第 10 章：密钥管理和其他公钥密码体制

本章要点

- 公钥密码方案是安全的，仅当公钥的真实性能够得到保证。公钥证书方案提供了必要的安全性。
- 一个简单的公钥算法是 Diffie-Hellman 密钥交换协议。这个协议使得通信双方利用基于离散对数问题的公钥算法建立秘密密钥。这个协议是安全的，仅当通信双方的真实性能够得到保证。
- 椭圆曲线算术可以用来开发许多椭圆曲线密码 ECC 方案，包括密钥交换，加密和数字签名。
- 就 ECC 而言，椭圆曲线算术是指使用定义在有限域上的椭圆曲线方程。方程里的系数和变量都是域里的元素。已经开发了很多使用 Z_p 和 $\mathrm{GF}(2^n)$ 的方案。

主要内容

1. 公钥分配
2. Diffie-Hellman 密钥交换
3. 椭圆曲线密码

主要内容

1. 公钥分配
2. Diffie-Hellman 密钥交换
3. 椭圆曲线密码

密钥管理之公钥分配

- 公开密码的主要作用之一就是解决密钥分配问题，公钥密码实际上可以用于以下两个不同的方面：
  - 公钥的分配
  - 公钥密码用于传统密码体制的密钥分配
- 几种公钥分配方案：
  - 公开发布
  - 公开可访问目录
  - 公钥授权
  - 公钥证书

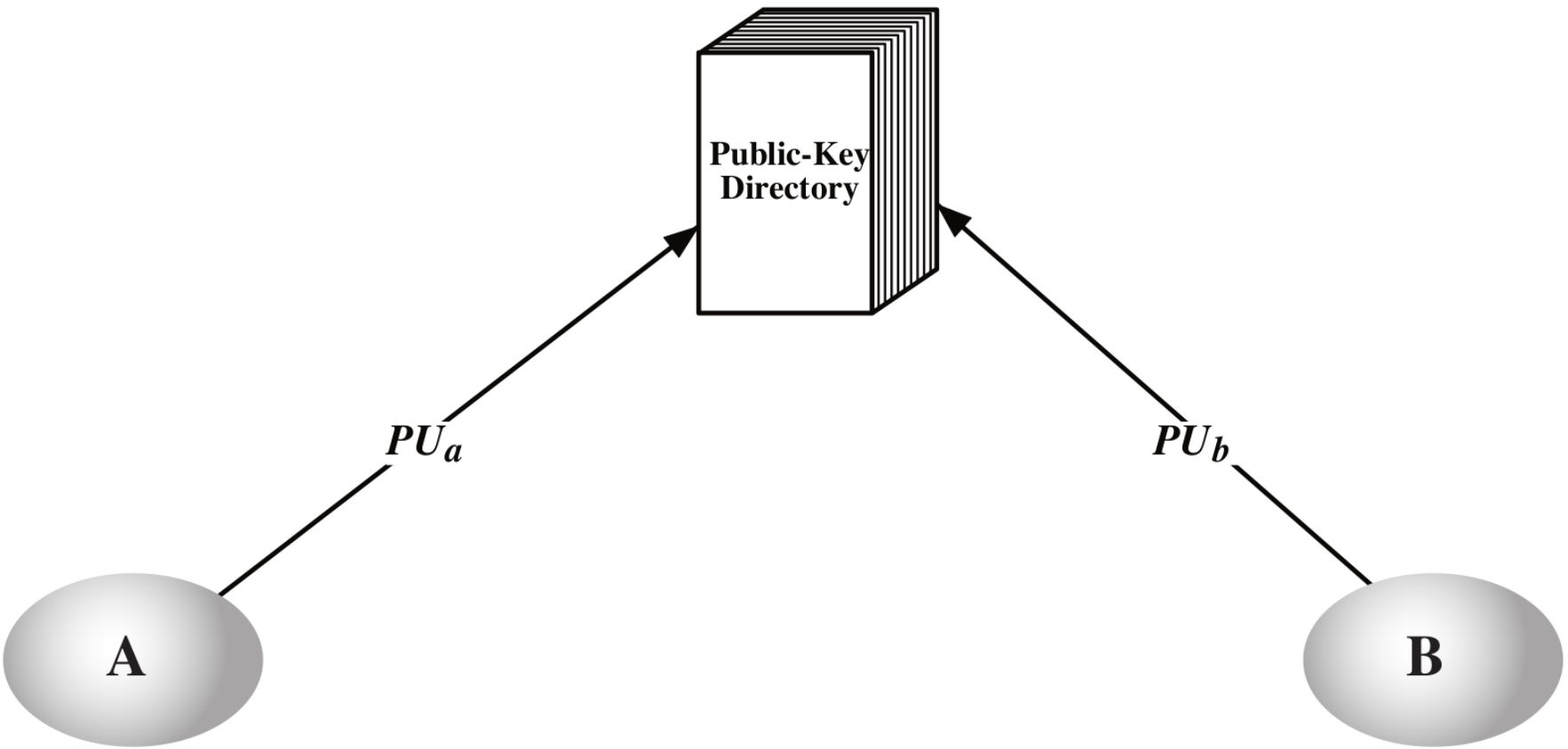
公钥的公开发布

- 用户将他的公钥发送给另一通信方，或者广播给通信各方，比如在电子邮件后附上 PGP 密钥，或者发布到邮件列表上。
- 最大问题在于任何人都可以伪造这种公钥的发布。

公开可访问目录

- 维护一个动态可访问的公钥目录可以获得更大程度的安全性。
- 一个可信实体或组织负责这个公开目录的维护和分配：
  - 目录包含 name, public-key 等项；
  - 每一通信方通过目录管理员以安全的方式注册一个公钥；
  - 通信方在任何时刻可以用新的密钥替代当前的密钥；
  - 目录定期更新；
  - 目录可通过电子方式访问。
- 一旦攻击者获得目录管理员私钥，则可传递伪造的公钥，可以假冒任何通信方以窃取消息，或者修改已有的记录。

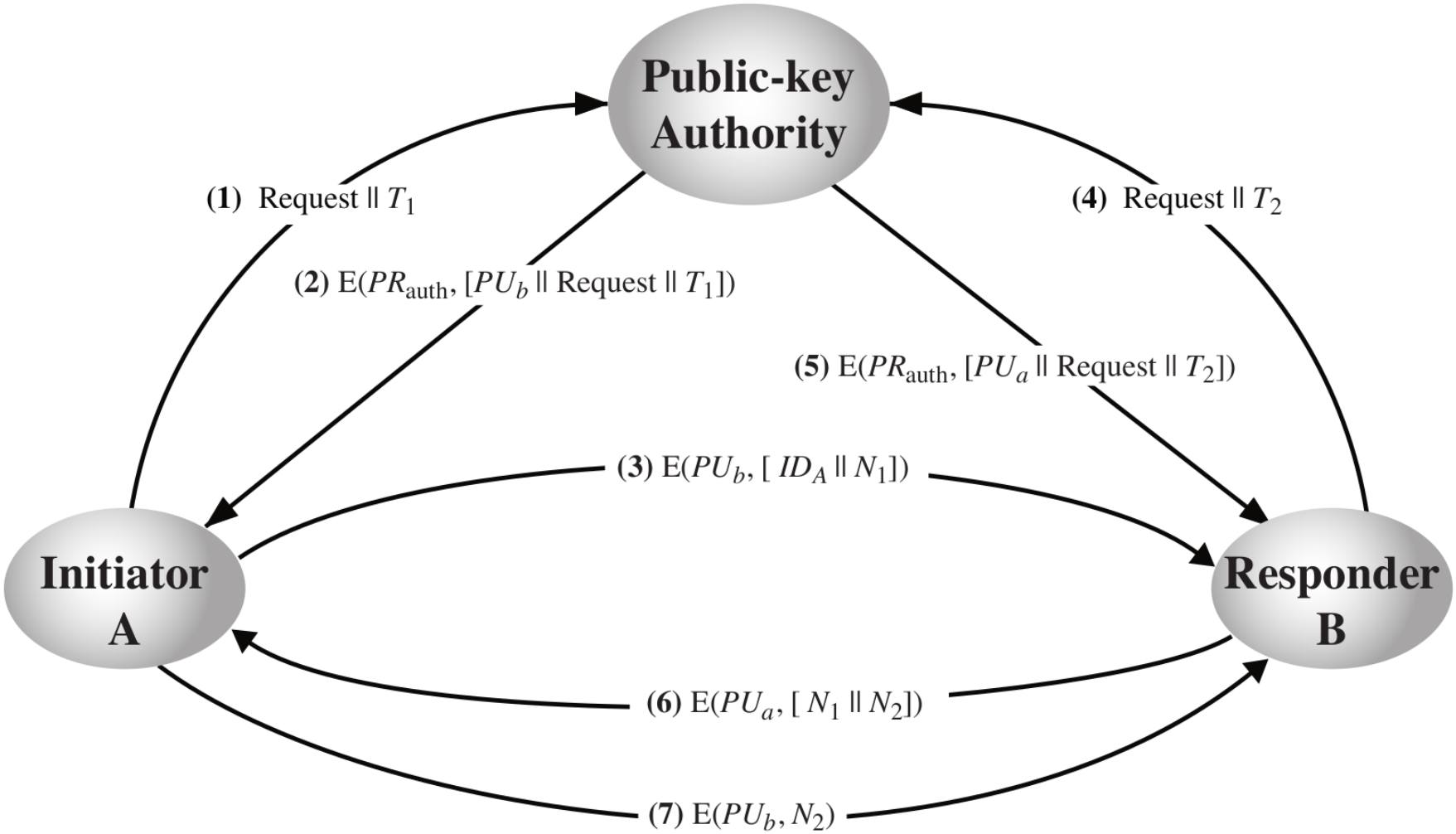
公开可访问目录



公钥授权

- A 发送带有时间戳的消息给公钥管理员, 请求 B 的公钥;
- 管理员给 A 发送用其私钥 $\mathrm{PR}_{\mathrm{auth}}$ 加密的消息, A 用管理员的公钥解密，可以确信该消息来自管理员:
  - B 的公钥 PU_b，用来加密;
  - 原始请求，A 可以验证其请求未被修改;
  - 原始时间戳, A 可以确定收到的不是来自管理员的旧消息。
- A 保存 B 的公钥, 并用它对包含 A 的标识 ID_A 和 Nonce_1 的消息加密, 然后发送给 B;
- B 以同样方式从管理员处得到 A 的公钥;
- B 用 PU_a 对 A 的 N_1 和 B 的 N_2 加密, 发送给 A;
- A 用 B 的公钥对 N_2 加密并发送给 B, 使 B 相信其通信伙伴是 A。

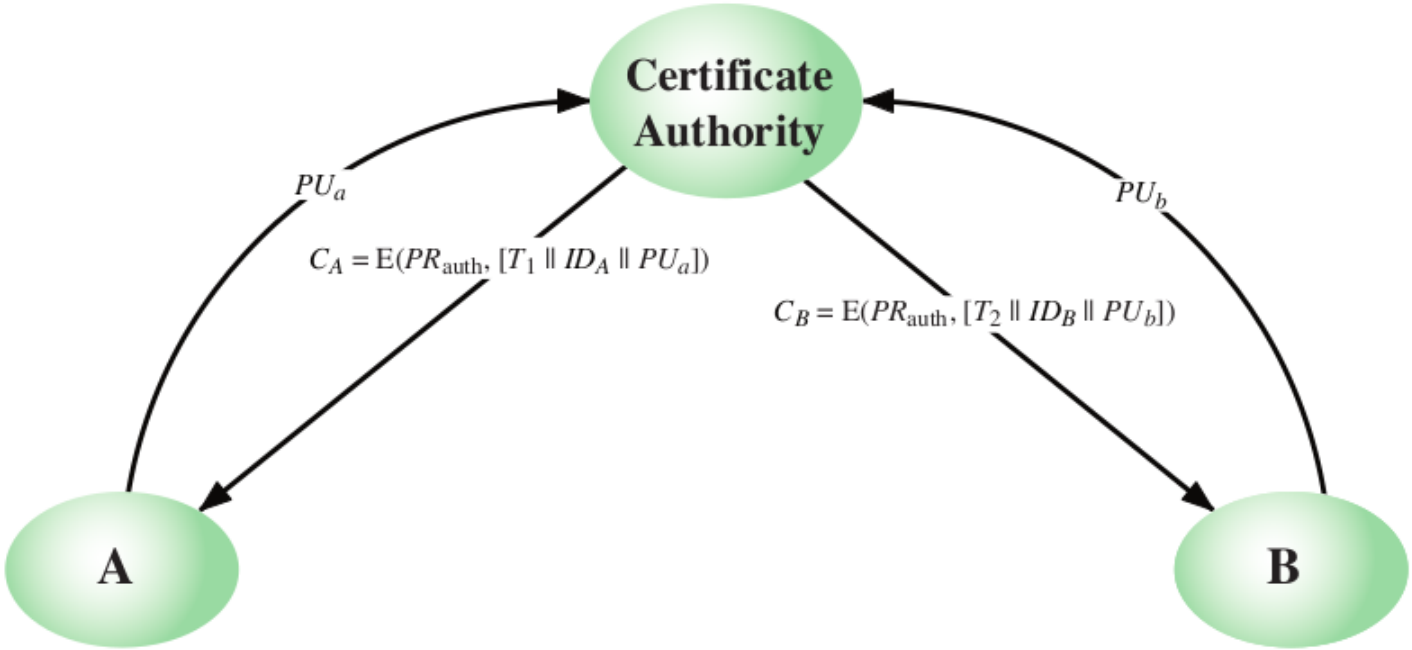
公钥授权


Public-key Authority
(1) Request || T_1
(2) E(PR_{auth}, [PU_b || Request || T_1])
(3) E(PU_b, [ID_A || N_1])
(4) Request || T_2
(5) E(PR_{auth}, [PU_a || Request || T_2])
Initiator A
Responder B
(6) E(PU_a, [N_1 || N_2])
(7) E(PU_b, N_2)

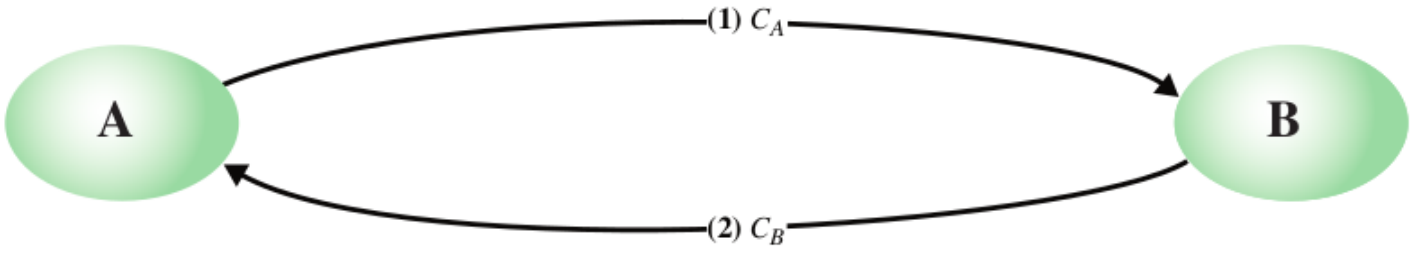
公钥证书

- 有了公钥证书使得不通过实时访问公钥授权部门而实现公钥交换成为可能；
- 公钥证书将一个通信方的身份与他的公开密钥绑定在一起，通常还包括有效期和使用方法等；
- 证书的所有内容必须经由可信公钥授权方或者证书授权方签名后方可生效；
- 知道公钥授权当局公开密钥的任何人都可以验证一个用户的公开密钥证书的有效性；
- 对于申请者 A，管理员提供的证书为：
  $C_A = E_{\mathrm{PR}_{\mathrm{auth}}}[T, \mathrm{ID}_A, \mathrm{PU}_a]$
- 其他人读取并验证：$D_{\mathrm{PU}_{\mathrm{auth}}}[C_A] = (T, \mathrm{ID}_A, \mathrm{PU}_a)$

公钥证书



(a) Obtaining certificates from CA



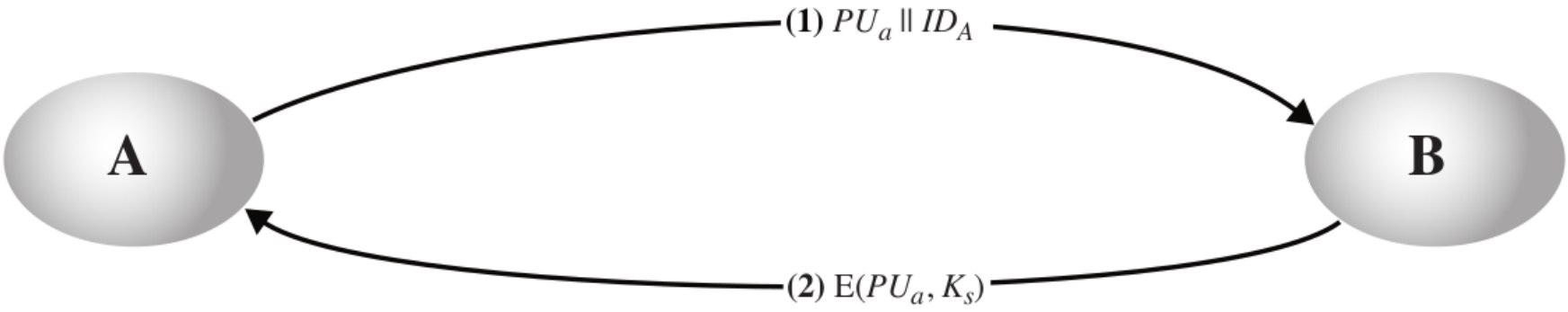
(b) Exchanging certificates

利用公钥密码分配传统密码体制的密钥

- 采用前述方法获得的公开密钥可以用于保密和认证之需
- 公钥密码算法速度较慢，因此更适合作为传统密码中实现秘密密钥分配的一种手段
- 因此，需要产生会话密码来加密
- 已经有一些方法用来协商适当的会话密钥

一种简单的秘密密钥分配方法

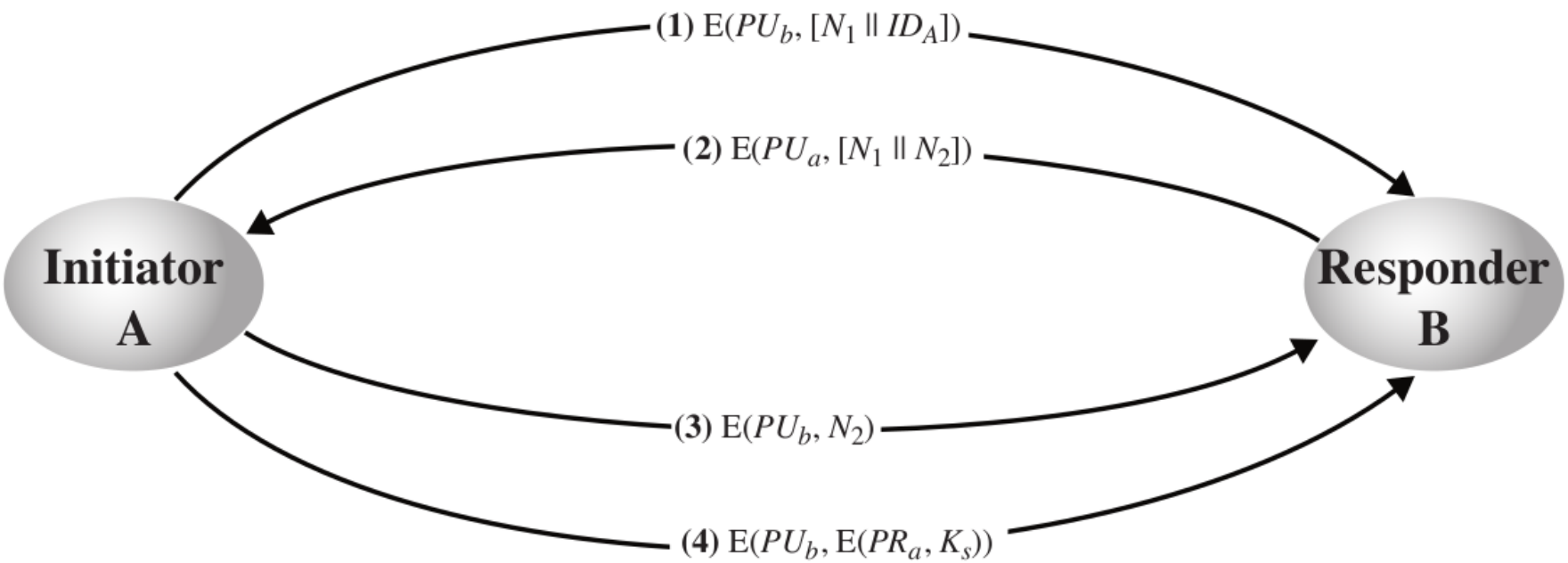
- Merkle 在 1979 提出一种简单的方法：
  - A 产生公/私钥对 $(\mathrm{PU}_a, \mathrm{PR}_a)$, 将含有 PU_a 和标识 ID_A 的消息发给 B；
  - B 产生秘密密钥 (会话密钥)K_s, 并用 A 的公钥加密后发给 A；
  - A 解密 $D(\mathrm{PR}_a, E(\mathrm{PU}_a, K_s))$, 得到 K_s, 这样双方即可通信。
- 这个协议不安全，因为会受到中间人攻击。



具有保密性和身份认证的密钥分配

- A 用 B 的公钥加密包含 ID_A 和一个临时交互号 N_1 的消息发送给 B，其中 N_1 唯一的标识本次消息传递；
- B 用 A 的公钥加密包含 N_1 和新的临时交互号 N_2 的消息给 A。因为只有 B 可解密上一条消息，故 N_1 在第二条消息中出现使 A 确信该消息来自 B；
- A 使用 B 的公钥加密 N_2 发送给 B 使 B 可以确信消息来自 A；
- A 选择密钥 K_s 后发送 $M = E(\mathrm{PU}_b, E(\mathrm{PR}_a, K_s))$ 给 B。用 B 的公钥加密确保只有 B 可以 q 读取该消息，用 A 的私钥加密确保只有 A 可发送该消息；
- B 计算 $D(\mathrm{PU}_a, D(\mathrm{PR_b}, \mathrm{M}))$ 得到密钥 K_s。

具有保密性和身份认证的密钥分配



主要内容

1. 公钥分配
2. Diffie-Hellman 密钥交换
3. 椭圆曲线密码

Diffie-Hellman 密钥交换

- Diffie-Hellman 密钥交换算法是一种公钥分发机制：
  - 它不是用来加密消息的；
  - 所生成的是通信双方共享的会话密钥，必须保密，其值取决于通信双方的私钥和公钥信息。
- Diffie-Hellman 密钥交换算法是基于有限域 GF 中的指数运算的（模一素数或多项式）；
- Diffie-Hellman 密钥交换算法的安全性依赖于求解离散对数问题 DLP。

离散对数问题 Discrete Logarithm Problem

- 如果 a 是素数 p 的一个本原根（本原元素），则 $a \bmod p, a^2 \bmod p, \ldots, a^{p-1} \bmod p$ 生成模 p 的完全剩余集 $\{1, 2, \ldots, p-1\}$。
- 对于所有素数，其本原根必定存在，即对于一个整数 b 和素数 p 的一个本原根 a，可以找到唯一的指数 i，使得 $b = a^i \bmod p$，其中 $0 \leq i \leq p-1$。指数 i 称为 b 的以 a 为基数的模 p 的离散对数或者指数。
- 离散对数密码体制的安全性基于 DLP 问题。在已知 C 和 p 的情况下，由 d 求 $M = C^d \bmod p$ 很容易；由 M 求 d 很困难。最快的算法需要 $T = \exp((\ln p \ln \ln p)^{1/2})$ 次运算。当 p 是 200 位时，$T = 2.7 \times 10^{11}$，如果 $1\mu s$ 解一次，需要 $2 \sim 3$ 天；如果 $p = 664$ 位，则 $T = 1.2 \times 10^{23}$，约 10^{12} 天，即 2.7 亿年。
- 只要 p 足够大，可以达到足够安全。

Diffie-Hellman 密钥交换协议

- 通信双方约定一个大素数 p, 和模 p 的一个素根 α;
- 各方产生公开密钥:
  - 选择一个秘密钥（数值），如 $x_A < p$，$x_B < p$;
  - 计算公钥 $y_A = \alpha^{x_A} \bmod p$, $y_B = \alpha^{x_B} \bmod p$, 并相互交换。
- 双方共享的会话密钥 K 可以如下算出:

用户 A

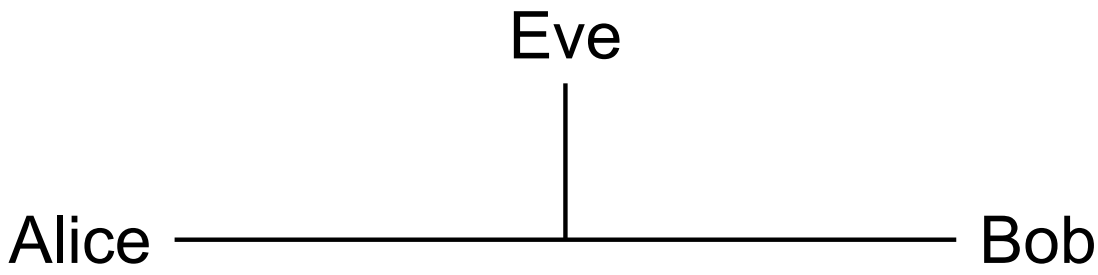
$$\begin{aligned} K &= y_B^{x_A} \bmod p \\ &= \alpha^{x_B x_A} \bmod p \end{aligned}$$

用户 B

$$\begin{aligned} K &= y_A^{x_B} \bmod p \\ &= \alpha^{x_A x_B} \bmod p \end{aligned}$$

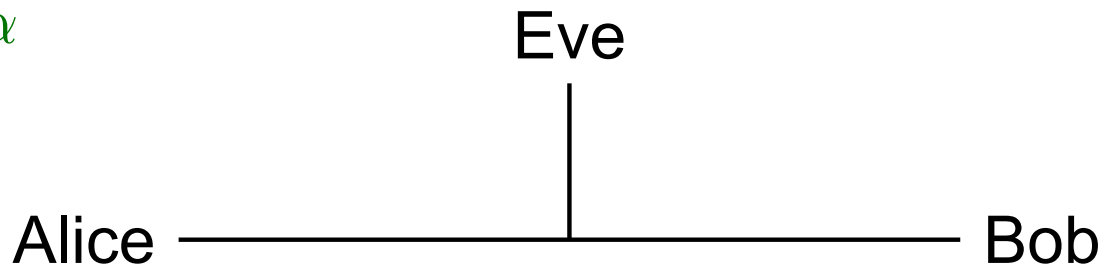
- K 是双方用对称密码通信时共享的密钥;
- 攻击者如果想要获得 K, 则必须解决 DLP 问题。

Diffie-Hellman 密钥交换协议



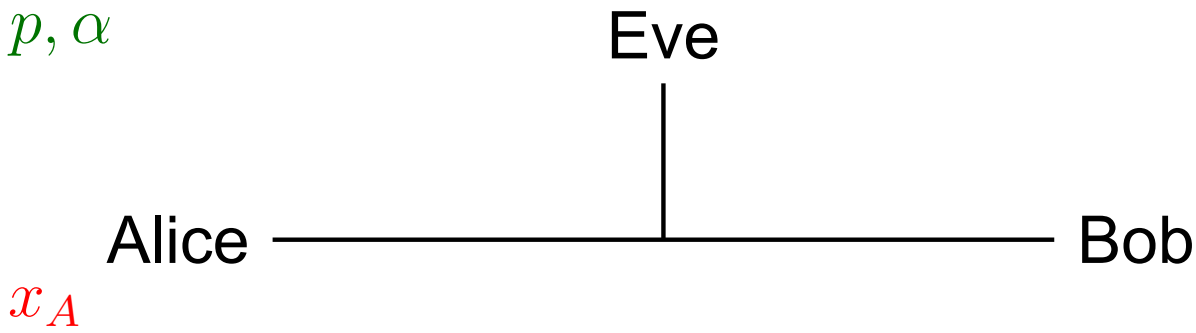
- Alice 和 Bob 选择大素数 p 和它的一个素根 α；
- Alice 选择一个随机数 x_A 作为自己的私钥，并计算公钥 $y_A = \alpha^{x_A} \bmod p$，并将 y_A 发送给 Bob；
- Bob 选择一个随机数 x_B 作为自己的私钥，并计算公钥 $y_B = \alpha^{x_B} \bmod p$，并将 y_B 发送给 Alice；
- Alice 计算 $K = y_B^{x_A} \bmod p$ 作为会话密钥；
- Bob 计算 $K = y_A^{x_B} \bmod p$ 作为会话密钥。

Diffie-Hellman 密钥交换协议



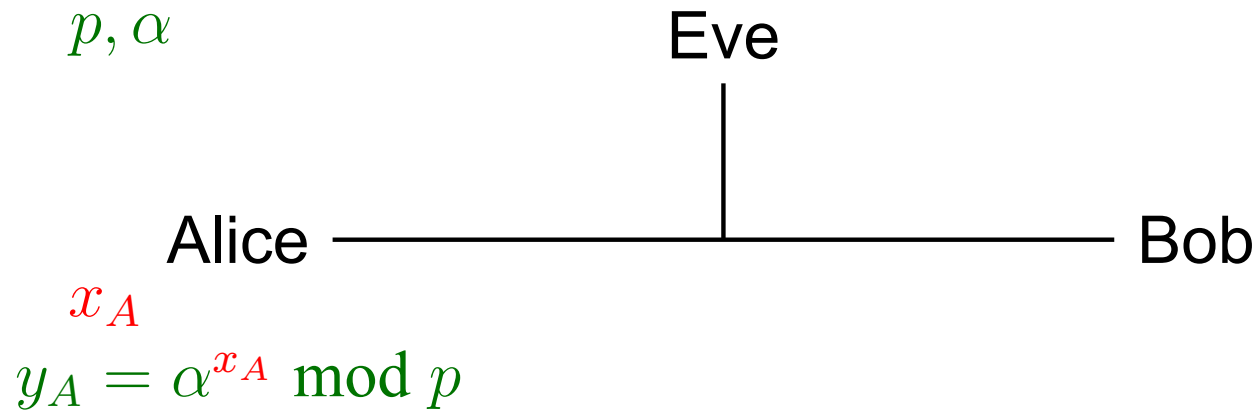
- Alice 和 Bob 选择大素数 p 和它的一个素根 α；
- Alice 选择一个随机数 x_A 作为自己的私钥，并计算公钥 $y_A = \alpha^{x_A} \bmod p$，并将 y_A 发送给 Bob；
- Bob 选择一个随机数 x_B 作为自己的私钥，并计算公钥 $y_B = \alpha^{x_B} \bmod p$，并将 y_B 发送给 Alice；
- Alice 计算 $K = y_B^{x_A} \bmod p$ 作为会话密钥；
- Bob 计算 $K = y_A^{x_B} \bmod p$ 作为会话密钥。

Diffie-Hellman 密钥交换协议



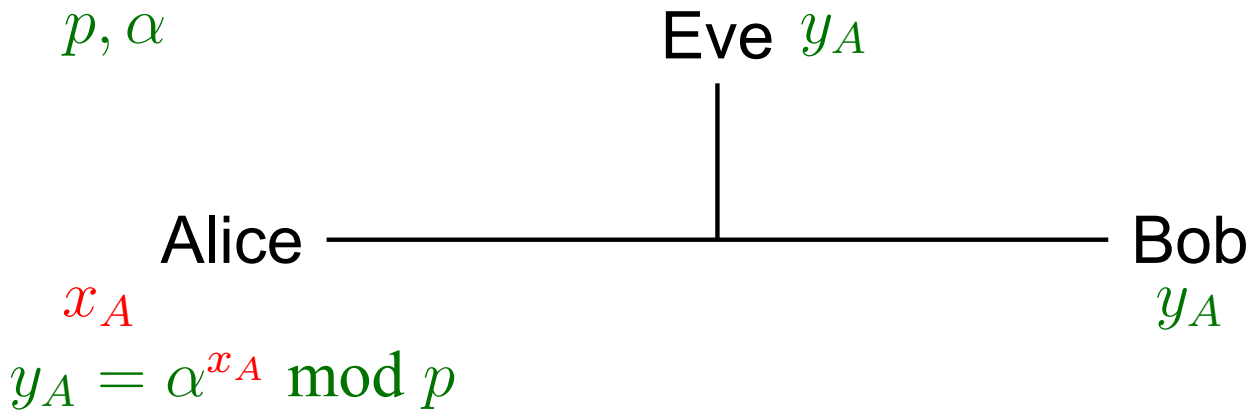
- Alice 和 Bob 选择大素数 p 和它的一个素根 α；
- Alice 选择一个随机数 x_A 作为自己的私钥，并计算公钥 $y_A = \alpha^{x_A} \bmod p$，并将 y_A 发送给 Bob；
- Bob 选择一个随机数 x_B 作为自己的私钥，并计算公钥 $y_B = \alpha^{x_B} \bmod p$，并将 y_B 发送给 Alice；
- Alice 计算 $K = y_B^{x_A} \bmod p$ 作为会话密钥；
- Bob 计算 $K = y_A^{x_B} \bmod p$ 作为会话密钥。

Diffie-Hellman 密钥交换协议



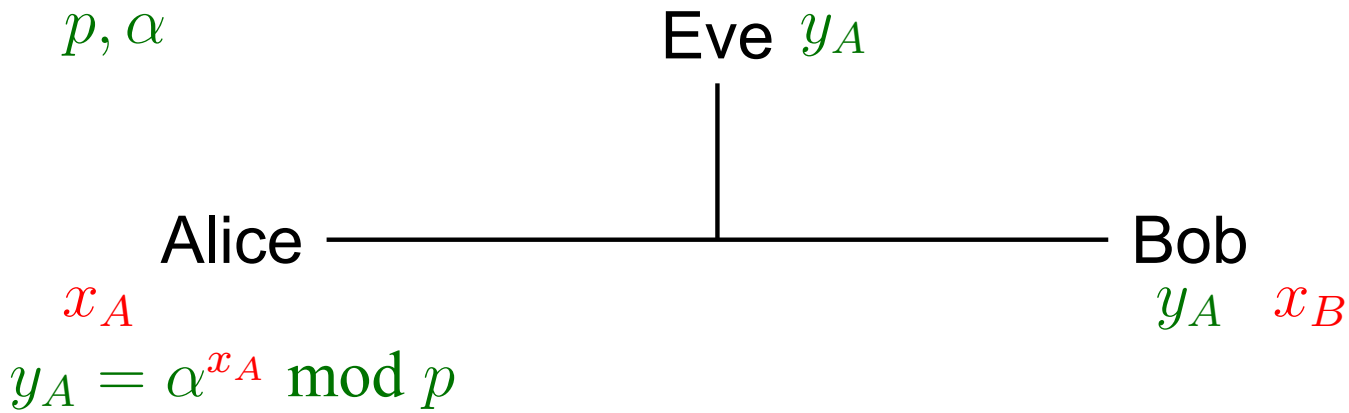
- Alice 和 Bob 选择大素数 p 和它的一个素根 α；
- Alice 选择一个随机数 x_A 作为自己的私钥，并计算公钥 $y_A = \alpha^{x_A} \bmod p$，并将 y_A 发送给 Bob；
- Bob 选择一个随机数 x_B 作为自己的私钥，并计算公钥 $y_B = \alpha^{x_B} \bmod p$，并将 y_B 发送给 Alice；
- Alice 计算 $K = y_B^{x_A} \bmod p$ 作为会话密钥；
- Bob 计算 $K = y_A^{x_B} \bmod p$ 作为会话密钥。

Diffie-Hellman 密钥交换协议



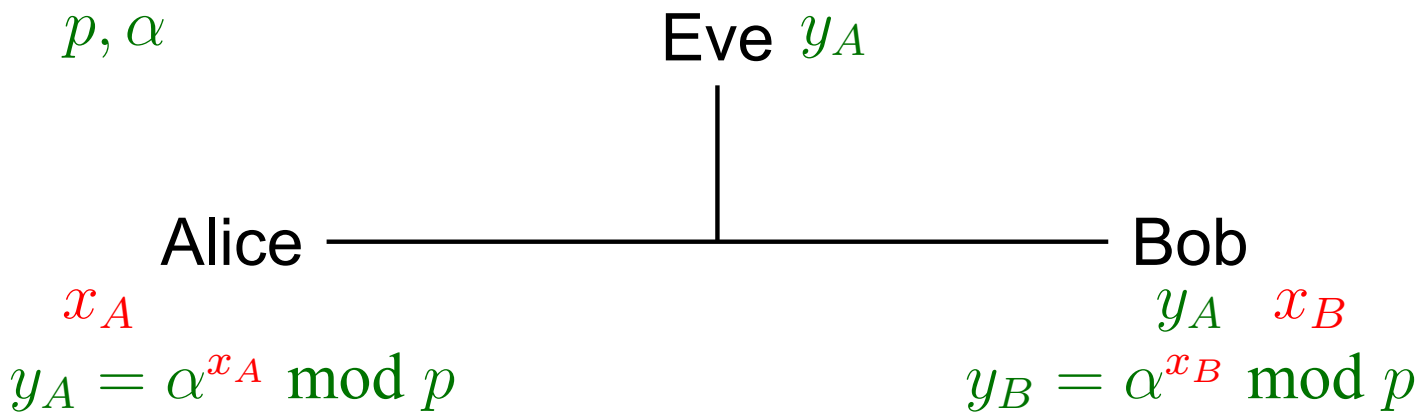
- Alice 和 Bob 选择大素数 p 和它的一个素根 α；
- Alice 选择一个随机数 x_A 作为自己的私钥，并计算公钥 $y_A = \alpha^{x_A} \bmod p$，并将 y_A 发送给 Bob；
- Bob 选择一个随机数 x_B 作为自己的私钥，并计算公钥 $y_B = \alpha^{x_B} \bmod p$，并将 y_B 发送给 Alice；
- Alice 计算 $K = y_B^{x_A} \bmod p$ 作为会话密钥；
- Bob 计算 $K = y_A^{x_B} \bmod p$ 作为会话密钥。

Diffie-Hellman 密钥交换协议



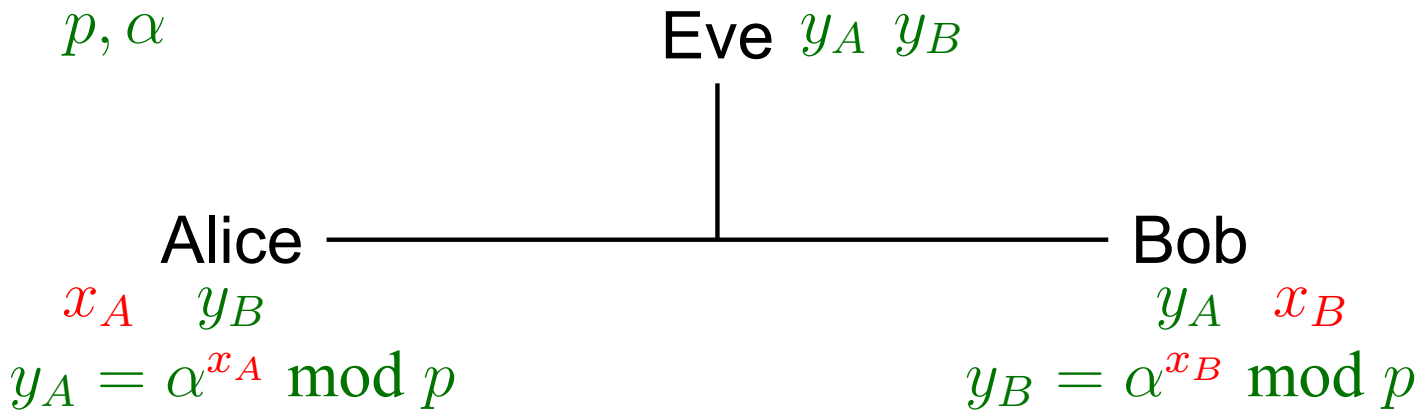
- Alice 和 Bob 选择大素数 p 和它的一个素根 α；
- Alice 选择一个随机数 x_A 作为自己的私钥，并计算公钥 $y_A = \alpha^{x_A} \bmod p$，并将 y_A 发送给 Bob；
- Bob 选择一个随机数 x_B 作为自己的私钥，并计算公钥 $y_B = \alpha^{x_B} \bmod p$，并将 y_B 发送给 Alice；
- Alice 计算 $K = y_B^{x_A} \bmod p$ 作为会话密钥；
- Bob 计算 $K = y_A^{x_B} \bmod p$ 作为会话密钥。

Diffie-Hellman 密钥交换协议



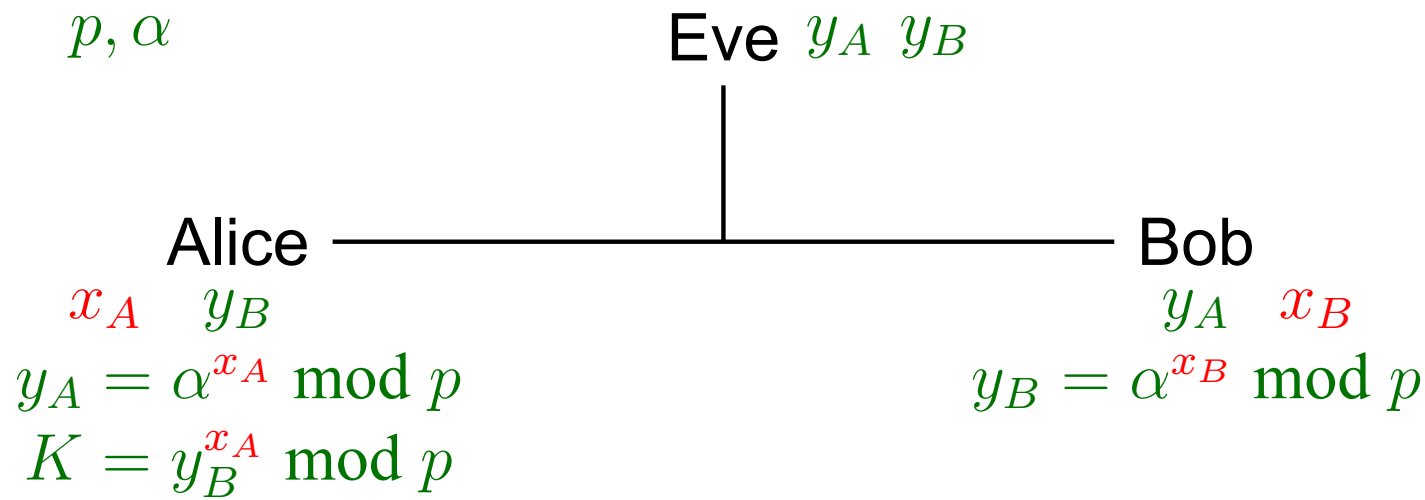
- Alice 和 Bob 选择大素数 p 和它的一个素根 α；
- Alice 选择一个随机数 x_A 作为自己的私钥，并计算公钥 $y_A = \alpha^{x_A} \bmod p$，并将 y_A 发送给 Bob；
- Bob 选择一个随机数 x_B 作为自己的私钥，并计算公钥 $y_B = \alpha^{x_B} \bmod p$，并将 y_B 发送给 Alice；
- Alice 计算 $K = y_B^{x_A} \bmod p$ 作为会话密钥；
- Bob 计算 $K = y_A^{x_B} \bmod p$ 作为会话密钥。

Diffie-Hellman 密钥交换协议



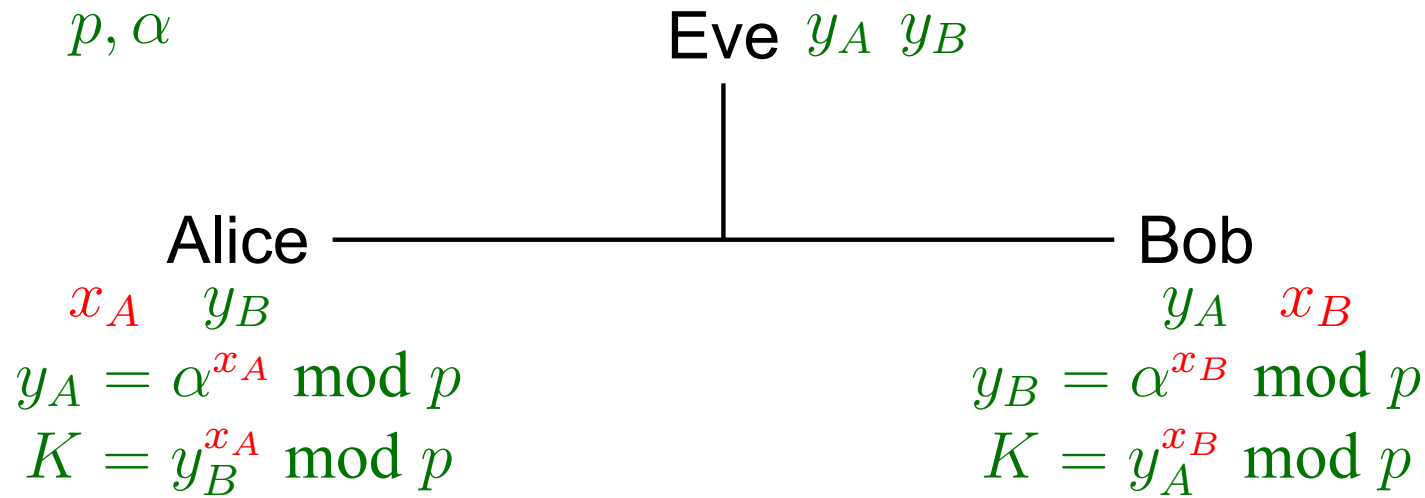
- Alice 和 Bob 选择大素数 p 和它的一个素根 α；
- Alice 选择一个随机数 x_A 作为自己的私钥，并计算公钥 $y_A = \alpha^{x_A} \bmod p$，并将 y_A 发送给 Bob；
- Bob 选择一个随机数 x_B 作为自己的私钥，并计算公钥 $y_B = \alpha^{x_B} \bmod p$，并将 y_B 发送给 Alice；
- Alice 计算 $K = y_B^{x_A} \bmod p$ 作为会话密钥；
- Bob 计算 $K = y_A^{x_B} \bmod p$ 作为会话密钥。

Diffie-Hellman 密钥交换协议



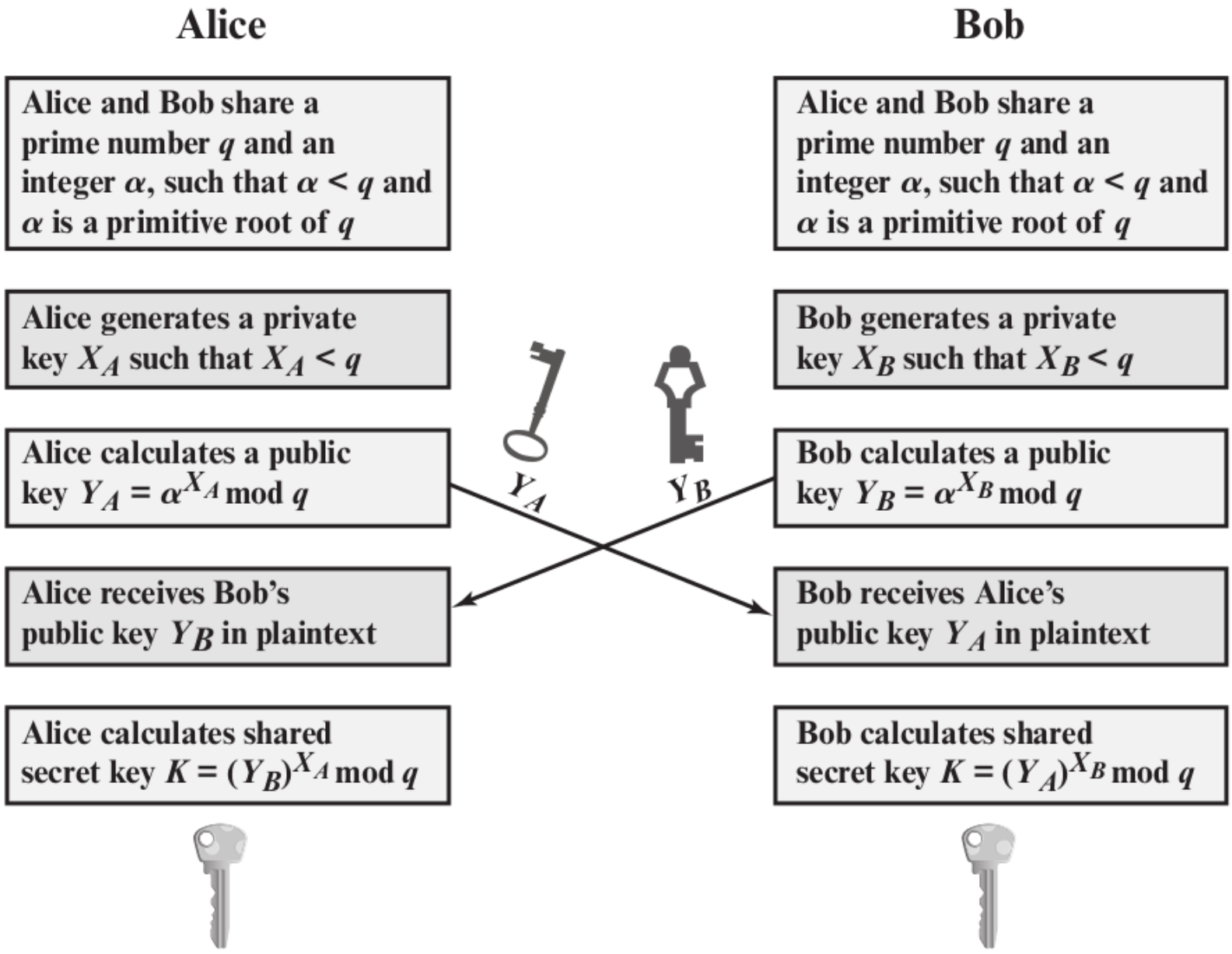
- Alice 和 Bob 选择大素数 p 和它的一个素根 α；
- Alice 选择一个随机数 x_A 作为自己的私钥，并计算公钥 $y_A = \alpha^{x_A} \bmod p$，并将 y_A 发送给 Bob；
- Bob 选择一个随机数 x_B 作为自己的私钥，并计算公钥 $y_B = \alpha^{x_B} \bmod p$，并将 y_B 发送给 Alice；
- Alice 计算 $K = y_B^{x_A} \bmod p$ 作为会话密钥；
- Bob 计算 $K = y_A^{x_B} \bmod p$ 作为会话密钥。

Diffie-Hellman 密钥交换协议



- Alice 和 Bob 选择大素数 p 和它的一个素根 α；
- Alice 选择一个随机数 x_A 作为自己的私钥，并计算公钥 $y_A = \alpha^{x_A} \bmod p$，并将 y_A 发送给 Bob；
- Bob 选择一个随机数 x_B 作为自己的私钥，并计算公钥 $y_B = \alpha^{x_B} \bmod p$，并将 y_B 发送给 Alice；
- Alice 计算 $K = y_B^{x_A} \bmod p$ 作为会话密钥；
- Bob 计算 $K = y_A^{x_B} \bmod p$ 作为会话密钥。

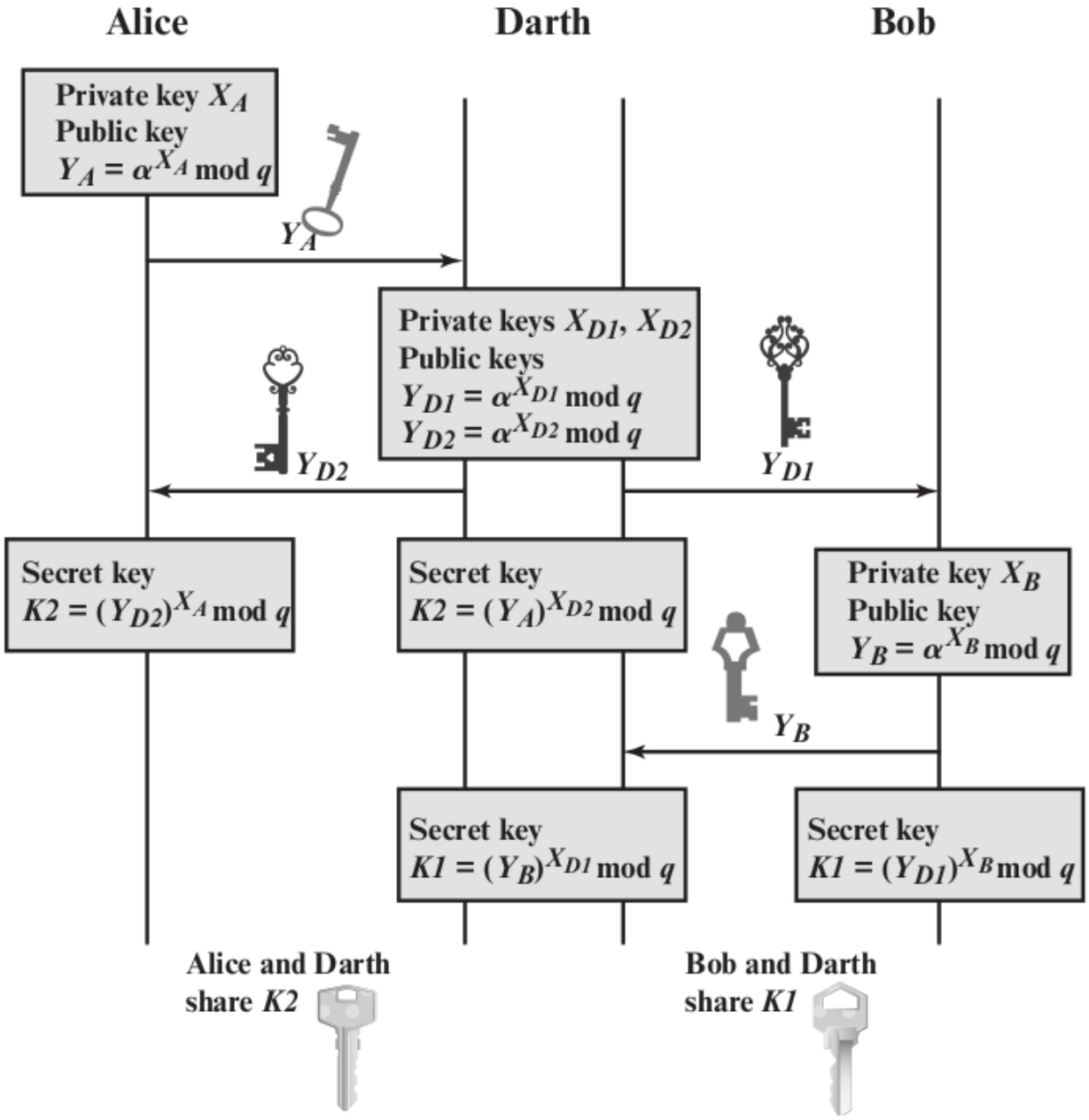
Diffie-Hellman 密钥交换协议



Diffie-Hellman 密钥交换举例

- Users Alice & Bob who wish to swap keys;
- Agree on prime $p = 353$ and $\alpha = 3$;
- Select random secret keys:
  - A chooses $x_A = 97$,
  - B chooses $x_B = 233$.
- Compute public keys:
  - $y_A = 3^{97} \bmod 353 = 40$ (Alice)
  - $y_B = 3^{233} \bmod 353 = 248$ (Bob)
- Compute shared session key as:
  - $K = y_B^{x_A} \bmod 353 = 248^{97} \bmod 353 = 160$ (Alice)
  - $K = y_A^{x_B} \bmod 353 = 40^{233} \bmod 353 = 160$ (Bob)

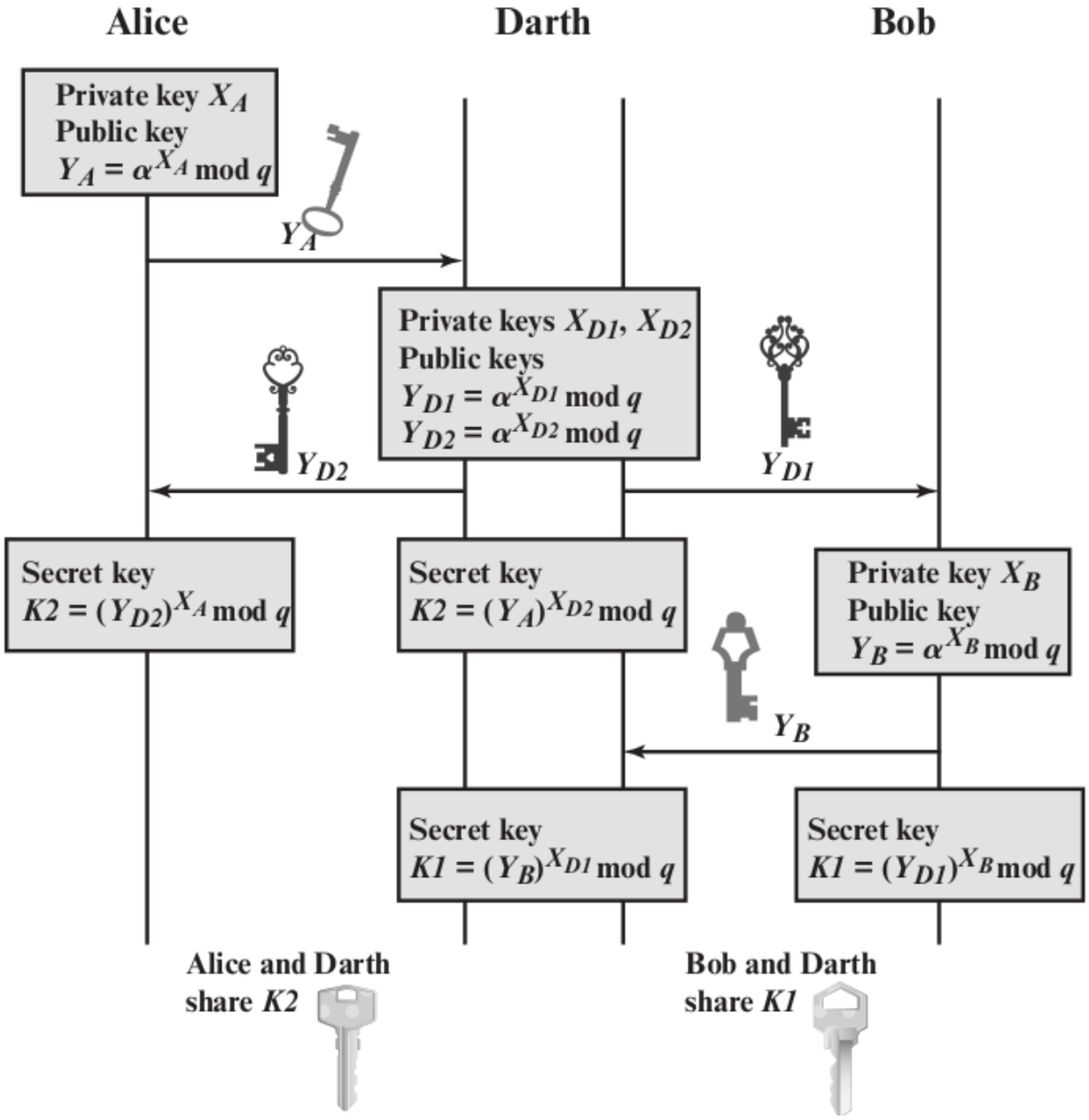
中间人攻击 Man-in-the-middle Attack



注意

密钥交换协议不能抵抗上述攻击，因为它未对通信参与方进行认证，可通过数字签名克服。

中间人攻击 Man-in-the-middle Attack



注意

密钥交换协议不能抵抗上述攻击，因为它未对通信参与方进行认证，可通过数字签名克服。

主要内容

1. 公钥分配

2. Diffie-Hellman 密钥交换

3. 椭圆曲线密码
3.1 椭圆曲线算术
3.2 有限域上的椭圆曲线
3.3 椭圆曲线密码学

Abel 群

Abel 群由元素的集合 G 及其上的二元运算 $\cdot$ 组成，记作 $\{G, \cdot\}$。满足以下公理：

- 封闭性：若 $a, b \in G$，则 $a \cdot b \in G$；
- 结合性：对 G 中的任意 a, b, c，$(a \cdot b) \cdot c = a \cdot (b \cdot c)$；
- 单位元：G 中存在元素 e，使得对 G 中所有元素 a，$e \cdot a = a \cdot e = a$；
- 逆元：对 G 中任何 a，存在 $a' \in G$，满足 $a' \cdot a = a \cdot a' = e$；
- 交换性：对 G 中任何 a, b，有 $a \cdot b = b \cdot a$。

注意

$\cdot$ 为通用运算，可以是乘法、加法或其他运算。如果是加法，单位元也可称为零元，逆元也可称为负元。

Abel 群

Abel 群由元素的集合 G 及其上的二元运算 $\cdot$ 组成，记作 $\{G, \cdot\}$。满足以下公理：

- 封闭性：若 $a, b \in G$，则 $a \cdot b \in G$；
- 结合性：对 G 中的任意 a, b, c，$(a \cdot b) \cdot c = a \cdot (b \cdot c)$；
- 单位元：G 中存在元素 e，使得对 G 中所有元素 a，$e \cdot a = a \cdot e = a$；
- 逆元：对 G 中任何 a，存在 $a' \in G$，满足 $a' \cdot a = a \cdot a' = e$；
- 交换性：对 G 中任何 a, b，有 $a \cdot b = b \cdot a$。

注意

$\cdot$ 为通用运算，可以是乘法、加法或其他运算。如果是加法，单位元也可称为零元，逆元也可称为负元。

椭圆曲线算术（实数域）

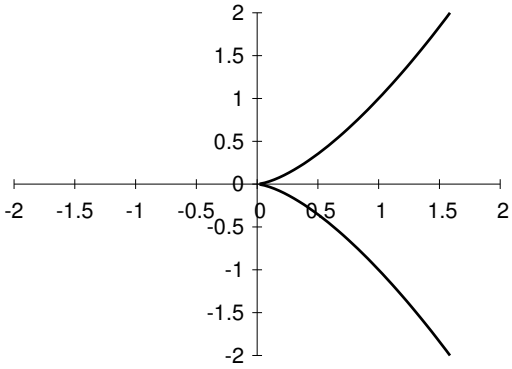
- 椭圆曲线并非椭圆，它指的是由威尔斯特拉斯（Weierstrass）方程所确定的平面曲线：

$$y^2 + axy + by = x^3 + cx^2 + dx + e$$

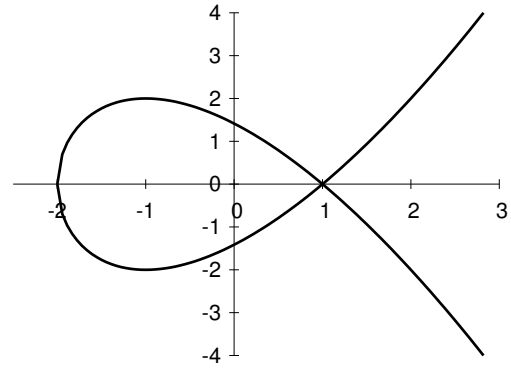
- 椭圆曲线密码主要考虑威尔斯特拉斯标准形式：

$$y^2 = x^3 + ax + b$$

- 满足上述方程的数偶 (x, y) 称为椭圆曲线 $E(a, b)$ 上的点；
- 要求 a, b 满足条件 $4a^3 + 27b^2 \neq 0$ 使 $x^3 + ax + b = 0$ 不含重根；否则椭圆曲线在重根处不存在切线。



$E(0, 0)$

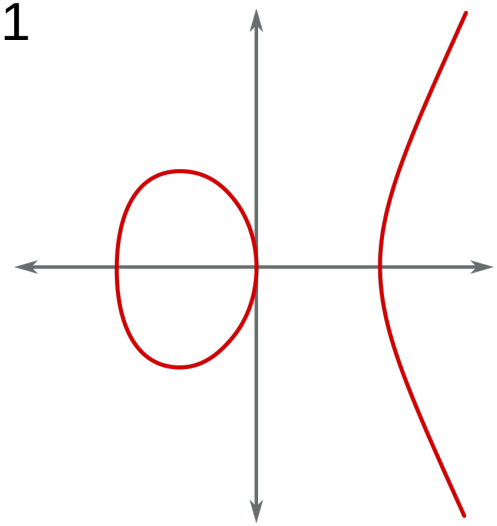


$E(-3, 2)$

（曲线在 0 点不光滑或者曲线在 $(1, 0)$ 点交叉的情况都无法定义切线）

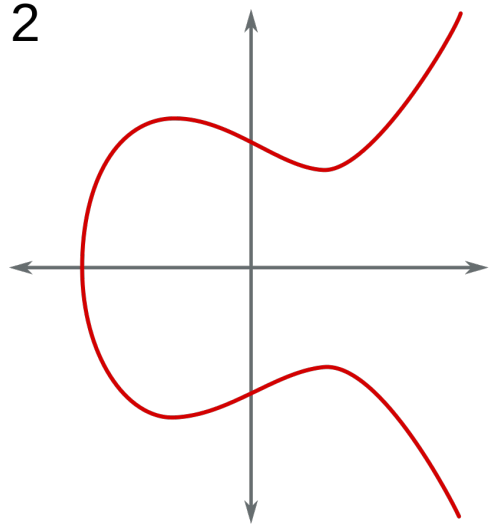
椭圆曲线举例

1

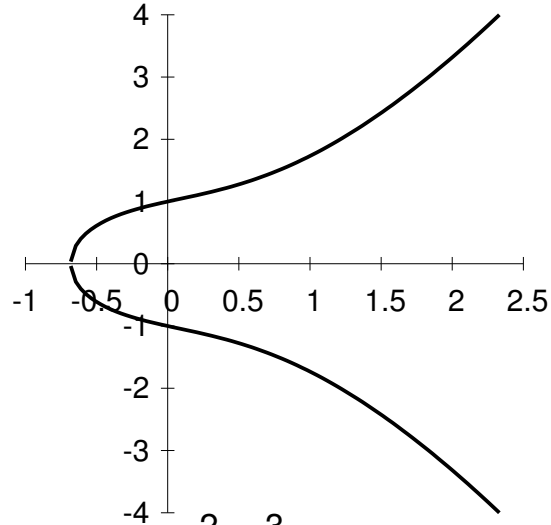


$y^2 = x^3 - x$

2



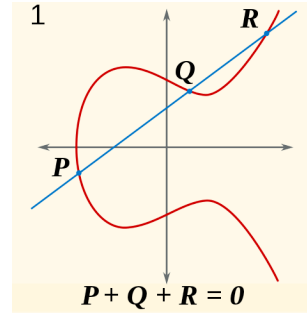
$y^2 = x^3 - x + 1$

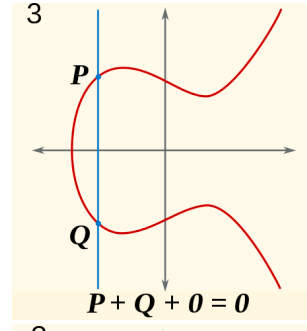


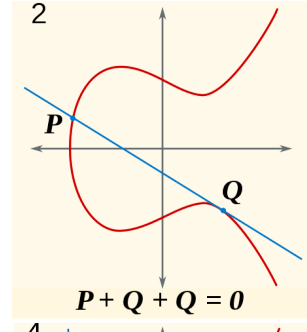
$y^2=x^3+x+1$

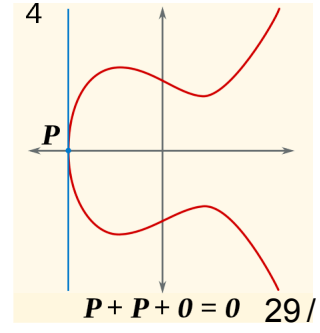
椭圆曲线上形式加法的定义

- 椭圆曲线包括一个称为无穷远点或零点的元素，记为 O；
- O 是加法的零元，对于椭圆曲线上的任意点 P，满足 $P + O = O + P = P$；
- 如果椭圆曲线上的三个点处于一条直线上，那么它们的和为 O；
- 一条垂直线与曲线相交于 $P = (x, y)$ 和 $Q = (x, -y)$，也相交于无穷点 O，有 $P + Q + O = O$，称 $Q = -P$ 为 P 的负元；
- 在点 Q 处画一切线求出另一交点 P，则 $Q + Q + P = O$，即 $2Q = -P$；
- 椭圆曲线上的点及其上的加法构成一个 Abel 群。



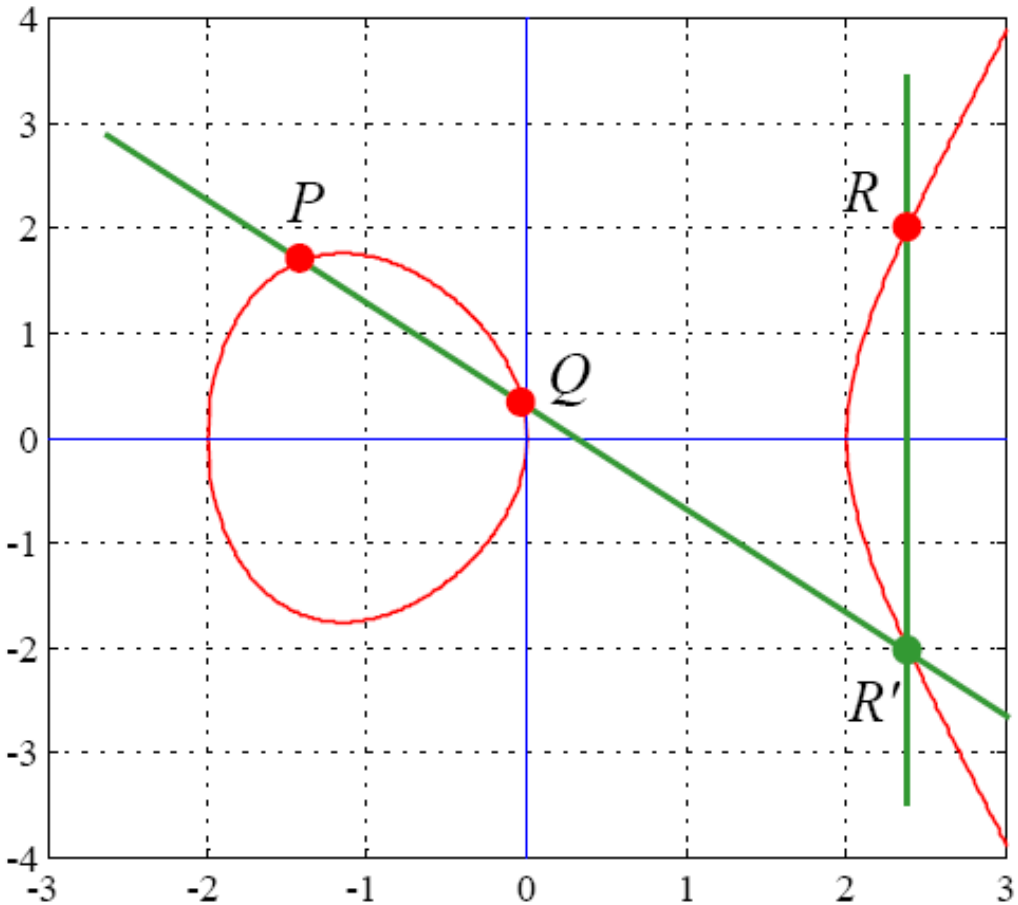






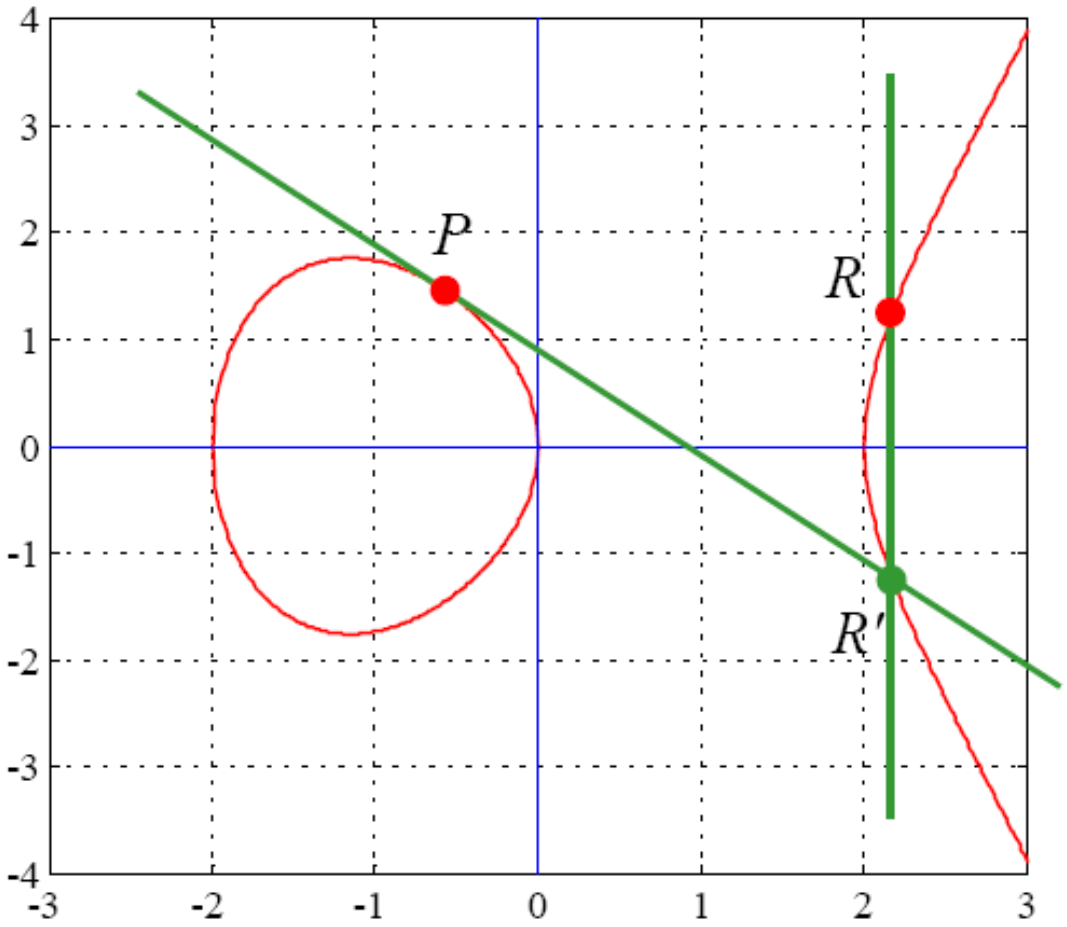
加法

$R = P + Q$ （或 $R = P \cdot Q$）



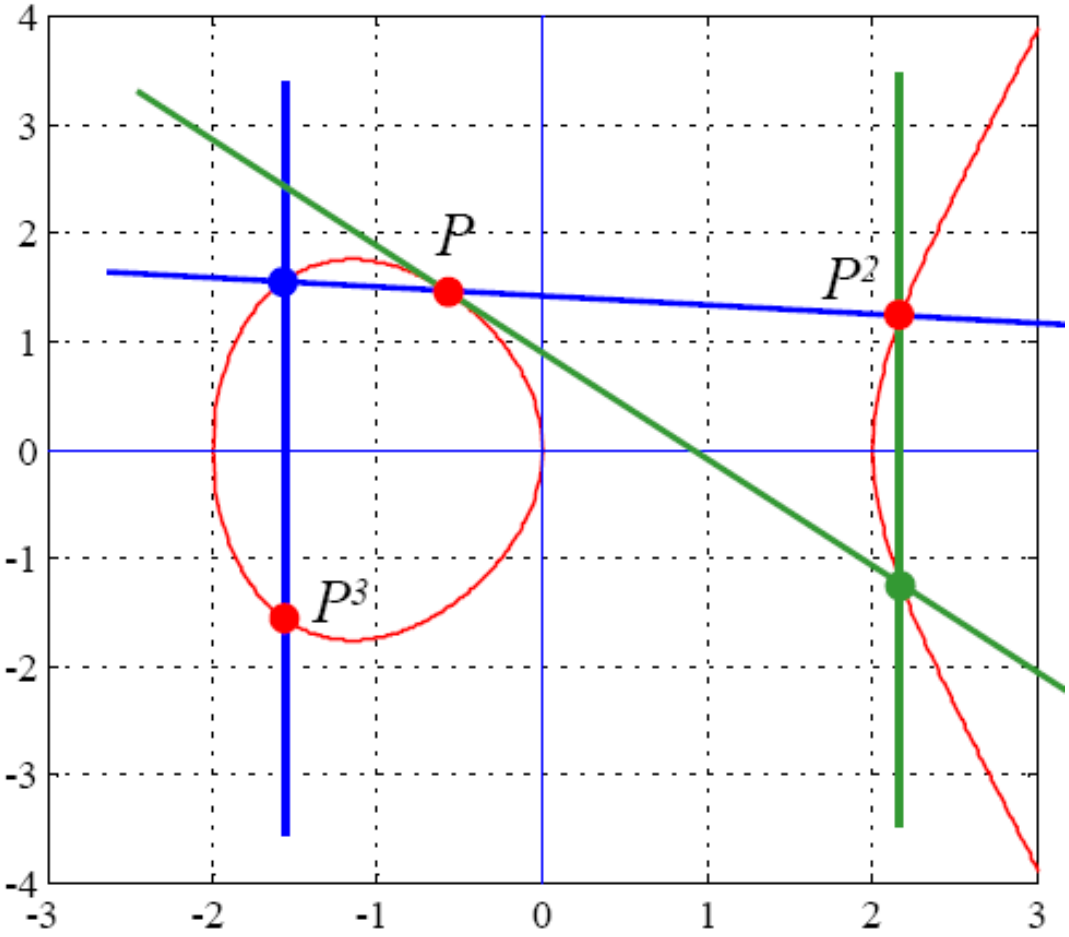
累加

$R = P + P = 2P$ （或 $R = P^2$）



累加

$R = P + P + P = 3P$ （或 $R = P^3$）



计算直线与椭圆曲线交点

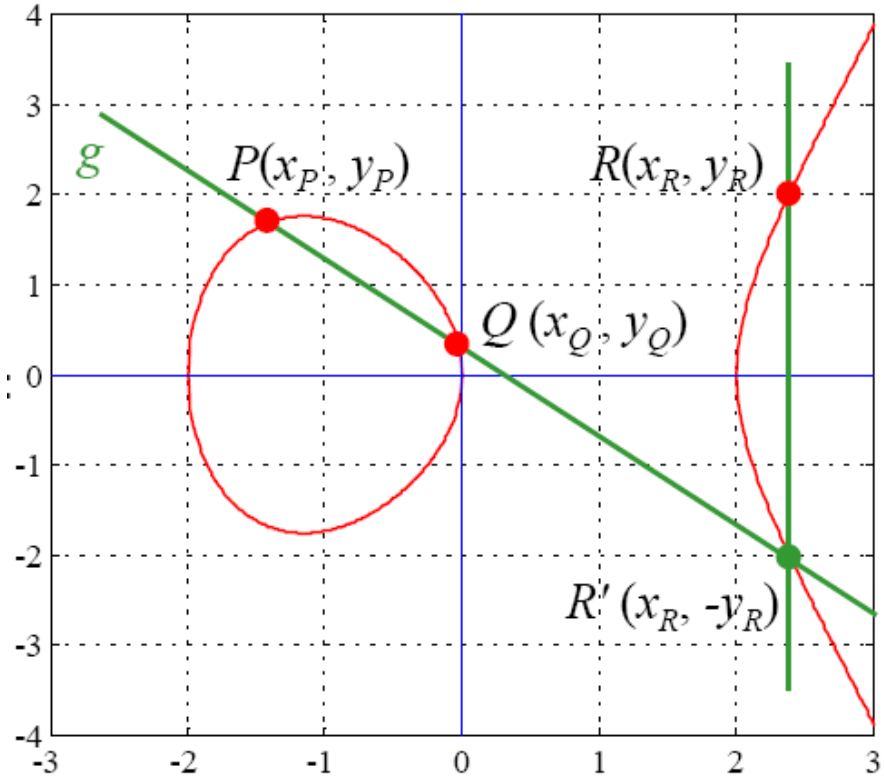
- 经过点 P 和 Q 的直线：

$$y = sx + y_0$$

其中

$$s = \frac{y_Q - y_P}{x_Q - x_P}$$

$$y_0 = y_P - sx_P$$



- 直线与曲线的另一个交点：

$$(sx + y_0)^2 = x^3 + ax + b$$

得到 R 点坐标：

$$x_R = s^2 - x_P - x_Q$$

$$y_R = -(sx_R + y_0)$$

计算切线与椭圆曲线交点

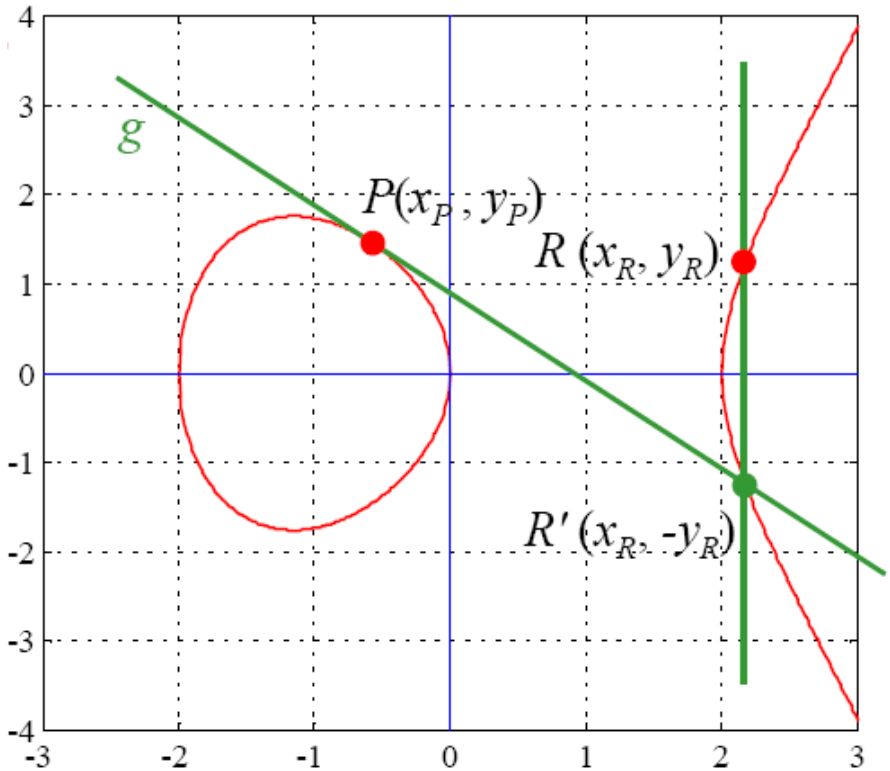
- P 点的切线：

$$y = sx + y_0$$

其中

$$s = \frac{dy}{dx} = \frac{3x_P^2 + a}{2y_P}$$

$$y_0 = y_P - sx_P$$



- 切线与曲线交点：

$$(sx + y_0)^2 = x^3 + ax + b$$

得到 R 点坐标：

$$x_R = s^2 - 2x_P$$

$$y_R = -(sx_R + y_0)$$

有限域上的椭圆曲线

- 椭圆曲线密码体制使用的是变元和系数均为有限域中元素的椭圆曲线。
- 定义在 $\mathrm{GF}(p)$ 上的椭圆曲线 $E_p(a,b)$：

$$y^2 = (x^3 + ax + b) \bmod p$$

  其中变元和系数均取自集合 $\{0, 1, \ldots, p-1\}$，模 p 运算；适合用软件实现。

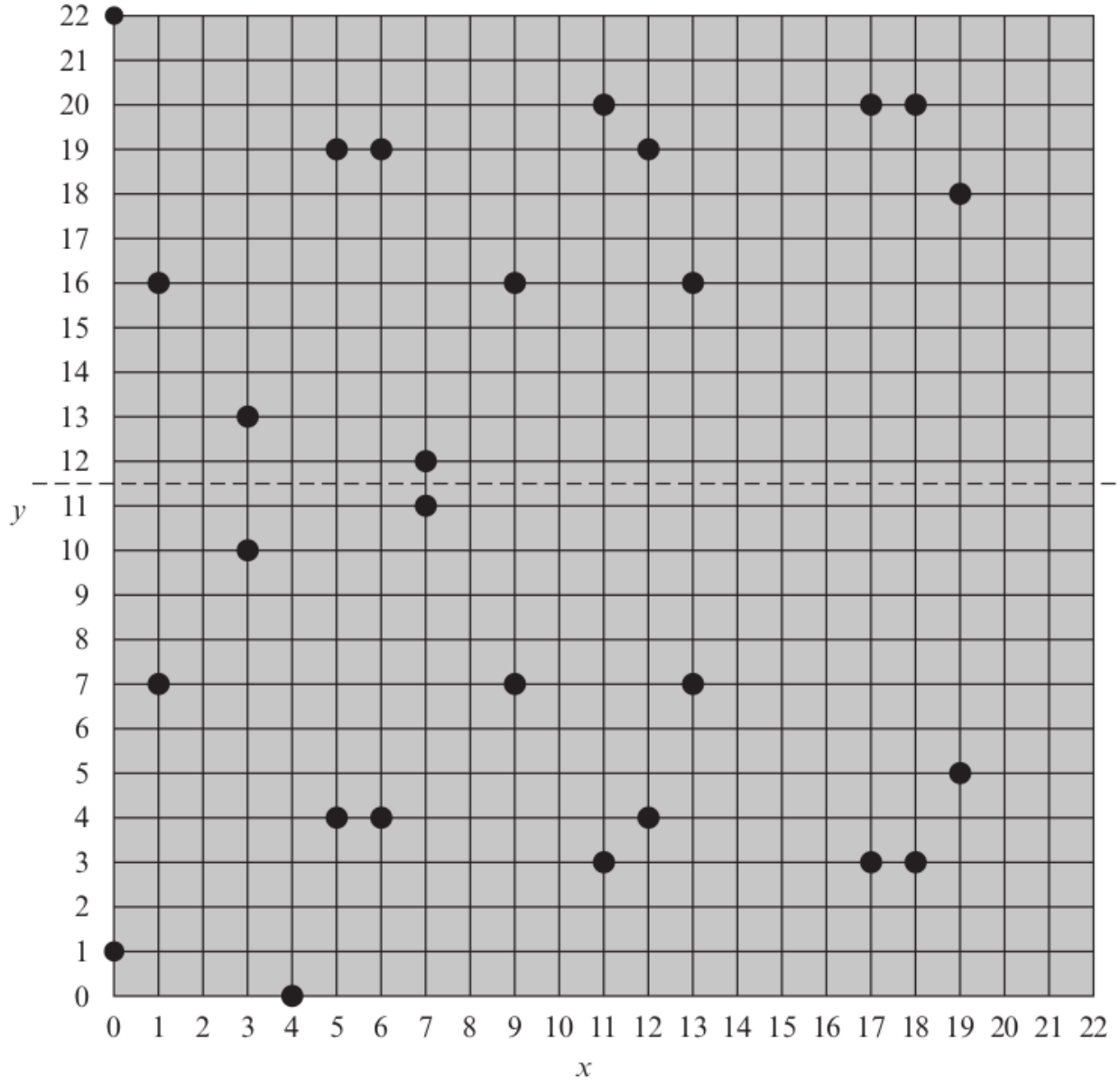
- 定义在 $\mathrm{GF}(2^n)$ 上的椭圆曲线 $E_{2^n}(a,b)$：
  - 变量和系数取自 $\mathrm{GF}(2^n)$，模素多项式（二进制多项式）；
  - 最适合硬件实现。

椭圆曲线 $E_{23}(1,1)$ 上的点

- 对于每个 $x \in \{0, \ldots, p-1\}$, 计算 $y^2 = x^3 + x + 1 \bmod p$;
- 对每个结果确定它是否有一个模 p 的平方根;
- 如果没有, 在 $E_{23}(1,1)$ 中就没有具有这个 x 值的点;
- 如果有, 就有两个满足平方根是 y 的值（除非这个值是单个的 y 值 0）。这些点就是 $E_{23}(1,1)$ 上的点（外加无穷远点）。

(0, 1)	(6, 4)	(12, 19)
(0, 22)	(6, 19)	(13, 7)
(1, 7)	(7, 11)	(13, 16)
(1, 16)	(7, 12)	(17, 3)
(3, 10)	(9, 7)	(17, 20)
(3, 13)	(9, 16)	(18, 3)
(4, 0)	(11, 3)	(18, 20)
(5, 4)	(11, 20)	(19, 5)
(5, 19)	(12, 4)	(19, 18)

椭圆曲线 $E_{23}(1,1)$ 上的点（除无穷远点外）



椭圆曲线 $E_{11}(1,6)$ 上的点

- 在 $\mathrm{GF}(11)$ 上找出椭圆曲线 $y^2 = x^3 + x + 6 \bmod 11$ 的点；
- 有 12 个点, 加上无穷远点 O 共有 $n = 13$ 个元素；
- n 称为椭圆曲线群的阶或序（Order），与参数 a, b 有关。

x	y^2	$y_{1,2}$	P(x,y)	P'(x,y)
0	6	-		
1	8	-		
2	5	4,7	(2,4)	(2,7)
3	3	5,6	(3,5)	(3,6)
4	8	-		
5	4	2,9	(5,2)	(5,9)
6	8	-		
7	4	2,9	(7,2)	(7,9)
8	9	3,8	(8,3)	(8,8)
9	7	-		
10	4	2,9	(10,2)	(10,9)

椭圆曲线点加运算

- 将椭圆曲线 $E_{11}(1,6)$ 上的点 $P(2,4)$ 反复累加
- 计算 $2P = P + P$

$$s = \frac{dy}{dx} = \frac{3x_P^2 + a}{2y_P} \qquad x_R = s^2 - 2x_P$$

$$y_0 = y_P - sx_P \qquad y_R = -(sx_R + y_0)$$

- 计算 $3P = P + P + P = 2P + P$

$$s = \frac{y_Q - y_P}{x_Q - x_P} \qquad x_R = s^2 - x_P - x_Q$$

$$y_0 = y_P - sx_P \qquad y_R = -(sx_R + y_0)$$

- 所有运算均在 $\mathrm{GF}(11)$ 上进行

椭圆曲线点加运算

- 取 $P(2,4)$, 计算 $2P = P + P$

$$s = \frac{3 \times 4 + 1}{2 \times 4} = \frac{13}{8} = 7 \times 2 = 3 \qquad x_R = 9 - 2 \times 2 = 5$$

$$y_0 = 4 - 3 \times 2 = -2 = 9 \qquad y_R = -(3 \times 5 + 9) = 9$$

所以 $2P = (5, 9)$

- 再计算 $3P = P + P + P = 2P + P$

$$s = \frac{9 - 4}{5 - 2} = \frac{5}{3} = 4 \times 5 = 9 \qquad x_R = 81 - 2 - 5 = 8$$

$$y_0 = 4 - 9 \times 2 = -3 = 8 \qquad y_R = -(9 \times 8 + 8) = -3 = 8$$

所以 $3P = (8, 8)$

定义在 $\mathrm{GF}(2^n)$ 上的椭圆曲线 $E_{2^n}(a,b)$

- 有限域 $\mathrm{GF}(2^n)$ 由 2^n 个元素及定义在多项式上的加法和乘法运算组成；
- 给定某 n，对于 $\mathrm{GF}(2^n)$ 上的椭圆曲线，使用变元和系数均在 $\mathrm{GF}(2^n)$ 上取值的三次方程，且利用 $\mathrm{GF}(2^n)$ 中的算术运算规则来进行计算；
- 通常，$\mathrm{GF}(2^n)$ 上适合椭圆曲线密码应用的三次方程为

$$y^2 + xy = x^3 + ax^2 + b$$

其中变元 x 和 y 以及系数 a 和 b 是 $\mathrm{GF}(2^n)$ 中的元素，计算在 $\mathrm{GF}(2^n)$ 中进行。

$\mathrm{GF}(2^n)$ 上的椭圆曲线

- 考虑所有整数对 (x, y) 和无穷远点 O 组成的集合 $E_{2^n}(a, b)$
- 例如，使用不可约多项式 $f(x) = x^4 + x + 1(10011)$ 定义的有限域 $\mathrm{GF}(2^4)$，其生成元 g 满足 $f(g) = 0$，即 $g^1 = 0010$，$g^4 = g + 1$，或二进制 0011，$g^5 = g^4 g = g^2 + g = 0110$

$g^0 = 0001$	$g^4 = 0011$	$g^8 = 0101$	$g^{12} = 1111$
$g^1 = 0010$	$g^5 = 0110$	$g^9 = 1010$	$g^{13} = 1101$
$g^2 = 0100$	$g^6 = 1100$	$g^{10} = 0111$	$g^{14} = 1001$
$g^3 = 1000$	$g^7 = 1011$	$g^{11} = 1110$	$g^{15} = 0001$

椭圆曲线 $E_{2^4}(g^4, 1)$

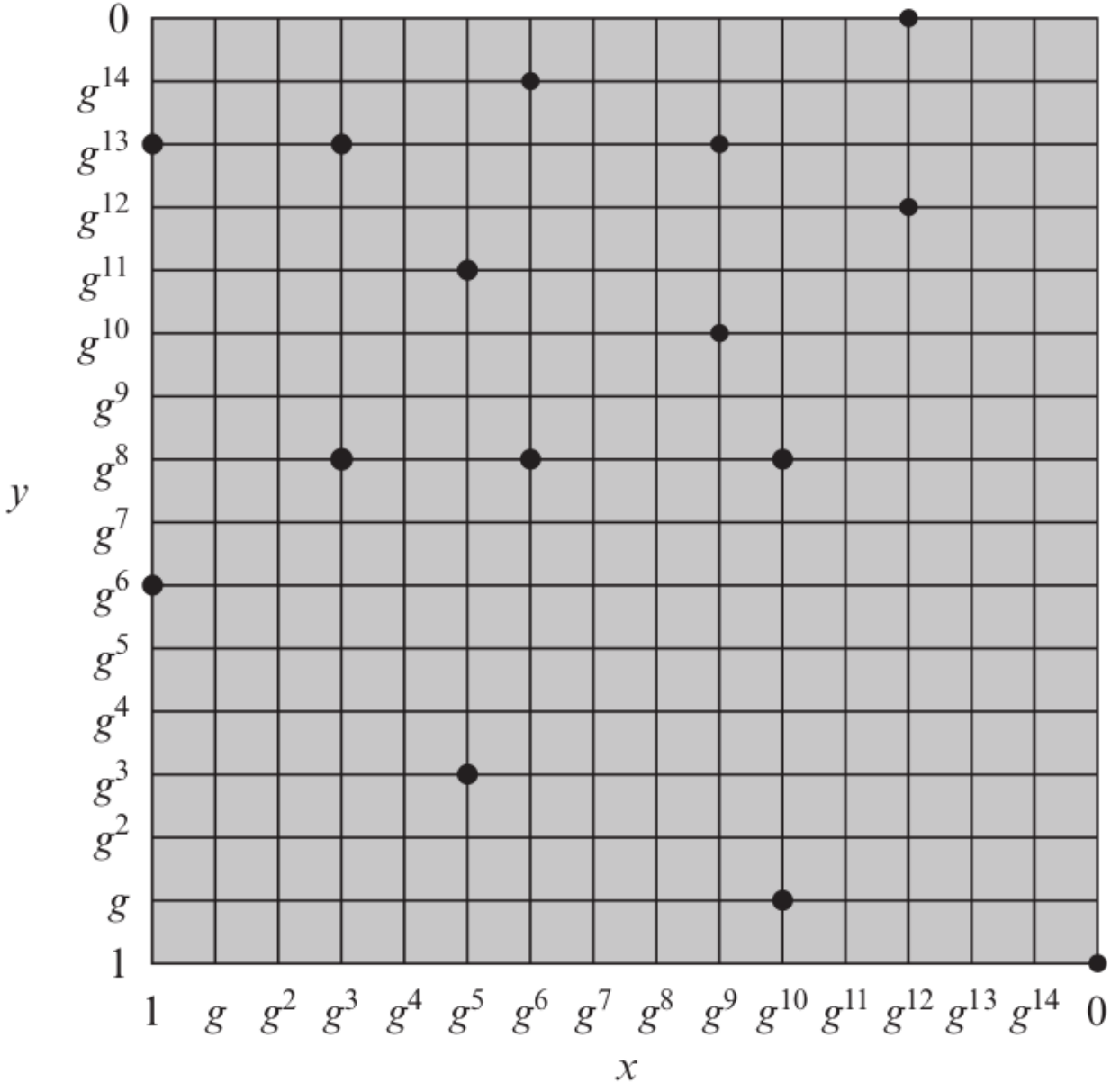
- 例如，考虑椭圆曲线 $y^2 + xy = x^3 + g^4x^2 + 1$，$a = g^4 = 0011$，$b = g^0 = 0001$, 满足该方程的一个点 (x, y) 为 (g^5, g^3)：

$$(g^3)^2 + (g^5)(g^3) = (g5)^3 + (g^4)(g^5)^2 + 1$$
$$g^6 + g^8 = g^{15} + g^{14} + 1$$
$$1100 + 0101 = 0001 + 1001 + 0001$$
$$1001 = 1001$$

- $E_{2^4}(g^4, 1)$ 上的点（除无穷远点外）：

$(0, 1)$	(g^5, g^3)	(g^9, g^{13})
$(1, g^6)$	(g^5, g^{11})	(g^{10}, g)
$(1, g^{13})$	(g^6, g^8)	(g^{10}, g^8)
(g^3, g^8)	(g^6, g^{14})	$(g^{12}, 0)$
(g^3, g^{13})	(g^9, g^{10})	(g^{12}, g^{12})

$E_{2^4}(g^4, 1)$ 上的点（除无穷远点外）



椭圆曲线密码学

- 大多数公开密钥密码系统如 RSA、D-H 都使用具有非常大数目的整数或多项式，计算量大，密钥和报文存储量也极大。
- 使用椭圆曲线密码系统 ECC，可以达到同样安全但密钥位数要小得多。
- ECC 的加类似于模乘，ECC 的重复加类似于模指数；
- ECC 需要有对应于 DLP 的难解问题：
  - $Q = kP$，Q，P 属于 $E_p(a, b)$，$k < P$；
  - 给定 k，P，容易计算 $Q = kP$；
  - 但是给定 Q，P，求 k 难；
  - 这就是椭圆曲线对数问题。

椭圆曲线对数问题

- 给定曲线 $y^2 = x^3 + ax + b \bmod p$ 及其上一点 P，我们可以通过连续自加 $k-1$ 次计算 $Q = kP$，目前存在这样的快速算法。
- 问题：当 Q 已知时能否计算 k?
- 答案：这是一个被称为椭圆曲线对数的难题。

k	P^k	s	y_0
1	(2,4)	3	9
2	(5,9)	9	8
3	(8,8)	8	10
4	(10,9)	2	0
5	(3,5)	1	2
6	(7,2)	4	7
7	(7,9)	1	2
8	(3,6)	2	0
9	(10,2)	8	10
10	(8,3)	9	8
11	(5,2)	3	9
12	(2,7)	∞	-
13	O	∞	-

椭圆曲线对数问题

- 例：$E_{23}(9,17)$, 即 $y^2 = x^3 + 9x + 7 \bmod 23$, 以 $P = (16,5)$ 为底的 $Q = (4,5)$ 的离散对数 k 为多少?
- 可以通过穷举攻击方法，多次计算 P 的倍数直至找到 Q 为止，这样 $P = (16,5); 2P = (20,20); 3P = (14,14); 4P = (19,20); 5P = (13,10); 6P = (7,3); 7P = (8,7); 8P = (12,17); 9P = (4,5)$.
- 所以, 以 $P = (16,5)$ 为底的 $Q = (4,5)$ 的离散对数 k 为 9。
- 实际应用中，k 的值非常大，穷举攻击不可行。

椭圆曲线对数问题

- 椭圆曲线密码系统的定义：
  - 域标识：定义椭圆曲线采用的有限域椭圆曲线：系数 a 和 b；
  - 基准点（base point)：指定的椭圆曲线上的点 G；
  - 阶（order)：G 点的阶 n，使得 $nG = O$。
- 椭圆曲线公钥系统：
  - $E_p(a, b)$, $\mathrm{GF}(p)$；
  - Base point G: (x, y)；
  - 选择 e 作为私有密钥；
  - 公开密钥为 $Q = eG$。

ECC Diffie-Hellman 密钥交换

- 类似于 D-H，ECC 也可以实现密钥交换
- 用户选择合适的 ECC, $E_p(a, b)$
- 选择基点 $G = (x_1, y_1)$, 满足 $nG = O$ 的最小 n 是一个大整数
- A 和 B 之间的密钥交换如下
  - A 和 B 选择私钥 $n_A < n$，$n_B < n$；
  - 计算公钥 $P_A = n_A G$，$P_B = n_B G$；
  - A 与 B 交换 P_A 和 P_B；
  - 计算共享密钥 $K = n_A P_B = n_B P_A$，因为 $K = n_A n_B G$，所以这两个密钥是一样的。

举例：ECC Diffie-Hellman 密钥交换

- $E_p(0,-4)$，即 $y^2=x^3-4$，$G=(2,2)$，$p=211$,$n=240$；
- 计算 $240G=O$；
- $n_A=121$，$P_A=121(2,2)=(115,48)$；
- $n_B=203$，$P_B=203(2,2)=(130,203)$；
- $K=121(130,203)=203(115,48)=(161,69)$。

椭圆曲线加密

- 首先将明文消息 m 编码为 (x, y) 的点 P_m，点 P_m 就是要进行加密的点。注意不能简单地把消息编码为点的 x 坐标或 y 坐标，因为并不是所有的坐标都在 $E_p(a, b)$ 中。
- 类似于 D-H 密钥交换，加解密系统也需要点 G 和椭圆群 $E_p(a, b)$ 这些参数。
- 用户 A 选择私钥 $n_A < n$，并产生公钥 $P_A = n_A G$；
- 用户 B 选择私钥 $n_B < n$，并产生公钥 $P_B = n_B G$；
- A 要给 B 发送 P_m；
- A 随机选择一个正整数 k，加密点 P_m 产生密文：$C = \{kG, P_m + kP_B\}$；
- B 解密 C，计算：$P_m + kP_B – n_B(kG) = P_m + k(n_B G) – n_B(kG) = P_m$。

举例：椭圆曲线加密

- $E_p(-1, 188)$, 即 $y^2 = x^3 - x + 188$, $G = (0, 376)$, $p = 751$;
- A 要向 B 发送消息 $P_m = (562, 201)$;
- A 首先随机选择 $k = 386$, 并获得 B 的公钥 $P_B = (201, 5)$;
- 计算 $kG = 386(0, 376) = (676, 558)$;
- $P_m + kP_B = (562, 201) + 386(201, 5) = (385, 328)$;
- 这样, 密文即为 $C = kG, P_m + kP_B = (676, 558), (385, 328)$;
- B 要解密
  $P_m + kP_B – n_B(kG) = P_m + k(n_BG) – n_B(kG) = P_m$。

椭圆曲线密码的安全性

- ECC 的安全性是建立在由 kP 和 P 确定 k 的难度之上的，即椭圆曲线对数问题。
- ECC 可以使用比 RSA 短得多的密钥。
- 密钥长度相同时，ECC 与 RSA 所执行的计算量也差不多。
- 与具有同等安全性的 RSA 相比，由于 ECC 使用的密钥更短，所以 ECC 所需的计算量比 RSA 少。

Symmetric	56	80	112	128	192	256
RSA n	512	1024	2048	3072	7680	15360
ECC p	112	161	224	256	384	512
Key size ratio	5:1	6:1	9:1	12:1	20:1	30:1

小结

1. 公钥分配
2. Diffie-Hellman 密钥交换
3. 椭圆曲线密码