构造从字符串到 C_{34} 曲线的散列函数

于 伟[1),2)] 王鲲鹏[2)] 李 宝[2)] 田 松[2)]

[1)](中国科学技术大学电子工程与信息科学系 合肥 230027)

[2)](中国科学院信息工程研究所 北京 100093)

摘 要 该文利用求立方根的方法构造了一个从有限域 $\mathbb{F}_q$ 映射到 C_{34} 曲线上的确定函数,其时间复杂性是 $\mathcal{O}(\log^3 q)$. 利用这个确定的函数构造了从字符串到 C_{34} 曲线上的散列函数. 在相同的素域上,与 2009 年 Icart T 构造的到椭圆曲线上的散列函数相比,开立方的方法在计算速度上提高超过 30%. 并且作者利用该确定函数构造了与随机谕言不可区分的函数.

关键词 散列;随机谕言;椭圆曲线;C_{34} 曲线

中图法分类号 TP309 **DOI 号**: 10.3724/SP.J.1016.2012.01868

Construct Hash Function from Plaintext to C_{34} Curves

YU Wei[1),2)] WANG Kun-Peng[2)] LI Bao[2)] TIAN Song[2)]

[1)](*Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei* 230027)

[2)](*Institute of Information Engineering, Chinese Academy of Sciences, Beijing* 100093)

Abstract This paper proposes a deterministic construction of an encoding from a finite filed $\mathbb{F}_q$ to a C_{34} curve using finding cube roots method, where the time complexity is $\mathcal{O}(\log^3 q)$. Based on the deterministic encoding, we construct a hash function from plaintext to C_{34} curves. The new method provides up more than 30% speed improvements compare to Icart T.'s hash function in Crypto 2009 on the same prime filed. Moreover, we provide a new function indifferentiable from a random oracle based on our deterministic encoding.

Keywords hash; elliptic curves; random oracle; C_{34} curves

1 引 言

公钥密码体制成功地解决了对称密码面临的 3 个难题:密钥分发、密钥管理和提供不可否认性. 公钥密码一般基于所依赖的数学难题来分类. 目前被认为安全有效的主要有 3 类:

(1) 基于大整数分解问题的 RSA 公钥密码体制.

(2) 基于有限域上离散对数问题的 Elgamal 型的公钥密码体制.

(3) 基于代数曲线的 Jacobian 群的离散对数问题的代数曲线公钥密码体制,主要包括椭圆曲线公钥密码体制、超椭圆公钥密码体制和 C_{ab} 曲线公钥密码体制.

由于 C_{ab} 曲线密钥短、曲线丰富、有更多的密码学结构,并且不存在亚指数时间攻击算法,故而受到越来越多的关注.

Andreas[1] 指出了离散对数在亏格比较高的曲

收稿日期:2012-05-10;最终修改稿收到日期:2012-07-20. 本课题得到国家自然科学基金(60970153)、中国科学院战略性先导专项基金(XDA06010702)资助.于 伟,男,1987 年生,博士研究生,目前主要从事椭圆曲线密码学方面的研究. E-mail: yuwei_1_yw@163.com. 王鲲鹏,男,1971 年生,博士,副研究员,主要研究方向为椭圆曲线密码学. 李 宝,男,1962 年生,博士,研究员,博士生导师,主要研究领域为椭圆曲线密码学、可证明安全方法的理论与应用、安全协议的设计与分析. 田 松,男,1987 年生,博士研究生,目前主要从事椭圆曲线密码学方面的研究.

线上是不安全的. 在实际应用中，取亏格小于 5 的代数曲线. C_{ab} 曲线的亏格是 $\frac{(a-1)(b-1)}{2}$，本文主要考虑亏格为 3 的 C_{34} 曲线. 文献[2]中给出了 C_{34} 曲线上如何进行快速有效计算的方法.

Boneh-Frankin 构造的基于身份的加密算法[3]中，公钥需要映射到代数曲线上的点. 他们构造了一个从基域 $\mathbb{F}_q$ 到超奇异椭圆曲线上点的一一映射 f，可以证明当 h 是一个传统散列函数时，$f(h(m))$ 也相当于一个传统散列函数. 其它的一些系统，如基于身份的加密算法[4-6]、基于身份的签名算法[7-11]以及基于身份的签密算法[12-13]都需要散列进代数曲线.

基于身份的加密[3]一文中指出：映射 $f: S \mapsto R$ 具有如下 3 条性质时，$f \circ h$ 是散列函数.

(1) 可计算性：f 在确定的多项式时间内是可计算的.

(2) 映射的原象数目可计数：对于任意的 $r \in R$，$|f^{-1}(r)| \leq l$.

(3) 对于任意的象，都存在多项式时间的算法，以一定的概率返回其中的一个原象.

$a=2, b=3$ 时，C_{23} 曲线即为椭圆曲线. 散列进椭圆曲线的方法有 Boneh，Lynn，Shacham[14]短签名方案中的非确定性时间的算法、Shallue 和 Woestijne[15]基于开平方根的算法和 Icart T[16]的算法. 散列进超椭圆曲线的方法有文献[17-18]中的方法.

散列进 C_{34} 曲线像散列进椭圆曲线一样，构造一个从基域 $\mathbb{F}_q$ 到 C_{34} 曲线上的映射 f，当 h 是一个传统散列函数时，需要证明 $f(h(m))$ 也满足传统散列函数的条件. 至今还没有散列进 C_{34} 曲线的研究. 本文给出了构造从字符串到 C_{34} 曲线上的一一映射 f 的方法，当 h 是传统散列函数时，$f \circ h$ 是散列函数. 并且利用该函数构造了到 C_{34} 曲线 Jacobian 群上与随机谕言不可区分的函数.

本文所讨论的问题都是在域 $\mathbb{F}_q$ 上，其中 $q=p^n$，p 是素数并且 $p>3$，n 为任意的正整数.

本文第 2 节给出 C_{34} 曲线的基础知识；第 3 节论述如何构造从有限域到 C_{34} 曲线上的映射；第 4 节证明函数 $f \circ h$ 是一个散列函数；第 5 节给出在 NIST 素域和 SM2 素域上的计算实例和算法的时间和空间复杂性分析；第 6 节给出构造字符串到 C_{34} 曲线 Jacobian 群上的与随机谕言不可区分的函数；第 7 节给出文章的结论.

2　背景知识

2.1　C_{34} 曲线

C_{ab} 曲线最先由 Miura S[19-21]提出. 令 K 是一个完全域，$\overline{K}$ 为 K 的代数闭包，$\mathcal{X}$ 是定义在 $\overline{K}^2$ 上的不可约的仿射代数集. x, y 是仿射平面 $\overline{K}^2$ 的两个坐标. a, b 是互素的两个整数，则下面的两个条件是等价的(证明见文献[22]定理 1)：

(1) $\mathcal{X}$ 是绝对不可约的代数曲线，恰好有一个 K 无穷远有理点 Q. aQ 是 x 的唯一极点，bQ 是 y 的的唯一极点.

(2) $\mathcal{X}$ 是由如下所示定义的双变量的多项式：

$$\alpha_{b,0}x^b+\alpha_{0,a}y^a+\sum_{ia+jb<ab}\alpha_{i,j}x^iy^j\alpha_{i,j}\in K,\quad \alpha_{b,0},\ \alpha_{0,a}\neq 0 \tag{1}$$

式(1)所定义的代数曲线叫做 C_{ab} 曲线(文献[22]的定义 2).

在 C_{ab} 曲线中，令 $a=3, b=4$，得到 C_{34}[22-23]曲线的方程为

$$\alpha_{4,0}x^4+\alpha_{3,0}x^3+\alpha_{2,0}x^2+\alpha_{1,0}x+\alpha_{0,0}+\alpha_{0,3}y^3+\alpha_{0,2}y^2+\alpha_{0,1}y+\alpha_{2,1}x^2y+\alpha_{1,2}xy^2+\alpha_{1,1}xy=0 \tag{2}$$

实际应用中，常使用形式简单的 C_{34} 曲线：

$$\alpha_{4,0}x^4+\alpha_{3,0}x^3+\alpha_{2,0}x^2+\alpha_{1,0}x+\alpha_{0,0}=\alpha_{0,3}y^3.$$

这里以一条 C_{34} 曲线：$y^3=x^4+2x^2-3x-1$ 为例，在实数域上的图形如图 1 所示.

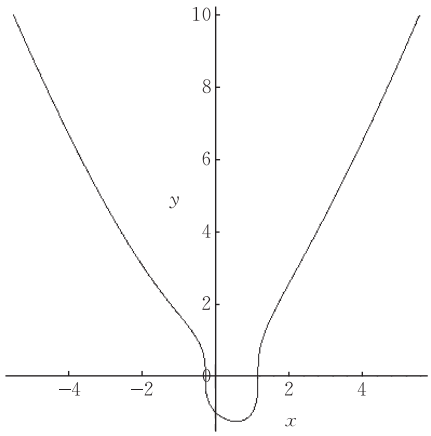


图 1　C_{34} 曲线 $y^3=x^4+2x^2-3x-1$

C_{34} 有唯一的无穷远点. 其亏格是 3，是除椭圆曲线和超奇异椭圆曲线外亏格最小的 C_{ab} 曲线.

2.2　C_{34} 曲线上的 Jacobian 群

令 C 表示一条 C_{34} 曲线，$K(C)$ 是 C 的函数域.

除子类群 $Div(C)$定义为由 C 上的点 P 生成的自由阿贝尔群. 除子 $D\in Div(C)$定义为

$$D=\sum_{P\in C}n_P P,\ n_P\in\mathbb{Z},$$

并且只有有限个 n_P不为 0. 除子 D 的次数定义为

$$\deg D=\sum_{P\in C}n_P.$$

零次除子的集合

$$D_C^0(K)=\{D\in Div(C)\mid \deg D=0\}.$$

对于 $f\in\overline{K}(C)^*$,定义除子

$$(f)=\sum_{P\in C}\mathrm{ord}_P(f),$$

其中 $\mathrm{ord}_P(f)$表示 f 在 $P\in C$ 的阶. 如果存在 $f\in\overline{K}(C)^*$ 使得 $D=(f)$,则称 D 为主除子. C 上的主除子类群是 $D_C^0(K)$的子群,记做 $P_C(K)$. Jacobian 群定义如下

$$J_C(K)=D_C^0(K)/P_C(K).$$

C_{34} 曲线上的 Jacobian 群的具体运算见文献[24]. 关于 C_{34} 曲线,更多的介绍,请参考文献[24-26].

2.3 构造散列函数

短签名中的构造散列函数算法[14]的运行时间不是常数,还可能导致时间攻击(具体分析见文献[16]). 当 q 是模 4 余 3 的情况下,平均时间复杂性是 $\mathcal{O}(\log^3 q)$;其余情况下的平均时间复杂性是 $\mathcal{O}(\log^4 q)$. 后来,密码学研究者寻找到了运行时间可以确定的散列函数. Shallue 和 Woestijne[15]的利用 Skalba 的等式[27]的散列算法是基于开平方根的,在域 $\mathbb{F}_q$(q 模 4 余 3)中,时间复杂性是 $\mathcal{O}(\log^3 q)$;其它情况下的时间复杂性是 $\mathcal{O}(\log^4 q)$. 在 $q\equiv 2 \bmod 3$ 情况下,Icart 的确定散列算法[16]的时间复杂性是 $\mathcal{O}(\log^3 q)$.

Ulas[18]给出了到形如 $y^2=x^n+ax+b$ 和 $y^2=x^n+ax^2+bx$ 超椭圆曲线上的散列算法. 其算法的本质是寻找二次剩余. 对于 C_{ab} 曲线,因为 x,y 次数的提高,构造散列的情况要复杂得多. 本文主要考察 C_{ab} 曲线中应用和研究较多的 C_{34} 曲线.

3 构造从有限域到 C_{34} 曲线的函数

当 $q=p^n\equiv 2 \bmod 3$ 时,映射 $x\mapsto x^3$ 是一个双射,并且有逆映射 $x\mapsto x^{1/3}=x^{(2q-1)/3}$. 由此可以构造从有限域到 C_{34} 曲线的一一映射.

对于域 $\mathbb{F}_q$ 中的任意元素 u,构造从 $\mathbb{F}_q$ 到 C_{34} 曲线的映射 f:

$$f:u\mapsto\left(u,\left(\frac{\alpha_{4,0}u^4+\alpha_{3,0}u^3+\alpha_{2,0}u^2+\alpha_{1,0}u+\alpha_{0,0}}{\alpha_{0,3}}\right)^{1/3}\right)\tag{3}$$

这是一个关于 u 的 1 次方程 $u=x$,即象集的点一定存在唯一的原象.

引理 1. $\mathbb{F}_q$ 表示一个域,$q=p^n\equiv 2 \bmod 3$,$p>3$. 对于任意的 $u\in\mathbb{F}_{p^n}$,$f(u)$是 $\alpha_{4,0}x^4+\alpha_{3,0}x^3+\alpha_{2,0}x^2+\alpha_{1,0}x+\alpha_{0,0}=\alpha_{0,3}y^3$ 中的点.

证明.

$$x=u,$$

$$y=\left(\frac{\alpha_{4,0}u^4+\alpha_{3,0}u^3+\alpha_{2,0}u^2+\alpha_{1,0}u+\alpha_{0,0}}{\alpha_{0,3}}\right)^{1/3}.$$

代入 $u=x$,可以得到 x,y 之间的关系

$$y=\left(\frac{\alpha_{4,0}x^4+\alpha_{3,0}x^3+\alpha_{2,0}x^2+\alpha_{1,0}x+\alpha_{0,0}}{\alpha_{0,3}}\right)^{1/3},$$

得到

$$y^3=\frac{\alpha_{4,0}x^4+\alpha_{3,0}x^3+\alpha_{2,0}x^2+\alpha_{1,0}x+\alpha_{0,0}}{\alpha_{0,3}}.$$

所以,

$$\alpha_{4,0}x^4+\alpha_{3,0}x^3+\alpha_{2,0}x^2+\alpha_{1,0}x+\alpha_{0,0}=\alpha_{0,3}y^3.$$

这样就证明了引理 1.

根据 f 的定义,可以发现不同的原象映射到不同的象,原象的 q 个元素恰好映射到象集中的 q 个不同的点(x 坐标互不相同). 这说明 $\mathbb{F}_q$中的元素映射到 C_{34} 曲线中的 q 个不同的点.

4 构造从字符串到 C_{34} 曲线上点的散列函数

设映射 $h:\{0,1\}^*\mapsto\mathbb{F}_q$是一个传统散列函数,$m$ 是消息(表示成 0,1 字符串). 我们构造了一个一一映射 f,将域 $\mathbb{F}_q$中的元素映射到 C_{34} 曲线上的点. 下面证明 $f\circ h$ 是一个散列函数,即 $f\circ h$ 具有单向性和抗碰撞性. 首先,定义散列函数的单向性和抗碰撞性.

4.1 单向性

定义 1. 一个散列函数被称作是(t,ε)单向:任给一个 $y\in Im(h)$,其中 $Im(h)$是象集,对于任意算法运行时间 t,输出 m,满足 $h(m)=y$ 的概率最多是 ε. 可以证明这是一个从域中元素映射到椭圆曲线的散列.

引理 2. 如果 h 是一个(t,ε)的单向函数,f 如式(3)所定义,则 $H=f\circ h$ 是一个(t',ε)的单向函数.

证明. 令 $u=h(m)$,则

$$H(m)=\left(u,\left(\frac{\alpha_{4,0}u^4+\alpha_{3,0}u^3+\alpha_{2,0}u^2+\alpha_{1,0}u+\alpha_{0,0}}{\alpha_{0,3}}\right)^{1/3}\right).$$

f 的逆映射 $f^{-1}:C_{34}$ 上的点$\to\mathbb{F}_q$，即是 $P=(x,y)\mapsto x$，则 $H^{-1}(P)=h^{-1}(x)$. 因为 h 是一个(t,ε)的单向函数，所以 $H=f\circ h$ 是一个(t',ε)的单向函数. 证毕.

4.2　抗碰撞性

定义 2.　一个散列函数簇 $\mathbb{H}$ 被称作是(t,ε)抗碰撞的：对于任意算法运行时间为 t，任给一个 $h\in\mathbb{H}$，输出(m,m')满足 $m\neq m'$，$h(m)=h(m')$的概率最多是 ε.

对于给定的散列函数簇 $\mathbb{H}$，任意的 $h\in\mathbb{H}$，定义函数 $H=f\circ h$，f 为 $\mathbb{F}_q$到 C_{34} 曲线上点的函数. H 中发生碰撞当且仅当：

(1)存在 m 和 m'，使得 $h(m)=h(m')$，这是 h 的一个碰撞.

(2)或者对于 $u=h(m)$，$u'=h(m')$，$u\neq u'$满足 $f(u)=f(u')$，这是 f 造成的一个碰撞.

引理 3.　f 如式(3)所定义，如果 h 是(t,ε)抗碰撞的，那么 H 是(t',ε)抗碰撞的.

证明.　设(m,m')是 H 的碰撞，由于 f 是一一映射，所以 H 发生碰撞只能是 h 也发生了碰撞. 同理，h 的碰撞也是 H 的一个碰撞. 如果 h 是(t,ε)抗碰撞的，那么 H 是(t',ε)抗碰撞的.　　证毕.

5　实验结果与分析

本节以 C_{34} 曲线 $y^3=x^4+2x^2-3x-1$（如图 1）定义在 NIST 和 SM2 推荐使用的素域 $\mathbb{F}_p$ 上为例，给出具体的计算方法和计算时间.

5.1　NIST 素域和 SM2 素域上的 C_{34} 曲线的散列

美国国家标准与技术协会 NIST 推荐的 192 bit 素域 P192 和 384 bit 素域 P384. 中国国家密码管理局给出了椭圆曲线密码标准 SM2，推荐了二进制长度为 256 位的素域 S256. 通过计算得知 P192$\equiv$2 mod 3，P384$\equiv$2 mod 3，S256$\equiv$2 mod 3. 用 u 表示 $\mathbb{F}_p$中的元素，映射 f 的构造如下. 在 3 个素域上，$u=1$ 的运行结果如表 2，$u=2$ 的运行结果如表 3，随机选取 u 之后的平均运行时间如表 4.

$$f:u\mapsto\left(u,\left(\frac{\alpha_{4,0}u^4+\alpha_{3,0}u^3+\alpha_{2,0}u^2+\alpha_{1,0}u+\alpha_{0,0}}{\alpha_{0,3}}\right)^{1/3}\right).$$

计算 $x^{1/3}$ 的方法如下：$x^{1/3}=x^{(2q-1)/3}$.

表 1　NIST 素数和 SM2 素域

素域名称	素域大小
P192	$2^{192}-2^{64}-1$
P384	$2^{384}-2^{128}-2^{96}+2^{32}-1$
S256	$2^{256}-2^{224}-2^{96}+2^{64}-1$

表 2　$u=1$ 时的运行结果

素域名称	运行结果(十六进制)
P192	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFFFFFFFFFFFFE
P384	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFFFFF0000000000000000FFFFFFFE
S256	FFFFFFFEFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF00000000FFFFFFFFFFFFFFFE

表 3　$u=2$ 时的运行结果

素域名称	运行结果(十六进制)
P192	13CCC34C326FC4E767F52212D966D8CCE4D17D573E254055
P384	C63B40D90223F1CE8D961BF39394224E8D25154647434B30042A053074CA4ECD97CA0EFC54378593C3BC8D339C0A1DC5
S256	4196DFF089DB32EC4AED31786B72D56B806C3898FEAF270C46F90BEF464D4FA7

表 4　平均运行时间

素域名称	平均运行时间/ms	
	开立方	Icart T 函数
P192	1.1	1.9
P384	2.1	2.9
S256	2.5	3.8

使用大数运算库 LibTom[28] 编程实现 Icart T 的散列函数，平台是 Intel Core2 2.66 GHz 四核处理器. Icart T 的散列函数的平均运行时间如表 4 所示. 在相同的素域上，与 Icart T 散列函数的运行时间相比，我们的算法平均提高 30%以上.

5.2　时间复杂性

域 $\mathbb{F}_q$的乘法时间用 M 表示，域中的元素乘以一个常数用 M_A表示，平方用 S 表示，求逆用 I 表示，开立方根用 E 表示.

散列进 C_{34} 曲线的算法中，主要是计算映射

$$f:u\mapsto\left(u,\left(\frac{\alpha_{4,0}u^4+\alpha_{3,0}u^3+\alpha_{2,0}u^2+\alpha_{1,0}u+\alpha_{0,0}}{\alpha_{0,3}}\right)^{1/3}\right).$$

计算 f 的时间为 $I+5M_A+M+2S+E$. 在域 $\mathbb{F}_q$中，当 $q=p^n\equiv2$ mod 3 时，$x^{1/3}=x^{(2q-1)/3}$. 也就是说开立方就是计算一个数的$(2q-1)/3$ 次方，时间复杂性为$\mathcal{O}(\log^3 q)$.

实际上，取

$$\alpha'_{4,0}=\frac{\alpha_{4,0}}{\alpha_{0,3}},\ \alpha'_{3,0}=\frac{\alpha_{3,0}}{\alpha_{0,3}},\ \alpha'_{2,0}=\frac{\alpha_{2,0}}{\alpha_{0,3}},$$

$$\alpha'_{1,0}=\frac{\alpha_{1,0}}{\alpha_{0,3}},\ \alpha'_{0,0}=\frac{\alpha_{0,0}}{\alpha_{0,3}},$$

则原 C_{34} 曲线的方程化为

$$y^3=\alpha'_{4,0}x^4+\alpha'_{3,0}x^3+\alpha'_{2,0}x^2+\alpha'_{1,0}x+\alpha'_{0,0},$$

映射 f 变为

$$f:u\mapsto(u,(\alpha'_{4,0}u^4+\alpha'_{3,0}u^3+\alpha'_{2,0}u^2+\alpha'_{1,0}u+\alpha'_{0,0})^{1/3}),$$

时间复杂性为 $4M_A+M+2S+E$. Icart T[16]的时间复杂性为 $I+2M+3S+E$. 在素域上,一般取 $I+2M+3S+ES=0.8M$,利用重复平方法计算 $E: x^{1/3}=x^{(2q-1)/3}$,需要 $\left(S+\frac{M}{2}\right)\log p$. 选取比较小的参数 $\alpha'_{i,0}$,$0\leq i\leq 4$,则 M_A 相对于 M 可以忽略. 开立方的方法的时间复杂性是 $(2.6+1.3\log p)M$. Icart T 的散列函数[16]的时间复杂性为 $(104.4+1.3\log p)M$. 如果 p 的位数为 192 位,则节省 $101.8/(104.4+1.3\times 192)\times 100\%\approx 30\%$.

6 构造随机谕言

在文献[29]中指出了如何利用 Icart T 的确定函数构造与随机谕言模型不可区分的函数. 文献[30]中指出了如何构造到亏格大于 1 的曲线上与随机谕言不可区分的函数. 利用文献[30]中给出的方法构造出到 C_{34} 曲线 Jacobian 群上的与随机谕言不可区分的函数. 构造如下:

$$H(m)=f(h_1(m))+f(h_2(m))+f(h_3(m))+\cdots+f(h_i(m)).$$

其中,$H_j(1\leq j\leq i)$是传统的散列函数. 根据文献[30]中给出的结论,只要 i 大于 C_{34} 曲线的亏格,该构造就是与随机谕言不可区分的函数. 即时,我们取 $i=4$,得到构造为 $H(m)=f(h_1(m))+f(h_2(m))+f(h_3(m))+f(h_4(m))$.

7 结 论

当 $q=p^n\equiv 2 \bmod 3$ 时,映射 $x\mapsto x^3$ 是 $\mathbb{F}_q$ 到自身的一个双射. 利用逆映射 $x\mapsto x^{1/3}=x^{(2q-1)/3}$,本文给出了从域 $\mathbb{F}_q$ 到定义在 $\mathbb{F}_q$ 上 C_{34} 曲线 $\alpha_{4,0}x^4+\alpha_{3,0}x^3+\alpha_{2,0}x^2+\alpha_{1,0}x+\alpha_{0,0}=\alpha_{0,3}y^3$ 的散列函数 f. 与 Icart T 构造的到椭圆曲线上的散列函数相比,我们构造的散列函数在时间复杂性上有较大的优势. 在相同的素域上,计算速度至少提高 30%. 由于 f 是一个双射,由传统散列函数 $h:\{0,1\}^*\mapsto\mathbb{F}_q$ 可以得到从字符串 $\{0,1\}^*$ 到 C_{34} 曲线的散列函数 $f\circ h$ 和与随机谕言不可区分的函数.

参 考 文 献

[1] Andreas Enge. Computing discrete logarithms in high-genus hyperelliptic Jacobians in provably subexponential time. Mathematics of Computation, 2002, 71(238): 729-742(electronic). MR 1885624(2003b:68083)

[2] Arita S. An addition algorithm in Jacobian of C_{34} curve// Proceedings of the Information Security and Privacy, ACISP 2003. Lecture Notes in Computer Science 2727. Springer-Verlag, 2003: 93-105

[3] Boneh D, Franklin M K. Identity-based encryption from the weil pairing//Kilian J ed. Proceedings of the CRYPTO. Lecture Notes in Computer Science 2139. Springer, 2001: 213-229

[4] Baek J, Zheng Y. Identity-based threshold decryption//Proceedings of the PKC 2004. Springer, 2004: 262-276

[5] Gentry C, Silverberg A. Hierarchical id-based cryptography//Proceedings of the ASIACRYPT 2002. Springer, 2002: 548-566

[6] Horwitz J, Lynn B. Toward hierarchical identity-based encryption//Knudsen L R ed. Proceedings of the EUROCRYPT. Lecture Notes in Computer Science 2332. Springer, 2002: 466-481

[7] Boldyreva A. Threshold signatures, multisignatures and blind signatures based on the Gap-Diffie-Hellman-group signature scheme//Proceedings of the PKC 2003. Springer, 2003: 31-46

[8] Boneh D, Gentry C, Lynn B, Shacham H. Aggregate and verifiably en-crypted signatures from bilinear maps//Proceedings of the EUROCRYPT 2003. Springer, 2003: 416-432

[9] Boneh D, Lynn B, Shacham H. Short signatures from the weil pairing//Boyd C ed. Proceedings of the ASIACRYPT. Lecture Notes in Computer Science 2248. Springer, 2001: 514-532

[10] Cha J C, Cheon J H. An identity-based signature from gap Diffie-Hellman groups//Proceedings of the PKC 2003. Springer, 2003: 18-30

[11] Zhang F, Kim K. Id-based blind signature and ring signature from pairings//Proceedings of the ASIACRYPT 2002. Springer, 2002: 533-547

[12] Boyen X. Multipurpose identity-based signcryption (a swiss army knife for identity-based cryptography)//Boneh D ed. Proceedings of the CRYPTO. Lecture Notes in Computer Science 2729. Springer, 2003: 383-399

[13] Libert B, Quisquater J-J. Efficient signcryption with key privacy from gap Diffie-Hellman groups//Proceedings of the PKC 2004. Springer, 2004: 187-200

[14] Boneh D, Lynn B, Shacham H. Short signatures from the weil pairing. Journal of Cryptology, 2004, 17(4): 297-319

[15] Shallue A, Woestijne C. Construction of rational points on elliptic curves over finite fields//Hess F, Pauli S, Pohst M eds. Proceedings of the ANTS 2006. Lecture Notes in Computer Science 4076, Heidelberg: Springer, 2006: 510-524

[16] Icart T. How to hash into elliptic curves//Proceedings of the Crypto' 2009. Lecture Notes in Computer Science 5677, Springer-Verlag, 2009: 303-316

[17] Fouque P-A, Tibouchi M. Deterministic encoding and hashing to odd hyperelliptic curves. http://eprint. iacr. org/2010/382. pdf, 2010

[18] Ulas M. Rational points on certain hyperelliptic curves over finite fields. Bulletin of the Polish Academy of Sciences Mathematics, 2007, 55: 97-104

[19] Miura S. Algebraic geometric codes on certain plane curves. Transactions of IEICE, 1992, J75-A(11): 1735-1745

[20] Miura S. Algebraic geometric codes on certain plane curves [Ph. D. dissertation]. University of Tokyo, Tokyo, 1997

[21] Miura S. Linear codes on affine algebraic curves. Transactions of IEICE, 1998, J81-A(10): 1398-1421

[22] Matsumoto R. The C_{ab} curve—A generalization of the Weierstrass form to arbitrary plane curves. http://www.rmatsumoto.org/cab.html

[23] Miura S. Linear codes on affine algebraic curves. Transactions of IEICE, 1998, J81-A(10): 1398-1421

[24] Seigo A. An addition algorithm in Jacobian of 34 Curve. Lecture Notes in Computer Science 2727, Springer, 2003: 93-105

[25] Basiri A, Enge A, Faugèere J-C, Gürel N. Implementing the arithmetic of C_{34}-curves//Proceedings of the Algebraic Number Theory—ANTS VI. Lecture Notes in Computer Science, Berlin: Springer-Verlag, 2004: 87-101

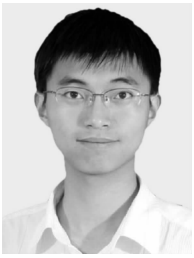
[26] Kim S. Computing in the Jacobian of a C_{34} curve. Tohoku Mathematical Publications, 2004, 30(30): 1-63

[27] Skalba M. Points on elliptic curves over finite fields. Acta Arithmetica, 2005, 117: 293-301

[28] LibTom. http://libtom.org/

[29] Brier E, Coron J-S, Icart T, Madore D, Randriam H, Tibouchi M. Efficient indifferentiable hashing into ordinary elliptic curves//Rabin T ed. Proceedings of the CRYPTO. Lecture Notes in Computer Science. Springer, 2010: 237-254

[30] Farashahi R R, Fouque P A, Shparlinski I E, Tibouchi M, Felipe Voloch J. Indifferentiable deterministic hashing to elliptic and hyperelliptic curves. http://eprint.iacr.org/2010/539.pdf, 2010



YU Wei, male, born in 1987, Ph. D. candidate. His currently working on public key cryptosystems.

WANG Kun-Peng, born in 1971, Ph. D., associate professor. His main research interests include elliptic curve cryptography.

LI Bao, born in 1962, Ph. D., professor, Ph. D. supervisor. His main research interests include elliptic curve crypto- graphy, the theory and application of provable security, and design and analysis of cryptographic protocols.

TIAN Song, born in 1987, Ph. D. candidate. His main research interest is elliptic curve cryptography.

Background

Many algebraic curve cryptosystems require to hash message into an algebraic curve. Boneh-Franklin identity based encryption scheme proposes a one-to-one mapping f from the base field $\mathbb{F}_p$ to a particular supersingular elliptic curve. This enables to hash using $f(h(m))$ where h is a classical hash function. There are many other identity based schemes need hashing into an algebraic curve, such as encryption schemes, signature schemes, signcryption schemes and so on.

The algorithm proposed by Boneh D. et al. mapping an element of $\mathbb{F}_{p^n}$ into an elliptic curve is probabilistic which fails to return a point for a fraction 2^{-k} of the inputs, where k is a predetermined bound. One drawback of the algorithm is that the number of steps of the algorithm depends on the input u. Thus the number of operations is not constant. If the input u has to be secret in practice, this may lead to a timing attack.

The algorithms mapping $\mathbb{F}_{p^n}$ into an elliptic curve in deterministic polynomial time were published by Shallue and Woestijne in ANTS 2006 and Icart T. in Crypto 2009. Shallue and Woestijne's algorithm is based on Skalba's equality and uses a modification of the Tonelli-Shanks algorithm for computing square roots. This algorithm runs in time $\mathcal{O}(\log^3 q)$ when $q \equiv 3 \bmod 4$ and in time $\mathcal{O}(\log^4 q)$ otherwise. Icart T.'s algorithm is based on computing a cube root, which can be implemented in $\mathcal{O}(\log^3 q)$ time and in a constant number of operations over $\mathbb{F}_q$, when $q = p^n \equiv 2 \bmod 3$. Their algorithms encode an element of a finite field into Weierstrass form elliptic curves.

There are some methods hashing into hyperelliptic curves. Their main idea is computing square roots.

This paper proposes a new deterministic construction of an encoding from a finite filed $\mathbb{F}_q$ to a C_{34} curve using finding cube roots method, where the time complexity is $\mathcal{O}(\log^3 q)$. Based on the deterministic encoding, we construct a hash function from plaintext to C_{34} curves. The new method provides up more than 30% speed improvements compare to Icart T.'s hash function in Crypto 2009 on the same prime filed. Moreover, we provide a new function indifferentiable from a random oracle based on our the deterministic encoding.