Something about cryptography 3

汇报时间：2018年1月　　讲师：1phan

目 录

CATALOG

目录

CATALOG

01 离散对数公钥体制

- 离散对数
- 公钥

离散对数问题

- Y = pow(g, a, N)
- 已知Y, g, N 求a

- 常见的群？
  - 乘法群

基于离散对数的困难问题

- DLP(Discrete logarithm problem)
  - parallel Pollard rho method(O(√r))
  - 应用：
    - Schnorr signatures
    - DSA signatures
- CDH(computational Diffie-Hellman problem)
  - 已知的最快算法是计算DLP
  - 应用：
    - Diffie-Hellman key exchange
    - Elgamal
    - BLS signatures

基于离散对数的困难问题

- DDH(decision Diffie-Hellman problem)
  - 已知的最快算法是解决DLP问题
  - 不过在一些 pairing groups 中这个问题是简单的
  - 应用：
    - Diffie-Hellman key exchange
    - Elgamal

DHKE

- N = 23，g = 5 (order 22)

- Alice:
  - a = 4, A = g^a mod N = 5^4 mod 23 = 4
- Bob:
  - b = 3, B = g^b mod N = 5^3 mod 23 = 10

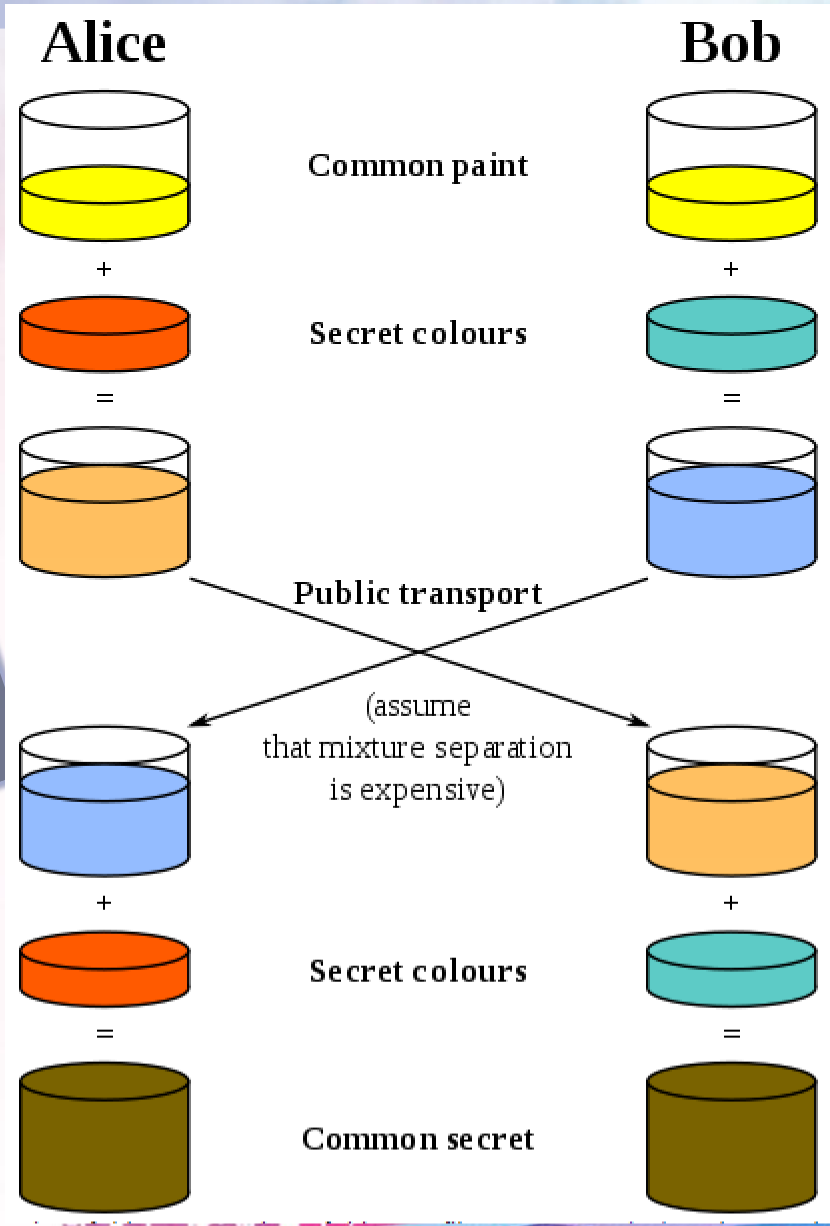
- Final Key:
- 10^4 mod 23 = 4^3 mod 23 = 18

关于order

- pow(3,2,5)，pow(4, 2, 5) v.s. pow(2, 4, 5)

- 4 = 2^2
- 3 = 2^3

DHKE



Alice
Bob
Common paint
+
Secret colours
=
Public transport
(assume
that mixture separation
is expensive)
+
Secret colours
=
Common secret

ElGamal Enc

- 私钥x([1, q-1])， 公钥G (order q)，g，h （h = g^x）
- 安全性假设：DDH，CDH (它们的区别？语义安全？)
- Alice:
  - 待加密信息m
  - 随机选择 r -> [1, q-1]
  - c1 = g^r
  - s = h^r = g^(x*r)
  - c2 = m*s
  - cipher text:
    - (c1, c2)

ElGamal DEC

- Bob :
  - $s = c1^x$
  - $m = c2 * s^{(-1)}$

ElGamal Signature

- 私钥x([1, q-1])， 公钥G (order q)，g，h （h = g^x）

- Alice：
  - 随机选择k [1, q-1], (gcd(k, q) = 1)
  - r = g^k
  - s = (H(m) - x*r)*k^(-1)

  - cipher text:
    - (r, s)

ElGamal Verification

- g^H(m) == h^r * r^s

- g^(x*g^k) * g^(k*(H(m) - x*r)*k^(-1))
- = g^(x*g^k + k*(H(m) - x*r)*k^(-1))
- = g^(x*g^k + H(m) - x*r)
- = g^(x*g^k + H(m) - x*g^k)
- = g^H(m)

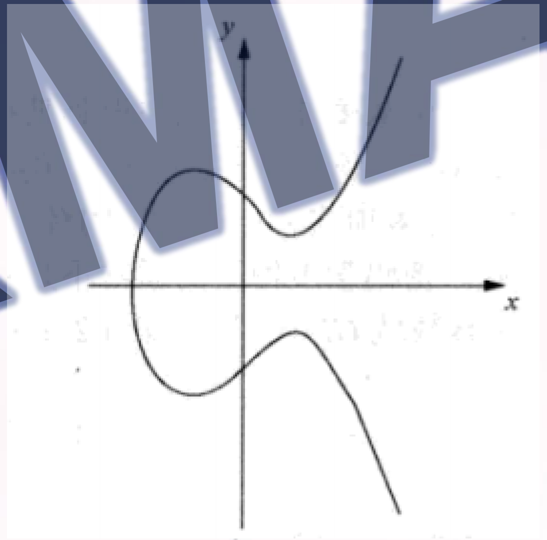
目 录

CATALOG

02 椭圆曲线

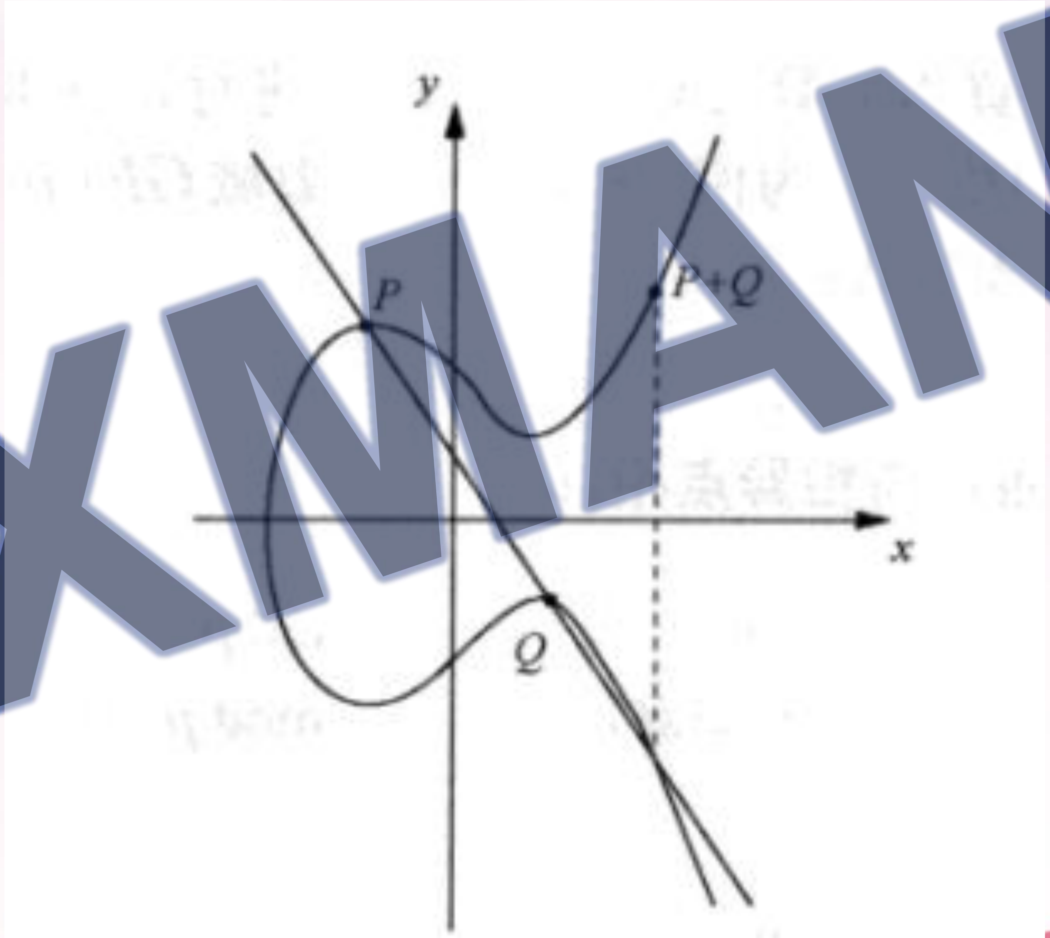
什么是椭圆曲线

- ECC
  - $y^2 = x^3 + a*x + b \bmod p$， a,b 属于 Z_p，$(4*a^3 + 27*b^2 \neq 0 \bmod p)$
  - 一个无穷远点

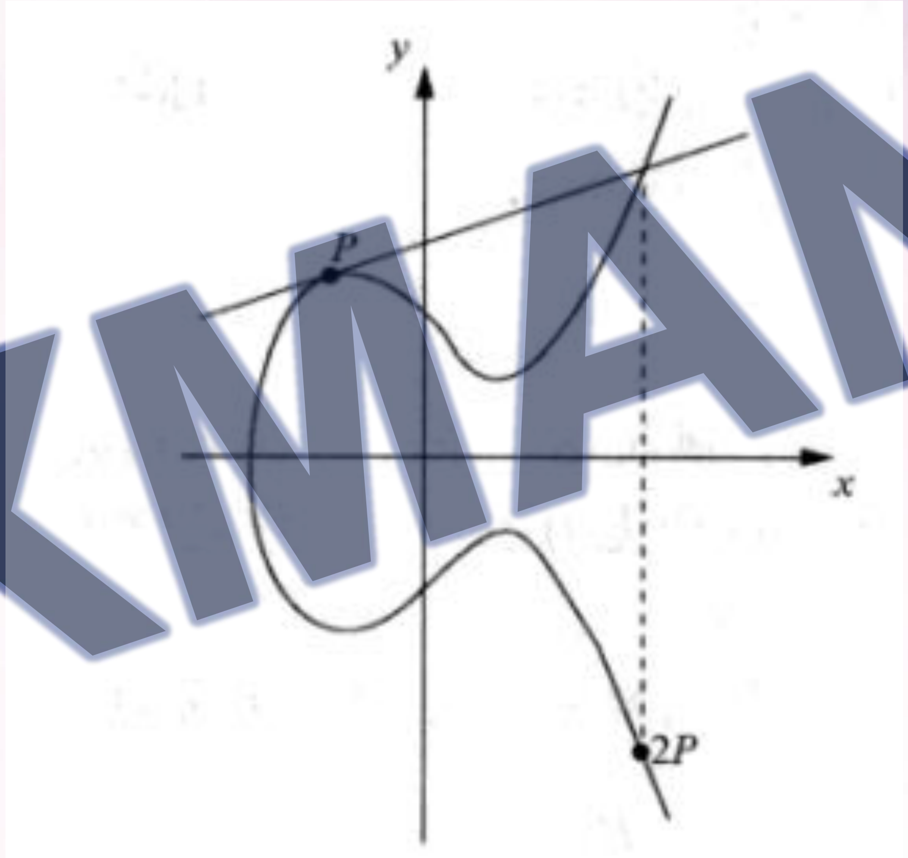


- $y^2 = x^3 - 3*x + 3$

椭圆曲线上的加法



椭圆曲线上的加法



椭圆曲线上加法的计算方式

- P(x1, y1), Q(x2, y2), R(x3, y3)
- P+Q=R

- x3 = s^2 - x1 - x2 mod p
- y3 = s*(x1 - x3) - y1 mod p
- s:斜率
  - (y2 - y1)/(x2 - x1)
  - (3*x1^2 + a)/(2*y1) mod p

test

- P(x1, y1), Q(x2, y2), R(x3, y3)
- P+Q=R
- x3 = s^2 - x1 - x2 mod p
- y3 = s*(x1 - x3) - y1 mod p
- s:斜率
  - (y2 - y1)/(x2 - x1)
  - (3*x1^2 + a)/(2*y1) mod p

  - y^2 = x^3 + 2*x + 2 mod 17
  - P = (5, 1)
  - 2P = ?

test

- 2P = (5, 1)+(5, 1) = (6, 3)
- 把2P 带入到y^2 = x^3 + 2*x + 2 mod 17中？

椭圆曲线上的群

- 闭合性
- 结合性
- 单位元
- 逆元

- 【某些条件下】椭圆曲线上的点可以构成一个循环群

椭圆曲线上的群

- y^2 = x^3 + 2*x + 2 mod 17
- P = (5, 1)
- 2P = (6, 3)
- 3P = (10, 6)
- 4P = (3, 1)
- 5P = (9, 16)
- 7P = (0, 6)
- 8P = (13, 7)
- 9P = (7,6)
- 10P = (7, 11)
- 11P = (13, 10)
- 12P = (0, 11)
- 13P = (16, 4)
- 14P = (9, 1)
- 15P = (3, 16)
- 16P = (10, 11)
- 17P = (6, 14)
- 18P = (5, 16)
- 19P = ?

曲线上有多少个点？

- [p + 1 - pow(p, 1/2), p +1 + pow(p, 1/2)]

为什么使用椭圆曲线

- ECC V.S. RSA = 160~256 V.S. 1024~3072

- 性能上的优势？
  - 来源于密钥长度

椭圆曲线上的离散对数问题

- Q = aP, P为曲线上一点，a为Zp内一元素
- 已知Q，P，曲线方程，求解a是困难的
- 为什么乘法的逆会是困难的?

怎么算aP？

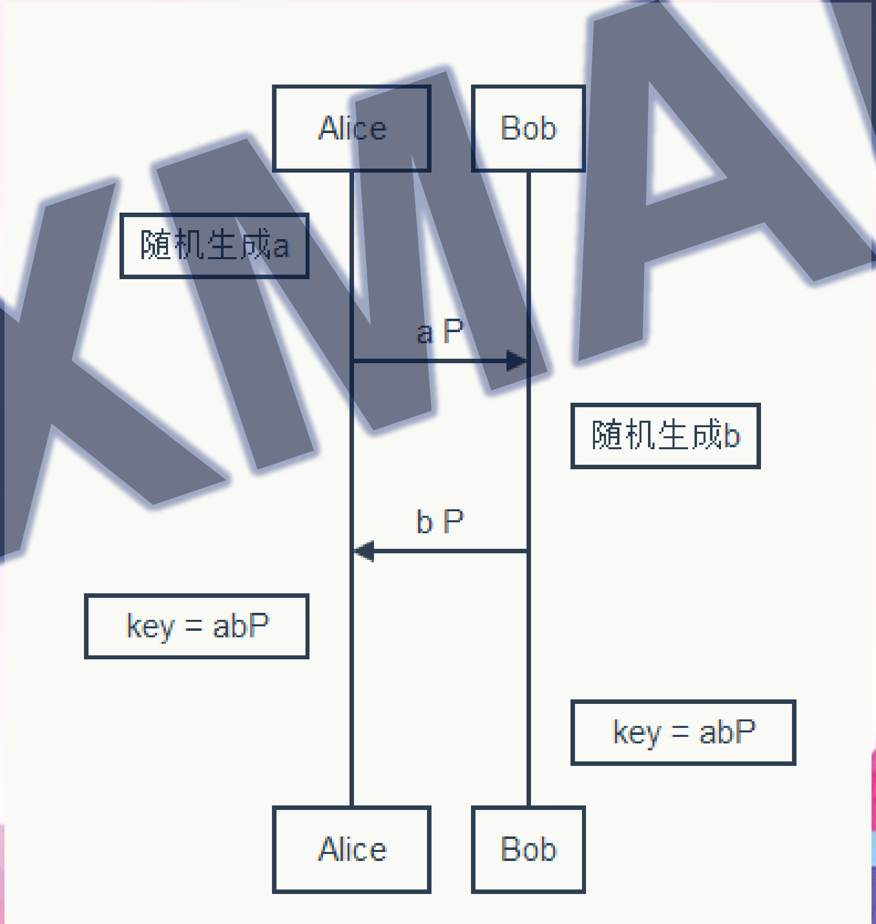
- RSA中提到的快速幂

- 26P
- 11010P

#0	$P = 1_2 P$	初始化设置，被处理的位为：d_4=1
#1a	$P + P = 2P = 10_2 P$	DOUBLE，被处理的位为：d_3
#1b	$2P + P = 3P = 10_2 P + 1_2 P = \mathbf{11_2 P}$	ADD，因为 d_3=1
#2a	$3P + 3P = 6P = 2(11_2 P) = \mathbf{110_2 P}$	DOUBLE，被处理的位为：d_2
#2b		没有 ADD，因为 d_2=0
#3a	$6P + 6P = 12P = 2(110_2 P) = \mathbf{1100_2 P}$	DOUBLE，被处理的位为：d_1
#3b	$12P + P = 13P = 1100_2 P + 1_2 P = \mathbf{1101_2 P}$	ADD，因为 d_1=1
#4a	$13P + 13P = 26P = 2(1101_2 P) = \mathbf{11010_2 P}$	DOUBLE，被处理的位为：d_0
#4b		没有 ADD，因为 d_0=0

椭圆曲线上的DHKE

- 首先，双方公开协商好用的曲线、模数，与基点P



不同代数结构中的困难问题

- 双线性对中：
  - e(g^a, h^b) == e(g, h)^(a*b)

- CDH ? DDH ?

目 录

CATALOG

03 数字签名 & MAC

RSA

- 签名：
  - s = pow(p, d, n)

- 验证
  - p = pow(c, e, n)

ElGamal Signature

- 私钥x([1, q-1])， 公钥G (order q)，g，h （h = g^x ）

- Alice：
  - 随机选择k [1, q-1], (gcd(k, q) = 1)
  - r = g^k
  - s = (H(m) - x*r)*k^(-1)

  - cipher text:
    - (r, s)

ElGamal Verification

- g^H(m) == h^r * r^s

- g^(x*g^k) * g^(k*(H(m) - x*r)*k^(-1))
- = g^(x*g^k + k*(H(m) - x*r)*k^(-1))
- = g^(x*g^k + H(m) - x*r)
- = g^(x*g^k + H(m) - x*g^k)
- = g^H(m)

DSA

- Hash (SHA-2 in DSS)
- (L, N) = (1,024, 160), (2,048, 224), (2,048, 256), (3,072, 256) [N<=len(Hash(xxx))]
- N bits 素数q
- L bits 素数p，(p-1)|q
- 选择一个g，g在模p运算中的order是q
- x in (0, q)
- y = pow(g, x, p)

- (p, q, g)

DSA

- random k in (1, q)
- r = pow(g, k, p) % q
- s = k^(-1)*(H(m) + x*r) mod q

- (r, s)

DSA

- u1 = h(m)*s^(-1) mod q
- u2 = r*s^(-1) mod q

- assert r == (g^u1*y^u2 mod p) mod q

ECDSA

- G 基点
- n G的order, prime

- Alice
  - secrect key da in [1, n-1]
  - public key Qa = da*G
  - random k in [1, n-1]
  - r = k*G [0]
  - s = k^(-1)*(H(m)+r*da) mod n

ECDSA

- u1 = h(m)*s^(-1)
- u2 = r*s^(-1)

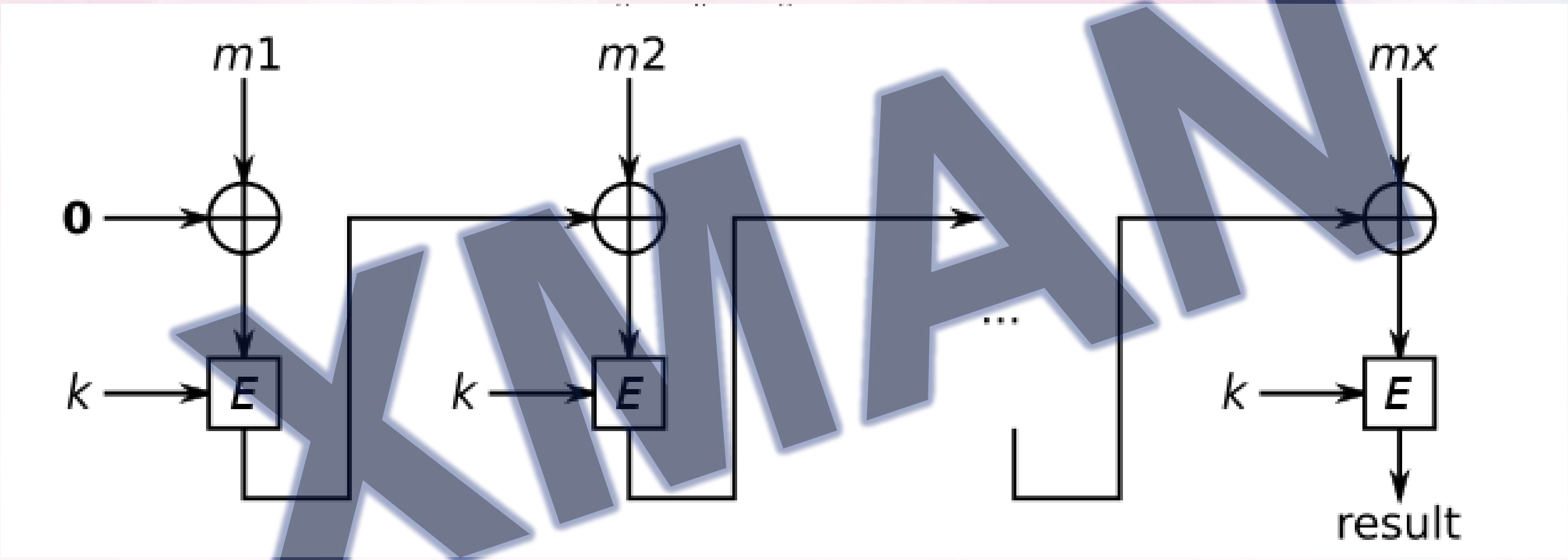
- (x1, y1) = u1*G + u2*Qa
- x1 = r

HMAC

- 不安全的HMAC V.S. 安全的HMACbu'an'qu

$$\mathrm{HMAC}(K,m) = H\Big((K' \oplus opad) \| H\big((K' \oplus ipad) \| m\big)\Big)$$

CBC-MAC



CBC-MAC

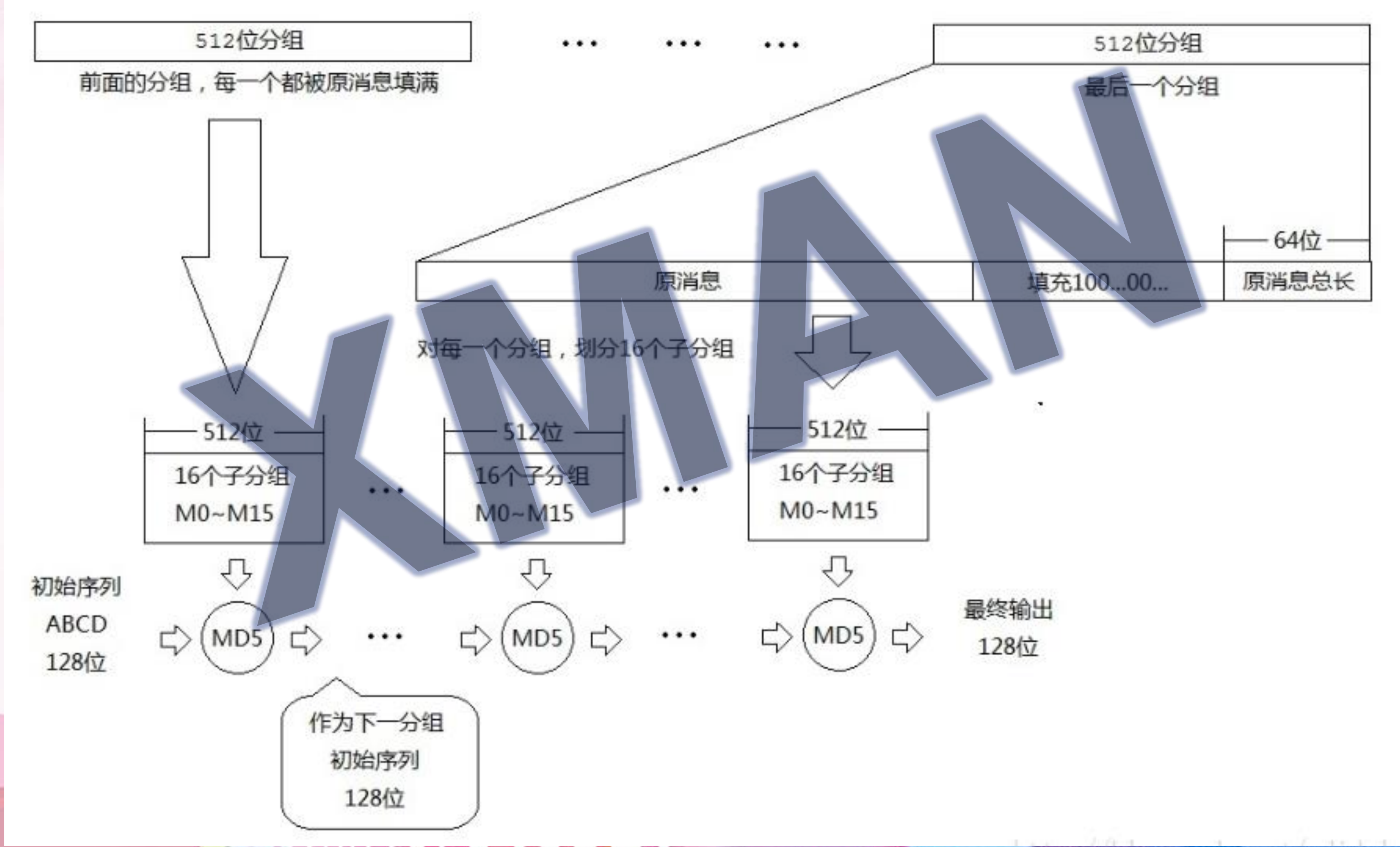
- Attack？
  - 加解密共用一个key
  - 允许选择iv

目 录

CATALOG

04 HASH

MD结构



512位分组
前面的分组，每一个都被原消息填满
512位分组
最后一个分组
64位
原消息
填充100...00...
原消息总长
对每一个分组，划分16个子分组
512位
16个子分组
M0~M15
512位
16个子分组
M0~M15
512位
16个子分组
M0~M15
初始序列
ABCD
128位
MD5
MD5
MD5
最终输出
128位
作为下一分组
初始序列
128位

Hash扩展攻击

- hashpump

SHA家族与MD结构

目 录

CATALOG

05 密钥分发

线性秘密切割方案

- n元一次方程

树形权限切割

- m个n元一次方程

目录

CATALOG

06 MPC

MPC

- OT
- Yao's protocol

谢　谢

汇报时间：2018年1月　　汇报人：1phan