Elliptic curves and Cryptography: A Pseudorandom Bit Generator and Other Tools

Burton S. Kaliski, Jr.[1]

Submitted on January 1, 1988, to the Department of Electrical Engineering and Computer Science at the Massachusetts Institute of Technology in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science.

Abstract

In recent years there has been much research concerning randomness and the generation of pseudorandom strings of bits. The purpose of this thesis is to develop a new pseudorandom bit generator based on one of the most interesting topics in cryptography today, elliptic curves. But this is not the only result, and a number of other tools, relating both to elliptic curves and to cryptography, are developed as well in the course of constructing the generator. The other tools make use of other new results in cryptography and set the ground for a general approach to generating pseudorandom bits.

This thesis outlines the problems encountered in constructing new pseudorandom bit generators and presents methods for solving them. For elliptic curves, each of the problems is solved completely in the thesis. In general there remain open questions for further research.

Among the secondary results are the following:

1. a probabilistic polynomial time algorithm for testing whether the order of an element is maximum in a finite commutative group whose size is known;

2. a new reduction which holds in any finite commutative group of bounded rank showing that if logarithms are hard to compute in the group then the most significant bit of the logarithm cannot be predicted in polynomial time with probability significantly more than 1/2.

Thesis supervisor: Ronald L. Rivest
Title: Professor of Electrical Engineering and Computer Science

[1]Author's address: 20B Foxberry Drive, Getzville, NY 14068.

Chapter 3

Elliptic Curves: Survey

Elliptic curves have long been objects of mathematical interest, and recently have become the basis for significant applications of computational number theory—factorization, primality testing, and so on. Indeed these applications are considered the first use of "20th century mathematics" in modern cryptography. While most of the work on elliptic curves relevant to recent applications has been done in this century, the study of elliptic curves is much older, beginning with certain computational problems in the 17th century. As a consequence elliptic curves have a rich history; the purpose of this chapter is to survey the major results as they result to present applications.

The chapter is divided into three parts. In Section 3.1 a survey of past work on elliptic curves is presented. The section studies elliptic integrals, elliptic functions and finally elliptic curves. Section 3.2 gives the traditional introduction to elliptic curves, including the group law and properties of elliptic curves over finite fields. In Section 3.3 recent computational results are presented. These results include Lenstra's factorization algorithm, the methods of Goldwasser-Kilian and Adleman-Huang for testing primality, Schoof's algorithm for computing the number of points on an elliptic curve and Miller's and Koblitz' cryptosystems based on the difficulty of computing elliptic logarithms.

The reader interested only in understanding the results of the thesis should begin with Section 3.2.

3.1 Survey of past work

In this section, the history of elliptic curves is surveyed. The section begins with a discussion of elliptic integrals. It continues with a presentation of two types of elliptic functions—Jacobi's and Weierstrass'. Finally it shows how elliptic curves

result from Weierstrass' elliptic functions. This section is not needed to understand the thesis, but it does help to explain why elliptic curves have been studied so much.

3.1.1 Elliptic integrals

Elliptic integrals are a nice generalization of trigonometric integrals, where one studies ellipses rather than circles. Like their trigonometric counterparts, elliptic integrals have no closed form solution. (Recall that sine and cosine are only abbreviations for infinite series.) Two problems in which elliptic integrals arise were both studied in the 17th century, and perhaps earlier. These are the computation of the arc length of an ellipse and the period of a pendulum, both of which are described next. The analysis is taken from Alling's book [All81].

Arc length of an ellipse

One finds the length of the arc running from the point $(1,0)$ to the point $(\sqrt{1-y^2}, y)$ on the unit circle as the value of the integral

$$s = \int_0^v \frac{dy}{\sqrt{1-y^2}} = \sin^{-1}(y), \quad -1 \le y \le 1. \tag{3.1}$$

In the same way one finds the arc length of the ellipse

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = 1 \tag{3.2}$$

as the difference of elliptic integrals of the first and second kinds

$$s = b\int_0^{v/b} \frac{dt}{\sqrt{(1-t^2)(1-k^2t^2)}} - bk^2\int_0^{v/b} \frac{t^2dt}{\sqrt{(1-t^2)(1-k^2t^2)}}, \quad -b \le y \le b, \tag{3.3}$$

where $k = \sqrt{a^2-b^2}$ is the eccentricity of the ellipse. In each case, the result of the integral is an inverse function of the limits of the integral, where the function is defined in such a way as to satisfy a differential equation relating to the integrand. Thus Equation 3.1 can be checked by substituting $\sin(s)$ for y.

Period of a pendulum

One application of elliptic functions is the computation of the period of a pendulum. Suppose the pendulum swings an arc of 2α radians. Let k, which is called the

modulus, take the value $sin(\alpha/2)$. Then one arrives at a definite integral, called the complete elliptic integral of the first kind:

$$\tau_0 = 4 \int_0^{\pi/2} \frac{d\phi}{\sqrt{1 - k^2 \sin^2 \phi}} \tag{3.4}$$

This integral gives the period of the pendulum in normalized units. For small values of α it is interesting to note that the modular angle is near 0 and thus the period τ_0 is roughly constant. For large values of α (near π) the period tends to infinity.

3.1.2 Elliptic functions

Neither of the two kinds of integrals mentioned (and there is a third) is solvable in closed form. Each has a solution as an infinite series—an elliptic function. Like the trigonometric functions, elliptic functions are generally found in a table of approximations, such as the CRC handbook [Bey73]. (One clever way of computing the value of the complete elliptic integral involves the arithmetic-geometric mean. This is discussed in [All81] and is given as an example of "numerical analysis with a calculator" in [Hen77].)

Many elliptic functions have been studied by mathematicians in the last three centuries. In this section, two types of elliptic functions due to Jacobi and Weierstrass are described. Jacobi's work is earlier and is a natural generalization of the trigonometric functions. Weierstrass' work is much more general and incorporates all other elliptic functions. There are other functions, such as the sine lemniscate, due to Gauss, and the theta functions, due to Abel. The primary sources for the material which follows are [All81] and [Nam84].

Jacobi's sinus amplitudinus

Jacobi's sinus amplitudinus function generalizes the trigonometric sine to allow for an eccentricity k of an ellipse; for a circle, the eccentricity is 0. The function, denoted sn_k, satisfies the equation

$$\mathrm{sn}_k \left(\int_0^{\phi} \frac{d\Phi}{\sqrt{1 - k^2 \sin^2 \Phi}} \right) = \sin \phi. \tag{3.5}$$

With the substitutions $\xi = \sin \Phi$ and $x = \sin \phi$, the equation becomes

$$\mathrm{sn}_k \left(\int_0^{x} \frac{d\xi}{\sqrt{(1 - \xi^2)(1 - k^2 \xi^2)}} \right) = x. \tag{3.6}$$

Traditionally the eccentricity or modulus k is assumed and is not written; here it is included as a parameter to the sn function for clarity.

It is worthwhile to compare the definition of the sn function with a similar definition of the trigonometric sine, which is in fact the limit of Jacobi's sn_k as k nears 0 (*i.e.*, the ellipse becomes a circle). In this case, one obtains the more familiar equation

$$\sin\left(\int_0^x \frac{d\xi}{\sqrt{1-\xi^2}}\right) = x. \tag{3.7}$$

An interesting exercise is to show that the limit of sn_k as k nears 1 is the hyperbolic tangent.

If the limit x in the integral in Equation 3.6 is allowed to be a complex number, then the integral becomes a line integral in the complex plane which may evaluate to a complex number. Thus it makes sense to allow the sn_k function to have a complex argument. The function then has an infinite series expansion defined for all complex arguments z as

$$\mathrm{sn}_k(z) = z - (1+k^2)\frac{z}{3!} + (1+14k^2+k^4)\frac{z^5}{5!} - (1+135k^2+135k^4+1)\frac{z^7}{7!} + \cdots. \tag{3.8}$$

Periodicity

It turns out that the sinus amplitudinus function so defined is periodic with periods $4K_k$ and $2iK'_k$, where

$$K_k = \int_0^1 \frac{d\xi}{\sqrt{(1-\xi^2)(1-k^2\xi^2)}} \quad \text{and} \quad K'_k = K_{\sqrt{1-k^2}}. \tag{3.9}$$

Not surprisingly, as the eccentricity k approaches 0, the periods approach 2π and an infinite imaginary value, as one would expect for the trigonometric sine. As the eccentricity approaches 1, the periods approach an infinite real value and πi, which are correct for the hyperbolic tangent. The double periodicity is essentially what sets elliptic functions apart from trigonometric functions.

Addition law and differential equation

Two other properties are of interest. First, the sn_k function satisfies an addition law in terms of the associated functions

$$\mathrm{cn}_k z = \sqrt{1-\mathrm{sn}_k^2 z} \quad \text{and} \quad \mathrm{dn}_k z = \sqrt{1-k^2\mathrm{sn}_k^2 z}. \tag{3.10}$$

The addition law is

$$\mathrm{sn}_k(z_1+z_2) = \frac{\mathrm{sn}_k z_1 \mathrm{cn}_k z_2 \mathrm{dn}_k z_2 + \mathrm{sn}_k z_2 \mathrm{cn}_k z_1 \mathrm{dn}_k z_1}{1-k^2\mathrm{sn}_k^2 z_1 \mathrm{sn}_k^2 z_2}. \tag{3.11}$$

Second, the sinus amplitudinus function satisfies the differential equation

$$\left(\frac{d}{dz}\mathrm{sn}_k z\right)^2 = \left(1 - \mathrm{sn}_k^2 z\right)\left(1 - k^2\mathrm{sn}_k^2 z\right). \tag{3.12}$$

For comparison with the trigonometric sine, observe the familiar equations

$$\sin(z_1 + z_2) = \sin z_1 \cos z_2 + \sin z_2 \cos z_1 \tag{3.13}$$

$$\left(\frac{d}{dz}\sin z\right)^2 = 1 - \sin^2 z. \tag{3.14}$$

Weierstrass form

Weierstrass observed in the mid-19th century that any elliptic integral could be transformed into the form

$$\int \frac{ds}{\sqrt{4s^3 - g_2 s - g_3}} \tag{3.15}$$

by a linear fractional transformation. He also found a general solution for this integral in terms of the coefficients g_2 and g_3. His solution is the famous Weierstrass $\wp$ (pronounced "pe") function.

As an example, the integral for the sinus amplitudinus function function

$$\int_0^x \frac{d\xi}{\sqrt{(1-\xi^2)(1-k^2\xi^2)}} \tag{3.16}$$

becomes the Weierstrass form integral

$$\int_{\frac{-k^2-5}{12}}^{\frac{(5k^2-1)x+k^2-5}{12(x-1)}} \frac{ds}{\sqrt{4s^3 - \frac{k^4+14k^2+1}{12}s - \frac{k^6-33k^4-33k^2+1}{216}}} \tag{3.17}$$

with the substitution

$$\xi = \frac{12s + k^2 - 5}{12s - 5k^2 + 1}. \tag{3.18}$$

The $\wp$ function obtains its generality by being defined modulo a lattice in the complex plane defined as follows. Given two complex numbers ω_1 and ω_2 such that the imaginary part of ω_1/ω_2 is non-zero (and, by convention, positive), a lattice $\mathcal{L}(\omega_1, \omega_2)$ is the set of all linear combinations of integral multiples of ω_1 and ω_2. The complex numbers ω_1 and ω_2 are a basis of the lattice.

The function $\wp_{\omega_1,\omega_2}$ is defined as the sum

$$\wp_{\omega_1,\omega_2}(z) = \frac{1}{z^2} + \sum_{\substack{\omega \in \mathcal{L}(\omega_1,\omega_2) \\ \omega \neq 0}} \left(\frac{1}{(z-\omega)^2} - \frac{1}{\omega^2} \right) = \sum_{\omega \in \mathcal{L}(\omega_1,\omega_2)} \frac{1}{(z-\omega)^2} - \sum_{\substack{\omega \in \mathcal{L}(\omega_1,\omega_2) \\ \omega \neq 0}} \frac{1}{\omega^2}. \tag{3.19}$$

It is not difficult to check that the function $\wp_{\omega_1,\omega_2}$ has periods ω_1 and ω_2. Because of the double periodicity, one need only be concerned with the complex numbers modulo the lattice, i.e., $\mathbf{C}/\mathcal{L}(\omega_1,\omega_2)$.

The relationship between the lattice $\mathcal{L}(\omega_1,\omega_2)$ and the coefficients g_1 and g_2 of the integral is described next.

Relationship between lattice and coefficients

The lattice for which the function $\wp_{\omega_1,\omega_2}$ is defined depends on the values of the coefficients g_2 and g_3. For a given lattice $\mathcal{L}(\omega_1,\omega_2)$, the coefficients satisfy

$$g_2 = 60 \sum_{\substack{\omega \in \mathcal{L}(\omega_1,\omega_2) \\ \omega \neq 0}} \frac{1}{\omega^4} \quad \text{and} \quad g_3 = 140 \sum_{\substack{\omega \in \mathcal{L}(\omega_1,\omega_2) \\ \omega \neq 0}} \frac{1}{\omega^6}. \tag{3.20}$$

The "inverses" of these summations make it possible to derive a unique lattice (but not unique periods ω_1 and ω_2) from the coefficients. As a consequence, one may write $\wp_{g_2,g_3}$ to refer to the function corresponding to the lattice $\mathcal{L}_{g_2,g_3}$ derived from coefficients g_2 and g_3.

To determine the lattice from the coefficients, let e_1, e_2, e_3 denote the roots of the cubic polynomial $4s^3 - g_2 s - g_3$, where $e_1 \geq e_3 \geq e_2$. (This assignment of subscripts is common though not the only one used in the literature.) Dutta and Debrath give the identity [DD65]

$$\wp_{g_2,g_3}(z) = e_2 + \frac{e_1 - e_2}{\mathrm{sn}_k^2(z\sqrt{e_1 - e_2})}, \tag{3.21}$$

where the modulus k takes the value

$$k = \sqrt{\frac{e_3 - e_2}{e_1 - e_2}}. \tag{3.22}$$

The lattice corresponding to coefficients g_2 and g_3 is thus related to the periods of the function sn_k. Notice that although the real period of the function sn_k is $4K_k$ (Equation 3.9), the period of its square, sn_k^2, is $2K_k$. Thus one assignment to the periods ω_1 and ω_2 is

$$\omega_1 = \frac{2K_k}{\sqrt{e_1 - e_2}}, \quad \omega_2 = \frac{2iK'_k}{\sqrt{e_1 - e_2}}. \tag{3.23}$$

Integral, addition law, differential equation

Like the sinus amplitudinus function, the $\wp$ function solves an integral, has an addition formula, and satisfies a differential equation. These are as follows.

$$\wp_{g_2,g_3}\left(\int_x^\infty \frac{ds}{\sqrt{4s^3 - g_2 s - g_3}}\right) = x \tag{3.24}$$

$$\wp_{g_2,g_3}(z_1 + z_2) = \frac{1}{4}\left(\frac{\wp'(z_1) - \wp'(z_2)}{\wp(z_1) - \wp(z_2)}\right)^2 - \wp(z_1) - \wp(z_2) \tag{3.25}$$

$$(\wp_{g_2,g_3}(z))^2 = \wp^3(z) - g_2\wp(z) - g_3 \tag{3.26}$$

Some other properties of the $\wp$ function are summarized in [Nam84].

Notice that Equation 3.38 is identical to the addition formula for the Weierstrass $\wp$ function given in Equation 3.25, except for a factor of 1/4 due to an alternate representation.

3.1.3 Elliptic curves

The addition laws and differential equations of the various elliptic functions lead naturally to a geometric interpretation of elliptic functions, the elliptic curve. It is this point with which most 20th century and almost all recent work has been concerned. Thus elliptic curves are often studied, not in terms of the $\wp$ function but rather in terms of points on an elliptic curve resulting from Equation 3.26,

$$y^2 = 4x^3 - g_2 x - g_3, \quad \Delta = g_2^3 + 27g_3^2 \neq 0. \tag{3.27}$$

The quantity Δ, called the discriminant, is constrained to be nonzero so that the polynomial $4x^3 - g_2 x - g_3$ has distinct roots. In the literature the alternate form $y^2 = x^3 + ax + b$ is often used.

A geometric interpretation is possible because the mapping

$$z \in \mathbf{C}/\mathcal{L}_{g_2,g_3} \mapsto (\wp_{g_2,g_3}(z), \wp'_{g_2,g_3}(z)) \in \mathbf{C} \times \mathbf{C} \tag{3.28}$$

is invertible; that is, every point (x, y) on the curve defined by Equation 3.27 corresponds to a unique value z modulo the lattice $\mathcal{L}_{g_2,g_3}$. As a direct consequence, the addition law (Equation 3.25) becomes a group law for points on the curve. The identity of the group, *i.e.*, $(\wp_{g_2,g_3}(0), \wp'_{g_2,g_3}(0))$, is a point at infinity in $\mathbf{C} \times \mathbf{C}$ and is usually denoted O.

Figure 3.1 contains an example of an elliptic curve. The curve corresponds to the Weierstrass integral resulting from Jacobi's sinus amplitudinus function $\mathrm{sn}_{1/2}$, with a slight change in scale.

Group law

The group law for an elliptic curve E corresponds to addition modulo the lattice as follows:

$$\left(\wp_{g_2,g_3}(z_1), \wp'_{g_2,g_3}(z_1)\right) + (\wp(z_2), \wp'(z_2)) = (\wp(z_1+z_2), \wp'(z_1+z_2)). \qquad (3.29)$$

The outer addition is the group law for points on the curve; the inner additions are in the complex plane modulo the lattice $\mathcal{L}_{g_2,g_3}$.

This group law can be expressed as rational polynomials and therefore can be adapted to an arbitrary field K. The set of all points on an elliptic curve E defined over a field K is denoted $E(K)$; this forms a group using the rational polynomials as group law. Two types of fields for which elliptic curves have been studies are rational numbers and finite fields. These are described next.

Over rational numbers

The study of elliptic curves over rational numbers and integers is much like ancient Diophantine analysis, which asks whether an equation or system of equations has non-trivial rational or integral solutions.[1] In the 1920's Mordell and also Weil showed that the group $E(\mathbf{Q})$ could be generated by applying the group law repeatedly to a fixed, finite set of points on the curve. This is in contrast to the field $\mathbf{Q}$ itself, which has an infinite number of generators (essentially, the primes).

Over finite fields

The work over the rational numbers leads naturally to finite fields through a "reduction" modulo a prime p. In this reduction, an element r of the field $\mathbf{Q}$ is transformed to an element of the finite field with p elements, denoted $\mathbf{F}_p$, by modulo p division of the numerator of r by the denominator of r. This procedure transforms the group $E(\mathbf{Q})$ to the group $E(\mathbf{F}_p)$ if all of the divisions are defined.

Elliptic curves may also be defined over the field $\mathbf{F}_q$ with q elements, where q is a power of a prime, and over the algebraic closure $\overline{\mathbf{F}_q}$ of the finite field.[2] In either case, the coefficients of the equation defining the elliptic curve are usually restricted to the finite field $\mathbf{F}_q$, and the discriminant $g_2^3 + 27g_3^2$ is required to be nonzero. The group $E(\overline{\mathbf{F}_q})$ is analogous to the group $E(\mathbf{C})$ (over the complex plane) since every element of the field is an x-coordinate of at least one point. (The same is true for

[1] An example is Fermat's Last "Theorem" that the equation $x^n + y^n = x^n$ has no non-trivial solutions if $n > 2$.

[2] The *algebraic closure* $\overline{\mathbf{F}_q}$ of a finite field $\mathbf{F}_q$ consists of all roots of polynomials $a_n x^n + \cdots + a_1 x + a_0$ with coefficients a_i in the finite field.

y-coordinates.) In contrast, the group $E(\mathbf{F}_q)$, a subgroup of $E(\overline{\mathbf{F}_q})$, is analagous to the group $E(\mathbf{R})$, a subgroup of $E(\mathbf{C})$, since not every element of the field is necessarily an x-coordinate of at least one point.

The relationship of the complex numbers modulo a lattice $\mathcal{L}$ with the elliptic curve makes subgroups of the elliptic curve over the closure $\overline{\mathbf{F}_q}$ doubly periodic. The curve has these properties over the closure, rather than the finite field itself, since the closure contains the roots of all polynomials defined over the finite field. The expansion of the group law gives degree l^2 polynomials whose roots are the x- or the y-coordinates of the points of order dividing an integer l on the elliptic curve. Thus in general there are l^2 points in the subgroup denoted $E(\overline{\mathbf{F}_q})[l]$, assuming l is relatively prime to q, and the subgroup has the structure

$$E(\overline{\mathbf{F}_q})[l] \cong (\mathbf{Z}/l\mathbf{Z}) \times (\mathbf{Z}/l\mathbf{Z}). \tag{3.30}$$

For integers l not relatively prime to q, the group structure is degenerate.

Returning to the finite field, consider the set of points in the subgroup $E(\overline{\mathbf{F}_q})[l]$ with coordinates in the finite field $\mathbf{F}_q$. Since the composition operations are rational, the set is closed under composition and thus it forms a subgroup $E(\mathbf{F}_q)[l]$ of $E(\overline{\mathbf{F}_q})[l]$. As a result, for some integers l_1 and l_2 dividing l, one has the isomorphism

$$E(\mathbf{F}_q)[l] \cong (\mathbf{Z}/l_1\mathbf{Z}) \times (\mathbf{Z}/l_2\mathbf{Z}). \tag{3.31}$$

Without loss of generality one may assume that l_2 divides l_1.

3.2 Modern introduction

This section gives a modern introduction to elliptic curves, which is generally the one found in the literature. It is divided into three parts. The first defines elliptic curves generally and presents the group law for Cartesian coordinates. The same group law holds over finite fields. The second part shows the important properties of elliptic curves over finite fields. In the first and second parts an example elliptic curve is given; this example is carried into Chapter 6. The third part mentions the properties of elliptic curves over rings, as used in Lenstra's and other algorithms.

The reader interested in a description of the origins of elliptic curves should turn to Section 3.1.

3.2.1 Definition and group law

An *elliptic curve* E is the set of solutions (x, y) to the equation

$$E: y^2 = x^3 + ax + b, \quad 4a^3 - 27b^2 \neq 0, \tag{3.32}$$

together with a point at infinity denoted O. The points form a commutative group under the "tangents and chords" group law described next.

Tangents and chords operation

The group law is traditionally called the "tangents and chords" operation, based on a geometric interpretation of Equation 3.25. The "sum" of two points $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$, written $P_1 + P_2$, is the reflection across the x-axis of the unique additional point of intersection of a line through points P_1 and P_2 with the elliptic curve E. If the points P_1 and P_2 are distinct than the line is a chord; if they are identical, then the line is a tangent. If the points are reflections of one another across the x-axis, their sum the point O at infinity.

There are three cases for the application of the group law to add points P_1 and P_2.

Case 1 (a) $P_1 = O$. Then $P_1 + P_2 = P_2$.

(b) $P_2 = O$. Then $P_1 + P_2 = P_1$.

Case 2 $P_1 = (x_1, y_1) \neq O$, $P_2 = (x_2, y_2) \neq O$, $P_1 \neq P_2$.

(a) $x_1 = x_2$ and $y_1 = -y_2$. Then $P_1 + P_2 = O$. (Chord is vertical.)

(b) $x_1 \neq x_2$ or $y_1 \neq -y_2$. Then $P_1 + P_2 = (x_3, y_3)$ where x_3 and y_3 are obtained as follows.

The chord through points P_1 and P_2 satisfies the equation

$$y = \lambda x + \mu \tag{3.33}$$

where the slope λ is

$$\lambda = \frac{y_1 - y_2}{x_1 - x_2} \tag{3.34}$$

and the y-intercept μ is

$$\mu = \frac{x_1 y_2 - x_2 y_1}{x_1 - x_2}. \tag{3.35}$$

The chord intersects the curve $y^2 = x^3 + ax + b$ at points with x-coordinates satisfying

$$x^3 - \lambda^2 x^2 + (a - 2\lambda\mu)x + (b - \mu^2) = 0. \tag{3.36}$$

The three values x_1, x_2 and x_3 thus satisfy the sum

$$x_1 + x_2 + x_3 = \lambda^2, \tag{3.37}$$

so that the x-coordinate of the sum is

$$x_3 = \lambda^2 - x_1 - x_2 = \left(\frac{y_1 - y_2}{x_1 - x_2}\right)^2 - x_1 - x_2. \tag{3.38}$$

The y-coordinate is reflected across the x-axis, *i.e.*

$$y_3 = -(\lambda x_3 + \mu). \tag{3.39}$$

Case 3 $P_1 = (x_1, y_1) \neq O$, $P_2 = (x_2, y_2) \neq O$, $P_1 = P_2$.

(a) $y_1 = 0$. Then $P_1 + P_2 = O$.

(b) $y_1 \neq 0$. Then $P_1 + P_2 = (x_3, y_3)$ where x_3 and y_3 are obtained as follows. The tangent at point P_1 satisfies the equation

$$y = \lambda x + \mu \tag{3.40}$$

where the slope λ is

$$\lambda = \frac{3x_1^2 + a}{2y_1} \tag{3.41}$$

and the y-intercept μ is

$$\mu = y_1 - \frac{3x_1^3 + ax_1}{2y_1}. \tag{3.42}$$

The x- and y-coordinates of the sum are thus

$$x_3 = \lambda^2 - 2x_1 = \left(\frac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1 \tag{3.43}$$

and

$$y_3 = -(\lambda x_3 + \mu). \tag{3.44}$$

Notice that the only points P which are self-inverse ($P + P = O$) are O and points with y-coordinate 0 (case 3(b)). This fact is used in the proof of Lemma 6.2.

An example

As an example of an elliptic curve, consider the following equation. Section 3.1.3 explains its significance.

$$y^2 = x^3 - \frac{73}{768}x + \frac{595}{55296} = \left(x + \frac{17}{48}\right)\left(x - \frac{7}{48}\right)\left(x - \frac{5}{24}\right). \tag{3.45}$$

As is typically done, the curve is plotted for real values of x and y. Notice in Figure 3.1 that the points on the x-axis are clearly self-inverse, since their tangents intersect the point at infinity.

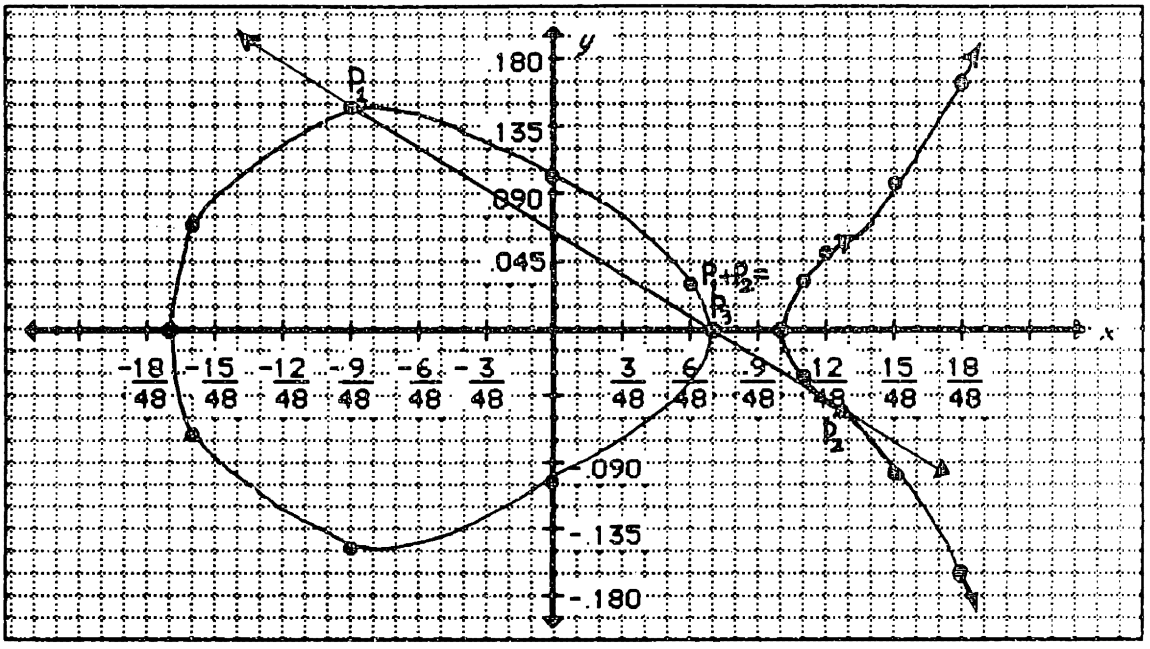

Figure 3.1: An elliptic curve. Point P_3 is the sum of points P_1 and P_2 and is self-inverse.

3.2.2 Properties over finite fields

When the equation for an elliptic curve is considered over a finite field $\mathbf{F}_q$ where q is a prime, rather than over the real or complex numbers, one obtains a finite group denoted $E(\mathbf{F}_q)$. In particular, the equation for an elliptic curve may be considered modulo a prime number p, giving a finite group $E(\mathbf{F}_p)$. The finite groups have many interesting properties. Two of them are very useful in the thesis: bounds on the number of points in the group and on its structure.

Size of elliptic curves

Some of the early work on elliptic curves over finite fields includes Hasse's important inequality (proved in the 1930's) concerning the number of points n, including the point at infinity, on an elliptic curve $E(\mathbf{F}_q)$. Hasse showed that

$$|n - (q+1)| \leq 2\sqrt{q}. \tag{3.46}$$

The result also holds in a related form for Abelian varieties, which are generalizations of elliptic curves. Hasse's result is commonly called a Riemann Hypothesis for elliptic curves.

point	image in $(\mathbf{Z}/4\mathbf{Z}) \times (\mathbf{Z}/2\mathbf{Z})$
O	(0,0)
(0,0)	(0,1)
(2,5)	(1,0)
(6,6)	(1,1)
(3,0)	(2,0)
(4,0)	(2,1)
(2,2)	(3,0)
(6,1)	(3,1)

Table 3.1: Structure of the elliptic curve $y^2 = x^3 + 5x$ over $\mathbf{F}_7$.

Group structure

For some integers n_1 and n_2, where n_2 divides n_1, the elliptic curve $E(\mathbf{F}_q)$ satisfies the isomorphism

$$E(\mathbf{F}_q) \cong (\mathbf{Z}/n_1\mathbf{Z}) \times (\mathbf{Z}/n_2\mathbf{Z}) \tag{3.47}$$

Thus the group has rank at most 2. The group structure is represented by the symbols n_1 and n_2, with both the curve and finite field implied. It is well known (and will be shown in Chapter 4) that the value n_2 in the group structure divides $q-1$.

The example over a finite field

As an example of elliptic curves over a finite field, consider Equation 3.45 over the finite field $\mathbf{F}_7$:

$$y^2 = x^3 + 5x \bmod 7. \tag{3.48}$$

The curve has eight points, satisfying Equation 3.46; it is isomorphic to the group $(\mathbf{Z}/4\mathbf{Z}) \times (\mathbf{Z}/2\mathbf{Z})$, satisfying Equation 3.47. Table 3.1 illustrates the structure of the group. Notice that the points $(x, 0)$ in Figure 3.1 become points $(0, 0)$, $(3, 0)$ and $(4, 0)$ when reduced modulo 7.

3.2.3 Projective coordinates and elliptic curves over rings

When the elliptic curve is expressed in projective coordinates the addition laws can be implemented more efficiently.[3] The equation defining an elliptic curve in

[3]Projective coordinates partition the nonzero elements of the set K^3 into equivalence classes where two elements (x_1, y_1, z_1) and (x_2, y_2, z_2) are considered equivalent if and only if for some value

projective coordinates is homogeneous:

$$y^2z = x^3 + axz^2 + bz^3. \tag{3.49}$$

In projective coordinates, the point at infinity has coordinates $(0:y:0)$ for $y \neq 0$. A point with affine coordinates (x, y) has projective coordinates $(xz:yz:z)$ for any value $z \neq 0$.

The improvement in efficiency comes from the observation that since there are many representatives for the sum $P_1 + P_2$, the addition formulas can be redefined so that no divisions are required. The lack of divisions also generalizes the group law to work over rings, not only fields. The properties of elliptic curves over rings (specifically, the integers modulo a composite number) are useful in Lenstra's and other algorithms described in Section 3.3.

An elliptic curve defined over a ring $(\mathbf{Z}/n\mathbf{Z})$ has the following important property. If the integer n has the factorization $p_1^{\alpha_1} \cdot \cdots \cdot p_k^{\alpha_k}$, then one has for each prime factor p_i the homomorphism

$$h_i: E(\mathbf{Z}/n\mathbf{Z}) \to E(\mathbf{F}_{p_i}) \tag{3.50}$$

The homomorphism is simply reduction of the projective coordinates of a point modulo the factor p_i. As a result of the homomorphism, the number of points on the curve $E(\mathbf{Z}/n\mathbf{Z})$ is divisible by the number of points on each of the curves $E(\mathbf{F}_{p_i})$.

3.3 Recent results

Since 1985 elliptic curves have been of great interest in computational number theory, with several major results. Lenstra's algorithm for integer factorization using elliptic curves has become the best algorithm for noncryptographic applications. The primality testing algorithms of Goldwasser-Kilian and Adleman-Huang have solved the problem of producing a proof that a number is prime. A third result is Schoof's algorithm for computing the number of points on an elliptic curve. In addition, Miller's and Koblitz' adaptations of cryptosystems based on discrete logarithms have introduced elliptic curves into cryptography. Lenstra's, Schoof's and Miller's work all have direct applications to new applications obtained in this thesis.

For a survey which is quite thorough in describing the recent developments in computational number theory using elliptic curves, the reader is referred to a paper by Lenstra and Lenstra [LL87], scheduled for publication in *Handbook of Theoretical Computer Science.*

$\alpha \in K$, $x_1 = \alpha x_2$, $y_1 = \alpha y_2$, $z_1 = \alpha z_2$. An equivalence class is denoted by replacing commas by colons in some member of the class.

3.3.1 Lenstra's factorization algorithm

Elliptic curves can be said to have gained importance in the last three years because of Lenstra's new algorithm for integer factorization [Len]. The algorithm prompted all the other recent computational results involving elliptic curves. (Although Schoof's result appeared first, Lenstra played a role in motivating Schoof's as well [Sch85a]). Lenstra's algorithm is asymptotically no faster than other algorithms in the worst case, but it is now the method of choice for factoring "naturally occurring" integers, as opposed to those generated for cryptographic purposes. In addition, Lenstra's algorithm proves useful in the new results of Chapter 4 for computing the order of an element in an arbitrary commutative group. For this reason Lenstra's algorithm is described in more detail than the other results.

Pollard's $p-1$ method

Before describing Lenstra's algorithm, it is useful to describe a similar but less effective algorithm due to Pollard [Pol74]. The algorithm, known as the $p-1$ method.

In this method, one obtains a factorization of an integer n by working in the multiplicative group $(\mathbf{Z}/n\mathbf{Z})^*$. Analogous to Equation 3.50, for each prime factor p_i of n there is a reduction modulo p_i homomorphism

$$h_i\colon (\mathbf{Z}/n\mathbf{Z})^* \to (\mathbf{Z}/p_i\mathbf{Z})^*. \tag{3.51}$$

Consequently, if some prime factor p_i satisfies the (unlikely) condition that all prime factors of $p_i - 1$ are small, then one can determine p_i from the value $\gcd(x^m, n)$, where m is the product of small prime numbers and x is an element selected at random. However, since such factors are unlikely, Pollard's method is not effective on arbitrary integers n.

Lenstra's method

The essential property of Lenstra's algorithm is Equation 3.50, the homomorphism of the group $E(\mathbf{Z}/n\mathbf{Z})$. Let P be a point on the elliptic curve $E(\mathbf{Z}/n\mathbf{Z})$. Then for any integer m and factor p_i of n, the homomorphism and repeated composition can be done in any order, *i.e.*, $h_i(mP) = mh_i(P)$. Suppose the integer m is divisible by the size of one of the groups $E(\mathbf{F}_{p_i})$. Then the point $mh_i(P)$ is the identity $(0 : y : 0)$ of the group $E(\mathbf{F}_{p_i})$. As a consequence, the z-coordinate of the point $mP \in E(\mathbf{Z}/n\mathbf{Z})$ is divisible by p_i.

By proper choice of the integer m, it is possible to extract a factor p_i simply by taking the gcd of the integer n and the z-coordinate of the point mP. (For simplicity, one may assume that the integer m is divisible by the sizes of only one of

the groups $E(\mathbf{F}_{p_i})$.) This method of extracting factors leads naturally to a recursive algorithm for finding all the factors p_i of an integer n. Thus the algorithm can be summarized by describing how to find a single factor. This involves two steps:

1. Choose an elliptic curve $E(\mathbf{Z}/n\mathbf{Z})$ and a point P on the curve at random. This can be accomplished by choosing a point $P = (x : y : 1)$ (since the identity need not be considered), and then a curve $E(\mathbf{Z}/n\mathbf{Z})$ on which the point P lies. The curve can be selected by choosing a coefficient a and solving for the coefficient b as
$$b = y^2 - x^3 - ax \bmod n, \tag{3.52}$$
subject to the condition $\gcd(4a^3 - 27b^2, n) \neq 0$. The condition guarantees that the groups $E(\mathbf{F}_{p_i})$ are defined for each factor p_i of n, and of course if the condition is not met then a factor of n has been found.

2. Compute the point mP by successive doubling. If the z-coordinate and the integer n have a common factor, output the factor and exit; else go to step 1.

The random selection of an elliptic curve gives Lenstra's algorithm its power. Unlike Pollard's $p - 1$ method and others related to it, in which the sizes of the groups corresponding to the factors p_i are fixed, in Lenstra's algorithm the sizes may vary and therefore may divide a predetermined integer m. The probability that one of the groups divides an integer m depends on the distribution of smooth numbers near each factor p_i. Let M, the "bound," be an integer, and define m as the product
$$m = \prod_{\substack{q=2 \\ q \text{ prime}}}^{M} q^{\lfloor \log_q(n+1+2\sqrt{n}) \rfloor}. \tag{3.53}$$
To find a factor p_i, the size of the group $E(\mathbf{F}_{p_i})$ must be smooth with respect to the bound M. That is, all the prime factors of the size must be M or smaller. The exponent in Equation 3.53 handles the possible multiplicity of prime factors in the size of the group, using the observation based on Equation 3.46 that
$$\#E(\mathbf{F}_{p_i}) \leq p_i + 1 + 2\sqrt{p_i} \leq n + 1 + 2\sqrt{n}. \tag{3.54}$$
Lenstra shows that the possible sizes of a group $E(\mathbf{F}_{p_i})$ are fairly evenly distributed across the interval allowed by Equation 3.46. Thus the probability that the size is smooth with respect to the bound M is directly related to the probability that a randomly selected integer near the factor p_i is smooth with respect to the same bound. Under certain assumptions, this probability is
$$\left(\log_{p_i} M\right)^{\log_M p_i}. \tag{3.55}$$

The assumptions include heuristic arguments on the distribution of smooth numbers in an interval. More rigorous results concerning the asymptotic distribution of smooth numbers can be found in papers by Pomerance and others [Pom85a] [CEP83] [KT76]. Whether there are enough smooth numbers near a given factor p_i is unknown, and for this reason Lenstra's algorithm may fail to find a given factor. Such factors are probably rare, if any exist.

Running time

The expected running time of the algorithm is proportional to the time to compute the point mP, multiplied by the expected number of iterations to find a curve with smooth size. To find the prime factor p_i, the time is

$$O\left(\frac{\log m \log^2 n}{\left(\log_{p_i} M\right)^{\log_M p_i}}\right) \approx O\left(M \log^3 n (\log_M p_i)^{\log_M p_i}\right). \tag{3.56}$$

The running time is minimized (see [Bac85] for details) by choosing the bound M such that

$$\log_M p_i \approx \sqrt{\frac{2 \ln p_i}{\ln \ln p_i}}; \tag{3.57}$$

Since the factor p_i is not known in advance, one may systematically increase the bound M at each iteration of the algorithm. For purposes of analysis, one may assume the factor p_i is known. Typically, Lenstra's algorithm will find the smallest prime factor of the integer n first, since the expected running time is least for this factor.

The total running time for Lenstra's algorithm depends on the time to recover all prime factors of the integer n except the largest. In general this time is bounded by the time to find the second largest prime factor, call it p_2. Using the standard notation for expressing running times of factorization algorithms [Pom85a],

$$L(x) = \exp(\sqrt{\ln x \ln \ln x}), \tag{3.58}$$

one obtains a running time of $O(L(p_2)^{\sqrt{2}+o(1)} \log^3 N)$ for Lenstra's algorithm. Thus Lenstra's algorithm is faster than other methods for factoring numbers with small prime factors. In cryptanalytic uses, such as factoring an RSA modulus [RSA78], one has $p_2 \approx \sqrt{n}$ and the running time is the same as other algorithms, namely $O(L(n)^{1+o(1)})$. Still, it has advantages in that it uses very little memory and is easily performed in parallel.

Most of the recent additions to the table containing factorizations of number of the form $2^\nu \pm 1$, $3^\nu \pm 1$, etc. have been obtained using Lenstra's algorithm. Such

factorizations using other algorithms are described in [BLS*83]. An early implementations is that of Chudnovsky and Chudnovsky [CC86]. Lenstra's algorithm also useful as a subroutine of the quadratic sieve algorithm [Pom85b] and in computing partial factorizations efficiently (Chapter 4).

3.3.2 Goldwasser-Kilian's and Adleman-Huang's primality testing algorithms

In both theoretical and practical senses, it is desirable to have proofs rather than probabilistic evidence that a number is prime. An example of a proof of primality is that used in Pratt's $\mathcal{NP}$ algorithm for recognizing primes [Pra75]. Until recently, all polynomial time algorithms for recognizing primes, such as that of Solovay and Strassen [SS77], produced proofs of *compositeness* and probabilistic evidence of primality based on the lack of finding a proof of compositeness quickly. However, if one assumes the Extended Riemann Hypothesis, as in [Mil76], proofs of compositeness and primality can be found in polynomial time. The results described in this section are not of direct bearing to the new results in the thesis.

Pollard-like method

As in the case with Lenstra's algorithm, there is a simpler but less effective method for solving the problem (in this case, primality testing) which does not involve elliptic curve. For factorization this is Pollard's $p-1$ method. Here there is no chronological predecessor of the same nature, but in retrospect the following method is suggested.

Observe that if a number n is composite then at least one of its factors, say p_1, must be no greater than $\sqrt{n}$. Consequently every element a in $(\mathbf{Z}/n\mathbf{Z})^*$ for which $\gcd(a-1,n)=1$ and whose order is prime must have order at most $\sqrt{n}$. The proof is as follows. Suppose the prime order is q. Then one has

$$a^q \equiv 1 \bmod n, \tag{3.59}$$

and thus

$$a^q \equiv 1 \bmod p_1. \tag{3.60}$$

Since q is prime and $a \not\equiv 1 \bmod p$, one must have $q < p_1 \leq \sqrt{n}$.

This suggests a proof of primality: Find an element $a \in (\mathbf{Z}/n\mathbf{Z})^*$ with $\gcd(a-1,n) = 1$ and prime order q exceeding $\sqrt{n}$; then n must be prime. The proof is recursive, since one must also show that q is prime. The reason this proof of primality was never proposed is that not all numbers n have elements a of the proper type. The advantage of elliptic curves is that one can select many curves

corresponding to the integer n, and with high probability a curve which has a point of the right type can be found.

Goldwasser-Kilian's method

Goldwasser and Kilian [GK86] use a variant of the Pollard-like method in which one looks for an elliptic curve $E(\mathbf{Z}/n\mathbf{Z})$ whose size is twice a prime q, together with a point P of order q. Once q is shown to be prime, it follows that n is prime. And since q is about half of n, the recursion terminates in $O(\lg n)$ steps.

The expensive part of the method is computing the size of the elliptic curve $E(\mathbf{Z}/n\mathbf{Z})$ using Schoof's algorithm (see next section). The theoretical drawback to the method is that for some integers n there may not be enough elliptic curves $E(\mathbf{Z}/n\mathbf{Z})$ whose sizes are twice a prime. One way to overcome this difficulty is to consider elliptic curves whose sizes are three times a prime, four times a prime, and so on. The problem is not with the algorithm but rather with what is known about the distribution of prime numbers, not only around n but around all numbers encountered in the recursion. So it is possible that there are some integers n which cannot be shown to be prime by this method, though none is known.

Adleman-Huang's method

Adleman and Huang [AH87] solve the problem in the previous algorithm by an upward recursion. The upward step generates an *increasing* sequence of numbers whose primality is to be proved. The step involves the Jacobians of hyperelliptic curves, basically a generalization of divisors of elliptic curves. The analysis is quite complicated, but the main result is that the increasing sequence reaches in expected polynomial time a number which Goldwasser and Kilian's downward recursion can prove is a prime.

3.3.3 Schoof's algorithm for computing size of an elliptic curve

For many practical uses of elliptic curves over finite fields, it is important to be able to determine the number of points on the elliptic curve. For instance, to find the group structure and a generating pair in Chapter 4, the number of points is needed. It is clear that counting the number of pairs (x, y) which satisfy the equation defining the elliptic curve is not efficient. Using the algebra of endomorphisms, however, such counting is unnecessary and Schoof has developed a algorithm [Sch85b] for computing the number of points on an elliptic curve $E(\mathbf{F}_q)$ which has running time

$O(\log^{6+o(1)} q)$. Schoof's result is used in the algorithms for selecting elliptic curves and pairs of generators (Chapter 6).

The idea behind Schoof's algorithm is to study the properties of certain mappings called endomorphisms defined as follows. An endomorphism f maps an elliptic curve to itself, usually over the closure $\overline{\mathbf{F}_q}$, such that for all points $P, Q \in E(\overline{\mathbf{F}_q})$,

$$f(P) + f(Q) = f(P + Q). \tag{3.61}$$

Endomorphisms can be added and "multiplied" (*i.e.*, composed), and in this sense every elliptic curve has a ring of endomorphisms. The Frobenius endomorphism, denoted ϕ, maps points $(x, y) \in E(\overline{\mathbf{F}_q})$ to points (x^q, y^q). The points (x^q, y^q) are also on the curve $E(\overline{\mathbf{F}_q})$ since the coefficients a and b of the equation defining the curve are in the finite field. This can be checked as follows:

$$(y^q)^2 = (y^2)^q = (x^3 + ax + b)^q = (x^3)^q + a^q x^q + b^q = (x^q)^3 + ax^q + b. \tag{3.62}$$

For points in the elliptic curve $E(\mathbf{F}_q)$ over the finite field, the Frobenius endomorphism acts as the identity.

Allowing an integer i to represent the endomorphism $P \mapsto iP$, the Frobenius endomorphism satisfies, for some integer t, the equation in the ring of endomorphisms

$$\phi^2 - t\phi + q = 0, \quad \textit{i.e.}, \ (x^{q^2}, y^{q^2}) - t(x^q, y^q) + q(x, y) = O. \tag{3.63}$$

The integer t for which the equation holds is called the *trace* of Frobenius. In the elliptic curve over the finite field, $E(\mathbf{F}_q)$, Equation 3.63 becomes

$$(x, y) - t(x, y) + q(x, y) = O. \tag{3.64}$$

In other words, every point is "annihilated" by $1 - t + q$. In fact, it turns out that the value $1 - t + q$ is the number of points on the curve.

Schoof shows how to compute the value of the trace t modulo small prime numbers l. Given enough such computations, one can construct the integer value of t. It is possible to "guess" the value of t modulo l, say τ, simply by checking for which value τ the equation

$$\phi^2 - \tau\phi + (q \bmod l) = 0 \tag{3.65}$$

holds for all points in the group $E(\overline{\mathbf{F}_p})[l]$. Since all points in the group can be characterized by expanding the composition formulas l times, it is possible to do the checking with rational polynomial manipulations.

Deterministic algorithm for square roots

An application of Schoof's algorithm, also described in his paper, is the computation of square roots of small values x in the field $\mathbf{F}_q$. The algorithm is deterministic and has running time proportional to the absolute value of x and the time to run Schoof's main algorithm. It is significant primarily because it is deterministic, and it is in fact useful in some situations.

3.3.4 Miller's and Koblitz' cryptosystems

Inasmuch as the difficulty of computing elliptic logarithms is the basis for the pseudorandom bit generator constructed in the thesis as well as the cryptosystems proposed by Miller and Koblitz, it is worthwhile to review why elliptic logarithms are considered so difficult. Before this discussion, three cryptosystems based on elliptic curves are presented.

Elliptic cryptosystems

The cryptosystems based on elliptic curves, each an adaptation of conventional cryptosystems, are the following. For illustration, let Alice and Bob be the parties involved in the cryptosystem. The adaptations are discussed in the papers of Miller [Mil86] and Koblitz [Kob87].

1. *Diffie-Hellman key exchange* [DH76] [Mil86]*:* Alice and Bob agree on a key for use in a private key cryptosystem by the following exchange. An elliptic curve and a point P are public information. Alice selects an integer a at random and sends the point aP to Bob; Bob likewise selects an integer b and sends bP. Each then computes the point abP from which the key can be extracted (perhaps from the x-coordinate).

2. *Massey-Omura cryptosystem* [WW84] [Kob87]*:* An elliptic curve and its size n are public information. Alice sends Bob a point P as follows. Alice selects an integer c and its modulo n inverse c'; Bob likewise selects an integer d and its inverse d'. Alice first sends the point cP and Bob replies with the point dcP. Alice then sends the point $c'dcP = dP$ and Bob computes the point $d'dP = P$.

3. *ElGamal cryptosystem* [ElG85] [Kob87]*:* An elliptic curve, a point G, and points aG (for Alice) and bG (for Bob) are public information; the integers a and b are Alice and Bob's respective secret information. Alice sends Bob a point P by selecting an integer k at random and sending the pair $(kG, P + kaG)$, from which Bob, knowing the integer a, can extract the point P.

Breaking the adapted Massey-Omura and ElGamal cryptosystems is as hard as computing elliptic logarithms. Breaking the adapted Diffie-Hellman key exchange is considered to be as hard as computing elliptic logarithms, but this is not known to be true.

One issue in adapting a cryptosystem to elliptic curves is the embedding of messages into points. In general one deals with a consecutive set of integers (or even all integers represented by a given number of bits), rather than points directly. Koblitz suggests three methods, two of them probabilistic, for embedding messages into points. In these methods Koblitz focuses on finding points which correspond to a given message, rather than on constructing a mapping between points and messages. A new method which involves all points on an elliptic curve (or two curves) is presented in Chapter 6.

Elliptic logarithms

The discussion of the difficulty of computing elliptic logarithms is divided into several parts. First, a general paradigm for computing logarithms in a finite, commutative group with n elements is given, leading to an $O(\sqrt{n})$ algorithm. Second, the method by which Adleman computes discrete logarithms in $O(\exp(\sqrt{\ln n \ln \ln n}))$ is discussed. Finally, Miller's arguments for why Adleman's method cannot be extended to elliptic curves are presented.

Basic paradigm

The basic paradigm for computing logarithms in a finite, commutative group involves two steps. In the first step, a database is constructed, allowing one to compute logarithms "efficiently" for some fraction of "good" elements in the group. In the second step, the element whose logarithm is desired is composed with random elements whose logarithms are known, and the database is applied to the result. With some probability the result is "good," and the correct logarithm, which can be checked, is computed.

To be specific, let A be the group, let G be the generator, and let X be the element. Suppose initializing the database takes $i(n)$ steps, and with the database one can compute logarithms in expected time $t(n)$ for a fraction $p(n)$ of elements in the group A. Then the expected time to compute the logarithm $\log_G(X)$ is $i(n)+t(n)/p(n)$. (Note that this value ignores the time required to compose X with random elements.) A tradeoff is clear: One would like to maximize the fraction of good elements while minimizing initialization and database computation time.

Exponential time algorithm

A straightforward application of the paradigm is the following. Select any $O(\sqrt{n})$ elements with known logarithms at random and store these elements, together with their logarithms, in a lookup table. Initialization time is $\sqrt{n}$ group operations; expected database computation time is constant, if the lookup table is a hash table of size $2\sqrt{n}$; and the fraction of good elements is $1/\sqrt{n}$. Thus the expected time to compute the logarithm of any element is $O(\sqrt{n})$. This method is essentially the same as the "baby step, giant step" technique in which the values $G, 2G, 3G, \ldots, \sqrt{n}G$ (note additive notation) are stored in a hash table, and the values $X, X+\sqrt{n}G, X+2\sqrt{n}G, \ldots, x+(n-\sqrt{n})G$ are looked up. The method is easily generalized to noncyclic groups, leading to an $O(\sqrt{n})$ time algorithm for computing logarithms in any finite, commutative group, including elliptic curve groups.

Subexponential time algorithm for discrete logarithms

In the case of discrete logarithms one can do much better. (Here the group A is $(\mathbf{Z}/p\mathbf{Z})^*$ where p is a prime.) In this case the database consists of the logarithms of small prime numbers modulo p. Good elements are those which, considered as integers, are divisible only by small prime numbers. The value of $t(n)$ is then just the time to factor a good or "smooth" element, and the fraction $p(n)$ is the density of smooth elements. Initialization involves finding enough smooth numbers with known logarithms to compute, by Gaussian elimination, the logarithms of the small primes. Underlying the method is the logarithmic identity (note multiplicative notation)

$$\log_g(p_1^{\alpha_1} \cdots p_k^{\alpha_k}) = \alpha_1 \log_g(p_1) + \cdots + \alpha_k \log_g(p_k). \tag{3.66}$$

With proper definition of what it means for a prime number to be small, it is possible to obtain a subexponential, $O(\exp(\sqrt{\ln n \ln\ln n}))$ time algorithm [Adl79]. Methods involving Equation 3.66 are termed "index calculus" [Odl85].

Lifting

Adleman's method relies on a subtle property of the group $(\mathbf{Z}/p\mathbf{Z})^*$, which is that factoring makes sense. One can factor elements of the group $(\mathbf{Z}/p\mathbf{Z})^*$ because of their dual representation as integers. Essentially one is *lifting* elements of the group $(\mathbf{Z}/p\mathbf{Z})^*$ to the integers. Let l denote this lifting map and let r be the map reducing integers modulo p. Then, for all elements x in $(\mathbf{Z}/p\mathbf{Z})^*$ and integers y, z, one has the following obvious but important properties:

$$r(l(x)) = x; \tag{3.67}$$

$$r(yz) = r(y)r(z). \tag{3.68}$$

In a sense a homomorphism is taking place between the integers and the group $(\mathbf{Z}/p\mathbf{Z})^*$. As a result, if one can obtain an integer factorization

$$l(x) = yz, \tag{3.69}$$

one also obtains the $(\mathbf{Z}/p\mathbf{Z})^*$ factorization

$$x = r(l(x)) = r(yz) = r(y)r(z). \tag{3.70}$$

In general, one must lift to a set in which *height* is defined for Adleman's method to work. Such sets include the integers and elliptic curves over the rationals. A notion of height is necessary because factorization involves breaking an element into the composition of elements of smaller height. The height of an element is usually related to the number of bits needed to represent the element. Since computation time is also related to the number of bits, the result of lifting must be an element of relatively small height for the algorithm to run in subexponential time.

Difficulty with elliptic logarithms

The main problem with lifting points from an elliptic curve over a finite field to an elliptic curve over the rationals is that few points on the curve $E(\mathbf{Q})$ have small heights. This is a result of the Mordell-Weil Theorem that the rank of the group $E(\mathbf{Q})$ is finite (Section 3.1.3). In contrast, the group $\mathbf{Z}$ has infinite rank: the basis is the set of primes, which itself is infinite, and there are many elements with small height. In [Mil86] Miller gives convincing arguments that there are few points in the group $E(\mathbf{Q})$ whose coefficients are small enough to manage efficiently—and thus lifting is not an option. Adleman's algorithm does not apply, and the best method for computing elliptic logarithms runs in time $O(\sqrt{n})$.

About the author

The author, Burton Stephen Kaliski, Jr., attended Central High School in Manchester, New Hampshire prior to beginning his studies at M.I.T. in the fall of 1981. He received a bachelor's degree in 1984 and a master's degree in 1987 in computer science. Ronald Rivest supervised all three of his theses.

Burt and his spouse, the former Michele Fichtl, reside near Buffalo, New York. Michele received a master's degree in 1987 in manufacturing engineering from M.I.T. and is employed by Harrison Radiator Division of General Motors. Burt expects to be employed as a contract programmer during their two-year stay in the Buffalo area. After this time they intend to move to Japan to help young Christian churches grow, while Burt continues work in computer science. His interest in this type of career began when he decided as a senior at M.I.T. to acknowledge Christ as his Lord.

Burt enjoys punning, hiking, traveling, cooking and singing.