# 一种新的基于椭圆曲线的数字签名方案

## A new digital signature plan based on elliptic curve

潘晓君　(桂林电子科技大学计算机与控制学院　桂林 541004)

**摘　要：** 首先阐述了有限域上的椭圆曲线数字签名方案。椭圆曲线密码体制中影响数字签名效率的最主要的因素之一是模逆操作,本文提出了一个新的基于椭圆曲线的数字签名方案,该方案无需进行模逆操作,大大提高了签名效率,同时也增强了安全性。

**关键词：** 数字签名　椭圆曲线　模逆　密钥

## 1　引言

椭圆曲线密码系统(ECC)于 1985 年由 Neal Koblitz[1]和 Victor Miller[2]发明,基于有限域椭圆曲线离散对数困难性的椭圆曲线密码算法,是迄今为止安全性最高的一种公钥加密算法。由于椭圆曲线密码体制相对其他公钥体制具有密钥短、占用带宽、存储空间少,单位密钥安全性高等优点,非常适合于现今计算资源有限的终端设备,越来越受到人们的关注,逐渐成为研究热点。

椭圆曲线密码体制是基于有限域上,椭圆曲线点集 E 所构成的群上定义的离散对数系统。对有限域上椭圆曲线的选择,应尽量避免使用超奇异曲线,以保证足够的安全性。椭圆曲线的运算为给定椭圆曲线 E 上的一个基点 G 和一个整数 k($1 \leqslant k \leqslant n-1$),求数乘 $kG = Q(\mathrm{mod}\, p)$,Q 也是上的一点,计算 $kG = G + G + \cdots + G$(k 个 G 相加)相对容易;但若给定椭圆曲线上两点 G 和 Q,求一整数 k,使 $kG = Q(\mathrm{mod}\, p)$,特别是当 G 是较高阶的基点时,则非常困难,这就是通常所说的椭圆曲线离散对数问题。基于椭圆曲线离散对数问题的难解性,形成了椭圆曲线密码体制。

椭圆曲线数字签名技术在现代网络数据通信处理过程中扮演着非常重要的角色,在网络资源非常有限的情况下,如何提高网络数据通信效率及安全,提高签名效率,就显得尤为重要。基于此种原因,本文提出了一种改进的椭圆曲线数字签名方案,无需进行模逆操作,不仅增强了网络数据通信安全性,还大大提高了签名效率,对模逆操作运算的研究可能是未来椭圆曲线密码体制提高效率的一个热点研究方向。

## 2　椭圆曲线数字签名方案

### 2.1　ECDSA(椭圆曲线数字签名算法)的主域参数

ECDSA 的主域参数主要由如下几个部分组成:$T = (q, FR, a, b, G, n, h)$。

(1) 选定一个域 q,$q = p$(素数域)或 $q = 2^m$(二元域,即以 2 为底,m 为素数的指数域);当为素数域 GF(p),且 $p > 3$ 时,其方程为 $y^2 = x^3 + ax + b$(其中 $a, b \in GF(p)$ 且须满足 $4a^3 + 27b^2 (\mathrm{mod}\, p) \neq 0$ 以确保椭圆的非超奇异性;对于二元域 $GF(2^m)$,其方程为 $y^2 + xy = x^3 + ax^2 + b$(其中 $a, b \in GF(2^m)$,且 $b \neq 0$)。椭圆曲线 E 定义为满足以上方程的点加上无穷远点 O 构成加法阿贝尔群。

(2) FR 域表示法来表示 F 中的元素;

(3) 两个 F 中的元素 a 和 b 来表示以上椭圆曲线方程的系数;

(4) 两个 F 中的元素 x 和 y 来表示 E(F)中的一个基点 G = (x,y)(用户 A 选定一条椭圆曲线 Ep(a,b),并取椭圆曲线上一点,作为基点 G,且要求基点满足 nG = O,O 为满足椭圆曲线方程的无穷远点。);

(5) 基点 G 的阶为 n,即 nG = O(其中 nG 表示为 n 个 G 相加,即 $nG = G + G + \cdots + G$ 相加,O 表示满足以上某种域方程的无穷远点),n 为素数,$n > 4q^{1/2}$ 且 $n > 2$;

(6) 椭圆曲线伴随因子 $h = \#E(Fq)/n$。

### 2.2　ECDSA 密钥对

椭圆曲线域参数与椭圆曲线数字签名算法的密钥对特定集相互关联。确定椭圆曲线的主域参数 T = (q,TR,a,b,G,n,h)后,便可确定密钥对。签名实体 A 必须在产生公钥之前确定主域参数的有效性。公钥是基点的随机倍数;而私钥则是用来生成这个倍数的整数。为了生成 ECDSA 密钥对,要求成员 A 通常操作的算法步骤如下:

(1) 在区间[1,n-1]中选择一个随机整数或伪随机整数 $d_A$(私钥);

(2) 计算 $Q_A = d_A G$;

(3) A 的公钥是 $Q_A$,私钥是 $d_A$。

### 2.3 ECDSA 签名过程

签名方 A 为了对消息 m 进行签名,将会利用域参数 T = (q,FR,a,b,G,n,h)及其相关密钥对($d_A$,$Q_A$)作如下操作:

(1) 选择一个随机或伪随机整数 k,使得 k 满足[1,n-1];

(2) 计算 kG = ($x_1$,$y_1$),且将 $x_1$ 转换成整数 x;

(3) 计算 r = $x_1$ mod n,如果 r=0 则回到第 a 步;

(4) 计算 $k^{-1}$ mod n;

(5) 计算 SHA-1(m)消息摘要值,并将该位串转换成整数 e;

(6) 计算 s = $k^{-1}$(e + $d_A$r) mod n,如果 s=0 则回到第 a 步;

(7) 发送(r,s)。

### 2.4 ECDSA 验证过程

验证方 B 接到 A 对消息 m 的签名后,要验证 A 在消息 m 上的签名为(r,s),需要取得 A 的相关公钥 $Q_A$ 和域参数 T = (q,FR,a,b,G,n,h)的可信副本,并且要验证 T 和 QA 的有效性,然后 B 做如下的操作:

(1) 验证 r 和 s 是区间[1,n-1]内的整数;

(2) 计算 SHA-1(m) 消息摘要值,并将该位串转换成整数 e;

(3) 计算 w = $s^{-1}$ mod n;

(4) 计算 u1 = ew mod n 和 $u_2$ = rw mod n;

(5) 计算 X = $u_1$G + $u_2$$Q_A$;

(6) 如果 X =O 则拒绝签名;否则转换 X 的 x 坐标 $x_1$ 为整数 $x_1$',并计算 v = $x_1$' mod n;

(7) 当且仅当 v=r 时 B 才接受签名。

## 3 改进的 ECDSA 方案

### 3.1 主域参数及密钥对

本方案的域参数及其相关密钥对的产生与前面介绍的 ECDSA 方案相同。

### 3.2 签名过程

签名方 A 为了对消息 m 进行签名,将会利用域参数 T = (q,FR,a,b,G,n,h)及其相关密钥对($d_A$, $Q_A$)作如下操作:

(1) 选择一个随机或伪随机整数 k ,使得 k 满足[1,n-1];

(2) 计算 kG = ($x_1$, $y_1$),且将 $x_1$ 转换成整数 x;

(3) 计算 r = $x_1$ mod n,如果 r=0 则回到第 a 步;

(4) 计算 SHA-1(m) 消息摘要值,并将该位串转换成整数 e;

(5) 计算 s = k - e$d_A$,如果 s=0 则回到第 a 步;

(6) 发送(s,r,m)给验证者。

### 3.3 验证过程

验证方 B 接到 A 对消息 m 的签名后,要验证 A 在消息 m 上的签名为(r,s),需要取得 A 的相关公钥 $Q_A$ 和域参数 T = (q,FR,a,b,G,n,h)的可信副本,并且验证 T 和 $Q_A$ 的有效性,然后 B 做以下的操作,具体的验证步骤如下:

(1) 验证 r 和 s 是区间[1,n-1]内的整数;

(2) 计算 SHA-1(m,r),并将该位串转换成整数 e;

(3) 计算 r' = sG + e$Q_A$;

(4) 当且仅当 r' =r 时 B 才接受签名。

## 4 不需模逆操作证明

整个计算过程不需要进行模逆操作,签名与验证过程如下:

签名方程:

$s = k - ed_A$　(1)

验证方程:

$r = sG + eQ_A$　(2)

验证证明:

$sG + eQ_A = (k - ed_A)G + eQ_A = kG = r$

本改进方案将需要验证的消息 m 与 r 一起进行 hash 摘要,这样就使得签名与验证过程分别少了一步

乘法，加快了签名的运算速度。下面的试验结果表明了本方案比改进前的方案签名速度有较大提高，证明了该方案的可行性与有效性。

## 5 性能分析评估

参照文献[3]，G 是椭圆曲线 E(Fq) 上的一个基点，E 是定义在域 Fq 上的椭圆曲线，且要求 $q \approx 2^{160}$。给定 mG，m 是一个随机的 160 位整数。因此，时间复杂度换算关系可按如下关系式运算：

$T_{EC-MUL} \approx 29Tmul$　(1)

$T_{EC-ADD} \approx 0.12Tmul$　(2)

$T_{INV} \approx 0.4T_{EC-MUL} \approx 11.6\ Tmul$　(3)

为了便于比较所提出的签名方案的计算时间效率，不妨设 Tmul 为在模意义下 2 个整数相乘的计算时间，Tinv 为在模意义下计算逆元素所需时间，$T_{EC-MUL}$ 为椭圆曲线模意义下数乘的计算时间，$T_{EC-ADD}$ 为椭圆曲线模意义下模加的计算时间比较而言，椭圆曲线上点加运算和在模意义下的加运算的计算时间均远远小于相应的乘运算，所以可以忽略不计。

在椭圆曲线的加密或者签名过程中，求逆是比较耗时的一种运算，如在 2.3 小节 ECDSA 签名过程 d) 计算 $k^{-1} \bmod n$ 和 2.3 小节验证过程 c) 计算 $w = s^{-1} \bmod n$ 中，对于大整数 k 这个运算是比较慢的。使用扩展欧几里德算法平均也需完成 $0.843\log_2 n + 1.47$ 次除法。如表 1 所示：改进的方案没有模逆操作，省略了一些操作步骤。由表 2 可知，该方案相对 ECDSA 方案，安全性、时间效率有了较大提高。

表 1　本文方案与 ECDSA 方案比较

| 方案 / 签名/验证过程 | ECDSA 方案 | 本文方案 |
|---|---|---|
| 产生签名 | 需要进行模逆操作计算 $k^{-1} \bmod n$<br>$kG = (x_1, y_1)$<br>$r = x_1 \bmod n$<br>$s = k^{-1}(e + d_A r) \bmod n$ | 不需要进行模逆操作计算<br>$kG = (x_1, y_1)$<br>$r = x_1 \bmod n$<br>$s = k - ed_A$ |
| 验证签名 | 需要进行模逆操作计算<br>$w = s^{-1} \bmod n$;<br>$u_1 = ew \bmod n$ ;<br>$u_2 = rw \bmod n$;<br>$X = u_1 G + u_2 Q_A$; | 不需要进行模逆操作计算<br>$X = sG + eQ_A$ |

表 2　时间复杂度比较

| 时间复杂度 / 签名方案 | 生成签名时间复杂度 | 生成签名时间复杂度粗略估计 | 验证签名时间复杂度 | 验证签名时间复杂度粗略估计 |
|---|---|---|---|---|
| ECDSA 方案 | $T_{mul} + T_{EC-MUL} + T_{INV} + T_{EC-ADD}$ | $41.6\ T_{mul}$ | $2T_{mul} + 2T_{EC-MUL} + T_{INV} + T_{EC-ADD}$ | $71.22\ T_{mul}$ |
| 本文方案 | $2T_{mul} + T_{EC-MUL} + T_{EC-ADD}$ | $31\ T_{mul}$ | $T_{mul} + 2T_{EC-MUL} + T_{EC-ADD}$ | $59.12\ T_{mul}$ |

## 6 结束语

本文在深入分析了基本的椭圆曲线数字签名方案的基础上，针对模逆操作是影响签名效率比较重要的因素之一，提出了一个新的基于椭圆曲线的数字签名方案，该方案无需模逆操作，通过试验分析得出，该方案不仅在签名效率上与 ECDSA 方案相比的确有所提高，并且安全性也有所增强。

### 参考文献


1 N Koblitz. Elliptic curve cryptosystems[J]. Mathematics of Computation, 1987, (48): 203 – 209.

2 V Miller. Uses of elliptic curves in cryptography [A]. Advances in Cryptology – Crypto'85, Lecture Notes in Computer Science [C], 1986, 218: 387 – 398.

3 Jurisic A, Menezes A. Elliptic Curves and Cryptography [EB/OI ]. [2003 04 02]. http://www. certicom. corn/whitepapers.

4 D Pointcheval, J Stern. Security proofs for signature schemes [A]. Advances in Cryptology – Eurocrypt '96, Lecture Notes in Computer Science [C], 1996, 1070: 417 – 426.