密 码 学

第五章 公钥密码算法

网络空间安全学院
朱丹 戚明平
zhudan/mpqi@nwpu.edu.cn

知识回顾-离散对数问题

- 离散对数问题（Discrete Logarithm Problem, DLP）
  - 设p为素数，则模p的剩余构成有限域

$$F_p = GF(p) = \{0,1,2,\dots\dots,p-1\}$$

F_p的非零元素构成乘法循环群$F_p^* = \{1,2,\dots\dots,p-1\} = \{a,a^2,\dots,a^{p-1}\}$，

则称a是F_p^*的生成元或模p的本原元。

  - 求a的模幂运算为$y = a^x \bmod p, 1 \le x \le p-1$，求对数$x$的运算为$x = \log_a y, 1 \le x \le p-1$，由于上述运算是定义在有限域$F_p$上的，所以称为离散对数运算。
  - 从x计算y是容易的。从y计算x就困难得多，利用目前最好的算法，对于小心选择的p将至少需用$O(p^{\frac{1}{2}})$次以上的运算，只要p足够大，求解离散对数问题是相当困难的。

知识回顾-离散对数问题

离散对数问题（Discrete Logarithm Problem, DLP）

- 离散对数具有良好的单向性，在公钥密码中得到了广泛的应用，如
  - Diffie-Hellman密钥交换协议
  - ElGamal公钥密码算法
  - 美国数字签名算法（DSA, Digital Signature Algorithm）
  - ElGamal公钥密码算法改进了Diffie-Hellman密钥交换协议，提出了基于离散对数的公钥密码和数字签名体制

知识回顾-Diffie-Hellman密钥交换协议

参数选取：

选取大素数p，再选取Z_p的本原元a，并将p和a公开，全网公用。

密钥协商：

$1 \le x_A \le p-2$

$1 \le x_B \le p-2$

Alice  x_A

$$y_A = a^{x_A} \bmod p \quad (\text{Alice} \rightarrow \text{Bob})$$

$$y_B = a^{x_B} \bmod p \quad (\text{Bob} \rightarrow \text{Alice})$$

Bob  x_B

$$k = (y_B)^{x_A} \quad \boxed{= a^{x_A x_B} =} \quad k = (y_A)^{x_B}$$

将k作为Alice和Bob协商出来的密钥，同时不再保留x_A和x_B；攻击者如果想要获得x_A或者x_B，必须解决DLP问题

知识回顾-Diffie-Hellman密钥交换协议

安全性:

- 有限域上的离散对数问题是密码学中的困难问题，目前没有快速求解有限域上离散对数的数学方法。最快算法的复杂度为亚指数级别，当p为200位时，求解需要2~3天；而当p为664位时，求解约需2.7亿年。只要p足够大，D-H密钥交换协议就可以达到足够安全。

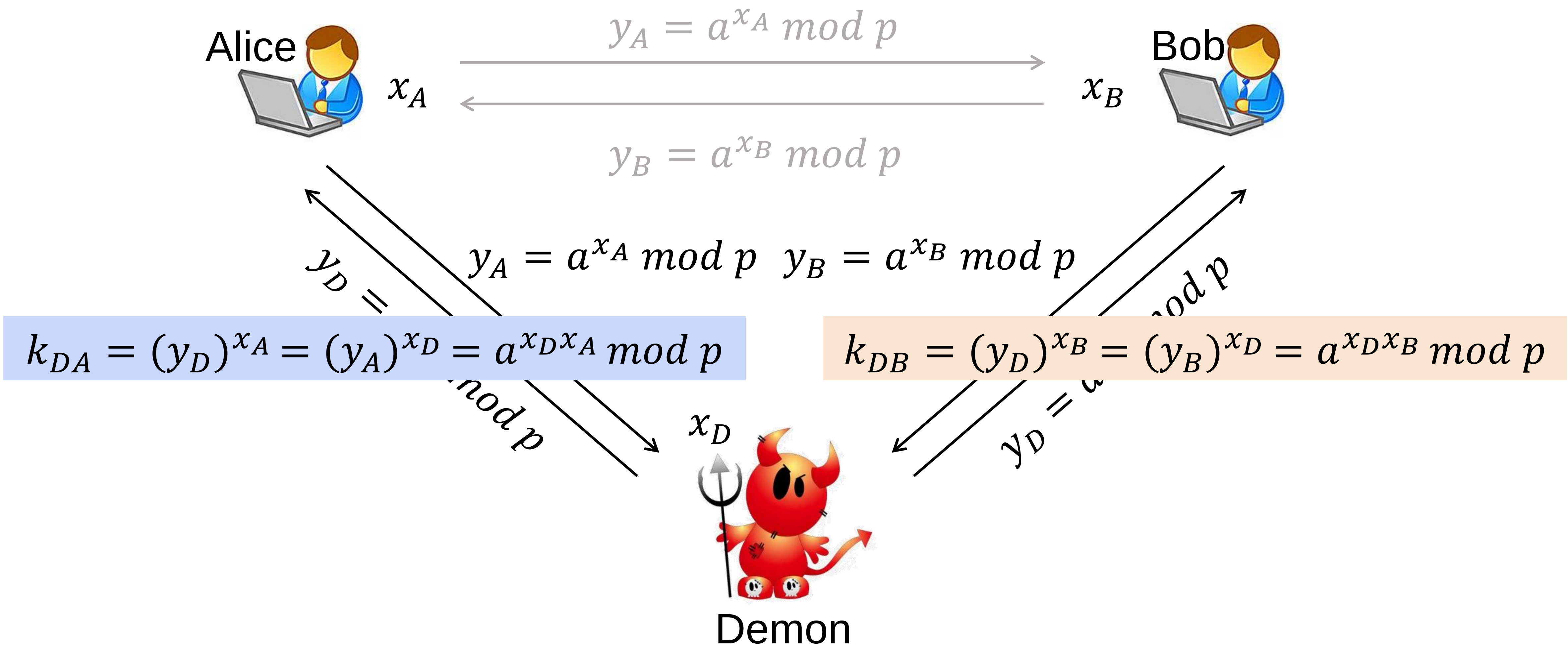
优点:

- 任何两人都可以协商出会话密钥
- 降低通信双方的密钥管理负担

缺点:

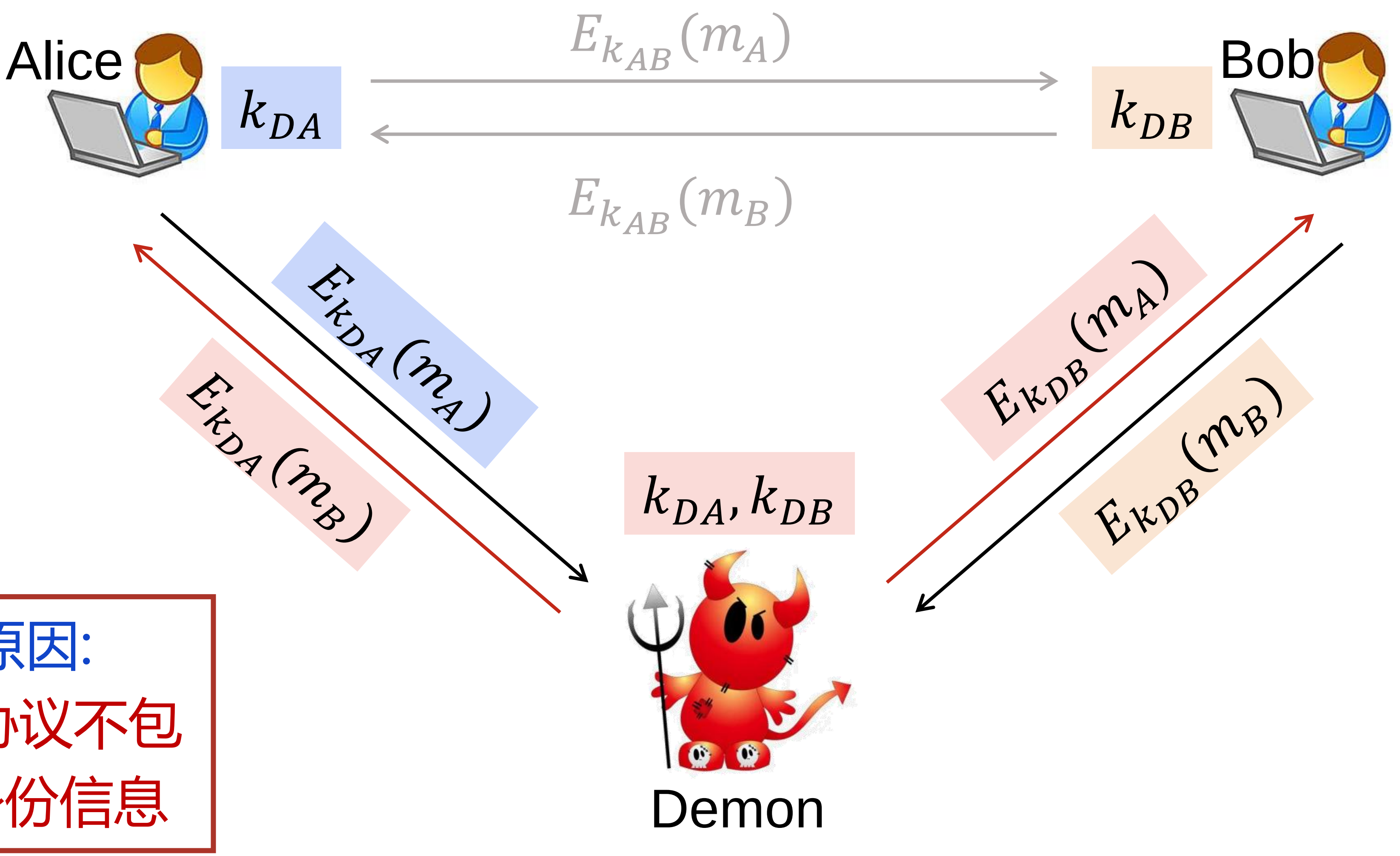
- 易遭受中间人攻击

知识回顾-中间人攻击


Alice
Bob
Demon
x_A
x_B
x_D
$y_A = a^{x_A} \bmod p$
$y_B = a^{x_B} \bmod p$
$y_A = a^{x_A} \bmod p$
$y_B = a^{x_B} \bmod p$
$k_{DA} = (y_D)^{x_A} = (y_A)^{x_D} = a^{x_D x_A} \bmod p$
$k_{DB} = (y_D)^{x_B} = (y_B)^{x_D} = a^{x_D x_B} \bmod p$

知识回顾-中间人攻击

Man-In-The-Middle Attack



攻击原因:
DH交换协议不包含任何身份信息

身份认证:
公钥证书
数字签名

知识回顾-ElGamal公钥密码算法

参数选取：

随机地选择一个大素数p，且要求$p-1$有大素数因子。再选择一个模p的本原元a。

将p和a公开作为算法的基础参数。

密钥生成：

- 用户随机地选择一个整数d作为自己保密的解密密钥k_d，$2 \le d \le p-2$;
- 用户计算$y = a^d \bmod p$，并取y作为自己公开的加密密钥k_e。

显然，由公开密钥y计算解密密钥d，必须计算离散对数，这是极困难的。

知识回顾-ElGamal公钥密码算法

加密算法：

将明文消息$M(0 \leq M \leq p-1)$加密成密文的过程如下：

- 随机地选择一个整数k，$2 \leq k \leq p-2$；
- 计算：

$$U = y^k \ mod \ p$$

$$C_1 = a^k \ mod \ p$$

$$C_2 = UM \ mod \ p$$

- 取$C = (C_1, C_2)$作为密文。

解密算法：

将密文消息(C_1, C_2)解密成明文的过程如下：

- 计算：

$$V = C_1^d \ mod \ p$$

- 计算：

$$M = C_2 V^{-1} \ mod \ p$$

- 获得明文M。

知识回顾-ElGamal公钥密码算法

- ElGamal公钥密码算法的安全性
  - 由于ElGamal公钥密码的安全性建立在$GF(p)$离散对数的困难性之上，目前尚无求解$GF(p)$离散对数有效算法，所以在p足够大时Elgamal密码是安全的。
  - 为了安全p应为150位以上的十进制数，而且$p-1$应有大素因子。
  - d和k都不能太小，应为高质量的随机数。
  - 为了安全加密和签名所使用的k必须是一次性的。

素数域上的椭圆曲线问题

- 0元素设p是大于3的素数，且$4a^3+27b^2 \neq 0 \bmod p$，称

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

$a, b \in GF(p)$为$GF(p)$ 上的椭圆曲线。

- 由椭圆曲线可得到一个同余方程：

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

其解为一个二元组$< x, y >, x, y \in GF(p)$，将此二元组描画到椭圆曲线上便为一个点，称其为一个解点。

知识回顾-椭圆曲线离散对数问题

素数域上的椭圆曲线问题

- 定义单位元：引进一个无穷点$O(\infty,\infty)$，简记为O，作为0元素

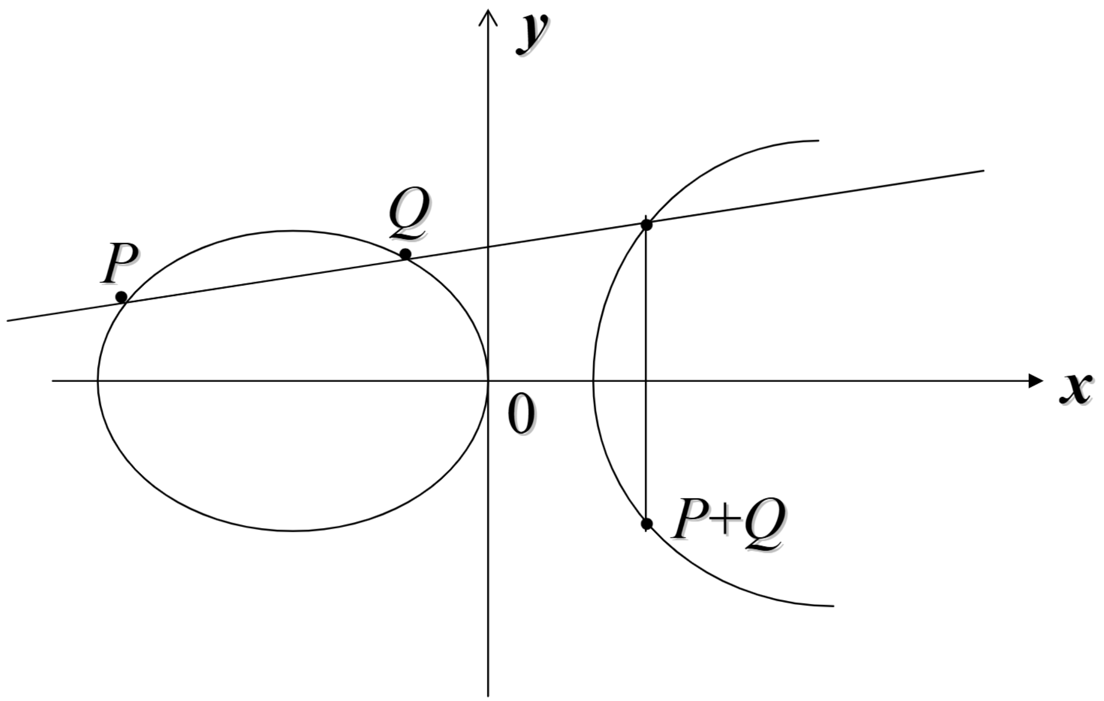
$$O(\infty,\infty)+O(\infty,\infty)=0+0=0$$

- 定义逆元素：任何解点$R(x,y)$的逆就是$R(x,-y)$，规定无穷远点的逆是本身。
- 定义加法：$P(x_1,y_1)+Q(x_2,y_2)=R(x_3,y_3)$，其中

$$\begin{cases} x_3=\lambda^2-x_1-x_2 \\ y_3=\lambda(x_1-x_3)-y_1 \\ \lambda=\dfrac{(y_2-y_1)}{(x_2-x_1)}, P\neq Q \\ \lambda=\dfrac{(3x_1^2+a)}{2y_1}, P=Q \end{cases}$$

椭圆曲线解点加法运算的几何意义

- 设$P(x_1,y_1)$和$Q(x_2,y_2)$是椭圆曲线上的两个点，则连接$P(x_1,y_1)$和$Q(x_2,y_2)$的直线与椭圆曲线的另一交点关于横轴的对称点即为$P(x_1,y_1)+Q(x_2,y_2)$点



椭圆曲线离散对数问题

- 由于是加法群，n个元素G相加表示为：

$$G+G+\cdots+G=nG$$

称为倍点运算

$$\begin{cases} x_3=\lambda^2-x_1-x_2 \\ y_3=\lambda(x_1-x_3)-y_1 \\ \lambda=\dfrac{(y_2-y_1)}{(x_2-x_1)}, P\neq Q \\ \lambda=\dfrac{(3x_1^2+a)}{2y_1}, P=Q \end{cases}$$

- 椭圆曲线$y^2=x^3+x+6\ mod\ 11$，取$G=(2,7)$为生成元，2倍点计算如下：

$$2G=(2,7)+(2,7)=(5,2)$$

因为$\lambda=\left(3\times 2^2+1\right)(2\times 7)^{-1}\ mod\ 11=2\times 3^{-1}\ mod\ 11=2\times 4\ mod\ 11=8$

于是，$x_3=8^2-2\times 2\ mod\ 11=5$，$y_3=8\times(2-5)-7\ mod\ 11=2$

椭圆曲线离散对数问题

G	$2G$	$3G$	$4G$	$5G$	$6G$	$7G$
$(2,7)$	$(5,2)$	$(8,3)$	$(10,2)$	$(3,6)$	$(7,9)$	$(7,2)$

$8G$	$9G$	$10G$	$11G$	$12G$	$13G$
$(3,5)$	$(10,9)$	$(8,8)$	$(5,9)$	$(2,4)$	$O(\infty,\infty)$

- 上例中，p较小，使得$GF(p)$也较小，故可以利用穷举法求出所有解点。但是，对于一般情况要计算椭圆曲线解点数N的准确值比较困难
- N满足不等式 $p+1-2p^{1/2} \le N \le p+1+2p^{1/2}$

椭圆曲线离散对数问题

- 在上例中椭圆曲线上的解点所构成的交换群恰好是循环群，但是一般并不一定。于是我们希望从中找出一个循环子群E_1
- 现有研究已经证明：当循环子群E_1的阶n是足够大的素数时，这个循环子群中的离散对数问题是困难的
- 除了$GF(p)$上的椭圆曲线，还有定义在$GF(2^m)$上的椭圆曲线。基于这两种椭圆曲线都可以设计出安全的椭圆曲线密码

椭圆曲线离散对数问题

设P和Q是椭圆曲线上的两个解点，t为一正整数，且$1 \leq t < n$

- 对于给定的P和t，计算$tP = Q$是容易的
- 但若已知P和Q点，要计算出t则是极困难的。这便是椭圆曲线群上的离散对数问题，简记为ECDLP (Elliptic Curve Discrete Logarithm Problem)
- 除了几类特殊的椭圆曲线外，对于一般ECDLP目前尚没有找到有效的求解方法。因子分解和DLP问题都有亚指数求解算法，而ECDLP尚没有发现亚指数求解算法
- 于是可以在这个循环子群E_1中建立任何基于离散对数困难性的密码，并称这个密码为椭圆曲线密码

章 节 安 排

Outline

章节安排

Outline

5.10(2) 椭圆曲线公钥密码

- 一些国际标准化组织已把椭圆曲线密码作为新的信息安全标准。如，IEEE P1363/D4，ANSI F9.62，ANSI F9.63等标准，分别规范了椭圆曲线密码在Internet协议安全、电子商务、Web服务器、空间通信、移动通信、智能卡等方面的应用
- 它密钥短，软件实现规模小、硬件实现节省电路。由于椭圆曲线离散对数问题尚没有发现亚指数算法，所以普遍认为，椭圆曲线密码比RSA和ElGamal密码更安全
- 椭圆曲线密码已成为RSA之外呼声最高的公钥密码之一

5.10(2) 椭圆曲线公钥密码

- 160位的椭圆曲线密码的安全性相当于1024位的RSA密码，而且运算速度也较快
- ElGamal密码建立在有限域$GF(p)$的乘法群的离散对数问题的困难性之上。而椭圆曲线密码建立在椭圆曲线群的离散对数问题的困难性之上。两者的主要区别是其离散对数问题所依赖的群不同。因此两者有许多相似之处
- 基于$GF(p)$和$GF(2^m)$上的椭圆曲线，都可以构成安全的椭圆曲线密码
- 我国商用密码采用了椭圆曲线密码，并具体颁布了椭圆曲线密码标准算法SM2

5.10(2) 椭圆曲线公钥密码

$GF(p)$上椭圆曲线密码算法

$GF(p)$上椭圆曲线密码的基础参数

- $T=<p,a,b,G,n,h>$，p为大于3素数， p确定了有限域$GF(p)$
- 元素$a,b\in GF(p)$，a和b确定了椭圆曲线：$y^2=x^3+ax+b$，$a,b\in GF(p)$
- G为循环子群E_1的生成元，n为素数且为生成元G的阶， G和n确定了循环子群E_1
- $h=\frac{|E|}{n}$，并称为余因子，h将交换群E和循环子群E_1联系起来

5.10(2) 椭圆曲线公钥密码

$GF(p)$上椭圆曲线密码算法

$GF(p)$上椭圆曲线密码的密钥

- 用户的私钥定义为一个随机数d，$d \in \{1,2,\cdots,n-1\}$
- 用户的公钥定义为Q点，$Q = dG$
- 由公钥Q求私钥d是求解椭圆曲线离散对数问题，当p足够大时，这是困难的

5.10(2) 椭圆曲线公钥密码

$GF(p)$上椭圆曲线密码算法

- 设d为用户私钥， Q为用户公钥，明文数据为M， $0 \le M \le n-1$
- 加密过程
  - S1: 选择一个随机数k，且$k \in \{1, 2, \cdots, n-1\}$
  - S2: 计算点$X_1(x_1, y_1) = kG$
  - S3: 计算点$X_2(x_2, y_2) = kQ$，如果分量$x_2 = 0$，则转S1
  - S4: 计算密文$C = Mx_2 \bmod n$
  - (X_1, C)为最终的密文数据

5.10(2) 椭圆曲线公钥密码

$GF(p)$上椭圆曲线密码算法

- 设d为用户私钥， Q为用户公钥，密文数据为(X_1, C)
- 解密过程
  - S1: 利用私钥d求出$X_2(x_2, y_2)$

$$dX_1 = d(kG) = k(dG) = kQ = X_2$$

  - S2: 利用$X_2(x_2, y_2)$计算得到明文M

$$M = Cx_2^{-1}\ mod\ n$$

5.10(2) 椭圆曲线公钥密码

$GF(p)$上椭圆曲线密码算法

- 例题：椭圆曲线$y^2 = x^3 + x + 6 \, mod \, 11$，以$G = (2,7)$为生成元构造椭圆曲线密码，设明文$M = 3$，进行加密和解密计算。

  - 密钥选取：选取私钥$d = 7$，计算公钥$Q = dG = 7G = (7,2)$
  - 消息加密：选择一个随机数$k = 3$，计算点$X_1(x_1, y_1) = kG = 3G = (8,3)$,计算点$X_2(x_2, y_2) = kQ = 3Q = (1,3)$

$$2Q = Q + Q = (2,7)$$

$$3Q = 2Q + Q = (3,5)$$

5.10(2) 椭圆曲线公钥密码

$GF(p)$上椭圆曲线密码算法

- 例题：椭圆曲线$y^2 = x^3 + x + 6 \: mod \: 11$，以$G = (2,7)$为生成元构造椭圆曲线密码，设明文$M = 3$，进行加密和解密计算。

  - 消息加密：计算点$X_1(x_1, y_1) = 3G = (8,3)$，点$X_2(x_2, y_2) = 3Q = (3,5)$

$$C = Mx_2 \: mod \: n = 9$$

    最中密文数据为[(8,3), 9]

  - 消息解密：计算点$dX_1 = 21G = 8G = (3,5) = X_2(x_2, y_2)$

$$M = Cx_2^{-1} \: mod \: n = 9 \times 4 \: mod \: n = 3$$

5.10(2) 椭圆曲线公钥密码

$GF(p)$上椭圆曲线密码算法

- $GF(p)$上椭圆曲线密码的实现：由于椭圆曲线密码所依据的数学基础比较复杂，从而使得其工程实现也比较困难。虽然目前椭圆曲线密码的实现技术已经成熟，但仍有些难度问题值得研究和改进。

- 实现的难点问题
  - 安全椭圆曲线的产生：美国NIST公布了15条曲线，我们应对其进行验证这之外还有好曲线吗？如何产生？
  - 倍点运算

章节安排

Outline

5.11 SM2公钥密码算法

- 推荐使用256位素域$GF(p)$上的椭圆曲线 $y^2 = x^3 + ax + b$，曲线参数如下：

$$p = FFFFFFFE\ FFFFFFFF\ FFFFFFFF\ FFFFFFFF\ FFFFFFFF\ 00000000\ FFFFFFFF\ FFFFFFFF$$
$$a = FFFFFFFE\ FFFFFFFF\ FFFFFFFF\ FFFFFFFF\ FFFFFFFF\ 00000000\ FFFFFFFF\ FFFFFFFC$$
$$b = 28E9FA9E\ 9D9F5E34\ 4D5A9E4B\ CF6509A7\ F39789F5\ 15AB8F92\ DDBCBD41\ 4D940E93$$
$$n = FFFFFFFE\ FFFFFFFF\ FFFFFFFF\ FFFFFFFF\ 7203DF6B\ 21C6052B\ 53BBF409\ 39D54123$$
$$x_G = 32C4AE2C\ 1F198119\ 5F990446\ 6A39C994\ 8FE30BBF\ F2660BE1\ 715A4589\ 334C74C7$$
$$y_G = BC3736A2\ F4F6779C\ 59BDCEE3\ 6B692153\ D0A9877C\ C62A4740\ 02DF32E5\ 2139F0A0$$

- 密钥：
  - 私钥是随机数$d, d \in [1, n-1]$
  - 公钥$P = dG = d(x_G, y_G)$

SM2的加密算法:

- S1: 产生随机数k，$1 \leq k \leq n - 1$
- S2: 计算椭圆曲线点$C_1 = kG = (x_1, y_1)$
- S3: 计算椭圆曲线点$kP = (x_2, y_2)$
- S4: 计算$t = KDF(x_2 \| y_2, klen)$，若$t$为全0比特串，则返回S1

  //$KDF(Z, klen)$是密钥派生函数，它利用HASH函数从数据Z产生出长度为$klen$的密钥数据
- S5: 计算$C_2 = M \oplus t$;
- S6: 计算$C_3 = Hash(x_2 \| M \| y_2)$
- S7: 输出密文$C = C_1 \| C_2 \| C_3$

5.11 SM2公钥密码算法

SM2的解密算法：

- S1: 计算$dC_1 = (x_2, y_2)$
- S2: 计算椭圆曲线点$t = KDF(x_2 \| y_2, klen)$，若$t$为全0比特串，则报错退出
- S3: 计算椭圆曲线点$M' = C_2 \oplus t$
- S4: 输出明文M'

5.11 SM2公钥密码算法

- 解密正确性:
  - 证明： $dC_1 = k(dG) = kP = (x_2, y_2)$，如果$(x_2, y_2)$是正确的，则$t = KDF(x_2 \| y_2, klen)$也将是正确的。又因为$C_2 = M \oplus t$，所以$M' = C_2 \oplus t$
  - 验证：根据解密得到的x_2, y_2和M'重新计算C_3，并于接收到的C_3比较，若相等则说明密文和解密正确，否则说明密文或解密不正确
- SM2的算法应用:
  - 我国二代居民身份证采用了SM2椭圆曲线密码
  - 我国的许多商用系统采用了SM2椭圆曲线密码

5.11 SM2公钥密码算法

- 传统ECC与SM2的比较：

传统ECC	SM2
计算点$X_2\ (x_2, y_2) = kQ$	计算点$t = KDF(x_2 \| y_2, klen)$
计算密文$C = Mx_2 \bmod n$	计算密文$C_2 = M \oplus t$
最终密文数据(X_1, C)	最终密文数据$C\ = C_1 \| C_2 \| C_3$

- 传统ECC用x_2加密，加密是乘法；SM2处理(x_2, y_2)产生t，用t加密，加密是模2加
- t与(x_2, y_2)相关，更安全。SM2利用C_3的验证，进一步确保密文和解密的正确性
- SM2的密文数据扩展较大

密 码 学

第六章 哈希函数

网络空间安全学院
朱丹 戚明平
zhudan/mpqi@nwpu.edu.cn

章 节 安 排

Outline

章 节 安 排

Outline

6.1 哈希函数简介

Hash函数的概念

- Hash函数又称哈希函数、杂凑函数、散列函数等，它能够将“任意长度”的消息变为某一固定长度的消息摘要（也称数字指纹、报文摘要、杂凑值、散列值等）。通常记为 $h = H(M)$或$Hash(M)$。

Hash函数的性质

- Hash函数的输入可以是“任意长度”的消息；
- Hash函数的输出位长固定，多数情况下输入的长度是大于输出的长度的，因此Hash函数具有压缩特性；
- 有效性，即对给定的m，计算h=H(m)的运算是高效的；
- 具有极强的错误检测能力，即输入有很小的不同，输出将有很大的不同

6.1 哈希函数简介

Hash函数的定义

- Hash函数将任意长的数据M变换为定长的码h，记为：

$$h = H(M)\text{或者}h = Hash(M)$$

一般，h的长度小于M的长度，因此HASH函数是一种压缩变换

- 安全性：
  - 随机性：哈希函数的输出具有伪随机性
  - 单向性：对给定的Hash值h，找到满足$H(x) = h$的x在计算上是不可行的

设h码为n位长，且Hash函数的输出值是等概分布的，那么任意输入数据x产生的$H(x)$恰好为h的概率是$\frac{1}{2^n}$。因此穷举攻击对于单向性求解的时间复杂度为$O(2^n)$

- **抗弱碰撞性**：对任何给定的x，找到满足$y \neq x$且$H(x) = H(y)$的y在计算上是不可行的

否则，攻击者可以截获报文M及其$H(M)$，并找出另一报文M'使得$H(M') = H(M)$。这样攻击者可用M'去冒充M，而收方不能发现。

抗弱碰撞又称为抗求第二原像。

从穷举分析的角度求解弱碰撞问题的难度等价于求解单向性的难度，时间复杂度为$O(2^n)$

- **抗强碰撞性**：找到任何满足$H(x) = H(y)$的偶对(x, y)在计算上是不可行的

平均需要尝试超过$2^{\frac{n}{2}}$个数据就能产生一个碰撞，复杂度$O(2^{\frac{n}{2}})$。

6.1 哈希函数简介

对Hash函数的基本要求

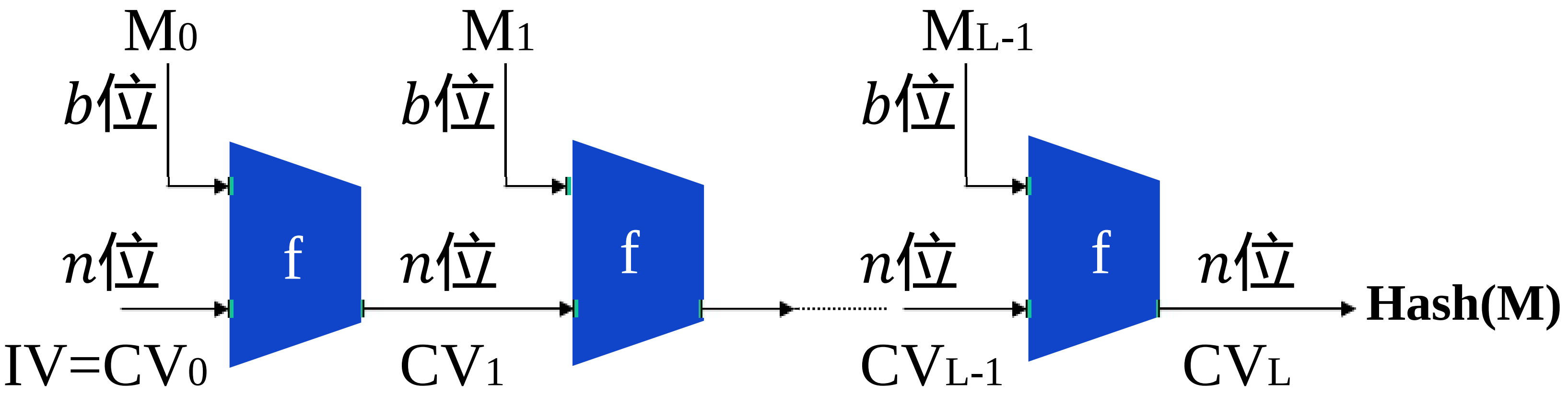
- 可应用于任意大小的数据块
- 能产生定长的输出
- 有效性
- 随机性、单向性、抗弱碰撞性、抗强碰撞性（安全性）

Hash函数的主要应用

- 用于完整性保护
- 利用函数构造消息认证码（MAC），用于认证
- 用于辅助公钥加密、数字签名、密钥交换等

安全Hash函数处理数据的一般模型

- Merkle最早提出了Hash函数处理数据M的一般模型，其中$M = M_0 \parallel \cdots \parallel M_{L-1}$



- b为分组长度，f为压缩函数，L为链接迭代轮数，n为输出位数。

6.1 哈希函数简介

安全Hash函数处理数据的一般模型

- 分组：将输入M分为$L-1$个大小为b位的分组
- 填充：若第$L-1$个分组不足b位，则将其填充为b位
- 附加：再附加上一个输入的总长度

填充和附加之后，共L个大小为b位的分组。

由于输入中包含长度，所以攻击者必须找出具有相同Hash值且长度相等的两条报文，或者找出两条长度不等但加入报文长度后Hash值相同的报文，从而增加了攻击的难度。

目前大多数Hash函数均采用这种数据处理模型。


感谢聆听!
THANK YOU FOR YOUR ATTENTION!