# 移动计算中基于椭圆曲线的匿名签名算法

熊 焰,王冬华,苗付友,杨寿保
(中国科学技术大学计算机系,安徽合肥 230026)

**摘 要**: 移动代码(例如移动代理)在异地执行签名时往往不希望暴露其所有者的私有密钥,本文提出了一种基于椭圆曲线的移动代码匿名签名算法,依据该算法,移动代码所有者可以利用椭圆曲线根据自己的身份信息为移动代码生成一个认证矢量和一个临时性密钥对,并通过它们实现了移动代码匿名签名以及签名后的不可否认性. 该算法除具有匿名性和不可否认性以外,还具有高效性、保密性和不可伪造性等特点,可广泛应用于各种具有代码移动特性的移动计算.

**关键词**: 移动计算; 椭圆曲线; 匿名签名
**中图分类号**: TP393 **文献标识码**: A **文章编号**: 0372-2112 (2003) 11-1651-04

# An Algorithm on Mobile Code Anonymity Signature Based on Elliptic Curve in Mobile Computation

XIONG Yan, WANG Dong-hua, MIAO Fu-you, YANG Shou-bao
*(Department of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui 230026, China)*

Abstract: It is not desired that private key of mobile code owner will be revealed when it signs in other site. An algorithm on mobile code anonymity signature based on elliptic curve is given. Based on the algorithm, mobile code owner can use elliptic curve to create an authentication vector and a temporary key pair for mobile code according to its identification information, and implement anonymity signature and unrepudiation through them. In addition, the algorithm has features of high efficiency, secrecy and unforgery, and is able to be widely used to mobile computation.

Key words: mobile computation; elliptic curve; anonymity signature

## 1 引言

移动计算从移动主体划分,大致可以分为代码移动和结点移动. 前者指代码可以透明地从网络中的一个结点移动到另一个结点,并在新的结点上执行,典型的有移动代理[1]以及主动网络[2]等. 后者则指移动结点可以方便地从网络的一点移动到另一点,移动过程中仍然能够维持现有的网络连接,典型的如移动 IP 网络等. 在移动计算中,当移动代码从源结点移动到目的结点上执行时,目的结点往往需要对移动代码身份的合法性进行确认,即需要移动代码提供相应的签名信息. 另一方面,移动代码提供签名往往意味着移动代码需要携带并利用其所有者的私有密钥,从而有可能泄露该私有密钥,造成严重的安全问题.

Sander 和 Tschudin 提出了使用函数复合的不可拆分签名方案[3,4]. 假设用户的签名函数为 $s$,则使用 $(f(\cdot), sof(\cdot))$ 作为移动代理的签名函数,从而隐藏了用户的私有密钥. 在文[5]中 Panayiotis Kotzanikolaou, Mike Burmester 以及 Vassilios Chrissikopoulos 首先通过 RSA 实现了这种方案. 在此方案中 $f(\cdot) = h^{(\cdot)} \bmod n$,其中 $h = \mathrm{Hash}(IDC, ReqC)$, $sof = k^{(\cdot)} \bmod n$,其中 $k = h^d \bmod n$(其中 $d$ 为 RSA 中的私钥). 但是文[5]中 RSA 实现的不可拆分签名方案破坏了用户的匿名性,而且 RSA 的实现效率不高.

因此针对于代码移动的情况,本文提出了一种基于椭圆曲线的匿名签名算法,依据该算法,移动代码所有者可以利用椭圆曲线根据自身的认证矢量为移动代码生成一个认证矢量和一个临时性密钥对,通过该密钥对实现移动代码签名而保护移动代码所有者的匿名性,同时通过认证矢量保证签名后移动代码所有者的不可否认性.

椭圆曲线[6] $E(a, b): Y^2 = X^3 + aX + b$, $delt = 4a^3 + 27b^2 \neq 0$. $a, b \in GF(p)$,其中 $GF(p)$ 为有限集, $p$ 为一个足够大的素数;令 $S$ 为点集$\{(x, y): (x, y) \in E(a, b)$或点 $O\}$(点 $O$ 为椭圆曲线的无穷远点也即沿 $y$ 坐标轴趋向无穷远). 定义 $S$ 上的'+'运算为:设 $P = (x_1, y_1)$, $Q = (x_2, y_2)$ 皆为 $S$ 中的点, $L$ 为 $PQ$ 的连线. 若 $P$ 和 $Q$ 重合为一点,即 $P = Q$,则 $L$ 便退化为过 $P$ 点的切线. 设 $L$ 和椭圆曲线相交与另一点 $R$, $R$ 点关于 $x$ 轴的对称点即为 $P + Q$.

收稿日期:2002-11-28;修回日期:2003-04-16
基金项目:安徽省自然科学基金(No. 00043105);国家 863 计划(No. 2002AA121066)

$\langle S, '+' \rangle$是一 Abel 群(交换群),简记$\underbrace{P+P+\cdots+P}_{m\text{个}}=mP$.

$C_x(A)$表示取点 $A$ 的 $x$ 坐标. Hash 是一个单向哈希函数: $\mathrm{Hash}:(0,1)^{*}\rightarrow(0,1)^{k}, (k=256)$; $(\cdot\|\cdot)$用来表示两个串的连接.

选择 $x\in GF(p)$为私有密钥, $G\in E(a,b)$, $y=xG$,当已知 $y$ 和 $G$,求解 $x$ 是公认的椭圆曲线离散对数问题,很难在有限时间内求解. 因此 $xG$ 可以作为相应的公开密钥.

## 2 算法模型

本文的算法模型包含四个组成部分:移动代码(MC ——Mobile Code)、移动代码所有者(MCO ——Mobile Code Owner)、被访问结点(VN ——Visited Node)和认证中心(AC ——Authentication Centre). 移动代码 MC 是指在系统运行过程中从一个结点迁移到一个新的结点,并在其上运行的可执行代码及其相关的数据;移动代码所访问的结点就是 VN;为了实现特定的功能,产生、发布移动代码并通过移动代码访问 VN 的系统称为 MCO;认证中心 AC 为 MCO 和 MC 提供注册和认证服务.

对于移动代码,无论是移动代理还是主动网络中的主动包,都可以依据特定信息从一个结点迁移到另一个结点,并在这些结点上运行. 每个 MCO 在使用 MC 执行特定的任务之前,必须向 AC 注册,并获得 AC 的确认. 之后,当一个 MCO 需要完成特定任务时(如通过主动网络在多个被访问结点上部署特定的功能,在基于移动代理的电子商务系统中,通过移动代理在某些商务网站上代表其所有者实现电子交易),就产生一个或几个移动代码 MC,并将 MC 发布到特定的 VN 上. 在 MC 到达 VN,开始访问 VN 之前,VN 必须验证其身份的合法性,在 MC 的身份得到确认后,MC 就可以对消息进行签名,在 VN 上运行并访问其资源了. 在 MC 的运行过程中,VN 如果发现 MC 企图对其实施攻击,则将 MC 在身份验证时提供的信息提交给认证中心 AC. AC 依据该 MC 的 MCO 在 AC 上注册的信息揭示 MCO 的身份,而使 MCO 不能否认其行为. 但是 VN 并不能根据 MC 提交的身份认证信息确定 MCO 的身份,从而保证了 MCO 的匿名性.

## 3 算法设计

本算法具体包含如下几个部分:AC 的建立、MCO 注册、MC 的生成、MC 访问 VN 和必要时通过 AC 揭示 MCO.

MCO 注册的过程是指 MCO 通过椭圆曲线生成一对密钥($x_{MCO}$, $P_{MCO}$),并将其身份信息提交给 AC,AC 对这些信息进行认证的过程;MC 的生成是指在完成某一任务之前,MCO 生成特定的 MC,并根据其自身的密钥对($x_{MCO}$, $P_{MCO}$)为 MC 生成一个认证矢量和一个临时性密钥对($x_{MC}$, $P_{MC}$)的过程;MC 在访问 VN 之前,必须将其签名的消息和认证矢量交由 VN 验证,只有在验证通过后,VN 才允许 MC 的访问. 在 MC 访问过程中,VN 如果发现受到 MC 的恶意攻击,则将 MC 的认证矢量提交 AC,AC 通过查找注册信息,可以确认并揭示 MC 的所有者 MCO 的身份.

### 3.1 认证中心 AC 的建立

认证中心是系统中各个部分都信赖的一个结点. 它为各个 MCO 提供注册和认证服务. 当 AC 建立时,它在有限域 $GF(p)$内随机选择一个值 $x_{AC}$,即 $x_{AC}\in GF(p)$, $p$ 为一个 256 比特的素数,将 $x_{AC}$作为 AC 的私有密钥;同时在椭圆曲线 $E(a,b)$上选择一点 $G$,其中 $a,b\in GF(p)$,令 $P_{AC}=x_{AC}G$,将 $P_A$作为 AC 的公开密钥. 从而 AC 就产生了一个密钥对($x_{AC}$, $P_{AC}$). 然后 AC 将矢量($p$, $G$, $a$, $b$, $P_{AC}$, Hash)向网络中的所有结点公开.

### 3.2 MCO 向 AC 注册

每一个需要通过移动代码实现匿名签名功能的 MCO 必须首先向 AC 注册,将它的标识信息提交给 AC,以保证在必要时 AC 可以根据这些标识信息确定一个移动代码是否是某个 MCO 产生的,从而实现 MCO 的不可否认性.

MCO 首先获取 AC 的公开矢量($p$, $G$, $a$, $b$, $P_{AC}$, Hash),然后开始如下的注册过程:

(1) MCO 随机选择 $x_{MCO}\in GF(p)$作为私有密钥,计算公开密钥

$$P_{MCO}=x_{MCO}G \tag{1}$$

将 $P_{MCO}$ 和 $ID_{MCO}$ 嵌入椭圆曲线 $E(a,b)$ 的一点 $PT(P_{MCO},ID_{MCO})$中,选择另一个随机数 $r\in GF(p)$,将数对($rG$, $PT_{(P_{MCO},ID_{MCO})}+rP_{AC}$)发送给 AC(此过程即对 $PT_{(P_{MCO},ID_{MCO})}$加密);

(2) 当 AC 收到数对后,计算

$$PT_{(P_{MCO},ID_{MCO})}=PT_{(P_{MCO},ID_{MCO})}+rP_{AC}-x_{AC}rG \tag{2}$$

(因为 $P_{AC}=x_{AC}G$,所以 $rP_{AC}=x_{AC}rG$,故上式成立),从而获得了明文 $PT_{(P_{MCO},ID_{MCO})}$,进而可以得到 MCO 的公开密钥 $P_{MCO}$ 和标识 $ID_{MCO}$;

(3) 针对特定的 MCO 的公开密钥 $P_{MCO}$和标识 $ID_{MCO}$,AC 随机选择一个数 $d_{mco}\in GF(p)$,计算

$$A_{mco}=d_{mco}G+P_{MCO} \tag{3}$$

$$b_{mco}=d_{mco}(x_{AC}+1) \bmod p \tag{4}$$

记录矢量($ID_{MCO}$, $P_{MCO}$, $A_{mco}$, $d_{mco}$, $b_{mco}$);

(4) AC 随机地选择一个数 $n\in GF(p)$,计算

$$N=nG,\ (n\text{必须保证}N\text{的横坐标值}C_x(N)\neq 0 \bmod p) \tag{5}$$

$$s=(C_x(A_{mco})\|C_y(A_{mco})\|b_{mco})\,n \bmod p+C_x(N)\,x_{AC} \bmod p \tag{6}$$

这样$\{N,s\}$就构成了 AC 对数对($A_{mco}$, $b_{mco}$)的签名,AC 将此签名$\{N,s\}$和加密后的数对($A_{mco}$, $b_{mco}$)发送给 MCO,数对($A_{mco}$, $b_{mco}$)的加密、解密过程分别类似于(1)中 MCO 对($P_{MAC}$, $ID_{MAC}$)的加密以及(2)中 AC 相应的解密过程;

(5) 当 MCO 收到 AC 的签名以后,解密数对($A_{mco}$, $b_{mco}$),验证

$$sG=(C_x(A_{mco})\|C_y(A_{mco})\|b_{mco})\,N+C_x(N)\,P_{AC} \tag{7}$$

是否成立,如果成立,则 MCO 可以确信数对($A_{mco}$, $b_{mco}$)是 AC 对它的公开密钥 $P_{MCO}$的签名,因而记录下数对($A_{mco}$, $b_{mco}$)并将其作为 AC 对它的认证矢量.

经过上述过程,MCO 生成了一对密钥($x_{MCO}$, $P_{MCO}$)并获得了 AC 对它的认证矢量($A_{mco}$, $b_{mco}$),因为是加密后由 AC 返回

给 MCO 的,所以这一认证矢量只有 MCO 和 AC 知道. 必要时,AC 可以通过验证等式

$$b_{mco}\ G = d_{mco}\ P_{AC} + A_{mco} - P_{MCO} \tag{8}$$

是否成立判断该 MCO 是否在本 AC 上注册过(通过式(2)和式(1)可得证该式),不难发现只有 AC 才可以验证式(8),因为只有 AC 知道 $d_{mco}$.

值得注意的是,MCO 的注册信息($P_{MAC}$, $ID_{MAC}$)必须加密后才可以传送给 AC,过程如(1)和(2)所示,而且 AC 在向 MCO 返回认证矢量前也必须对认证矢量签名. 这样可以防止恶意结点在 MCO 和 AC 之间发动中间人攻击.

### 3.3　MC 的生成

当 MCO 想要完成某个任务时,即生成一个 MC,MC 除包含有可执行代码、参数等普通信息外,还包含如下安全信息.

MCO 为 MC 选择一个随机数 $x_{MC}$ 作为 MC 的私有密钥: $x_{MC} \in GF(p)$,计算公开密钥

$$P_{MC} = x_{MC}\ G \tag{9}$$

($x_{MC}$, $P_{MC}$)就构成了 MC 的一个密钥对. 同时令

$$z = \mathrm{Hash}(m)\ x_{MC} \bmod p \tag{10}$$

$m$ 为移动代码信息,这样就生成了 MC 的临时密钥对($x_{MC}$, $P_{MC}$)和认证矢量($P_{MC}$, $P_{MCO}$, $A_{mco}$, $b_{mco}$, $z$, $t$), $t$ 为时间戳.

### 3.4　移动代码 MC 访问目标结点 VN

在移动代码 MC 访问目标结点 VN 之前,VN 必须能够确定 MC 的所有者是否在认证中心注册过,以及 MC 的完整性(保证 MC 没有被恶意修改过). 确认后,MC 即可进一步访问 VN. 具体过程如下:

(1)移动代码 MC 从 MCO 迁移到目标结点 VN;

(2)MC 将$\{P_{MC}, z, E(P_{MC}, P_{MCO}, A_{mco}, b_{mco}, t)\}$发送给某 VN 进程. $E(P_{MC}, P_{MCO}, A_{mco}, b_{mco}, t)$表示用 AC 的公开密钥加密后的矢量($P_{MC}$, $P_{MCO}$, $A_{mco}$, $b_{mco}$, $t$),其中 $t$ 表示时间戳. 具体地,MC 将矢量($P_{MC}$, $P_{MCO}$, $A_{mco}$, $b_{mco}$, $t$)的明文嵌入椭圆曲线 $E(a, b)$ 上的一点 $Pt_{(P_{MC}, P_{MCO}, A_{mco}, b_{mco}, t)}$,

$$E(P_{MC}, P_{MCO}, A_{mco}, b_{mco}, t) = (rG, Pt_{(P_{MC}, P_{MCO}, A_{mco}, b_{mco}, t)} + rP_{AC}) \tag{11}$$

$r$ 为 MCO 随机选择的一个数且 $r \in GF(p)$;

(3)VN 进程验证等式

$$zG = \mathrm{Hash}(m)\ P_{MC} \tag{12}$$

是否成立;

(4)若(3)中的等式成立,则 VN 进程将 $E(P_{MC}, P_{MCO}, A_{mco}, b_{mco}, t)$ 发送给 AC;

(5)AC 收到后,执行计算以解密 $E(P_{MC}, P_{MCO}, A_{mco}, b_{mco}, t)$:

$$Pt_{(P_{MC}, P_{MCO}, A_{mco}, b_{mco}, t)} = Pt_{(P_{MC}, P_{MCO}, A_{mco}, b_{mco}, t)} + ax_{AC}G - x_{AC}(aG) \tag{13}$$

从而获得矢量($P_{MC}$, $P_{MCO}$, $A_{mco}$, $b_{mco}$, $t$)的明文,记录该矢量;然后检验时间戳并验证等式

$$b_{mco}\ G = d_{mco}\ P_{AC} + A_{mco} - P_{MCO} \tag{14}$$

是否成立,若成立,则说明该 MCO 在本 AC 上注册过,反之则否(原因同式(8)的解释);

(6)AC 将验证结果签名后返回给 VN 进程;

(7)如果验证通过,VN 则允许 MC 访问;

(8)在访问过程中,如果 MC 需要向 VN 进程传递消息 $m$,MC 必须用其私有密钥 $x_{MC}$ 对消息 $m$ 签名后才可以发送给 VN 进程,这样就保证了 MC 不能否认它发送过的消息. MC 的签名过程如下,随机选取 $k \in GF(p)$,

$$K = kG \tag{15}$$

$$r = mk + C_x(K)\ x_{MC} \bmod p \tag{16}$$

则对消息 $m$ 的签名 $\mathrm{Sig}(m) = (K, r)$. VN 进程通过验证:

$$rG = mK + C_x(K)\ P_{MC} \tag{17}$$

可以确认该签名的真伪.

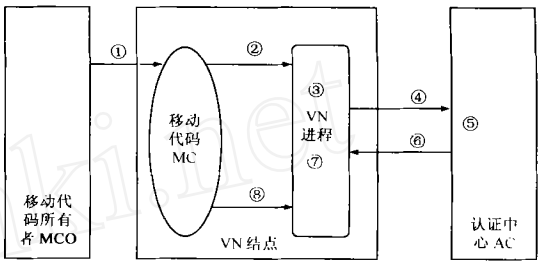


图 1　移动代码 MC 访问目的结点 VN 时的消息交互过程(参见描述部分)

在(3)中,如果式(12) $zG = \mathrm{Hash}(m)\ P_{MC}$ 成立,则式(10) $z = \mathrm{Hash}(m)\ x_{MC} \bmod p$ 成立. 式(12)的成立意味着这个 MC 在迁移过程中没有被篡改过,这由 $\mathrm{Hash}(m)$ 可以保证;式(12)结合(4)到(6)步保证了该移动代码 MC 由其真正的 MCO 产生而不是伪造的,因为只有该 MC 的 MCO 才能提供 MCO 的信息($P_{MCO}$, $A_{mco}$, $b_{mco}$, $t$).

(4)到(6)中,VN 通过 AC 确定来访 MC 的 MCO 是否在 AC 上注册过,从而可以保证 VN 不受 MC 的恶意攻击;但是 VN 却无法获知 MCO 本身的信息($P_{MCO}$, $A_{mco}$, $b_{mco}$, $t$),因为这些信息是加密的,这保证了 MCO 不致于因信息外泄而遭到攻击. 同时通过在信息中加入时间戳 $t$ 不仅可以较好地防止重放攻击,而且可以保证即使是同一个 MCO 发出的移动代码,每次提供的加密 MCO 信息也是不同的,这样就保证了 VN 不能通过比较不同移动代码提供的加密 MCO 信息判断它们是否由同一个 MCO 产生,即使 VN 获得了 $x_{MC}$,这进一步保证了 MCO 的匿名性. 为了保证 VN 和 AC 之间消息传输的安全性,在(4)中采取了加密措施保证只有 AC 才可以看到该消息,在(6)利用采取签名确保验证的结果确实来自 AC,进而保证了验证结果的真实性,要成功地解密和伪造签名,都必须解椭圆曲线离散对数难题.

(8)中,通过签名机制可以保证 MC 对其行为的不可否认性,而 MC 是由其 MCO 生成并且代表其 MCO(由上述分析可知). 因而在 MC 试图对 VN 进行攻击时,VN 可以向 AC 提交 MCO 的信息 $E(P_{MC}, P_{MCO}, A_{mco}, b_{mco}, t)$,通过 AC 揭示 MCO 的身份,而使 MCO 不能否认. 另一方面,同一个 MCO 生成的每一个移动代码都具有不同的密钥对,因而 VN 不可能通过比较 MCO 的密钥确定不同的移动代码是否来自同一个 MCO,这

也保证的 MCO 的匿名性.

假如 VN 试图通过伪造 MC 的签名 Sig′(*m*) = (*K*′, *r*′)攻击 MC,(设 MC 的真实签名为 Sig(*m*) = (*K*, *r*)),VN 可以令 *K*′ = *K*,由式(15)和式(16)可知,VN 必须根据已知的 *K* 和 *G* 计算出 *k*,进而根据 *k* 和 $x_{MC}$计算出 *r*′,才能使 *r*′= *r* 成功伪造出 MC 的签名. 而要根据 *K* = *kG* 和 $P_{MC} = x_{MC}\ G$ 计算出 *k* 和 $x_{MC}$,就必须解椭圆曲线离散对数难题.

通过上述分析,可以看出该算法能够比较好地保证 MCO 的匿名性,同时也保证了 MCO 的不可否认性.

## 4 算法特点

由以上算法描述可以看出,本算法具有以下特点:

(1)匿名性,移动代码所有者 MCO 发出的移动代码 MC 在访问目标结点 VN 时,VN 只能验证 MC 的合法性(MC 是否被篡改以及其 MCO 是否在 AC 上注册),但是无法知道 MCO 的身份信息;

(2)保密性,MC 在访问 VN 时,通过其临时性密钥进行签名;但不会泄露其 MCO 的密钥信息;

(3)不可否认性,当 VN 发现 MC 企图发动攻击,VN 可以将先前 MC 提供的信息{ $P_{MC}$, *z*, *E*( $P_{MC}$, $P_{MCO}$, $A_{mco}$, $b_{mco}$, *t*)}提交给认证中心 AC,由 AC 揭示生成该 MC 的 MCO 的身份;

(4)不可伪造性,对于 VN 而言,如果想成功地伪造 MC 的签名,它必须求解椭圆曲线离散对数问题,而这一问题目前尚无有效的解法;

(5)高效性,利用椭圆曲线实现匿名签名相对于其他类似方法,如 RSA 而言,由于其计算量较小,所以效率较高.

## 5 结论

为了解决具有代码移动特性的移动计算中签名的匿名性问题,本文提出了一种基于椭圆曲线的匿名签名算法. 依据该算法,移动代码所有者可以利用椭圆曲线根据自己的身份信息为移动代码生成一个认证矢量和一个临时性密钥对,并通过它们可以实现移动代码匿名签名以及签名后的不可否认性. 该算法除具有匿名性和不可否认性以外,还具有高效性,保密性,不可伪造性等特点,可广泛应用于各种具有代码移动特性的移动计算.

**参考文献**:


[1] Aglets workbench, IBM Japan Research Group[Z]. http://aglets.trl.ibm.co.jp/.

[2] Washington University Ants 2.0.2[Z]. http://www.cs.washington.edu/research/networking/ants/.

[3] Tomas Sander, Christian F. Tschudin. Protecting mobile agents against malicious hosts[A]. Mobile agent security[C]. LNCS, 1419, Springer-Verlag, 1998. 44 - 60.

[4] T Sander, C Tschudin. Towards mobile cryptography[J]. International Computer Security Institute(ICSI), TR-97-049, 1997.

[5] Panayiotis Kotzanikolaou. Mike Burmester and Vassilios Chrissikopoulos, secure transactions with mobile agents in hostile environments[A]. Information Security and Privacy[C]. Proceedings of the 5th Australasian Conference, ACISP 2000. LNCS 1841, Springer-Verlag, 2000. 289 - 297.

[6] 卢开澄. 计算机密码学 ——计算机网络中的数据保密与安全(第 2 版)[M]. 北京:清华大学出版社, 1998. 101 - 114.


**作者简介**:


**熊 焰** 男,1960 年 8 月生于安徽合肥,博士、副教授,研究方向为分布式处理、移动计算、移动通信、计算机网络及信息安全.



**王冬华** 男,1979 年 12 月生于江苏东台,硕士研究生,研究方向为计算机网络通信.



**苗付友** 男,1973 年 5 月生于河南驻马店,硕士、讲师,研究方向为计算机网络以及移动计算.