Tate 对的计算及其在密钥交换协议中的应用

Computation of Tate Pairing and Its Application In Key-exchange Protocol

专 业 名 称：　　计 算 机 软 件 与 理 论

学位申请人：　　郑　督

指 导 老 师：　　张治国

答辩委员会主席：　_____

委员：　_____

_____

_____

_____

_____

二零零九年五月

原创性及学位论文使用授权声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期：　　年　　月　　日

本人完全了解中山大学有关保留、使用学位论文的规定，即：学校有权保留学位论文并向国家主管部门或其指定机构送交论文的电子版和纸质版，有权将学位论文用于非赢利目的的少量复制并允许论文进入学校图书馆、院系资料室被查阅，有权将学位论文的内容编入有关数据库进行检索，可以采用复印、缩印或其他方法保存学位论文。

学位论文作者签名：　　导师签名：

日期：　　年　　月　　日　　日期：　　年　　月　　日

Tate对的计算及其在密钥交换协议中的应用

专　　业: 计 算 机 软 件 与 理 论
硕 士 生: 郑 督
指 导 老 师: 张治国

摘 要

由Kobliz [14]和Miller [20] 提出的椭圆曲线密码学是密码学中一个具有重要意义的研究课题。椭圆曲线上的双线性对在椭圆曲线密码中起着重要意义。一方面，椭圆曲线上的双线性对被用来攻击椭圆曲线上离散对数问题[18][10]，将椭圆曲线有理点群上的离散对数问题归约为有限域乘法群上的离散对数问题；另一方面，双线性对被用作构造密码协议[13][6]。因此，提高椭圆曲线上双线性对的计算效率至关重要。本文主要研究了双线性Tate对的计算及其在密钥交换协议中的应用, 获得如下结果：

1. 在雅可比坐标下，给出了计算两类非超奇异椭圆曲线上的双线性Tate对，Miller算法复杂度的估计值。
2. 给出了有限域$\mathbb{F}_p$上，求逆运算和乘法及平方运算的计算效率的测试结果和分析，用来支持给出上述估计值时用到的假设。
3. 指出了一类利用双线性Tate对构造的基于身份的密钥交换协议的不足，并给出证明。

关键词：双线性对、Tate对、密钥交换协议。

Computation of Tate Pairing and Its application In Key-exchange Protocol

Major: Computer Software and Theory
Name: Du Zheng
Supervisor: Zhiguo Zhang

Abstract

Elliptic curve cryptography proposed by Kobliz [14]and Miller[20] is a critical research topic in cryptography. Bilinear pairings on elliptic curve exert an imperative role in Elliptic Curve Cryptography. For one thing, bilinear pairings are used to attack the Discrete Logarithm Problem on elliptic curves[18][10], which reduce the Discrete Logarithm Problem on elliptic curves to the Discrete Logarithm Problem over finite field. For another thing, bilinear pairings are used to construct cryptographic protocols[13][6]. Therefore, it is imperative to improve the computation efficiency of pairings. In this thesis, the computation of Tate Pairing and its application in key-exchange protocol are discussed. The main results are listed as follows:

1. In Jacobian Coordinate, an estimated computation complexity of Miller algorithm is given concerning computing two types of Tate pairings constructed by using endomorphism on non-supersingular elliptic curves.

2. Test results about the computation efficiency of Inversion and Multiplication over finite field with big prime order are shown to support the hypothesis in the above mentioned estimation.

3. Point out and prove the deficiency of two key-exchange protocols constructed by using Tate pairing.

Keywords: Bilinear pairings, Tate pairing, key-exchange protocol.

目 录

第一章　绪论

椭圆曲线密码学是由Kobliz[14]和Miller[20]于1985年分别提出。椭圆曲线密码体制的优点在于，同其他传统的公钥密码体制相比[23]，在相同安全性的前提下，椭圆曲线密码体制可以使用较短的密钥达到同样的安全性。因此，椭圆曲线密码学成为密码学界的研究热点。

椭圆曲线上的双线性对在早期被用来攻击椭圆曲线有理点群上的离散对数问题。这些算法思想是将椭圆曲线有理点群上的离散对数问题归约为有限域乘法群上的离散对数问题[18] 。2002年以来，基于椭圆曲线上的双线性对构造的密码协议在密码学实际应用中起着越来越重要的作用。

下面给出，两种基于椭圆曲线上双线性对构造的密码体制，来更好的说明椭圆曲线上的双线性对是如何用来构造密码体制的，并在第三章给出基于身份的密钥交换协议的构造。

1.1　基于Weil对的身份加密体制

为了简单起见，我们考虑基于一类特殊椭圆曲线上Weil对[7]构造的基于身份的加密。

令椭圆曲线$E: y^2 = x^3 + 1$定义在有限域$\mathbb{F}_p$上，其中$p = 2 \mod 3$。再令，$\omega \in \mathbb{F}_{p^2}$是一个三次单位根。

定义如下映射，

$$\beta : E(\mathbb{F}_{p^2}) \to E(\mathbb{F}_{p^2})$$

$$(x, y) \to (\omega x, y)，\ \beta(\mathcal{O}_E) = \mathcal{O}_E$$

假设椭圆曲线上有理点P的阶为n，则$\beta(P)$的阶也为n。定义约化的Weil对为：

$e_n(P_1, P_2) = e_n(P_1, \beta(P_2))$，其中$e_n$如通常定义下的Weil对，$P_1$，$P_2 \in E[n]$。

当$3 \nmid n$，且$P \in E(\mathbb{F}_p)$的阶为n，则$e_n(P, P)$是单位元的n次本原根。

在该基于Weil对的身份加密密码体制中，我们假定，每个用户都有一个基于他们身份的公钥，例如，他们的邮件地址。一个可信赖的机构为每个用户分配一个对应其公钥的

私钥。在多数公钥密码体制当中，当Alice想要给Bob发送消息时，Alice首先要查找Bob的公钥。然而，Alice需要某种方法来确信这个公钥确实是属于Bob的，来防止某些人诸如Eve来冒充Bob。在该密码体制当中，认证发生在Bob和可信赖的机构初始通信阶段。在此之后，Bob是唯一拥有用其公开身份进行加密的信息解密密码的人。

一个很自然的问题是，为什么RSA[23]不能被用作来构建这样一个密码体制？例如，所有的用户能够使用同一大整数n，其唯一的素数分解只有可信赖的机构知道。Bob的身份，称作bobid，是其公开的加密指数。可信赖的机构将计算其秘密的解密指数，并把该解密指数传给Bob。当Alice想要向Bob发送一个消息m，Alice首先计算$m^{bobid} \mod n$。然后，Bob将用由可信赖的机构提供的解密指数来解密。问题在于，任一个用户，如同Bob，当知道一个加密和解密指数时，就可以采用消息重放攻击来分解大整数n。因此，就可以获得整个在该密码系统中传输的消息。因此，如此构建的密码系统，不能够提供对消息的保密。如果，对每一次的通讯，采取不同的大整数n，这样问题又回到原来的问题，如何保证用于每一次通讯的大整数n的安全传输。

为了解决上述问题，密码学家提出了基于身份的加密，来避免上述提到的诸多问题。

为了构建该密码体制，可信赖的机构做如下步骤：

1. 选择$E(\mathbb{F}_p)$中一个阶为l的点P。

2. 选择两个哈希函数H_1和H_2。其中哈希函数H_1将任意长度的比特串映射到椭圆曲线E上的一个阶为l的点。哈希函数H_2将$\mathbb{F}_{p^2}$ 中乘法群的阶为l的元素映射为一条长度为n的字符串，其中n为将要传送消息的长度。

3. 在$\mathbb{F}_l$的乘法群中选择一个随机数s，并计算$P_{pub} = sP$。

4. 公开P，H_1，H_2，n，P，P_{pub}，保留s。

如果一个身份用ID标识的用户想要获得一个私钥，可信赖的第三方做如下步骤：

1. 计算$Q_{ID} = H_1(ID)$，此时Q为椭圆曲线E上一点。

2. 令$D_{ID} = sQ_{ID}$。

3. 当可信赖的机构验证用户的身份标识无误后，可信赖的机构将D_{ID}发送给用户。

如果Alice想要发送消息M给Bob，Alice将做如下步骤：

1. Alice查找Bob的标识，例如，$ID = bob@computer.com$，然后计算$Q_{ID} =$

$H_1(ID)$。

2. 在$\mathbb{F}_l$的乘法群中选择一个随机数r。

3. 计算$g_{ID} = e(Q_{ID}, P_{pub})$。

4. 令密文信息为：$c = (rP, M \bigoplus H_2({g_{ID}}^r))$。

Bob解密信息$c = (u, v)$，做如下步骤：

1. 用Bob的私钥D_{ID}计算$h_{ID} = e(D_{ID}, u)$。

2. 计算$m = v \bigoplus H_2(h_{ID})$。

解密过程具体如下：$e(D_{ID}, u) = e(sQ_{ID}, rP) = e(Q_{ID}, P)^{sr} = e(Q_{ID}, P_{pub})^r = {g_{ID}}^r$。

因此，$m = v \bigoplus H_2(e(D_{ID}, u)) = (M \bigoplus H_2({g_{ID}}^r)) \bigoplus H_2({g_{ID}}^r) = M$。

1.2 三方Diffie-Hellman密钥交换

为了叙述的完整性，首先给出两方密钥协商的体制，然后，再给出基于双线性对的一轮三方密钥协商体制。

两方密钥协商体制是由Diffe-Hellman[9]提出，采用了有限域上的乘法群来构造。其主要步骤如下：

Alice和Bob协商一条定义在有限域$\mathbb{F}_q$上的椭圆曲线。同时，Alice和Bob 选定该椭圆曲线上上的一点P，要求由点P生成的子群$< P >$, 具有较大的阶数。

1. Alice选择一个秘密的整数a，并计算$P_a = aP$，发送P_a给Bob。

2. Bob选择一个秘密的整数b，并计算$P_b = bP$，发送P_b给Alice。

3. Alice计算$aP_b = abP$。

4. Bob计算$bP_a = baP$。

5. Alice和Bob使用某种公开的方法，从abP中提取一共享密钥。

一个监听者Eve所能够获得的信息只有Alice和Bob选取的椭圆曲线E，有限域$\mathbb{F}_q$，点P，aP，bP。但是，监听者Eve很难获得abP。如果监听者Eve能够获得abP，则相当于监听者Eve能够解决Computational Diffie-Hellman Problem。

Joux提三方Diffie-Hellman密钥协商[13]。

假设Alice，Bob，Chris想建立一个共享密钥。上述标准的Diffie-Hellman密钥共享协议需要两轮交互。使用约化后的Weil对可使交互的过程一次来完成。

令P为阶为n的超奇异椭圆曲线上的点，通常n被选作大素数。Alice，Bob，Chris，做如下操作：

1. Alice，Bob，Chris相应的选取秘密整数$a \mod n$，$b \mod n$，$c \mod n$。

2. Alice广播aP，Bob广播bP，Chris广播cP。

3. Alice计算$e_n(bP, cP)^a$, Bob计算$e_n(aP, cP)^b$，Chirs计算$e_n(aP, bP)^c$。

4. 因此，三个用户每一方通过计算得到同一数值，并使用预先协商的密钥提取算法提取会话密钥。

1.3 本章小结

本章通过给出两种常见的双线性对在密码协议中的应用，来说明提高双线性对的计算效率对于基于身份的密码协议的实现的重要性。

本论文将在下一章给出计算椭圆曲线上双线性对的所需的数学背景知识。

第二章　预备知识

在此章节，我们给出用于椭圆曲线上双线性对计算所需要的基本背景知识。首先，我们将给出有限域中的相关概念，及有限域中的基本代数运算；然后，介绍椭圆曲线和除子理论；最后，给出计算椭圆曲线上双线性Weil对和双线性Tate对的基本算法。

2.1　有限域

有限域是密码学、编码学和数字通信领域重要的数学工具之一，是代数学领域的一个重要分支。下面，我将给出将用于计算椭圆曲线上双线性对计算的，有限域中的一些基本性质。关于有限域理论更详细的介绍，可参考[1][15]。

有限域是指由有限个元素构成的域，有限域也叫加罗瓦（Galois）域。

有限域的定义如下：

设$\mathbb{F}$是一个非空的集合，并假定$\mathbb{F}$上定义了加法‘+’和乘法‘$*$’两种运算；并要求$\mathbb{F}$中任意两个元素经加法运算和乘法运算的结果仍是$\mathbb{F}$中的元素，即$\mathbb{F}$中任意两个元素a和b的和$a+b$，a和b的积$a*b$，仍是$\mathbb{F}$中的元素，这个性质通常称为$\mathbb{F}$对于在其上定义的加法和乘法运算是封闭的。并且我们要求$\mathbb{F}$对于所规定的加法和乘法运算满足如下规则成立：

1. 对任意$a \in \mathbb{F}, b \in \mathbb{F}$，有$a+b=b+a$ 成立。
2. 对任意$a \in \mathbb{F}, b \in \mathbb{F}$，$c \in \mathbb{F}$，有$(a+b)+c=a+(b+c)$ 成立。
3. $\mathbb{F}$ 有一个元素，把它记作0。对于任意$a \in \mathbb{F}$，有$a+0=a$ 成立。
4. 对于任意$a \in \mathbb{F}$，$\mathbb{F}$ 中有一个元素$-a$，有如下关系成立$a+(-a)=a$ 成立。
5. 对任意$a \in \mathbb{F}, b \in \mathbb{F}$，有$a*b=b*a$ 成立。
6. 对任意$a \in \mathbb{F}, b \in \mathbb{F}$，$c \in \mathbb{F}$，有$(a*b)*c=a*(b*c)$ 成立。
7. $\mathbb{F}$中有一个不等于0的元素，记为e，对任意$a \in \mathbb{F}$，具有性质$a*e=a$ 成立。
8. 对任意$a \in \mathbb{F}$，且$a \neq 0$，存在元素a^{-1}，满足$a*a^{-1}=e$ 成立。
9. 对任意$a \in \mathbb{F}, b \in \mathbb{F}$，$c \in \mathbb{F}$，有$a*(b+c)=a*b+a*c$ 成立。

下面给出有限域中一些重要的结论。但由于篇幅的限制，不给出具体证明。具体证明可参考[15].

1. 有限域的特征必然是素数。

2. 有限域$\mathbb{F}$的元素个数q，必然是某一素数的p的幂指数，即$q = p^m$，其中m为正整数。$\mathbb{F}_q$ 可看作有限域$\mathbb{F}_p$上的m维向量空间，$\mathbb{F}_q$的加法群是m个p阶循环群的直和。

3.有限域$\mathbb{F}$的非零元素的全体组成的群称作有限域$\mathbb{F}$的乘法群，记作$\mathbb{F}^*$。

4. 假设$\mathbb{F}_p$，$\mathbb{F}_{p'}$ 为两个有限域。若$\mathbb{F}_p$为$\mathbb{F}_{p'}$的子域，当且仅当，存在正整数m，使得$p' = p^m$。特别地，如$\mathbb{F}_{p^m}$ 是F_{p^n} 的子域，当且仅当$m|n$。

5. 对于整数$n \geq 1$，$\mathbb{F}_q[n]$ 中必然存在n次不可约多项式$g(x)$，并且对于$g(x)$在$\mathbb{F}_q$ 的代数闭包中的任一个根α，有$\mathbb{F}_q[\alpha] = \mathbb{F}_q(\alpha) = \mathbb{F}_{q^n}$。

有限扩域的具体构造

选取素域$\mathbb{F}_p$上的某个m次不可约多项式$f(x)$的根α，将α添加到素域$\mathbb{F}_p$中，即得到扩域$\mathbb{F}_{p^m} = \mathbb{F}_p(\alpha) \cong \mathbb{F}_p[x]/(f(x))$ 。有限扩域的构造，难点在于如何选取素域上的不可约多项式，可参考[1][16]。

2.2 椭圆曲线

数学界对椭圆曲线的研究已有200多年的历史。然而，真正将椭圆曲线应用到密码学领域，却是从20世纪80年代开始的。椭圆曲线最初被用作解决整数分解问题和素数检测问题。在20世纪80-90年代，椭圆曲线在证明费马大定理方面，起了重要作用。下面将给出椭圆曲线一些基本知识，可参考[28][29][30]。

2.2.1 定义在有限域$\mathbb{F}$上的Weierstrass方程

设$\mathbb{F}$为有限域，形如$y^2 = x^3 + Ax + B$的方程，其中，x，y，A，$B \in \mathbb{F}$，A，B为常数，被称作定义在有限域$\mathbb{F}$上的Weierstrass方程。该方程描述了一类常见的椭圆曲线。

如果，我们想描述那些属于域$\mathbb{F}$且包含于E的椭圆曲线上的点，记作$E(\mathbb{F})$。根据定义，这个集合包括点$\mathcal{O}_E$，这个点的定义将在后面给出。

$E(\mathbb{F})$=$\{\mathcal{O}_E\} \cup \{(x, y) \in \mathbb{F} \times \mathbb{F} \mid y^2 = x^3 + Ax + B, A, B \in \mathbb{F}\}$。

设r_1，r_2，r_3 为$Weierstrass$方程的三个根，由三次方程的判别式：$((r_1-r_2)(r_1-r_3)(r_2-r_3))^2 = -4A^3+27B^2$ 知；若定义在有限域$\mathbb{F}$上的Weistrass方程无根，则要求$-4A^3+27B^2 \neq 0$。

2.2.2 定义在有限域$\mathbb{F}$上的Generalized Weierstrass方程

形如$y^2+a_1xy+a_3y = x^3+a_2x^2+a_4x+a_6$的方程，其中$a_1 \ldots a_6$均为常数，$x$，$y$，$a_1 \ldots a_6 \in \mathbb{F}$，被称作定义在有限域$\mathbb{F}$上的Generalized Weierstrass方程。

如果，$\mathrm{Char}(\mathbb{F}) \neq 2$，方程可转化为$(y+a_1x/2+a_3/2)^2 = x^3+(a_2+a_1^2/4)x^2+(a_4+a_1a_3/2)x+(a_3^2+a_6)$。若记，$y_1 = y+a_1x/2+a_3/2$，$a_2' = (a_2+a_1^2/4$，$a_4' = a_4+a_1a_3/2$，$a_6' = a_3^2+a_6$。原方程可以转化为：$y_1^2 = x^3+a_2'x^2+a_4'x+a_6'$。

如果，$\mathrm{Char}(\mathbb{F}) \neq 3$，可令$x_1 = x+a_2'/3$，方程可转化为，$y_1^2 = x_1^3+Ax_1+B$。其中，$A = a_4'-a_2'^2/3$，$B = a_6'-a_2'^3/27$。

2.2.3 椭圆曲线上有理点加法

设椭圆曲线$E: y^2 = x^3+Ax+B$为定义在有限域$\mathbb{F}$上，其中$\mathrm{Char}(\mathbb{F}) = p$，$A$，$B \in \mathbb{F}$，并且$4A^3+27B^2 \neq 0$。以$E(\mathbb{F})$表示曲线在F上的所有的点构成的集合。

$$E(\mathbb{F}) = \{(x,y) \in \mathbb{F} \times \mathbb{F} \mid y^2 = x^3+Ax+B\} \cup \{\mathcal{O}_E\}.$$

这是一个有限集。可以按照切割线法则，定义两点间的加法运算。可以证明，$\langle E(\mathbb{F}), + \rangle$ 构成一个加法群。具体证明可参考[30]。下面给出群律。

设$E: y^2 = x^3+Ax+B$为一条定义在有限域$\mathbb{F}$上的椭圆曲线。令$P_1 = (x_1, y_1)$，$P_2 = (x_2, y_2)$为椭圆曲线E上的点，且P_1, $P_2 \neq \mathcal{O}_E$。定义$P_1+P_2 = P_3 = (x_3, y_3)$，如下：

1. 如果$x_1 \neq x_2$，则，$x_3 = m^2-x_1-x_2y_3 = m(x_1-x_3)-y_1$, 其中，$m = (y_2-y_1)/(x_2-x_1)$.

2. 如果$x_1 = x_2$，但，$y_1 \neq y_2$，则，$P_1+P_2 = \mathcal{O}_E$。

3. 如果$P_1 = P_2$，且$y_1 \neq 0$，则，$x_3 = m^2-2x_1$，$y_3 = m(x_1-x_3)-y_1$, 其中$m = (3x_1^2+A)/2y_1$。

4. 如果$P_1 = P_2$，且$y_1 = 0$，则，$P_1+P_2 = \mathcal{O}_E$。

5. 定义$\mathcal{O}_E$为椭圆曲线外一点，对所有椭圆曲线E上的点P，有$P+\mathcal{O}_E = P$成立。

椭圆曲线E上的点的加法满足如下的性质：

1. （交换律）对于椭圆曲线E上的所有点P_1，P_2，有$P_1 + P_2 = P_2 + P_1$。

2. （存在单位元）对于椭圆曲线E上的所有点P有，$P + \mathcal{O}_E = P$。

3. （存在逆元）给定椭圆曲线E上的点P，在E上存在点P'，满足$P + P' = \mathcal{O}_E$。这个点通常被记作$-P$。

4. （结合律）$(P_1 + P_2) + P_3 = P_1 + (P_2 + P_3)$。

2.2.4 扭映射

设ϕ是指一个从$E(\mathbb{F}_q)$到$E(\mathbb{F}_{q^k})$的同态映射。$\phi : E(\mathbb{F}_q) \to E(\mathbb{F}_{q^k})$。该映射将$E(\mathbb{F}_q)$中的点映射到$E(\mathbb{F}_{q^k})$中的某一点，且这两点线性无关[4][5]。

2.2.5 挠点

令椭圆曲线E定义在有限域$\mathbb{F}$上，n为一正整数。椭圆曲线上的挠点的定义：$E[n] = \{P \in E(\overline{\mathbb{F}}) \mid nP = \mathcal{O}_E\}$。所要强调的是，$E[n]$不只是包含坐标在F中的点，同样包括坐标在$\overline{\mathbb{F}}$ 中的点。

2.2.6 嵌入次数

假设椭圆曲线E定义在有限域$\mathbb{F}$上，其中，有限域$\mathbb{F}$的阶为q，椭圆曲线上的有理点群$\langle E(\mathbb{F}), + \rangle$的阶为$n$，存在最小的整数$k$，满足$n|q^k - 1$，那么，该整数$k$被称作该椭圆曲线的嵌入次数。

2.3 除子理论

令E为定义在有限域$\mathbb{F}$上的椭圆曲线。对于每一点$P \in E(\mathbb{F})$，定义形式符号$[P]$。椭圆曲线E上的除子D是以整数为系数的这类形式符号的有限线性组合。其形式如下：

$D = \sum a_j[P_j], a_j \in \mathbb{Z}$。

显然，除子是由形式符号$[P]$所生成的自由阿贝尔群中的一个元素。除子群记作Div(E)。

除子的度(Degree)和和(Sum)记作:

$\mathrm{Deg}(\sum a_j[P_j]) = \sum a_j$；

$\mathrm{Sum}(\sum a_j[P_j]) = \sum a_j P_j$。

度为0的除子形成$\mathrm{Div}(E)$的一个重要子群$\mathrm{Div}^0(E)$。

函数$\mathrm{Sum} : \mathrm{Div}^0(E) \to E(\mathbb{F})$定义了一个满同态映射。其中，满射的原因如下：$\mathrm{Sum}([P] - [\mathcal{O}_E]) = P$。

2.4 椭圆曲线上的有理函数

椭圆曲线上的函数是指有理函数$f(x,y)$至少在$E(\mathbb{F})$上有定义。显然，函数$1/(y^2 - x^3 + Ax + B)$，在椭圆曲线$E : y^2 = x^3 + Ax + B$上的任何一点都没有定义，因此，该函数在椭圆曲线$E : y^2 = x^3 + Ax + B$上无定义。如果一个函数$f(x,y)$。在椭圆曲线上的某点P的函数值为0，则称该点为函数f的零点；如果一个函数$f(x,y)$在椭圆曲线上的某点P的函数值为∞，则称该点为f的极值点。令P为椭圆曲线上的某一点，可以证明存在函数u_p，满足$u_p(p) = 0$ 且$f = u_p^r g$, 满足r属于$\mathbb{Z}$且$g(P) \neq 0, \infty$，则函数u_p被称作函数$f(x,y)$在点P的一致化子。

函数f在点P的价，记作$Ord_p(f) = r$。

如果f是E上的一个不恒等于0的函数，定义f的除子为：$div(f) = \sum ord_p(f)[P]$，其中$P \in E(\mathbb{F})$。

令E为一条椭圆曲线，f为椭圆曲线E上到函数，且f不恒等于0，则

1. f有有限个零点和极值点。

2. Deg $(div\ (f))= 0$。

3. 如果f既没有零点，也没有极值点，则f为常数。

令E为椭圆曲线。D为定义在E上的除子，且满足$deg(D) = 0$。那么在E上存在一个函数f，满足$div(f) = D$，当且仅当，$\mathrm{Sum}(D) = \mathcal{O}_E$。

设f为椭圆曲线E上的一个函数，假设除子$D = \sum a_j[P_j]$，满足$Deg(D) = 0$。f在除子D的赋值为：$f(D) = \prod f(P_j)^{a_j}$。

2.5 双线性对

从密码学协议设计的角度来说，协议设计者不需要了解椭圆曲线上双线性对的具体构造，只需了解其抽象定义。因此，此处先给出双线性对的抽象定义，然后再给出两类基于椭圆曲线有理点群构造的双线性对定义，双线性Weil对和双线性Tate对。

假设G_1和G_2是两个加法群，G_T是乘法群。如果存在映射$e: G_1 \times G_2 \rightarrow G_T$，并且对任意$P_1$，$P_2 \in G_1$，$Q_1$，$Q_2 \in G_2$，满足$e(P_1+P_2, Q_1) = e(P_1, Q_1)e(P_2, Q_1)$ 和$e(P_1, Q_1+Q_2) = e(P_1, Q_1)e(P_1, Q_2)$成立，则$e$被称作从$G_1 \times G_2$到$G_T$的一个双线性对。

在实际的密码应用中，还要求双线性对e满足如下性质：

非退化性：一定存在$P \in G_1$和$Q \in G_2$，使得$e(P, Q) \neq 1_{G_T}$，其中1_{G_T}为G_T中的单位元。

可计算性：存在有效的多项式时间算法可以计算出双线性对e的值。

由双线性对的定义可知如下性质：

假设$P \in G_1$和$Q \in G_2$，0_{G_1}和0_{G_2}分别表示G_1和G_2中的单位元，则有：

$e(P, 0_{G_2}) = e(0_{G_1}, Q) = 1_{G_T}$；

$e(-P, Q) = e(P, Q)^{(-1)} = e(P, -Q)$；

$e(jP, Q) = e(P, Q)^j = e(P, jQ)$。

2.5.1 基于椭圆曲线有理点群构造的双线性Weil对和Tate对

1. 双线性Weil对

令E为定义在有限域$\mathbb{F}_q$上的椭圆曲线，其中$Char(\mathbb{F}_q) = p$。设n为一正整数，且$p \nmid n$，存在$\mathbb{F}_q$的有限次扩域$\mathbb{F}_{q^k}$，使得$E[n] \in E(\mathbb{F}_{q^k})$。由$E[n] \in E(\mathbb{F}_{q^k})$，可以推出，$\mu_n \in \mathbb{F}_{q^k}$，其中$\mu_n$为n次单位根群。双线性Weil[19][21]对的定义如下：$e_n: E[n] \times E[n] \rightarrow \mu_n$。

双线性Weil对除了满足双线性性，还满足如下性质：

1. $e_n(T, T) = 1$，对任意$T \in E[n]$成立。

2. $e_n(P, Q) = e_n(Q, P)^{(-1)}$，对任意$P$，$Q \in E[n]$成立。

3. $e_n(\phi(P), \phi(Q)) = \phi(e_n(P, Q))$，其中$\phi$为$\overline{\mathbb{F}}_q$上的自同态映射。

4. 对于E上的所有可分自同态映射α，必有$e_n(\alpha(P),\alpha(Q)) = e_n(P,Q)^{deg(\alpha)}$。

2. **双线性Tate对**

令E为定义在有限域$\mathbb{F}_q$上的椭圆曲线，n为一整数且满足$n \mid (q-1)$。

令$E(\mathbb{F}_q)[n]$表示$E(\mathbb{F}_q)$中的阶整除n的元素的集合，$\mu_n = \{x \in \mathbb{F}_q \mid x^n = 1\}$。假设$E(\mathbb{F}_q)$中包含阶为$n$的元素，则存在非退化的双线性Tate对[11]。$< \cdot,\cdot >_n$: $E(\mathbb{F}_q)[n] \times E(\mathbb{F}_q)/nE(\mathbb{F}_q) \to {\mathbb{F}_q}^*/({\mathbb{F}_q}^*)^n$和$\tau_n : E(\mathbb{F}_q)[n] \times E(\mathbb{F}_q)/nE(\mathbb{F}_q) \to \mu_n$。其中，第一种pairing被称作Tate-Lichtenhaum Pairng，第二种pairing被称作约化的Tate-Lichtenhaum Pairng。显然，Tate-Lichtenhaum Pairng将$P \in E(\mathbb{F}_q)[n]$，$Q \in E(\mathbb{F}_q)/nE(\mathbb{F}_q)$映射到一陪集，既计算$\langle P,Q\rangle_n$的结果可以得到不同的数值，但这些数值均属于同一陪集。而计算经约化后的Tate-Lichtenhaum Pairng将得到集合$\mu_n = \{x \in \mathbb{F}_q | x^n = 1\}$中一个确定的数值。

2.5.2 Miller 算法

由双线性Weil对和双线性Tate对的定义可知，计算双线性Weil对和双线性Tate对的关键在于，首先，给出有理函数的确定形式，然后在将有理点代入进行计算。

首先，引用[30]中一例来说明如何计算双线性Weil对的。然后，在给出计算双线性对的Miller算法。

令$E : y^2 = x^3 + 2$为定义在有限域$\mathbb{F}_7$上的椭圆曲线。则有，$E(\mathbb{F}_7)[3] = Z_3 \bigoplus Z_3$，计算$e_3((0,3),(5,1))$。

令$D_{(0,3)} = [(0,3)] - [\mathcal{O}_E]$，$D_{(5,1)} = [(3,6)] - [(6,1)]$。其中，除子$D_{(5,1)}$，根据$(3,6) = (5,1) + (6,1)$得来。$div(y-3) = 3D_{(0,3)}$，$div((4x-y+1)/(5x-y-1)) = 3D_{f(5,1)}$。因此，取$f_{(0,3)} = y - 3$，$f_{(5,1)} = (4x-y+1)/(5x-y-1)$。则有，$f_{(0,3)}(D_{(5,1)}) = (f_{(0,3)}(3,6))/(f_{(5,1)}(6,1)) = (6-3)/(1-3) = 2 \mod 7$。类似有，$f_{(5,1)}(D_{(0,3)}) = 4$。因此，$e_3((0,3),(5,1)) = 4/2 = 2 \mod 7$。

1986年，Miller在其一篇手稿中[20]，第一次给出了计算双线性Weil对的多项式时间算法。其主要思想如下：

计算双线性Weil对，可归约为寻找函数f，满足$div(f) = n[P+R] - n[R]$，其中点$P \in E[n]$，点$R \in E$。对应点Q_1，Q_2计算$f(Q_1)/f(Q_2)$。引入除子$D_j = j[P+R] - j[P] - [jR] + [\mathcal{O}_E]$。显然，除子$D_j$是一个函数的除子。则存在函数$f_j$，满足$div(f_j) = D_j$。

假设，我们已经求得$f_j(Q_1)/f_j(Q_2)$和$f_k(Q_1)/f_k(Q_2)$。我们将给出如何计算$f_{(j+k)}(Q_1)/f_{(j+k)}(Q_2)$。

令$ax+by+c=0$为经过jP和kP的直线，令$x+d=0$为经过点$(j+k)P$的垂线。

由计算知：

$div(ax+by+c/x+d)=[jP]+[kP]-[(j+k)P]-[\mathcal{O}_E]$。

因此，

$div(f_{j+k})=D_{j+k}=D_j+D_k+div(ax+by+c/x+d)=div(f_jf_k(ax+by+c/x+d))$。

上式等价于，存在常数c，使得，$f_{j+k}=cf_jf_k(ax+by+c/x+d)$。

因此，

$$f_{j+k}(Q_1)/f_{j+k}(Q_2)=(f_j(Q_1)f_k(Q_1)(ax+by+c)/(x+d)|(x,y)=$$
$$Q_1)/(f_j(Q_2)f_k(Q_2)(ax+by+c)/(x+d)|(x,y)=Q_2)。$$

下面给出原始的Miller算法。

Algorithm 1 原始的Miller算法

输入: 素数p, 椭圆曲线$E/\mathbb{F}_p$及嵌入次数k. 素数$r=\sum_{i=0}^{l}r_i2^i$, $P\in E(\mathbb{F}_p)[r]$, $Q\in E(\mathbb{F}_{p^k})[r]$.

P与Q线性无关. $D_Q=[Q_1]-[Q_2]$.

输出: 双线性Tate对$e(P,Q)$

[1.] $R\leftarrow P$, $f\leftarrow 1$

[2.]For i from $l-1$ downto 0

[2.1] $f\leftarrow f^2\frac{l_{R,R}(Q_1)v_{2R}(Q_2)}{l_{R,R}(Q_2)v_{2R}(Q_1)}$. $R\leftarrow 2R$.

[2.2]If $r_i=1$

[2.3] $f\leftarrow f\frac{l_{R,P}(Q_1)v_{R+P}(Q_2)}{l_{R,P}(Q_2)v_{R+P}(Q_1)}$. $R\leftarrow R+P$.

[3.]Return $f^{\frac{q^k-1}{r}}$.

2.6 椭圆曲线上倍点和点加运算

2.6.1 仿射坐标下的倍点和点加运算

首先，记有限域$\mathbb{F}_p$上，其中p为素数，一次乘法运算的运算量为1M，一次平方运算的

运算量为1S，一次求逆运算的运算量为1I。

令$P=(x_1,y_1)$, $Q=(x_2,y_2)$, $R=(x_3,y_3)=P+Q$。如果$P\neq Q$，在仿射坐标下，关于椭圆曲线上的点加运算我们有如下公式：

$\lambda=\frac{y_2-y_1}{x_2-x_1}$ 1M + 1I

$x_3=\lambda^2-x_1-x_2$ 1S

$y_3=(x_1-x_3)\lambda-y_1$ 1M

因此，在仿射坐标下，完成点加运算需要2M+1S+1I次操作。

类似，当$P=Q$，关于椭圆曲线上倍点运算，我们有如下公式：

$\lambda=\frac{3x_1^2+a}{2y_1}$ 1M+1S+1I

$x_3=\lambda^2-2x_1$ 1S

$y_3=(x_1-x_3)\lambda-y_1$ 1M

因此，在仿射坐标下，完成倍点运算需要2M + 2S + 1I次操作。

2.6.2 雅克比坐标下的倍点和点加运算

首先，在雅克比坐标下，令$x=\frac{X}{Z^2}$，$y=\frac{Y}{Z^3}$，相应的椭圆曲线是：$Y^2=X^3+aXZ^4+bZ^6$。

令$P=(X_1,Y_1,Z_1)$，$Q=(X_2,Y_2,Z_2)$，$R=(X_3,Y_3,Z_3)$。如果$P\neq Q$，在雅克比坐标下，关于椭圆曲线上的倍点和点加运算，我们有如下算法[8]：

$\lambda_1=X_1{Z_2}^2$, 1M + 1S

$\lambda_2=X_2{Z_1}^2$, 1M + 1S

$\lambda_3=\lambda_1-\lambda_2$,

$\lambda_4=Y_1{Z_2}^3$, 2M

$\lambda_5=Y_2{Z_1}^3$, 2M

$\lambda_6=\lambda_4-\lambda_5$,

$\lambda_7=\lambda_1+\lambda_2$,

$\lambda_8=\lambda_4+\lambda_5$,

$Z_3=Z_1Z_2\lambda_3$, 2M

$X_3 = {\lambda_6}^2 - \lambda_7{\lambda_3}^2 \quad 1\mathrm{M} + 2\mathrm{S}$

$\lambda_9 = \lambda_7{\lambda_3}^2 - 2X_3$

$Y_3 = (\lambda_9\lambda_6 - \lambda_8\lambda_3^3)/2 \quad 3\mathrm{M}.$

因此，在雅克比坐标下，完成点击点加运算，所需操作为12M+4S。如果$Z_2 = 1$，所需操作为8M+3S。

如果$P = Q$，我们可利用如下公式完成倍点运算：

${\lambda_1} = 3{X_1}^2 + a{Z_1}^2 \quad 1\mathrm{M}+3\mathrm{S}$

$Z_3 = 2Y_1Z_1 \quad 1\mathrm{M}$

$\lambda_2 = 4X_1Y_12 \quad 1\mathrm{M}+1\mathrm{S}$

$X_3 = {\lambda_1}^2 - 2\lambda_2 \quad 1\mathrm{S}$

$\lambda_3 = 8{Y_3}^4 \quad 1\mathrm{S}$

$Y_3 = \lambda_1(\lambda_2 - X_3) - \lambda_3 \quad 1\mathrm{M}$

在雅克比坐标下，完成倍点运算所需操作为4M+6S。如果$a \equiv -3 \pmod p$，那么$\lambda_1 = 3(X_1 - Z_1^2)(X_1 + Z_1^2)$。计算$\lambda_1$仅需要1M+1S。此时，完成倍点运算所需操作为4M+4S。

2.7 本章小结

在本章给出了计算椭圆曲线上双线性对的数学背景知识，原始的Miller算法和仿射坐标、雅克比坐标下，点加和倍点运算在在基域上的复杂度。这些背景知识，为下一章的论述做了必要的铺垫。

在下一章中，将给出一些改进的Miller算法，以及两类椭圆曲线上双线性Tate对在在仿射坐标和雅克比坐标下计算效率的比较。

第三章　Tate对的计算

在此章节，首先，将从不同角度给出了加快Tate对计算的方法。其次，在雅可比坐标下，给出了计算两类非超奇异椭圆曲线上的双线性Tate对，Miller算法复杂度的估计值。最后，给出了有限域F_p上，求逆运算和乘法及平方运算的计算效率的测试结果和分析，用来支持给出上述估计值时用到的假设。

3.1　二次扩域上的运算

3.1.1　二次扩域上的基本运算

首先，我们来构造合适的不可约多项式，使其度数和我们选择的扩域的度数相等，并且在在基域上无因子。例如：考虑$k=2$的情形。在这种情况下，如果$p=3 \mod 4$，我们可以选择多项式x^2+1作为不可约多项式。因为模p下，-1是二次非剩余。因此，多项式x^2+1在基域$\mathbb{F}_p$上无因子。现在，有限域$\mathbb{F}_{p^2}$上的元素可以表示形如：$a+bx$的多项式，其中a，$b \in \mathbb{F}_p$。这样的元素可以像正常的元素一样进行乘法，然后在进行模x^2+1运算，即积中出现的x^2项可以用-1代替。注：x可以用不可约多项式的虚数根来替代，则扩域中的元素可以表示成$a+ib$的形式，其中i是-1的虚数平方跟根。这样，扩域中的加法、减法的算法看起来很明显。

3.1.2　二次扩域上的乘法和平方运算

1.　二次扩域上的乘法运算

有限扩域上的初等的乘法如下：$(a+ib)(c+id)=ac-bd+i(bc+ad)$。很明显，这个运算的完成，需要四次基域上的乘法模运算。

但是，采用如下方式：$(a+ib)(c+id)=ac-bd+i[(a+b)(c+d)-ac-bd]$。则只需要三次基域上的乘法模运算。

2.　二次扩域上的平方运算

我们知道，复数域上的算术运算中的平方运算如下：

$(a+ib)(a+ib)=(a+b)(a-b)-i2ab$，其中需要两次基域上的乘法模运算。

假设不可约多项式形如：x^3+n，考虑计算$(a+bx+cx^2)^2$。首先，可以预先计算$A=a^2$，$B=2bc$，$C=c^2$，$D=(a-b+c)^2$，$E=(a+b+c)^2$，然后计算$(a+bx+cx^2)^2=(A-Bn)+((E-D)/2-B-nC)x+(E-A-C-(E-D)/2)x^2$，其只需要四次模平方运算，一次模乘运算。

3.2 Miller算法的优化

下面将从不同方面给出优化Miller算法，从而加快椭圆曲线上双线性Tate对的的可能性。

3.2.1 椭圆曲线上的倍点运算

由上一章节给出的，原始Miiler算法，可以看出，椭圆曲线上的倍点运算，影响着整个Miller算法的计算效率。因此，提高椭圆曲线上倍点运算的计算效率，显然能够提高椭圆曲线上双线性Tate对的计算效率。下面将给出，两种常用的用来优化椭圆曲线上倍点运算的算法–倍点运算的二进制表示法和倍点运算的NAF算法。

1. 倍点运算的二进制法

二进制法是计算倍点运算的最常用方法，其具体算法如下：

Algorithm 2 倍点运算的二进制法

输入: 椭圆曲线上的点P，及二进制整数$k=\sum_{i=0}^{l-1}k_i2^i$，其中$k_i\in\{0,1\}$。

输出: 椭圆曲线上的点$Q=[k]P$.

[1.] Let $Q=P$

[2.]For i from $l-1$ downto 0

[2.1] $Q=2Q$

[2.2]If $k_i=1$, compute $Q=Q+P$

[3] **return** Q

2. **倍点运算的NAF法**

在椭圆曲线上有理点的计算中，点的减法运算和点的加法运算的计算效率相当。如果，基域的特征大于2，$P=(x,y)$为椭圆曲线上一点，那么$-P=(x,-y)$。如果，基域的特征为2，那么$-P=(x,x+y)$。

设整数k在二进制表示下的长度为l比特，那么k可以写成如下形式：

$\sum_{i=0}^{l} s_i 2^i$，其中$s_i \in \{-1,0,1\}$。

这种对整数的表示形式被称作带符号的二进制表示法。由于，对于同一整数k，存在3^l种表示方法，因此，对于整数k的表示形式不唯一。例如：$7=(0111)_2=(100\bar{1})_2$，其中$\bar{1}=-1$。但是，如果要求表达式中，没有任何两个非零的值相邻，也就是说对于任意的$i\geq 0$，有$s_j s_{j+1}=0$，则这种表达式被成为NAF(Non-Adjacent Form)。下面的定理表明，使用NAF表达式，能够减少椭圆曲线上倍点运算的运算量。

每一个整数k有唯一的NAF表达式；在k的所有的带符号的二进制表示法中，NAF表达式的Hamming Weight最小；NAF表达式的比特长度比二进制的比特长度最多长一个比特。

具体证明，可参考[2]。

可以证明的是，整数的NAF表达式的平均Hamming Weight为$\frac{1}{3}\log k$。因此，在NAF算法中，需要$\lfloor \log k \rfloor - 1$个倍点运算和平均$\frac{1}{3}\log k$个点加运算。（此处，将下面给出的倍点运算的NAF算法中的加法运算和减法运算的计算效率堪称是一致的。）因此，共需要$\lfloor \log k \rfloor - 1 + \frac{1}{3}\log k$个椭圆曲线上的有理点加法或减法运算。

但已经在上述定理中给出，NAF表达式的Hamming Weight最小；NAF表达式的比特长度比二进制的比特长度最多长一个比特。因此，在平均情况下，倍点的运算的NAF算法要好过倍点运算的二进制法。

下面给出利用NAF计算椭圆曲线上有理点倍点的具体算法，见Algorithm 3。

下面给出引入NAF，改进后的Miller算法，见Algorithm 4。

3.2.2 去分母法

在给出引入去分母法的Miller算法前，首先给出采取这种方法的可行性证明。

Algorithm 3 倍点运算的NAF算法

输入: 椭圆曲线上的点P，及整数的NAF表示：$k=\sum_{i=0}^{l-1} s_i 2^i$，其中$s_i \in \{-1,0,1\}$。

输出: 椭圆曲线上的点$Q=[k]P$.

[1.] Let $Q=P$

[2] For i from $l-1$ downto 0

[2.1] $Q=2Q$

[2.2]If $s_i=1$, compute $Q=Q+P$;

[2.3] If $s_i=-1$, compute $Q=Q-P$

[3.]Return Q

Algorithm 4 改进后的Miller算法（引入NAF）

输入: 素数p, 椭圆曲线$E/\mathbb{F}_p$及嵌入次数k. 素数$r=\sum_{i=0}^{l} r_i 2^i$, $r_i \in \{-1,0,1\}$, $P \in E(\mathbb{F}_p)[r]$, $Q \in E(\mathbb{F}_{p^k})[r]$. P与Q线性无关. $D_Q=[Q_1]-[Q_2]$.

输出: 双线性Tate对$e(P,Q)$

[1.] $R \leftarrow P$, $f \leftarrow 1$

[2.]For i from $l-1$ downto 0

[2.1] $f \leftarrow f^2 \frac{l_{R,R}(Q_1)v_{2R}(Q_2)}{l_{R,R}(Q_2)v_{2R}(Q_1)}$. $R \leftarrow 2R$.

[2.2]If $r_i=1$

[2.3] $f \leftarrow f \frac{l_{R,P}(Q_1)v_{R+P}(Q_2)}{l_{R,P}(Q_2)v_{R+P}(Q_1)}$. $R \leftarrow R+P$.

[2.4]If $r_i=-1$

[2.5] $f \leftarrow f \frac{l_{R,P}(Q_1)v_{R+P}(Q_2)}{l_{R,P}(Q_2)v_{R+P}(Q_1)}$. $R \leftarrow R-P$.

[3.]Return $f^{\frac{q^k-1}{r}}$.

引理：q^d-1 是q^k-1 的因子，其中k为嵌入度数。

证明：由割圆多项式的性质知：$q^k-1=(q^d-1)\sum_{i=0}^{k/d-1} q^{id}$。因为，嵌入度数$k>1$，显然有$r|(q^k-1)$，且$r \nmid q^d-1$。因此，$r \mid \sum_{i=0}^{k/d-1} q^{id}$。那么，$q^d-1$ 是$(q^k)-1$ 的因子。

推论：对$f(D)$乘以任何非零元素$x \in F_{q^d}$，并不影响双线性对的计算结果。

证明：为了完成双线性对的计算，在最后一步模指数运算中，需要计算$f(D)$的$(q^k-$

1)/r次幂。由上面的引理知，该指数包含因子q^d-1。由费马小定理知，在有限域F_{q^d}上，$x^{(q^k-1)/r}=1$。

定理：令$P \in E(F_q)[r]$，$Q \in E(\mathbb{F}_{q^k})$是线性无关的两个点。则，$e(P,Q)=f(Q)^{(q^k-1)/r}$。

证明：假设$R \in \{\mathcal{O}_E, -P, Q, Q-P\}$为椭圆曲线上的一点。令$f'$为一除子，满足$(f')=r(P+R)-r(R)$，满足$e(P,Q)=f'((Q)-(\mathcal{O}_E))^{(q^k-1)/r}$。由于$f'$在$\mathcal{O}_E$既没有零点也没有极点。则我们有$f'((Q)-(\mathcal{O}_E))=f(Q)/f(\mathcal{O}_E)$。由于点$P$的坐标在$\mathbb{F}_q$中，我们有$f'(\mathcal{O}_E) \in \mathbb{F}_q^*$。由上面的推论知，$f'(\mathcal{O}_E)$在Tate对计算的过程中没有意义。所以可以在Tate对的计算过程中忽略关于$f'(\mathcal{O}_E)$的计算。因此，我们有$e(P,Q)=f'(Q)^{q^k-1}$。我们知道，由于$(P+R)-(R) \sim (P)-(\mathcal{O}_E)$，那么$(f')=r((P+R)-(R))=r((P)-(\mathcal{O}_E)-(g))$，其中，$g$为椭圆曲线上有理函数，满足$Deg(g)=0$，且$Sum(g)=0$。因此，则有$f'=fg^r$。由于点$Q$既不是$f'$的零点和极值点，也不是$f$的零点和极值点。其中，$g(Q) \in \mathbb{F}_{p^k}$。再次使用有限域上的费马小定理，则有$f'(Q)^{(q^k-1)/r}=f(Q)^{(q^k-1)/r}g(Q)^{q^k-1}=f(Q)^{(q^k-1)/r}$。

由上面的定理可知，在双线性Tate对是实际计算过程中可以省略在扩域F_{p^k}上的求逆运算。从而的提高了椭圆曲线上双线性对的计算速度。因为，同乘法运算相比，即使采用一定技巧的求逆运算也是相对乘法运算来说是更加费时的。

下面给出，引入去分母法的，改进后的Miller算法Algorithm 5。

Algorithm 5 改进的Miller算法（引入去分母法）

输入: 素数p, 椭圆曲线$E/\mathbb{F}_p$及嵌入次数k. 素数$r=\sum_{i=0}^{l} r_i 2^i$, $P \in E(\mathbb{F}_p)[r]$, $Q \in E(\mathbb{F}_{p^k})$. P与Q线性无关.

输出: 双线性Tate对$e(P,Q)$

[1.] $R \leftarrow P$, $f \leftarrow 1$.

[2.]For i from $l-1$ downto 0

[2.1] $f \leftarrow f^2 \frac{l_{R,R}(Q)}{v_{2R}(Q)}$. $R \leftarrow 2R$.

[2.2]If $r_i=1$

[2.3] $f \leftarrow f \frac{l_{R,P}(Q)}{v_{R+P}(Q)}$. $R \leftarrow R+P$.

[3.]Return $f^{\frac{q^k-1}{r}}$.

3.2.3 点P和点Q的选择

如果限制P，Q在一些特殊的子群中，那么则有可能进一步提高双线性对的计算效率。如果选取$P \in E(\mathbb{F}_q)[r]$，显然有，点P的所有坐标在基域$\mathbb{F}_p$上。那么，关于点P任何标量乘的结果依然在基域$\mathbb{F}_p$上。若令，点Q的x坐标在基域$\mathbb{F}_p$上，y坐标在扩域$\mathbb{F}_{p^k}$中。若按上述方式选择点Q，则可使得$v_{2T}(Q)$和$v_{T+P}(Q)$的结果在基域$\mathbb{F}_p$上。在Miller算法的最后一步模幂运算中，由费马小定理知，$v_{2T}(Q)^{\frac{q^k-1}{r}} = 0$，$v_{T+P}(Q)^{\frac{q^k-1}{r}} = 0$。这就意谓着，通过技巧性的选择点$P$和点$Q$，可以避免在扩域中的除法运算，从而加快椭圆曲线上的双线性对的计算效率。

接下来，我们就来详细讨论这个问题，并给出一系列关于Q点合理的选择。

通常情况下，我们选择Q点为在有限域$\mathbb{F}_{q^k}$上的椭圆曲线上的一点，记作(x_Q, y_Q)，其中$x_Q = a + ib$，$y_Q = c + id$，a，b，c，$d \in \mathbb{F}_{q^k}$。现在，我们来约束Q点的形式，令$b = c = 0$。显然，此时，$\overline{v}_{2T}(Q)$和$\overline{v}_{T+P}(Q)$为有限域$\mathbb{F}_{p^d}$中的元素。这就意味着，$\overline{v}_{2T}(Q)$和$\overline{v}_{T+P}(Q)$的值，可以在最后一轮模指数运算中可以消掉。这就是上述提到的著名的"去分母"优化法。另外，可令$Q = (a, id)$为$E(\mathbb{F}_{p^k})$中的一点。那么，该点就可以被映射到二次扭曲线$E'(\mathbb{F}_{p^d})$上的有理点所构成的同构群上。因此，在双线性对计算之前，可以把点Q从基域$\mathbb{F}_{p^d}$映射到扩域$\mathbb{F}_{p^k}$上[26]。

令$even(\mathbb{F}_{q^k})$表示有限域$\mathbb{F}_{q^k}$的子集合，该集合由所有指数为偶数的一元多项式组成。即，$even(\mathbb{F}_{q^k}) = \{u \in \mathbb{F}_{q^k} : u(x) = a_{k-2}x^{k-2} + a_{k-4}x^{k-4} + \cdots + a_0\}$。

类似，令$odd(\mathbb{F}_{q^k})$表示有限域$\mathbb{F}_{q^k}$的子集合，该集合由所有指数为奇数的一元多项式组成。即，$odd(\mathbb{F}_{q^k}) = \{u \in \mathbb{F}_{q^k} : u(x) = a_{k-1}x^{k-1} + a_{k-3}x^{k-3} + \cdots + a_1\}$。显然，可以构造不可约多项式$R(x) = x^k + x^2 + \omega$，其中，$\omega \in \mathbb{F}_q$。那么，扩域$\mathbb{F}_{p^k}$可以表示为$\mathbb{F}_p[x]/R_k(x)$。

那么，如果$R(x) = x^k + x^2 + \omega$是$\mathbb{F}_q$上的不可约多项式，那么$r(x) = x^{k/2} + x + \omega$也是$\mathbb{F}_q$上的不可约多项式。

反证，若$r(x)$是$\mathbb{F}_q$上的可约多项式，则$r(x) = f(x)g(x)$，f，$g \in \mathbb{F}_q[x]$。则$R(x) = r(x^2) = f(x^2)g(x^2)$。与$R(x)$是不可约多项式的假设条件相矛盾。

因此，可以由上述结论建立如下映射关系：$\psi : \mathbb{F}_q[x]/r(x) \to \mathbb{F}_q[x]/R(x)$。显然有，$\psi(f) = F$，满足$F(x) = f(x^2)$。

这样，就在$\mathbb{F}_{q^{k/2}}$和$even(\mathbb{F}_{q^k})$之间建立了一种同构关系。

令$Q=(u,v)\in E(\mathbb{F}_{q^k})$，其中$Ev^2=f(u)u\in even(\mathbb{F}_{q^k})$，且$f(u)$为二次非剩余。那么，$v\in odd(\mathbb{F}_{q^k})$。

令$Q=(u,v)\in E(\mathbb{F}_{q^k})$，其中$Ev^2=f(u)u\in even(\mathbb{F}_{q^k})$，且$f(u)$为二次非剩余。如果$S=(s,t)\in<Q>$，那么$s\in even(\mathbb{F}_{q^k}),\ t\in odd(\mathbb{F}_{q^k})$。我们可令$S=mQ$。此时，$S$的阶为$\phi_k(l)$。对于任意$P\in E(\mathbb{F}_p)$。显然，$Q$与$P$线性无关。由于此时，可令$k$为偶数，所以可采用上文提到过的“去分母”法[12]。

3.2.4 Pairing-friendly Curve

对于上述提到过的加速双线性Tate对的计算方法，Miller循环内部的在扩域上的运算复杂的度，决定这整个算法的最终效率。因此，为了提高双线性Tate对的计算效率，要减小扩域中的计算量。但为了保证所构造协议的安全性，基域中的p 值不能够选取的过小。那么，则可以通过一定的方法，选取相应较小的k值，加快在有限扩域中的运算，从而加快整个双线性Tate 对的计算效率。

接下来的问题就是，如何选取合适的曲线，使其嵌入度数相对较小且为偶数，来满足上述定理提出的加速双线性对的算法。

在[27]中，作者给出了新的条件将$E/\mathbb{F}_q$上的ECDLP(Elliptic Curve Discrete Logarithm Problem)归约到诸如$\mathbb{F}_{p^3}$, $\mathbb{F}_{p^4}$, $\mathbb{F}_{p^6}$上的DLP(Discrete Logarithm Problem)更加明显的条件。对于超奇异椭圆曲线来说，这种规约很容易得到，因此存在对超椭圆曲线上的离散对数攻击。FR攻击，就是利用双线性Tate对的特殊性质，完成对超椭圆曲线上ECDLP的离散对数攻击。将椭圆曲线上的标量乘运算转化为有限域$\mathbb{F}_{p^3}$, $\mathbb{F}_{p^4}$, $\mathbb{F}_{p^6}$上的DLP。并可利用Index Attack和Pollard-lamda方法在亚指数时间内完成在$\mathbb{F}_{p^3}$, $\mathbb{F}_{p^4}$, $\mathbb{F}_{p^6}$上的DLP求解。

在这篇文章中，作者进一步给出了，一般曲线上，即非超奇异椭圆曲线上，关于Frobenius迹，易受到攻击的明显情况，给出了在FR攻击下，如何构造归约到更高阶扩域的曲线的算法。

然而，MNT给出的算法，只是给出了如何利用CM(Complex Method)构造嵌入次数小于6的非超奇异椭圆曲线。

在[3]中，作者利用割圆多项式和CM进一步推广了MNT的工作。

在给定嵌入度数k的情况下，如何找到素数q，整数t，满足$|t| \leq 2(q)^{1/2}$，大素数r，满足$r \mid q^k - 1$，但$r \nmid q^i - 1$，对所有$0 < i < k$；对任意一个椭圆曲线$E(\mathbb{F}_q)$，它的Frobenius迹是t，那么它的阶是$n = q + 1 - t$，且满足$r|n$，即$n = rh$。（实际工程实践中，希望h相对较小，这点可以使用Brent算法来实现对合数的素因子分解，从而找到找到合适的h值，确保有理点群的阶为一大素数。）

此处，忽略掉具体的证明工作。具体的证明可参看[3]的2.1节。关于2.1节用到的割圆多项式的性质，可以参看[15]的定理2.4.45。下面给出解CM方程的具体算法。

令l为一整数满足$|l| > 1$，r是割圆多项式$\phi_k(l)$的素因子，令d是一整数，且满足$1 \leqslant d \leqslant deg\phi_k(l)$。令$q = n + l^d$，$t = l^d + 1$。有上面的引理知，$r \mid (l^{kd} - 1)$，且$r \nmid (l^{id} - 1)$，对任意$0 < i < k$。

构造曲线的算法如下：选择l和h，根据CM方法确定的关于D和V的关系，$DV^2 = 4q - t^2$，找到素数q和t。因为$n = hr$，且$r \mid \phi_k(l)$，我们可令$n = m\phi_k(l)$。因此，对于上述给定的参数，CM方程可转化为如下形式：

$DV^2 = 4m\phi_k(l) - (l^d - 1)^2$。

Miyaji在$k = 3, 4, 6$，情况下求解出了该方程的解，因为他们发现该二次Diophantine方程可以归约为Pell方程。这种情况下，Pell方程的解是已知的。对于任意k值的求解，由于当$deg(\phi_k(l)) = p$，没有一般化的方法来对Diophantine方程进行求解，相对的比较困难。

数学家Tzanakis给出了解决形如:

$V^2 = al^4 + bl^3 + cl^2 + dl + e^2$椭圆方程的具体方法。由于当$k = 5, 8, 10, 12$时，$deg(\phi_k) = 4$，上述CM方程可以转化为如下形式：

$V^2 = al^4 + bl^3 + cl^2 + dl + f$。

对于该方程，可以使用代价较小的穷收索办法来对该方程求解。

3.3 两类非超奇异椭圆曲线上的Tate对的计算

3.3.1 椭圆曲线$E_1 : y^2 = x^3 + B$，其中$p \equiv 1 \mod 3$

假设β为$\mathbb{F}_p{}^*$上的三阶本原根。曲线E_1上的一个非平凡自同构为：

$$\phi : E_1 \to E_1$$

$$(x, y) \to (\beta x, y)$$

因为一个自同构显然是同种，那么ϕ的对偶同种为：

$$\widehat{\phi} : E_1 \to E_1$$

$$(x, y) \to (\beta^2 x, y)$$

不难验证，$\widehat{\phi} \circ \phi = 1$，$\widehat{\phi} = \phi^2$，$\widehat{\phi}^2 = \phi$。$\widehat{\phi}$也是曲线$E_1$上的一个非平凡自同构。关于$E_1$上的自同构群$Aut(E_1)$，有如下结论：

$$Aut(E_1) = \{\pm 1, \pm\phi, \pm\widehat{\phi}\}。$$

假设$P \in E_1(\mathbb{F}_P)$，且P的阶为r，其中r满足，$r \nmid \sharp E_1(\mathbb{F}_p)$。则，$\phi$和$\widehat{\phi}$在子群$< P >$上的作用，相当于标量乘$\lambda$和$\widehat{\lambda}$，其中，$\lambda$和$\widehat{\lambda}$是满足二次方程$x^2 + x + 1 = 0 \mod r$的根。则有，$\phi(P) = \lambda P$。那么，标量乘法运算$\lambda P$可以转化为$\mathbb{F}_p{}^*$上的一个乘法。

从新定义椭圆曲线E_1上的双线性Tate对，结果如下：

假设大素数p满足$p \equiv 1 \mod 3$。$E_1 y^2 = x^3 + B$是定义在有限域F_p上的椭圆曲线。ϕ和$\widehat{\phi}$为定义在E_1上的两个非平凡自同构映射。假设$P \in E_1(\mathbb{F}_p)$，且P的阶为r，其中，$r^2 \mid \sharp E_1(\mathbb{F}_p)$，并令$\lambda$为子群$< P >$的标量乘，且满足$\lambda P = \phi(p)$。假设$a$为满足$ar = \lambda^2 + \lambda + 1$的正整数。令$l_{\phi(P),\widehat{\phi}(P)}$为经过$\phi(P)$和$\widehat{\phi}(P)$的直线。那么，对于点$Q \in E_1(\mathbb{F}_{p^k})$，双线性Tate对满足如下关系式：

$$e(P, Q)^a = (f_{\lambda,P}(Q)^{\lambda+1} \cdot f_{\lambda,P}(\widehat{\phi}(Q) \cdot l_{\phi(P),\widehat{\phi}(P)}(Q))^{\frac{p^k-1}{r}}$$

证明可参考[31]。

由上述假设，可得到计算双线性对$e(P, Q)^a$的新算法。为了方便讨论，仅考虑当安全度（AES 80比特）的情形，即取p为512比特，嵌入度数$k = 2$。为了提高计算效率，可以取特殊Q点。令其x坐标在基域$\mathbb{F}_p$上，其y坐标在扩域$\mathbb{F}_{p^k}$上。因为，P，Q和$\widehat{\phi}(Q)$的x坐标在基域中，显然有$(p - 1) \mid \frac{p^2-1}{r}$。所以，Miller算法中的分母部分可以忽略不计算。对于曲线E_1上的双线性对$e(P, Q)^a$，有如下修改的Miller算法。

假设P的坐标为(x_P, y_P)，那么$\phi(P) = (\beta x_P, y_P)$和$\widehat{\phi}(P) = (\beta^2 x_P, y_P)$。所以直线$l_{\phi(P),\widehat{\phi}(P)}$的方程为$y - y_P = 0$。因此，$f_3$在$Q$点的赋值为$f_3(Q) = y_Q - y_P$，其中$y_P \in \mathbb{F}_p$，$y_Q \in \mathbb{F}_{p^2}$。

Algorithm 6 计算椭圆曲线$E_1 : y^2 = x^3 + B$上的双线性对$e(P,Q)^a$。

输入: 素数p, $\lambda = \sum_{i=0}^{n} l_i 2^i$, 其中，$l_i \in \{-1, 0, 1\}$椭圆曲线$E/\mathbb{F}_p$及嵌入次数$k = 2$. 素数$r = \sum_{i=0}^{l} r_i 2^i$, $P \in E(\mathbb{F}_p)[r]$, $Q \in E(\mathbb{F}_{p^k})[r]$. P与Q线性无关. Q点的x坐标在$E(\mathbb{F}_p)$上.

输出: 双线性对$e(P,Q)^a$

[1.] $R \leftarrow P$, $f_1 \leftarrow 1$,$f_2 \leftarrow 1$,$f_3 \leftarrow l_{\phi(P),\widehat{\phi}(P)}(Q))$

[2.]i from $l-1$ downto 0

[2.1] $f_1 \leftarrow f_1^2 \cdot l_{T,T}(Q)$, $f_2 \leftarrow f_2^2 \cdot l_{T,T}(\widehat{\phi}(Q))$，$T \leftarrow 2T$

[2.2] If $l_i = 1$ then ,

[2.3] $f_1 \leftarrow f_1 \cdot l_{T,P}(Q)$, $f_2 \leftarrow f_2 \cdot l_{T,P}(\widehat{\phi}(Q))$, $T \leftarrow T + P$

[2.4] If $l_i = -1$then ,

[2.5] $f_1 \leftarrow f_1 \cdot l_{T,-P}(Q)$, $f_2 \leftarrow f_2 \cdot l_{T,-P}(\widehat{\phi}(Q))$, $T \leftarrow T - P$

[3.] $f_1 \leftarrow f_1^{\lambda+1}$

[4.]Return $(f_1 \cdot f_2 \cdot f_3)^{\frac{p^2-1}{r}}$

由于β^2和Q点的x坐标在$\mathbb{F}_p*$上，因此，计算$\widehat{\phi}(Q)$仅需要$\mathbb{F}_p*$上的一个乘法。

假设T为$E(\mathbb{F}_p)$上的点，其坐标为(x_T, y_T)，设m为直线$l_{T,T}$的斜率。则直线$l_{T,T}$的方程为$(y-y_T)-m(x-x_T)=0$。那么，计算$l_{T,T}(Q) = (y_Q - y_T) - m(x_Q - x_T)$仅需要基域$\mathbb{F}_p$上的两个乘法。同理，计算$l_{T,P}(Q)$和$l_{T,P}(\widehat{\phi}(Q))$也同样只需要$\mathbb{F}_p$中的两个乘法。

算法效率分析在有限域上，加法运算、减法运算同乘法和除法运算比较起来，耗时相对较少。因此，在下面的效率分析中忽略有限域上的加法和减法运算。同时，定义基域$\mathbb{F}_p$上的一个乘法运算的计算量为$1M$。

为了减少上述算法中，倍点和和加法的元算量，我们可以寻找曲线，使其满足λ具有较低的Hamming Weight。在文献[25]中，讨论了这种曲线的生成方法，并给出了一个有效的例子。具体参数，如下表3-1：

1. **仿射坐标下的算法效率分析**

令$P = (x_P, y_p) \in E_1(\mathbb{F}_p)[r]$，$Q = (x_Q, y_Q) \in E_1(\mathbb{F}_{p^2})$，其中，$x_Q \in \mathbb{F}_p$ 和$y_Q \in \mathbb{F}_{p^2}$。

表 3-1 适合双线性对计算的椭圆曲线E_1的实例

曲线方程为，$E_1 : y^2 = x^3 + 5$，嵌入次数$k = 2$。
p为512比特大素数。
群的阶$r = \lambda^2 + \lambda + 1$，其中$\lambda = 2^{80} + 2^{16}$

接下来，我们将一行一行的来统计算法的计算量。

对于第一行来说，很容易验证$l_{\phi(P),\hat{\phi}(P)}(Q) = y_Q - y_P$。因为，$P$，$\phi(P)$，$\hat{\phi}(P)$具有相同的$y$坐标值。那么，在第一行中就不需要乘法运算。

在第二行中，我们所需要考虑的是，Miller循环中的计算量。由于$\hat{\phi}(P)$的值可以事先获得，完成两条直线的赋值运算需要两次乘法运算。在下一步中，有限域$\mathbb{F}_{p^2}$上的两次乘法不平方运算被执行，这可以转化成基域$\mathbb{F}_p$上的10次乘法运算。

在算法的2.1步中，所需计算量为；16M + 1I运算。因为，此处需要79次循环，2.1步总的计算量是：1264M + 79I。

2.3步的计算量为：11M+1I。

3步需要的计算量为：160M。

到现在为止，总的运算量为：1435M + 80I。

4步，需要2次$\mathbb{F}_{p^2}$上的乘法运算，可转化为$\mathbb{F}_p$上6M的乘法运算。

最后一步的模幂运算可以分为两步执行。利用Frobenius映射，$p-1$次模幂运算可以转化为基域$\mathbb{F}_p{}^*$上5M + 1I次运算。利用Lucas序列，我们需要$(512-161)\times 2M = 702M$次运算完成剩下的$(p+1)/r$模幂运算。

因此，第四行需要的运算量为6M + 5M + 1I + 702M。

总之，在仿射坐标下，完成上述运算需要的运算量为2148M + 81I。

若将1I=10M的假设带入，我们将获得和赵昌安师兄同样的结果。

2. **雅克比坐标下的算法效率分析**

首先，给出雅克比坐标下直线赋值的操作。

令μ为经过点P and $-2P$的直线$h_{P,P}$的斜率，且$\mu = \lambda_1/Z_3$。则有，$h_{P,P}(x,y) = (y - Y_1/{Z_1}^3) - \mu(x - X_1/{Z_1}^2)$。

因此，$h_{P,P}(-x_Q, iy_Q) = (iy_Q - {Y_1}/{Z_1}^3 + \mu(x_Q + {X_1}/{Z_1}^2)$。

定义，$\hat{h}_{P,P}(x,y) = (2Y_1{Z_1}^3)h_{P,P}(x,y)$，我们有如下结论：

$\hat{h}_{P,P}(-x_Q, iy_Q) = (2Y_1Z_1){Z_1}^2 iy_Q - 2{Y_1}^2 + (3{X_1}^2 + a{Z_1}^4)({Z_1}^2 x_Q + X_1) = Z_3 t_4 iy_Q - (2t_1 - t_5(t_4 x_Q + X_1))$。由于已知，最终的计算结果要求$(p^2-1)/r$次模幂，且$(2Y_1{Z_1}^3) \in \mathbb{F}_p$，因此，$(2Y_1{Z_1}^3)^{(p-1)} = 1$。

总的来说，在雅克比坐标下，完成倍点运算和直线赋值运算共需要8M + 6S的运算量。

同理，我们可以得出，在雅克比坐标下，完成点加运算和直线赋值运算共需要11M + 3S 的运算量。

下面，我们将给出在雅克比坐标下，上述算法的运算量估计值。

同在仿射坐标下的统计一样，对于第一行来说，不需要任何的乘法运算。

在第2.1步，需要的运算量为14M + 10S。有79次循环，因此，共需要1106M +790S。

在第2.3步，所需的运算量为17M + 3S。

同在仿射坐标下一样，对于第3步，我们需要160M运算量。对于底数，我们需要2M。对于前$p-1$次模幂运算，我们需要5M的运算量。同样，使用Lucas序列，完成后$(p-1)/r$次模幂运算，我们需要702M的运算量。

总的来所，在雅克比坐标下，我们需要2725 M的运算量完成上述运算，若假设$1I = 10M$，$1M = 1S$，则少于仿射坐标下的估计值。

3.3.2 椭圆曲线$E_2 : y^2 = x^3 + Ax$，其中$p \equiv 1 \mod 4$

假设α为$\mathbb{F}_p^*$的四阶元。由于，$p \equiv 1 \mod 4$，即有，$4 \mid (p-1)$。故，这样的α必然存在。

椭圆曲线E_2的一个非平凡自同构映射定义为：

$$\phi : E_2 \to E_2$$

$$(x, y) \to (-x, \alpha y)$$

自同构ϕ也是一个同种，其对偶同种为：

$$\widehat{\phi} : E_2 \to E_2$$

$$(x,y)\rightarrow(-x,-\alpha y)$$

假设$P\in E_2(\mathbb{F}_p)$，且P的阶为r，其中r满足$r^2\nmid\sharp E_2(\mathbb{F}_p)$。下面给出，自同构映射$\phi$和$\widehat{\phi}$在子群$<P>$上的作用。

$$\phi:\langle P\rangle\rightarrow\langle P\rangle \qquad \widehat{\phi}:\langle P\rangle\rightarrow\langle P\rangle$$
$$P\rightarrow\phi(P)=\mu P \quad P\rightarrow\widehat{\phi}(P)=\widehat{\mu}P$$

其中，μ和$\widehat{\mu}$为方程$x^2+1=0 \mod r$的根。

假设p为一个大素数，且满足$p\equiv 1 \mod 4$。令$E_2:y^2=x^3+ax$为有限域$\mathbb{F}_p$上的椭圆曲线。ϕ和$\widehat{\phi}$为E_2的两个非平凡自同构。$P\in E_2(\mathbb{F}_p)$，且P的阶为r，其中r满足$r^2\nmid\sharp E_2(\mathbb{F}_p)$。设$k$为$E_2$的嵌入次数。$\mu P=\phi(P)$。定义$a$为满足$ar=\mu^2+1$的正整数。对于$Q\in E_2(\mathbb{F}_{p^k})$，有满足如下关系的双线性Tate对：

$$e(P,Q)^a=(f_{\mu,P}(Q)^{\mu}\cdot f_{\mu,P}(\widehat{\phi}(Q)))^{\frac{p^k-1}{r}}$$

下面给出一条适合该算法的的曲线E_2的参数。

表 3-2 适合双线性对计算的椭圆曲线E_2的实例

曲线方程为，$E_2:y^2=x^3-3x$，嵌入次数$k=2$。
p为512比特大素数。
群的阶为$r=2377053510497684678496540544608070896366621137359721$，
且$\mu=689500327845851669903445 21$满足$\mu^2+1=2r$
μ的比特长度为86，用NAF形式表示后，非零的比特数为27。

下面给出对应的Miller算法，见Algorithm 7。

1. **仿射坐标下运算量分析**

算法的第一步，显然不需要任何运算量，略过。

由于，已知μ的长度为86，那么第二步的for循环所要执行步次数为85次。第2.1步，需要执行椭圆曲线上有理点的倍点运算，这需要消耗1I+4M。设直线$l_{T,T}$的方程为$y-y_T-m(x-x_T)=0$，其中，点T的坐标为(x_T,y_T)，m为直线$l_{T,T}$的斜率。设Q点

Algorithm 7 计算约化的Tate对的Miller算法

输入: 素数p, 椭圆曲线$E/\mathbb{F}_p$及嵌入次数k. 素数$r = \sum_{i=0}^{l} r_i 2^i$, $P \in E(\mathbb{F}_p)[r]$, $Q \in E(\mathbb{F}_{p^k})[r]$. P与Q线性无关. $D_Q = [Q_1] - [Q_2]$.

输出: 双线性对$e(P, Q)$

[1.] $T \leftarrow P$, $f_1 \leftarrow 1$,$f_2 \leftarrow 1$

[2.] for $i = n - 1$ downto 0

[2.1] $f_1 \leftarrow f_1^2 \cdot l_{T,T}(Q)$, $f_2 \leftarrow f_2^2 \cdot l_{T,T}(\widehat{\phi}(Q))$, $R \leftarrow 2R$

[2.2] If $r_i = 1$ then

[2.3] $f_1 \leftarrow f_1^2 \cdot l_{T,P}(Q)$, $f_2 \leftarrow f_2^2 \cdot l_{T,P}(\widehat{\phi}(Q))$, $R \leftarrow T + P$

[2.4] If $r_i = -1$ then

[2.5] $f_1 \leftarrow f_1^2 \cdot l_{T,-P}(Q)$, $f_2 \leftarrow f_2^2 \cdot l_{T,-P}(\widehat{\phi}(Q))$, $R \leftarrow T - P$

[3.] $f_1 \leftarrow f_1^{\mu}$

[4.] $(f_1 \cdot f_2)^{\frac{(p-1)(p+1)}{r}}$

的坐标为(x_Q, iy_Q)，其中x_Q，$y_Q \in \mathbb{F}_p$, $i \in \mathbb{F}_{p^2}$。直线赋值运算$l_{T,T}(Q)$和直线赋值运算$l_{T,T}(\widehat{\phi}(Q))$所消耗的计算量为2M。另外，计算$f_1^2 \cdot l_{T,T}(Q)$和$f_2^2 \cdot l_{T,T}(\widehat{\phi}(Q))$需要扩域$\mathbb{F}_{p^2}$上的两次乘法和两次平方运算，这可以转化成基域上的10M。因此，执行2.1步总的开销为1I+4M+2M+10M=1I+16M。由于，此处需要执行85次循环，2.1步的总开销为$(1I + 16M) \cdot 85 = 85I + 1360M$。

由于2.3步和2.5步所需的计算量几乎一样，所以，估计2.3步的计算量足矣。椭圆曲线上的有理点的加法需要1I+3M。直线值运算$l_{T,})(Q)$和直线赋值运算$l_{T,T}(\widehat{\phi}(Q))$所消耗的计算量为2M。计算$f_1 \cdot l_{T,T}(Q)$和$f_2 \cdot l_{T,T}(\widehat{\phi}(Q))$需要扩域$\mathbb{F}_{p^2}$上的两次乘法运算，这可以转化成基域上的6M。因此，执行2.3步所消耗的计算量为1I+11M。由于，此处给定的μ值的NAF表示后的Hamming Weight为27。所以，在整个算法执行中，2.3步和2.5步所消耗的计算量为$(27 - 1) \cdot (1I + 11M) = 26I + 286M$。

对于3步，由于一次扩域中的平方运算相当于基域上的2M，使用Lucas序列，步骤3的计算量为170M。

对于4步，$f_1 f_2$需要$\mathbb{F}_{p^2}$上的一个乘法，即3M。最后的幂运算，可分为两步。计算$(f_1 f_2)^{p-1}$需要一次基域上的求逆和5M，共要1I+5M。而$\frac{p+1}{r}$的比特长度为514-171=343。同样使用Lucas序列，计算$\frac{p+1}{r}$次幂要686M，4步共需要的运算量为3M+1I+5M+686M=1I+694M。

所以，执行上述算法所需的总运算量为$(85I + 1360M) + (26I + 286M) + (170M) + (1I + 694M) = (112I + 2510M)$。

2. **雅克比坐标下运算量分析**

在雅克比坐标下，完成倍点运算和直线赋值运算共需要8M + 6S的运算量; 完成点加运算和直线赋值运算共需要11M + 3S 的运算量。

下面，我们将给出在雅克比坐标下，上述算法的运算量。

同在仿射坐标下的统计一样，对于第一行来说，不需要任何的乘法运算。

在第2.1步，需要的运算量为(8M + 6S)+10M=18M+6S。有85次循环，因此，共需要$85 \cdot (18M + 6S) = 1530M + 510S$。

在第2.3步，所需的运算量为(11M+3S)+6M=17M+3S；循环26次，总的计算量为$26 \cdot (17M + 3S) = 442M + 78S$。

同在仿射坐标下一样，对于第3步，我们需要170M运算量。对于底数，我们需要3M。对于前$p-1$次模幂运算，我们需要5M的运算量。同样，使用Lucas序列，完成后$(p-1)/r$次模幂运算，我们需要686M的运算量。执行第4步需要3M+5M+686M=694M。

总的来所，在雅克比坐标下，我们需要$(1530M + 510S) + (442M + 78S) + (694M) = 2666M + 588S$的运算量完成上述运算。若假设$M = S$，$1I = 10M$，则需要$3254M$计算量，少于仿射坐标下的估计值。

3.4 $1I \stackrel{?}{=} 10M$

在上述给出的在雅克比坐标下，对椭圆曲线E_1和E_2上的两类特殊的Tate对的算法效率估计过程中用到了$1I \stackrel{?}{=} 10M$这样一个假设。如果，这个假设成立，那么，显然在雅克比坐标下，计算上述两类特殊的双线性Tate对的计算效率显然要好过在仿射坐标下计算

上述两类特殊双线性对的计算效率。

对于椭圆曲线E_1上构造的双线性Tate对的，具体比较结果如下：

仿射坐标	雅克比坐标
2148M+81I	2725M

对于椭圆曲线E_2上构造的双线性Tate对的，具体比较结果如下：

仿射坐标	雅克比坐标
2510M+112I	3254M

在[8]中，作者给出了$1I = 30M$的假设，在这种假设下，显然在上述比较过程中，将这种假设值代入后，在雅克比坐标下计算双线性Tate对的效率要明显好过在仿射坐标下的计算效率。

然而，实验结果显示，在p为512比特的大素数时，1次基域上的求逆运算的计算效率等价于基域18次的乘法运算效率。因此，上文给出的相对于$1I = 30M$的假设$1I = 10M$是合理的。因此，所得出的“在雅克比坐标下计算上述给出的两类双线性Tate对的计算效率好过在仿射坐标下计算上述两类双线性Tate对”的结论是可靠的。

下面给出关于p取不同的长度的大素数，基域$\mathbb{F}_p$上求逆运算和乘法运算的效率比的测试结果。

素数p的长度	有限域$\mathbb{F}_p$上求逆运算与乘法运算的计算效率比I/M
160$bits$	6.99425
320$bits$	10.7851
512$bits$	14.4476
1024$bits$	16.3938

上述的结果获得是在MIRACL上采用随机测试获得的结果。

具体的方法如下：

1. 随机生成320，512，1024比特的大素数，并生成相应的有限域。

2. 在上述有限域中，随机选则1000个大数。

3. 分别对上述所选大数进行乘法和求逆运算，并调用系统时钟完成计算世间的记录用来做算法效率的比较。(由于机器执行速度过快，所以在ms级很难准确记录，需要调用μs级的系统时钟。)

注：实际的测试结果是，当p为260比特的大素数时，有其构成的有限域上的求逆运算和乘法运算的比值平均大于10。

3.5 本章小结

本章，第一次利用实验数据分析具体给出了有限域上求逆和乘法运算效率的比值。给出了两类特殊Tate对在雅克比坐标下的计算效率估值。并利用实验数据支持了在估值过程中用到的假设。

第四章　Tate对在密钥交换协议中的应用

在本章节中，给出了两类基于双线性Tate对构造的密码体制的缺陷，并给出了相应的证明。

4.1　可验证的基于身份的密钥交换协议

在[27]这篇文章中，M. Scott 巧妙地构造了一种基于身份的密钥交换协议。这个协议精巧的地方在于，仅需要两次交互，四次计算就能够完成两方的密钥交换。其中，只需要一次pairing计算，并且这个pairing构造在超奇异椭圆曲线上，并且由于扭映射的引入，使得计算pairing计算的速度有所提高。

在这个密钥交换协议中，个体的秘密由存储在硬件中的$Token$和用户个人选择的PIN构成。值得一提的是，个体的秘密不能够从它的形式上看出来，也不能够轻易的构造出来。个体的秘密在形式上只是一个数字，而且可以是任何数字。个体的秘密和保存在硬件中的$Token$，以及用户个人选择的PIN存在着线性关系，即$D = N + PIN$，其中D代表个体秘密，N代表硬件$Token$上的数字，PIN代表用户的个人身份号码。

在这个协议中，可信机构知道主密钥s。用户的身份被一哈希函数映射到一特定的椭圆曲线上。例如，用户Alice的身份被一哈希函数映射到一个具有大素数阶的椭圆曲线上的一点A。用户Alice按自己的意愿随机选择一个PIN号α 。在用户Alice向可信机构注册之后，用户Alice从可信机构获得A和sA。用户Alice计算αA，并作减法，将A和$(s-\alpha)A$存储下来，并记录α。作为一个简单的秘密共享机制，这两部分秘密$(s-\alpha)A$和αA要同时获得，才能够恢复sA，即$sA = ((s-\alpha)A + \alpha)A$。很明显，用户Alice不可能获得主密钥s。因为，假设Alice能够获得主密钥s，那么Alice就能够解决椭圆曲线上离散对数问题。

下面，让我问来看下具体的AKE协议，见表4-1。

安全分析可以参照[27]Section 4.1。穷搜索攻击，以及特殊Tate pairng的选择。

下面给出，由可信机构发起的攻击方案。其中，可信机构用TA表示，见表4-2。

表 4-1 AKE协议

Alice	Bob
Generates Random $a < r$	Generates Random $b < r$
$ID_a \rightarrow$	$\leftarrow ID_b$
$P_a = \hat{e}((s-\alpha)A + \alpha A), B)^a$	$P_b = \hat{e}((s-\beta)A + \beta B), A)^b$
$P_a \rightarrow$	$\leftarrow P_b$
如果$P_b \leq 1$ or $(P_b)^r \neq 1$中止	如果$P_a \leq 1$ or $(P_a)^r \neq 1$
$key = (P_b)^a = (\hat{e}(A, B))^{abc}$	$key = (P_a)^b = (\hat{e}(B, A))^{abc}$

表 4-2 可信机构发起的攻击方案

Alice	TA(Bob)
随机选择$a < r$	随机选择$b < r$
$ID_a \rightarrow$	$\leftarrow ID_b$
$P_a = \hat{e}((s-\alpha)A + \alpha A), B)^a$	$P_b = \hat{e}(sB, A)^b$
$P_a \rightarrow$	$\leftarrow P_b$
如果$P_b \leq 1$ or $(P_b)^r \neq 1$ 中止	如果$P_a \leq 1$ or $(P_a)^r \neq 1$ 中止
$key = (P_b)^a = (\hat{e}(A, B))^{abc}$	$key = (P_a)^b = (\hat{e}(B, A))^{abc}$

显然，可信机构可以伪造Bob的身份，同Alice完成密钥交换。因为，可信机构不但拥有主密钥s；并且，在这个协议中，通讯双方不验证对方的身份。因此，当可信机构做出"某种"非法行为时，其他用户并不知道。

证明：标准模型下的中间人攻击[17]。可信机构可以在通信双方通信之前，若伪造通信双方一方的身份，同另一方通信，完成密钥交换。当通信双方进行通信时，此时"不再可信"的可信机构能够截取并伪造信息，同通信双方进行通信。而且，"不再可信"的可信机构在冒充通信双方进行通信时，不会留下任何可被查证的信息。因为，可信机构是唯一拥有主密钥s的实体，他它可以伪造任何用户身份，来完成通信。因此，在诸如Ad hoc等移动网络中，当某一"可信"节点受到攻击或不再可信时，网络节点间的可靠通信，

将受到严重的挑战。

4.2 无证书的密钥交换(Certificateless Key Exchange)

在[24][22]中，一种介于IBE和PKI之间的密码协议被提出。这种被称作无证书的协议，能够减少PKI 体制下用户对TA的注册，以及密钥托管问题；而且提供一定的用户身份认证机制，减少IBE体制下，用户身份认证不足的缺点。首先，给出CLK体制；然后，给出基于CLK的密钥交换机制；最后，给出这个机制的不足，以及该机制提供的当可信机构伪造用户身份进行通信时，在CLK机制下，TA不得不留下伪造的痕迹。

下面给出简单的CLK模型，因为该模型已经可以提供足够的参数信息，给出基于CLK的密钥交换协议。

令k为一个秘密参数，该参数传给Setup算法和$\mathcal{IG}$，其中$\mathcal{IG}$是一给定秘密参数k的BDH参数生成器。

Setup算法如下：

1. 对于输入k，$\mathcal{IG}$输出$< G_1, G_2, e >$，其中，G_1和G_2为素数阶群，其阶数为q。e为一双线性对，满足$e : G_1 \times G_1 \rightarrow G_2$。

2. 任意选择$P \in G_1$，且P为G_1的生成元。

3. 从$\mathbb{Z}_p{}^*$中随机均匀的选择一个主密钥s，且令$P_0 = sP$。

4. 选择两个哈希函数，令$H_1 : \{0,1\}^* \rightarrow G_1{}^*$，$H_2 : G_2 \rightarrow \{0,1\}^n$。

5. 输出系统参数。

因此，系统的参数为$params = \{G_1, G_2, e, n, P, P_0, H_1, H_2\}$。主密钥为$s \in \mathbb{Z}_p{}^*$。明文空间为$\mathcal{M} = \{0,1\}^n$，密文空间为$\mathcal{C} = G_1 \times \{0,1\}^n$。

部分私钥抽取算法如下：

输入为：用户$ID \in \{0,1\}^*$，例如：对于实体A，将其用户身份记作ID_A。

1. 计算$Q_A = H_1(ID_A) \in G_1{}^*$。

2. 输出部分私钥$D_A = sQ_A \in G_1{}^*$。

用户A将验证部分私钥抽取算法的正确性。计算$e(D_A, P) \stackrel{?}{=} e(Q_A, P_0)$。

设置秘密值。

这个算法，输入为系统参数和用户的身份ID。随机地选择$x_A \in \mathbb{Z}_p{}^*$，输出x_A作为用户A的秘密值。

设置私钥。

这个算法，将系统参数，用户A的部分私钥D_A，用户A的秘密值x_A，作为输入。将用户的部分私钥D_A转换成私钥S_A，通过计算$S_A = x_A D_Q = x_A s Q_A \in G_1{}^*$。

设置公钥。

这个算法，将系统的参数，用户A的秘密值$x_A \in \mathbb{Z}_p{}^*$，作为输入，来构造用户A的公钥对$< X_A, Y_A >$。其中，$X_A = x_A P$，$Y_A = x_A P_0 = x_A s P$。

加密算法。

1. 检查X_A，Y_A是否属于$G_1{}^*$，并验证$e(D_A, P) \stackrel{?}{=} e(Q_A, P_0)$。如果不成立，则终止加密。

2. 计算$Q_A = H_1(ID_A) \in G_1{}^*$。

3. 随机选择$r \in \mathbb{Z}_q{}^*$。

4. 计算并输出密文：$C = \langle rP, M \bigoplus H_2(e(Q_A, Y_A)^r \rangle$。

解密算法。

对于输入密文$C = \langle U, V \rangle$，解密过程如下。$V \bigoplus H_2(e(S_A, U))$。

基于CLK的两方密钥交换协议

对于基于CLK的两方密钥交换协的初始化同样需要，Setup，Paritial-Private-Key-Extract，Set-Secret-Value，Set-Private-Key，Set-Public-Key等一系列算法。

实体A和B进行密钥交互的协议机制如下：

1. 实体A和B，分别随机地选择a，$b \in \mathbb{Z}_q{}^*$。

2. $A \rightarrow B : T_A = aP,\ \langle X_A, Y_A \rangle$。

3. $B \rightarrow A : T_B = bP,\ \langle X_B, Y_B \rangle$。

在完成上述过程并将信息转发之后，通信双方都要检查对方的公钥是否正确。

例如：用户A将计算$e(X_B, P_0) = e(Y_B, P)$。其他用户，将执行类似步骤。但是，这轮每一通信方完成的计算只是对对方公钥正确性的一次检验，并没有对通信的另一方身份

进行任何校验，这就为后面提到的攻击埋下伏笔。在对对方的公钥完成校验之后，用户A计算$K_A = e(Q_B, Y_B)^a e(S_A, T_B)$；

用户B计算$K_B = e(Q_A, Y_A)^b e(S_B, T_A)$。

不难验证，$K = K_A = K_B$。

下面，就让我们来看看这个协议是如何抵挡身份伪造和攻击痕迹可寻的可信机构发起的中间人攻击。

首先，给出这个协议的抵挡身份伪造的证明。

假设，用户C想伪造用户B的身份，同A进行密钥交换。

那么，用户C将要向用户A发送：$T_B = bP, \langle X_B, Y_B \rangle$。

显然，用户A会检查用户C发送过来的公钥对$\langle X_B, Y_B \rangle$。但问题是，仅从C发送过来的公钥对，用户A不能够识别出用户C在伪造用户B进行通讯。因为，在公钥对中，不包含任何该公钥对持有者的身份信息。因此，该密钥交换协议将继续进行。用户A计算$K_A = e(Q_B, Y_B)^a e(S_A, T_B)$。用户C只有计算出$(Q_A, Y_A)^b e(S_B, T_A)$的值，才能够获得$K$。然而，$S_B$是用户B的私钥，用户C很难通过计算获得用户B的私钥。若用户C能够通过计算获得用户B的私钥，那么用户C显然能够通过计算解决椭圆曲线上的离散对数问题。

下面给出攻击痕迹可寻的可信机构发起的中间人攻击的证明。

若在此协议中，可信机构不再可信，伪造该体制中的任何用户发起攻击。虽然，基于CLK的密钥交换体制，不能够挡住住这种攻击；但是，当不再可信的可信机构发出这种伪造用户身份的攻击后，系统内的用户可以查找出这是由可信机构发出的攻击。

假设，可信机构TA伪造用户B的身份向用户A发送：

$T_B = bP, \langle X_B', Y_B' \rangle$。

用户A将计算$K_A = e(Q_B, Y_B')^a e(S_A, T_B)$。

TA将计算$K_C = e(Q_A, Y_A)^b e(S_B', T_A)$。

显然，$K_A = e(H_1(ID_B), x_B' P_0)^a e(x_A D_A, bP)$。

为了，给出明显的K_A的计算结果，带入具体数值。此时，主密钥s虽然不能够被系统中的用户推倒出，但是，作为等式证明，可将包含s的参数具体的代入。

$$K_A = e(H_1(ID_B), x_B{}'P_0)^a e(x_A D_A, bP) = e(H_1(ID_B), x_B{}'sP)^a e(x_A s H_1(ID_B), bP) = e(H_1(ID_B), P)^{a x_B{}' s} e(H_1(ID_B), P)^{x x_A b}。$$

$$K_B{}' = e(Q_A, Y_A)^b e(S_B{}', T_A) = e(H_1(ID_A), x_A sP)^b e(x_B{}' s H_1(ID_B), bP) = e(H_1(ID_A), P)^{x_A b s} e(H_1(ID_B), P)^{x_B{}' s b}。$$

显然，$K_A = K_B{}'$。

但是，虽然“不再可信”的可信机构能够伪造用户B的身份，欺骗用户A完成密钥交换。但是，在生成共享密钥的过程中，可信机构选择的用来构造密钥的秘密值以很小的概率同用户B选择的用来构造密钥的秘密值相等。因此，当用户A和真正的用户B进行密钥交换的过程后，可以验证之前用来交换的密钥值。显然，若用户B被伪装，则通过比较不同的共享密钥值，可以确定B是否被伪装。然而，系统中唯一具有伪装系统内任何用户来同其他用户进行通讯的只有可信机构。因为，可信机构是唯一拥有主密钥s的实体。若其他用户拥有主密钥s，相当于其他用户能够通过已知的系统参数中的P和P_0计算求得s。这显然是不可能的。如果可能，这用户能够解决椭圆曲线上离散对数问题(ECDLP)。因此，基于CLK的这个密钥交换协议虽然不能够抵挡由可信机构发起的中间人攻击，但是，当可信机构不再可信，发起中间人攻击后，该协议保证了，这种行为是可检验的。

证毕。

4.3 本章小结

“安全隐患往往来自内部”，这句话充分说明了，来自系统内部攻击的可能行性。当“可信机构”不再可信的时候，就出现了上述论证的问题。因此，提出更好的密钥交换协议来防止来自内部的攻击至关重要。最近，一些研究者基于零知识证明，提出了双机构的认证，一方认证用户身份，一方向匿名用户分配密钥来解决来自“可信机构”的攻击，即使两方“串通”也无法伪造用户身份。

第五章　总结和展望

5.1　研究成果

1 在雅可比坐标下，给出了计算两类非超奇异椭圆曲线上的双线性Tate对，Miller算法复杂度的估计值。

2 给出了有限域$\mathbb{F}_p$上，求逆运算和乘法及平方运算的计算效率的测试结果和分析，用来支持给出上述估计值时用到的'强'假设。

3 指出了一类利用双线性Tate对构造的基于身份的密钥交换协议的不足，并给出证明。

5.2　有待探讨的问题

1 给出在雅可比坐标下，计算其它双线性对Miller算法复杂度的估计值。

2 如何加快有限域$\mathbb{F}_p$上的求逆、乘法及平方运算。

3 如何基于MIRACL实现雅可比坐标下的双线性Tate对的计算。

4 如何利用双线性对构造更好的基于身份的密钥交换协议。

参 考 文 献

[1] 万哲先. *代数与编码*. 科学出版社, 北京, 12 1980.

[2] Gildas Avoine, Jean Monnerat, and Thomas Peyrin. Advances in alternative non-adjacent form representations. In *INDOCRYPT*, pages 260–274, 2004.

[3] P. S. L. M. Barreto and M. Naehrig. Pairing-friendly elliptic curves of prime order. In *In Selected Areas in Cryptography - SAC'2005*, page 319 – 331. LNCS 3897, Springer-Verlag, 2005.

[4] I. F. Blake, G. Seroussi, and N. P. Smart. *Elliptic Curves in Cryptography*, volume 265. Cambridge University Press, New York, NY, USA, 1999.

[5] I. F. Blake, G. Seroussi, and N. P. Smart. *Advances in Elliptic Curve Cryptography*, volume 317. Cambridge University Press, New York, NY, USA, 2005.

[6] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *In Advances in Cryptology - Crypto'2001*, pages 213–229. LNCS 1838, Springer-Verlag, 2001.

[7] D. Boneh and M. Franlin. Identity-based encryption from the weil pairing. *SIAM J. Comput.*, 32:586–615., 2003.

[8] S. Chatterjee, P. Sarkar, and R. Barua. Efficient computation of tate pairing in projective coordinate over general characteristic fields. In *In Information Security and Cryptology – ICISC'2004*, pages 168–181. LNCS 3506, Springer-Verlag, 2004.

[9] Whitfield Diffie. Cryptography, the next two decades. In Allen Gersho, editor, *Advances in Cryptology: A Report on CRYPTO 81*, pages 84–108. U.C. Santa Barbara Dept. of Elec. and Computer Eng., 1982. Tech Report 82-04.

[10] G. Frey and H.-G. Rück. A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. *Mathematics of Computation*, 62(206):865–874., Apr, 1994.

[11] S. D. Galbraith, K. Harrison, and D. Soldera. Implementing the tate pairing. In *In Algorithm Number Theory Symposium - ANTS 5*, pages 324–337. LNCS 2369, Springer-Verlag, 2002.

[12] R. P. Gallant, R. J. Lambert, and S. A. Vanstone. Faster point multiplication on elliptic curves with efficient endomorphisms. In *In Advances in Cryptography - Crypto'2001*, pages 190–200. LNCS 2139, Springer-Verlag, 2001.

[13] A. Joux. A one round protocol for tripartite diffie-hellman. In *In Algorithmic Number Theory Symposium – ANTS-4*, pages 385–394. LNCS 1838, Springer-Verlag, 2000.

[14] N. Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48:203–209., 1987.

[15] R. Lidel and H. Niederreiter. *Introduction to Finite Fields and their Applications*. Cambridge University Press, Cambridge, Great Britain, 1986.

[16] R. Lidl. On cryptosystems based on polynomials and finite fields. In T. Beth, N. Cot, and I. Ingemarsson, editors, *Proc. EUROCRYPT 84*, pages 10–15. Springer-Verlag, 1985. Lecture Notes in Computer Science No. 209.

[17] W. B. Mao. *Modern Cryptography: Theory and Practice*. Prentice Hall PTR, 2003.

[18] A. J. Menezes, T. Okamoto, and S. A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, 39:1639–1646., Sep, 1994.

[19] V. S. Miller. Short programs for functions on curves, 1986. http://crypto.stanford.edu/miller/miller.ps.

[20] V. S. Miller. Use of elliptic curves in cryptography. In *In Advances in Cryptology - Crypto'85*, pages 417–426. LNCS 218, Springer-Verlag, 1986.

[21] V. S. Miller. The weil pairing, and its efficient calculation. *Journal of Cryptology*, 17(4):235–261., Sep., 2004.

[22] K.G. Paterson. *Cryptography from Pairings-Advances in Elliptic Curve Cryptography*, volume 317. Cambridge University Press, New York, NY, USA, 2005.

[23] R. L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-key Cryptosystems. *Communications of the ACM*, pages 120–126., Feb, 1978.

[24] Kenney Paterson S.Al-Riyami. Certificateless public key cryptograph. Cryptology ePrint Archive, Report 2003/126, 2003.

[25] M. Scott. Faster pairings using an elliptic curve with an efficient endomorphism. In *In INDOCRYPT 2005*, pages 258–269. LNCS 3797, Springer-Verlag, 2005.

[26] M. Scott and P. S. L. M. Barreto. Compressed pairings. In *In Advances in Cryptology - Crypto'2004*, pages 140–156. LNCS 3152, Springer-Verlag, 2004.

[27] Micheal Scott. Authenticated id-based key exchange and remote log-in with simple token and pin. Cryptology ePrint Archive, Report 2002/164, 2007.

[28] J. H. Silverman. *The Arithmetic of Elliptic Curves*, volume 106 of *Graduate Texts in Mathematics*. Springer, 1986.

[29] J. H. Silverman. *Advances Topics in the Arithmetics of Elliptic Curves*, volume 151 of *Graduate texts in Mathematics*. Springer, 1994.

[30] L. C. Washington. *Elliptic Curves, Number Theory and Cryptography*. CRC Press, 2003.

[31] Chang'an Zhao. Efficient computation of bilinear pairings. Doctoral Thesis of Sun Yat-Sen Univ., 2008.

致 谢

首先，由衷感谢我的导师张治国老师，本文是在他的悉心指导下完成的。感谢张治国老师的夫人张鹏老师，感谢她在生活上对我无微不至的关怀；感谢他们对我一贯的支持和爱护!

其次，感谢赵昌安、林惜斌、伍春晖师兄的指导，感谢他们在过去一年中所提供的帮助和鼓励，我将铭记一生！

最后，谨以此文献给我的父母，以报养育之恩！