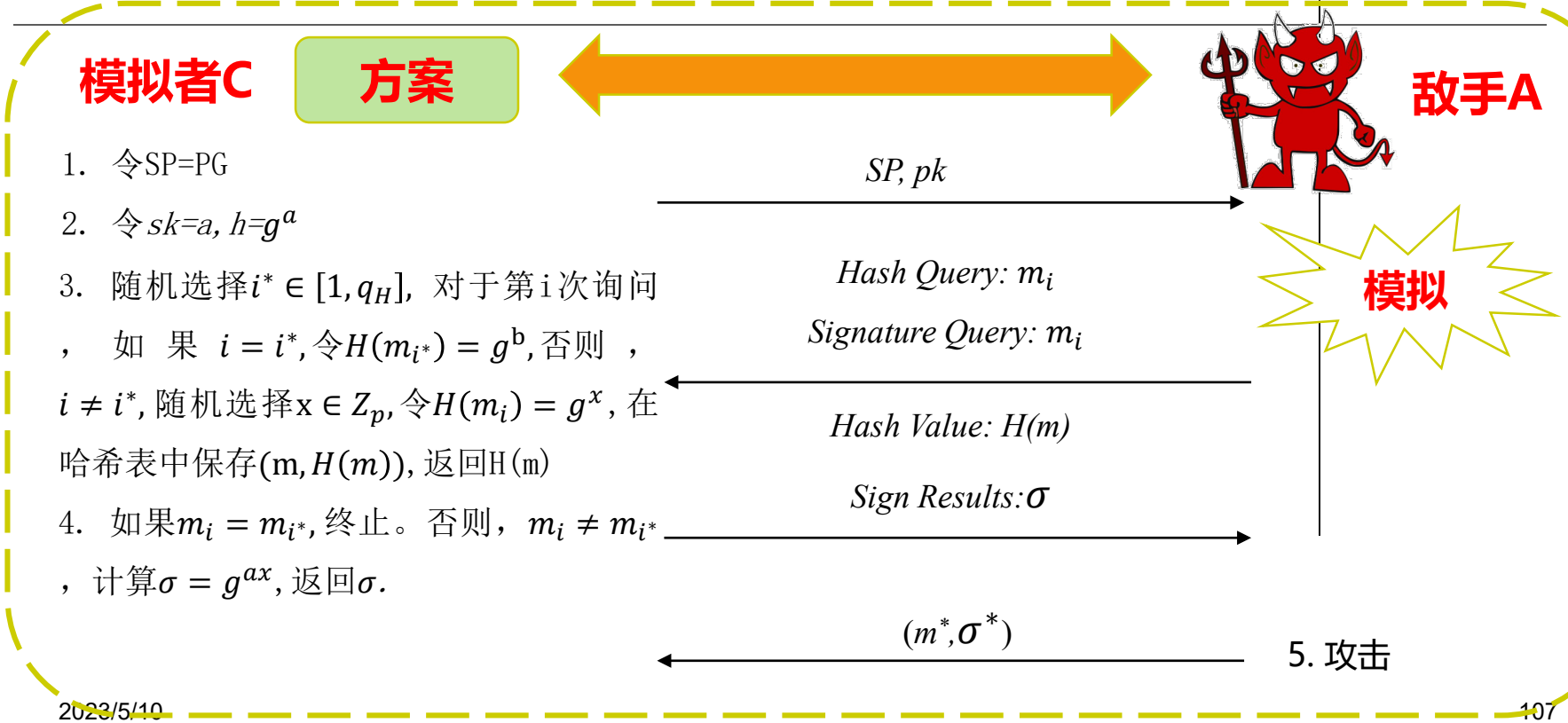
➢ BLS签名方案：安全归约证明

- 假设H是一个随机谕言机，BLS方案在EU-CMA安全模型下的安全性可被归约到CDH困难假设。
- 安全假设：CDH：给定： $g, g^a, g^b \in G$，求解： g^{ab}是困难的。
- ~~假设存在一个PPT敌手A可以在EU-CMA安全模型下攻破BLS方案，我们可~~以构造一个模拟者B解决CDH问题。



➢ BLS签名方案：安全归约证明

- 假设H是一个随机谕言机，假设存在一个PPT敌手A可以在EU-CMA安全模型下攻破BLS方案，我们可以构造一个模拟者B解决CDH问题。假设敌手A可以进行q_H次随机谕言机询问，q_S次签名询问。



➢ BLS签名方案：安全归约证明

- 假设H是一个随机谕言机，BLS方案在EU-CMA安全模型下的安全性可被归约到CDH困难假设。

安全假设：CDH：给定：$g, g^a, g^b \in G$，求解：g^{ab}是困难的。

攻击：敌手A返回(m^*, σ^*)，若$m^* \neq m_{i^*}$，终止。否则，$m^* = m_{i^*}$，$H(m^*) = g^b$，模拟者B输出

$$g^{ab} = \sigma^*$$

分析：1. 对于敌手A，模拟者模拟的方案与安全模型中的方案不可区分。

2. 攻击结果的正确性。

3. 模拟者输出正确攻击的概率是不可忽略的。

只有当敌手在攻击时选择$m^* \neq m_{i^*}$，模拟不终止且攻击有效，即概率为$\frac{1}{q_H}$.

假设在EU-CMA安全模型下攻破BLS方案的PPT敌手A的优势为ϵ，此时模拟者B可以解决CDH问题的优势为$\frac{\varepsilon}{q_H}$。

➢ BB^{RO}签名方案

- 概念回顾：数字签名算法定义

$Setup(\lambda) \rightarrow SP$

$KeyGen(SP) \rightarrow (pk, sk)$

$Sign(SP, sk, M) \rightarrow \sigma$

$Verify(SP, pk, M, \sigma) \rightarrow \{0,1\}$

数字签名的形式化算法定义

正确性：对于任意的$SP \leftarrow Setup(\lambda), (pk, sk) \leftarrow KeyGen(SP), M \in$ 消息空间$, \sigma \leftarrow Sign(SP, sk, M)$, 有$1 \leftarrow Verify(SP, pk, M, \sigma)$.

- BB^{RO}签名算法构造

SysGen: The system parameter generation algorithm takes as input a security parameter λ. It chooses a pairing group $\mathbb{PG} = (\mathbb{G}, \mathbb{G}_T, g, p, e)$ and a cryptographic hash function $H : \{0,1\}^* \rightarrow \mathbb{G}$. The algorithm returns the system parameter $SP = (\mathbb{PG}, H)$.

KeyGen: The key generation algorithm takes as input SP. It randomly chooses $g_2 \in \mathbb{G}$, $\alpha \in \mathbb{Z}_p$ and computes $g_1 = g^\alpha$. The algorithm returns a public/secret key pair (pk, sk) as follows.

$$pk = (g_1, g_2), \quad sk = \alpha.$$

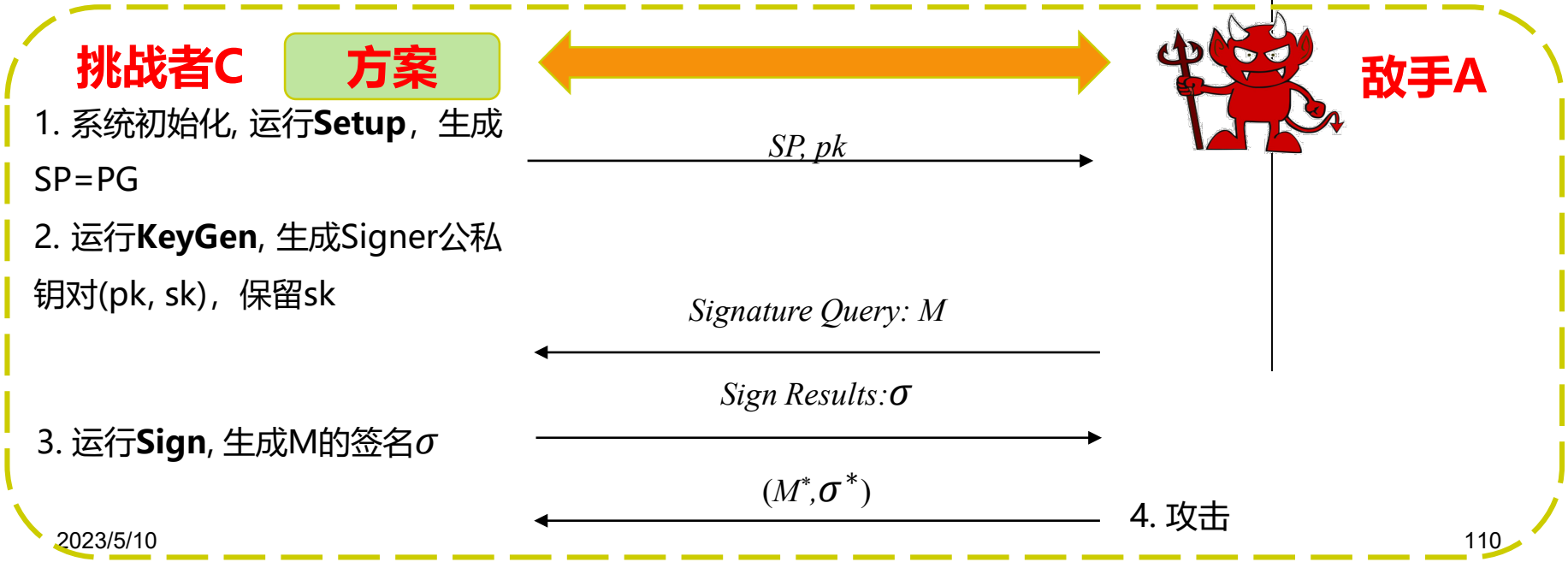
Sign: The signing algorithm takes as input a message $m \in \{0,1\}^*$, the secret key sk and SP. It chooses a random number $r \in \mathbb{Z}_p$ and returns the signature σ_m on m as

$$\sigma_m = (\sigma_1, \sigma_2) = \left(g_2^\alpha H(m)^r, g^r\right).$$

Verify: The verification algorithm takes as input a message-signature pair (m, σ_m), the public key pk and SP. Let $\sigma_m = (\sigma_1, \sigma_2)$. It accepts the signature if

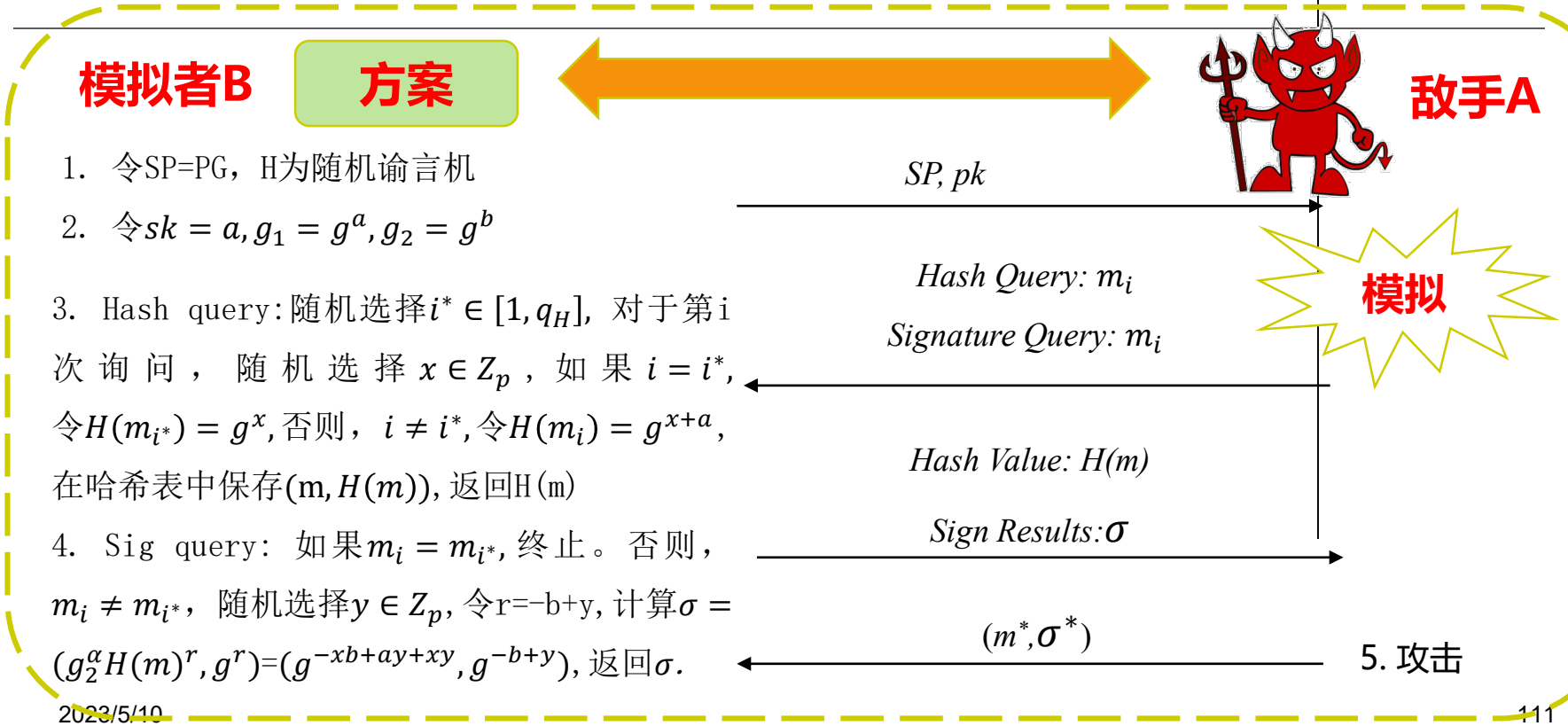
$$e(\sigma_1, g) = e(g_1, g_2)e\left(H(m), \sigma_2\right).$$

- BB^{RO}签名方案：安全归约证明
  - **定理：假设H是一个随机谕言机，BB^{RO}方案在EU-CMA安全模型下的安全性可被归约到CDH困难假设。**
  - 安全假设：CDH：给定：$g, g^a, g^b \in G$，求解：g^{ab}是困难的。
  - 归约证明：假设存在一个PPT敌手A可以在EU-CMA安全模型下攻破BB^{RO}方案，我们可以构造一个模拟者B解决CDH问题。



➢ BB^{RO}签名方案：安全归约证明

- 假设H是一个随机谕言机，假设存在一个PPT敌手A可以在EU-CMA安全模型下攻破BB^{RO}方案，我们可以构造一个模拟者B解决CDH问题。假设敌手A可以进行q_H次随机谕言机询问，q_S次签名询问。



➢ BBˆ{RO}签名方案：安全归约证明

- 假设H是一个随机谕言机，BB^{RO}方案在EU-CMA安全模型的安全性可被归约到CDH困难假设

安全假设：CDH：给定： $g, g^a, g^b \in G$，求解： g^{ab}是困难的。

攻击：敌手A返回(m^*, σ^*)，若$m^* \neq m_{i^*}$，终止。否则，$m^* = m_{i^*}$，$H(m^*) = g^x$，模拟者B输出

$$g^{ab} = \sigma_1^* \cdot \sigma_2^{-x}$$

分析：1. 对于敌手A，模拟者模拟的方案与安全模型中的方案是否不可区分？模拟成功的概率P_S？

2. 模拟者B发出有效攻击的概率P_U？

3. 模拟者B攻击困难问题成功的优势ϵ_R？

归约损失L=q_H

$$\varepsilon_R = P_S \cdot \varepsilon \cdot P_U = \frac{\varepsilon}{q_H}$$

只有当敌手在攻击时选择$m^* \neq m_{i^*}$，模拟不终止且攻击有效，即概率为$\frac{1}{q_H}$. 假设在EU-CMA安全模型下攻破BB^{RO}方案的PPT敌手A的优势为ϵ，此时模拟者B可以解决CDH问题的优势为$\frac{\varepsilon}{q_H}$。

4. 解决困难问题的时间消耗。

➢ ZSS签名方案

- 概念回顾：数字签名算法定义

$Setup(\lambda) \rightarrow SP$

$KeyGen(SP) \rightarrow (pk, sk)$

$Sign(SP, sk, M) \rightarrow \sigma$

$Verify(SP, pk, M, \sigma) \rightarrow \{0, 1\}$

数字签名的形式化算法定义

正确性：对于任意的$SP \leftarrow Setup(\lambda), (pk, sk) \leftarrow KeyGen(SP), M \in$ 消息空间$, \sigma \leftarrow Sign(SP, sk, M)$, 有$1 \leftarrow Verify(SP, pk, M, \sigma)$.

- ZSS签名算法构造

SysGen: The system parameter generation algorithm takes as input a security parameter λ. It chooses a pairing group $\mathbb{PG} = (\mathbb{G}, \mathbb{G}_T, g, p, e)$ and a cryptographic hash function $H : \{0,1\}^* \rightarrow \mathbb{Z}_p$. The algorithm returns the system parameter $SP = (\mathbb{PG}, H)$.

KeyGen: The key generation algorithm takes as input SP. It randomly chooses $h \in \mathbb{G}$, $\alpha \in \mathbb{Z}_p$ and computes $g_1 = g^\alpha$. The algorithm returns a public/secret key pair (pk, sk) as follows.

$$pk = (g_1, h), \quad sk = \alpha.$$

Sign: The signing algorithm takes as input a message $m \in \{0,1\}^*$, the secret key sk and SP. It returns the signature σ_m on m as

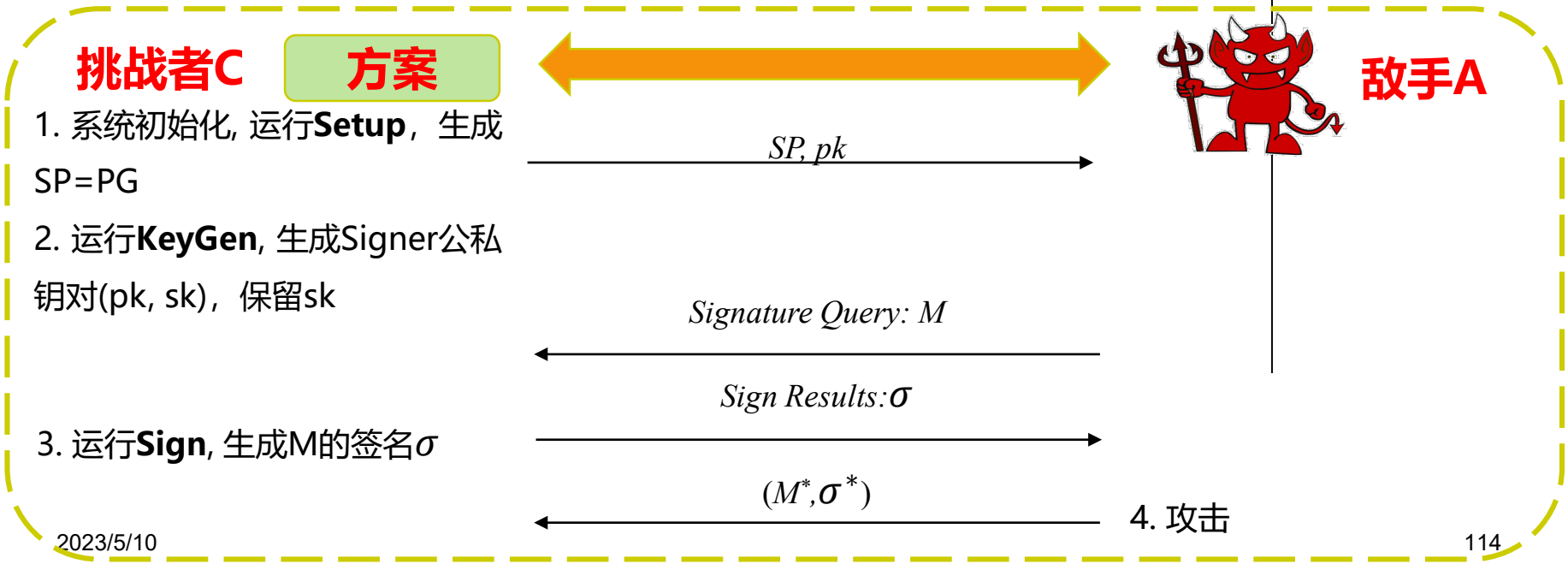
$$\sigma_m = h^{\frac{1}{\alpha + H(m)}}.$$

Verify: The verification algorithm takes as input a message-signature pair (m, σ_m), the public key pk and SP. It accepts the signature if

$$e\left(\sigma_m,\ g_1 g^{H(m)}\right) = e(h, g).$$

➢ ZSS签名方案：安全归约证明

- **定理：假设H是一个随机谕言机，ZSS方案在EU-CMA安全模型下的安全性可被归约到q-SDH困难假设。**
- 安全假设：q-SDH：给定 $g, g^a, g^{a^2}, \cdots, g^{a^q} \in G$，求解： $(s, g^{\frac{1}{a+s}}) \in Zp \times G$困难
- 归约证明：假设存在一个PPT敌手A可以在EU-CMA安全模型下攻破ZSS方案，我们可以构造一个模拟者B解决q-SDH问题。



➢ ZSS签名方案：安全归约证明

- 假设H是一个随机谕言机，假设存在一个PPT敌手A可以在EU-CMA安全模型下攻破ZSS方案，我们可以构造一个模拟者B解决q-SDH问题。假设敌手A可以进行q_H次随机谕言机询问，q_S次签名询问，令$q = q_H$。

模拟者B **方案** **敌手A**

1. 令SP=PG，H为随机谕言机
2. 令$sk = a, g_1 = g^a$，随机选择$x_1, x_2, \cdots, x_q \in Z_p$，计算$h = g^{(a+x_1)(a+x_2)\cdots(a+x_q)}$.
3. Hash query:随机选择$i^* \in [1, q_H]$，对于第i次询问，如果$i = i^*$,令$H(m_{i^*}) = x_i$,否则$i \neq i^*$,随机选择$x \in Z_p$,要求$x \neq x_i$,令$H(m_i) = x$,在哈希表中保存$(m, H(m))$,返回 H(m)
4. Sig query: 如果$m_i = m_{i^*}$，终止。否则，$m_i \neq m_{i^*}$，计算$\sigma = h^{\frac{1}{a+H(m)}} = g^{\frac{(a+x_1)(a+x_2)\cdots(a+x_q)}{a+x_i}} = g^{(a+x_1)\cdots(a+x_{i-1})(a+x_{i+1})\cdots(a+x_q)}$,返回$\sigma$.

SP, pk →

Hash Query: m_i

Signature Query: m_i ←

模拟

Hash Value: H(m)

Sign Results:σ →

(m^*, σ^*) ←

5. 攻击

➢ ZSS签名方案：安全归约证明

- 假设H是一个随机谕言机，ZSS方案在EU-CMA安全模型下的安全性可被归约到q-SDH困难假设

安全假设： q-SDH：给定 $g, g^a, g^{a^2}, \cdots, g^{a^q} \in G$，求解： $(s, g^{\frac{1}{a+s}}) \in Zp \times G$困难。

攻击：敌手A返回(m^*, σ^*)，若$m^* \neq m_{i^*}$，终止。否则，$m^* = m_{i^*}$，$H(m^*) = x$，

$$\sigma^* = h^{\frac{1}{a+x}} = g^{\frac{(a+x_1)(a+x_2)\cdots(a+x_q)}{a+x}} = g^{f(a)+\frac{F}{a+x}}$$

其中，$f(a)$是 q-1 阶的多项式，可计算，F为非零整数。模拟者B计算返回$(x, g^{\frac{1}{a+x}})$

分析：1. 对于敌手A，模拟者模拟的方案与安全模型中的方[illegible]

归约损失L=q_H

P_S ？

2. 模拟者B发出有效攻击的概率P_U ？

3. 模拟者B攻击困难问题成功的优势ϵ_R ？ $\varepsilon_R = P_S \cdot \varepsilon \cdot P_U = \frac{\varepsilon}{q_H}$

只有当敌手在攻击时选择$m^* \neq m_{i^*}$，模拟不终止且攻击有效，即概率为$\frac{1}{q_H}$. 假设在EU-CMA安全模型下攻破ZSS方案的PPT敌手A的优势为ϵ，此时模拟者B可以解决CDH问题的优势为$\frac{\varepsilon}{q_H}$。

4. 解决困难问题的时间消耗。

- 基于随机谕言机的签名方案的安全归约证明
  - BLS签名方案

    H-Type: $\sigma = H(m)^a$
  - BB^{RO}签名方案

    C-Type: $\sigma = (g^{ab}H(m)^r, g^r)$
  - ZSS签名方案

    I-Type: $\sigma = h^{\frac{1}{a+H(m)}}$

➢ Gentry签名方案

- 概念回顾：数字签名算法定义

$$Setup(\lambda) \rightarrow SP$$
$$KeyGen(SP) \rightarrow (pk, sk)$$
$$Sign(SP, sk, M) \rightarrow \sigma$$
$$Verify(SP, pk, M, \sigma) \rightarrow \{0, 1\}$$

数字签名的形式化算法定义

正确性：对于任意的$SP \leftarrow Setup(\lambda), (pk, sk) \leftarrow KeyGen(SP), M \in$ 消息空间$, \sigma \leftarrow Sign(SP, sk, M)$, 有$1 \leftarrow Verify(SP, pk, M, \sigma)$.

- Gentry签名算法构造

SysGen: The system parameter generation algorithm takes as input a security parameter λ. It chooses a pairing group $\mathbb{PG} = (\mathbb{G}, \mathbb{G}_T, g, p, e)$. The algorithm returns the system parameter $SP = \mathbb{PG}$.

KeyGen: The key generation algorithm takes as input SP. It randomly chooses $\alpha, \beta \in \mathbb{Z}_p$ and computes $g_1 = g^\alpha, g_2 = g^\beta$. The algorithm returns a public/secret key pair (pk, sk) as follows.

$$pk = (g_1, g_2), \quad sk = (\alpha, \beta).$$

Sign: The signing algorithm takes as input a message $m \in \mathbb{Z}_p$, the secret key sk and SP. It randomly chooses $r \in \mathbb{Z}_p$ and computes the signature σ_m on m as

$$\sigma_m = (\sigma_1, \sigma_2) = \left(r, \ g^{\frac{\beta - r}{\alpha - m}}\right).$$

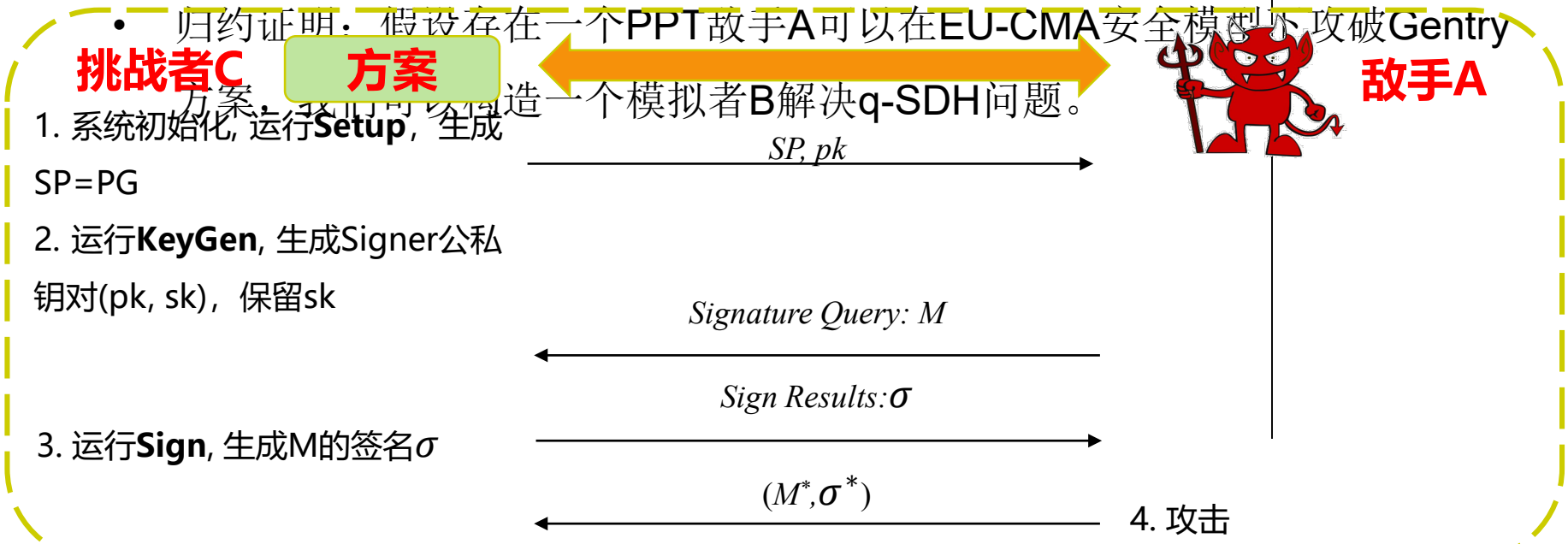
We require the signing algorithm always uses the same random number r for the signature generation on the message m.

Verify: The verification algorithm takes as input a message-signature pair (m, σ_m), the public key pk and SP. Let $\sigma_m = (\sigma_1, \sigma_2)$. It accepts the signature if

$$e\left(\sigma_2, \ g_1 g^{-m}\right) = e\left(g_2 g^{-\sigma_1}, g\right).$$

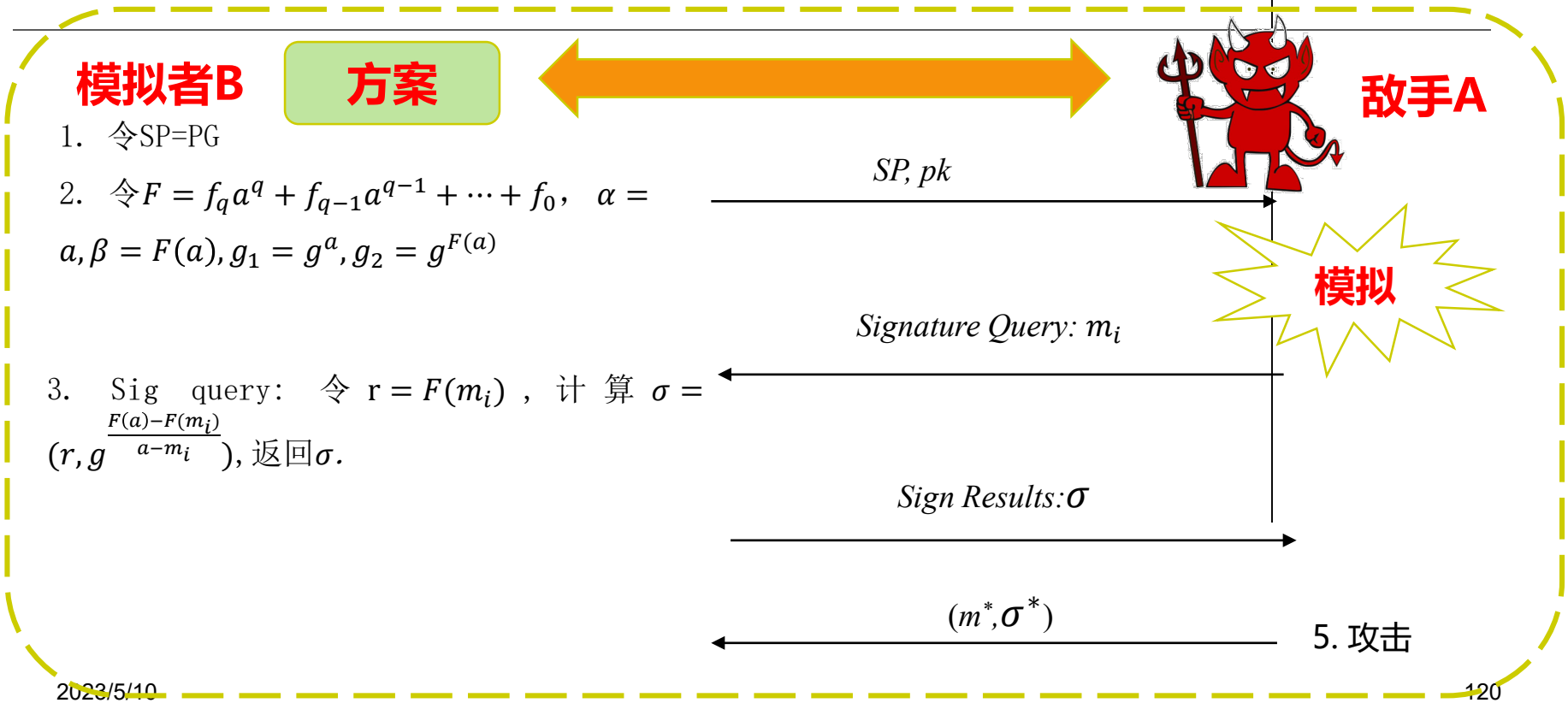
➢ Gentry签名方案：安全归约证明

- **定理：Gentry方案在EU-CMA安全模型下的安全性可被归约到q-SDH困难假设。**
- 安全假设：q-SDH: 给定 $g, g^a, g^{a^2}, \cdots, g^{a^q} \in G$，求解：$(s, g^{\frac{1}{a+s}}) \in Zp \times G$困难
- 归约证明：假设存在一个PPT敌手A可以在EU-CMA安全模型下攻破Gentry方案，我们可以构造一个模拟者B解决q-SDH问题。



➢ Gentry签名方案：安全归约证明

- 假设存在一个PPT敌手A可以在EU-CMA安全模型下攻破Gentry方案，我们可以构造一个模拟者B解决q-SDH问题。假设敌手A可以进行q_S次签名询问（令$q = q_S + 1$）。



➢ Gentry签名方案：安全归约证明

- Gentry方案在EU-CMA安全模型下的安全性可被归约到q-SDH困难假设

安全假设： q-SDH: 给定 $g, g^{a}, g^{a^2}, \cdots, g^{a^q} \in G$，求解：$(s, g^{\frac{1}{a+s}}) \in Zp \times G$困难。

攻击：敌手A返回(m^*, σ^*)，

$$\sigma_2{}^* = g^{\frac{F(a)-F(m_i)}{a-m_i}} = g^{f(a)+\frac{F}{a-m_i}}$$

其中， $f(a)$是 q-1 阶的多项式，可计算，F为非零整数。模拟者B计算返回$(-m_i, g^{\frac{1}{a-m_i}})$

分析：1. 对于敌手A，模拟者模拟的方案与安全模型中的方案是否不可区分？模拟成功的概率P_S？

2. 模拟者B发出有效攻击的概率P_U？

3. 模拟者B攻击困难问题成功的优势ϵ_R？

$$\varepsilon_R = P_S \cdot \varepsilon \cdot P_U = \varepsilon$$

4. 解决困难问题的时间消耗。

归约损失L=1

- 基于随机谕言机的签名方案的安全归约证明
  - BLS签名方案

    H-Type: $\sigma = H(m)^a$
  - BB^{RO}签名方案

    C-Type: $\sigma = (g^{ab}H(m)^r, g^r)$
  - ZSS签名方案

    I-Type: $\sigma = h^{\frac{1}{a+H(m)}}$
- 不基于随机谕言机的签名方案的安全归约证明
  - Gentry签名方案　I-Type: $\sigma = (r, g^{\frac{\beta-r}{a-m}})$