1998

An implementation of the El Gamal elliptic curve cryptosystem over a finite field of characteristic P

Samuel Thomas Arslanian

AN IMPLEMENTATION OF THE EL GAMAL ELLIPTIC CURVE CRYPTOSYSTEM OVER A FINITE FIELD OF CHARACTERISTIC P

By Samuel Thomas Arslanian

Thesis Advisor: Dr. Ali Ozluk

An Abstract of the Thesis Presented
in Partial Fulfillment of the Requirements for the
Degree of Master of Arts
(in Mathematics)
August, 1998

Since the earliest times, individuals and groups of individuals have been interested in communicating sensitive information in a manner which would guarantee that such information could not be arbitrarily received. Further, such information was to be received by select recipients and this required that a means of secure information transmission be found and employed. To these ends, methods of information encryption have ever since been sought and employed. The entire study and practice of this activity, cryptology, the science of message encryption and decryption, provides a framework for this thesis. In particular, the development of cryptology has been influenced by some specific areas of mathematics, employing abstract mathematical concepts and utilizing algebraic structures known as elliptic curves. It is with respect to these structures and their utilization in specific cryptosystems, called elliptic curve cryptosystems on which this thesis focuses. More specifically, this thesis is concerned with the implementation of such a cryptosystem and is a demonstration of that implementation. Additional pertinent examples, illustrations and supporting computer programs are included to present a self-contained work.

AN IMPLEMENTATION OF THE EL GAMAL ELLIPTIC CURVE CRYPTOSYSTEM OVER A FINITE FIELD OF CHARACTERISTIC P

By

Samuel Thomas Arslanian

B.S. University of Louisville, 1992

A THESIS

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Arts

(in Mathematics)

The Graduate School

University of Maine

August, 1998

Advisory Committee:

Ali E. Ozluk, Associate Professor of Mathematics, Advisor
William M. Snyder, Professor of Mathematics
Henrik Bresinsky, Professor of Mathematics

ACKNOWLEDGEMENTS

Since beginning this thesis, the writer has developed a greater appreciation for the extensive development of the theory of elliptic curves, to say nothing of its implementation. Though the material herein draws equally from a number of sources, each are worth citing for special mention. The historical development is treated well in Koblitz **[7]** from whom also the implementation **[5]** and the characterization over fields of characteristic 2 were drawn **[6]**. Silverman**[12],[13]** and Husemoller**[4]** give the basics for the mechanics of computing curve groups and structures. Lidl and Niederreiter **[8]** was indispensable for constructing group tables. The papers by Menezes and Vanstone **[9]** as well as Diffie and Hellman**[2]** and Agnew, et.al. **[1]** helped highlight and investigate special features such as supersingularity and practical aspects.

On a more personal note, special thanks to Doctors Henrik Brezinsky and William Snyder for steering the writer in the proper direction during his investigation. Finally, and above all, the writer is indebted to his advisor, Doctor Ali Ozluk, for having sparked an interest in the subject of Algebra, an appreciation of its sublime and enduring beauty and for having “cast the seed of germination” for this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CRYPTOGRAPHY- CLASSICAL AND MODERN

Background and Definitions

In order to introduce the topic at hand, a few basic definitions are in order. The following is a list of components common to any cryptosystem and their definitions.

Plaintext: message to be sent

Plaintexts are broken up into message units

Ciphertext: message disguised by some method of encryption

Encryption: process of converting plaintext to ciphertext

Decryption: process of converting ciphertext back to plaintext

Enciphering transformation: a function taking any plaintext message unit and mapping it to a ciphertext message unit in 1-1 correspondence.

Deciphering transformation: an inverse function of the enciphering transformation.

Hence, the following mapping:

$$\underset{P}{\quad} \overset{f}{\longrightarrow} \underset{C}{\quad} \overset{f^{-1}}{\longrightarrow} \underset{P}{\quad}$$

Cryptosystem: a messaging system with the above components

Classical Cryptography

An early example of a cryptosystem is one apparently used by Julius Caesar as follows. Using an N letter alphabet and some integer b, assign a shift transformation given by $C = f(P) \equiv P + b \bmod N$ where C is a ciphertext message unit in $\{0, 1, ..., N - 1\}$.

Decipher by C - b mod N to give the inverse. This system used a fixed set of values for ciphertexts, but eventually these were varied to add security. Decryption depends on frequency analysis to find b. We find the most frequently occurring ciphertext unit to assign it to the most frequently used letter in the plaintext language. Decryption can be accomplished by intercepting relatively short messages.

A slight improvement came with affine maps using a more general transformation of Z/NZ defined by $C \equiv aP + b \bmod N$ where a,b are fixed integers and (a,N) = 1. Decryption is given by solving for P in terms of C using $P \equiv a'C + b' \bmod N$ where $a' = a^{-1}$ in Z/NZ and $b' = -a^{-1}b$. If a = 1 we obtain the shift transformation above, and if b = 0 we have a linear transformation. We can solve the resulting system of congruent equations once we have done a frequency analysis as above to obtain two letters (most and second most frequently used letters).

Another approach, still better, involves using digraph transformations. Plaintext and ciphertext message units come in two letter blocks called digraphs. If the plaintext message has an odd number of letters we add one more that will not cause confusion, say a blank or some other dummy. Assign numerical equivalents to each digraph, say xN + y for example where x,y are numerically equivalent to the first and second letter in the digraph respectively. This gives a 1-1 correspondence between digraphs in the N letter alphabet and nonnegative integers less than N^2. Use an affine transformation over the integers modulo Z/N^2. The inversion process is as given above. Define the encrypted P to be the nonnegative integer less than N^2 satisfying the congruence $C \equiv aP + b \bmod N^2$. Then the inverse transformation is comparable to the above modulo N^2.

Continuing further, we can consider each digraph instead as a column vector in x and y say. Picture each digraph as a point on an N x N array with each axis as a copy of Z/NZ. This was the idea of the Vigenere cipher used for several centuries. The idea was to treat blocks of k letters (k, fixed) as vectors in $(Z/NZ)^k$. Some fixed vector b remembered as a key-word allowed one to encipher by vector translation C = P + b. This is of course almost as easy to solve as the first method.

An improvement over this method uses affine enciphering transformations on a digraph column vector (2X1) called P. Take a 2X2 matrix A over Z/NZ and adding a constant column vector (2X1) called B we have C = AP + B as the affine mapping. Thus

$$\begin{matrix} x' \\ \\ = \\ \\ y' \end{matrix} \quad \begin{matrix} a & b \\ & \\ & \\ & \\ c & d \end{matrix} \quad \begin{matrix} x \\ \\ \\ \\ y \end{matrix} \quad \begin{matrix} e \\ \\ = \\ \\ f \end{matrix} \quad \begin{matrix} ax + by + e \\ \\ \\ \\ cx + dy + f \end{matrix}$$

All the above methods share three properties. First, each cryptosystem corresponds to a choice of parameters whose values are necessary for encryption, known as an encryption key, and also a set of values known as a decryption key. Second, it is not really necessary to know the decryption key given the encryption key since the effort to determine the latter from the former is a relatively simple process given that the former is already known, with its appropriate algorithm. Finally, the amount of time for encryption and decryption share the same order of magnitude with respect to computation time. It is the elimination of part or all of these features that distinguishes modern cryptography from classical.

Public Key Cryptography

A public key cryptosystem (1976, Diffie & Hellman) differs from the above methods in that knowledge of the encryption key does not guarantee with any reasonable probability that one can also determine the decryption key as well without prohibitively long computations. Thus, the encryption function (called a trapdoor function) is easily computable while its inverse is from a computation standpoint, very difficult to compute. The function f is then said to be non-invertible, without additional information besides the encryption algorithm and key.

One way functions (Wilkes, 1968) are similar except that even given the additional information, computational ease is not achieved. Computation times for the two processes vary radically with decryption being the more difficult of the two. Using the idea of trapdoor and one-way functions, Rivest, Shamir & Adelman (1974) developed the concept of a public key cryptosystem (RSA method) based on factoring a large composite integer into primes factors.

RSA method: Each user chooses two extremely large primes, p,q of about 100 digits each, say. Set n = pq. Knowing how to factor n, it is easy to compute _(n) = (p - 1)(q - 1) = n + 1 - p - q. The user next chooses an integer x between 1 and _(n) such that (x,_(n)) =1. Note that we randomly choose x,p, and q as follows. Using a random number generator, generate a large integer m and check if m is even. If so, replace by m + 1 and test for primality. If it is not prime increment by 2 and check again. Continue until a prime is reached. The user computes also the multiplicative inverse of x modulo _(n), call

it d. The public enciphering key (n,x) is made known while the decryption key (n,d) is kept secret. The enciphering transformation is given by $f(P) \equiv P^x \bmod n$. The deciphering transformation is given by $f^{-1}(C) \equiv C^d \bmod n$, since these maps are inverses of each other by the choice of d.

Discrete Log Method

Another more modern method using the trapdoor idea involves using discrete logarithms over finite fields. Given a finite group, say Z/NZ or a finite field with multiplication, and an element $y = b^x$, how do we compute $x = \log_b y$? Discrete log cryptosystems based on the multiplicative group of a finite field have been shown as vulnerable to various index-calculus attacks**[11, p. 418]**. Hence these are not particularly secure given the present state of knowledge.

Elliptic Curves

By 1987, elliptic curves were being implemented in cryptosystems. An improvement over the discrete log method does not directly use the finite fields or groups but rather the elliptic curves defined over them**[5],[11]**. Elliptic curves have group structure that allows the encryption of message units to be implemented utilizing simple rational expressions. A correspondence with the location of points on these curves is established with the plaintext units after appropriate imbedding. Elliptic curves are known to provide a high degree of security and great variety for implementation**[1],[5],[11]**.

ELLIPTIC CURVES

DEFINITION, NORMAL FORMS AND ADDITION FORMULAS

A Weierstrass equation defined over a field K is a degree 3 homogeneous equation given by

$$Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3$$

where the $a_i \in K$, i =1,2,3,4,6. The equation is called smooth or non-singular if for all projective points $P = (X,Y,Z) \in P^2(\boldsymbol{K})$, $\boldsymbol{K}$ an algebraic closure of K, satisfying

$$F(X,Y,Z) = Y^2Z + a_1XYZ + a_3YZ2 - X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3 = 0$$

we have

$$(\partial F/\partial X(P),\partial F/\partial Y(P),\partial F/\partial Z(P)) \neq (0,0,0).$$

An elliptic curve, then, is the set of all solutions in $P^2(K)$ of a smooth Weierstrass equation defined over K. Using non-homogeneous coordinates x = X/Z, y = Y/Z, the Weierstrass equation becomes

$$E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \qquad a_i \in K, i = 1,2,3,4,6.$$

By a linear change of variables, the Weierstrass equation for an elliptic curve reduces to what is called the normal form, of which a table has been prepared on page 11. The points $(x, y) \in K \times K$ that are solutions of the equation, together with a point at infinity, can have a group structure imposed on them if addition and an identity element are defined as below. That the requisite properties of an abelian group are satisfied has been verified **[12].** To be able to add and double points in the curve group, necessary for any encryption procedure, one uses simple formulas derived directly from the equation of the particular curve. For the general case we have the following results**[12].**

First, let E be given by a Weierstrass equation as above. The group elements of E then consists of points P = (x,y) satisfying the equation together with an identity element identified with the point (0,1,0) at infinity. In projective space, a line L that intersects E in the space intersects it in three points counting multiplicity. Where two of these points are distinct in the y coordinates only, the third point of intersection is (0,1,0) at infinity and identified as the identity of the curve group which we will label ***O***.

Let $P_1 = (x_1,y_1)$ and $P_2 = (x_2,y_2)$. Then we define

$P_1 \oplus P_2 = P_3 = (x_3,y_3)$ where x_3 and y_3 are obtained via the equations below.

The symbol $\oplus$ is used here to indicate addition of points in the curve group. Henceforth, group addition of points will simply be indicated by +.

Notice first that if $x_1 = x_2$, then either $y_1 = y_2$ or $y_1 + y_2 + a_1x_2 + a_3 = 0$ **[12]**.

In order to add two points $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$, we consider the following exhaustive list of cases.

Case 1. If $x_1 = x_2$ and $y_1 + y_2 + a_1x_2 + a_3 = 0$ then $P_1 + P_2 = \boldsymbol{O}$ (the identity).

Otherwise, that is, if $y_1 + y_2 + a_1x_2 + a_3 \neq 0$, we have two more cases:

Case 2. If $x_1 \neq x_2$, set

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \qquad v = \frac{y_1x_2 - x_1y_2}{x_2 - x_1}$$

then a new point $P_3 = (x_3,y_3)$ is obtained by

$$x_3 = \lambda^2 + a_1\lambda - a_2 - x_1 - x_2 \qquad y_3 = -(\lambda + a_1)x_3 - v - a_3.$$

Case 3. If $x_1 = x_2$ and $y_1 = y_2$, set

$$\lambda = \frac{3x_1^2 + 2a_2x_1 + a_4 - a_1y_1}{2y_1 + a_1x_1 + a_3} \qquad v = \frac{-x_1^3 + a_4x_1 + 2a_6 - a_3y_1}{2y_1 + a_1x_1 + a_3}$$

Then a new point $P_3 = (x_3,y_3)$ is obtained by

$$x_3 = \lambda^2 + a_1\lambda - a_2 - x_1 - x_2 \qquad y_3 = -(\lambda + a_1)x_3 - v - a_3$$

Note that $v = y_i - \lambda x_i$, for i = 1,2 in cases 2 and 3. Furthermore, note that if $P = (x_0, y_0) \in E$,

then the additive inverse of P, $-P = (x_0, - y_0 - a_1x_0 - a_3)$.

[Geometrically, we are taking the line joining P_1 and P_2 (or the tangent line if $P_1 = P_2$), picking the third intersection point $\sim P_3$ and defining P_3 to be the intersection point of the line through ***O*** and $\sim P_3$ with the curve. (Note $\sim P = -P$ as defined).]

If in any computation, we find that the denominator vanishes, then the resulting point is taken to be point at infinity, namely ***O***, the identity. An alternate set of expressions for Case 3 to compute the doubling of a point are given as follows.

The x coordinate of $2P \neq \boldsymbol{O}$, namely x_{2P}, is given by

$$\frac{x^4 - b_4x^2 - 2b_6x - b_8}{4x^3 + b_2x^2 + 2b_4x + b_6}, \qquad \text{if } P = (x,y).$$

The y coordinate of 2P, namely y_{2P}, is given by

$$\frac{-x_{2P}\,[3x^2+2a_2x+a_4+a_1y+a_1^2x+a_3a_1]+x^3-a_4x-2a_6-a_3y-a_1a_3x-a_3^2}{2y+a_1x+a_3}$$

where b_i is defined in terms of the coefficients a_i of the curve equation:

$b_2 = a_1^2 + 4a_2$ $\quad$ $b_4 = a_1a_3 + 2a_4$ $\quad$ $b_6 = a_3^2 + 4a_6$

$b_8 = a_1^2a_6 - a_1a_3a_4 + 4a_2a_6 + a_2a_3^2 - a_4^2$

We also define for purposes of computation the following quantities.

(Discriminant) $\Delta = -b_2^2 b_8 - 8 b_4^3 - 27b_6^2 + 9b_2 b_4 b_6$

$c_4 = b_2^2 - 24b_4$ $\quad$ j-invariant $j(E) = c_4^3/\Delta$

The quantities Δ and j(E) are used in classifying elliptic curves in the following.

Definitions:

A curve E satisfying $\Delta = 0$ is said to be singular.

If $\Delta \neq 0$, then:

A curve E whose j-invariant $j(E) = 0$ is called a supersingular curve.

A curve E whose j-invariant $j(E) \neq 0$ is called a non-supersingular curve.

In cryptology, any elliptic curve satisfying $j(E) = 0$ has been shown to be vulnerable to certain methods of attack **[10],[11].** Such curves are a relative minority of all elliptic curves. For each class of curve with its respective j-invariant and discriminant Δ see Table T1. In particular settings we then have various forms of the above equations and formulas reduced below as follows.

The remainder of this section concludes with general formulas for point doubling

and addition in arbitrary curves over fields of characteristic p, prime$\neq$ 2,3, characteristic 2 and characteristic 3. In subsequent sections, the particular forms for some specific examples of curves over finite fields of characteristic p$\neq$ 2,3, characteristic 2 and characteristic 3 will be given. We will look at some features of the following curves:

1) $y^2 = x^3 - 36x$ defined over the rationals

2) $y^2 = x^3 + x + 1$ defined over F_5

3) $y^2 + y = x^3 + x + 1$, a supersingular curve defined over F_8

4) $y^2 + xy = x^3 + x^2 + 1$, a non-supersingular curve defined over F_8

5) $y^2 = x^3 + 2x^2 + 1$, a non-supersingular curve defined over F_{27}.

Included in the appendix is an implementation of the elliptic curve $y^2 = x^3 + x + 1$ defined over F_p, where $p = 3^{83} + 356$.

Table 1. Table of Normal Forms with Discriminant and j-invariant

Characteristic $\neq 2,3$

$y^2 = x^3 + a_4x + a_6$ $\quad \Delta = -16(4a_4^3 + 27a_6^2)$ $\quad j = 1728\dfrac{4a_4^3}{4a_4^3 + 27a_6^2}$

Characteristic 3 and $j(E) \neq 0$ (Non-supersingular)

$y^2 = x^3 + a_2x^2 + a_6$ $\quad \Delta = -a_2^3a_6$ $\quad j = -a_2^3/a_6$

Characteristic 3 and $j(E) = 0$ (Supersingular)

$y^2 = x^3 + a_4x + a_6$ $\quad \Delta = -a_4^3$ $\quad j = 0$

Characteristic 2 and $j(E) \neq 0$ (Non-supersingular)

$y^2 + xy = x^3 + a_2x^2 + a_6$ $\quad \Delta = a_6$ $\quad j = 1/a_6$

Characteristic 2 and $j(E) = 0$ (Supersingular)

$y^2 + a_3y = x^3 + a_4x + a_6$ $\quad \Delta = a_3^4$ $\quad j = 0$

We first consider curves over a field of characteristic $p \neq 2,3$. The Weierstrass normal form then becomes

$$y^2 = x^3 + a_4x + a_6$$

Setting $a_1 = a_2 = a_3 = 0$ we have the following addition formulas for distinct points.

Let $P_1 = (x_1,y_1)$ and $P_2 = (x_2,y_2)$, and $P_1 + P_2 \neq \boldsymbol{O}$: $\quad (x_1 \neq x_2)$

For

$$\lambda = \frac{y_2 - y_1}{}$$

$$x_2 - x_1$$

and

$$v = \frac{y_1x_2 - x_1y_2}{x_2 - x_1}$$

we have

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

$$y_3 = -\lambda x_3 - v.$$

Doubling formula for yielding x_{2P} where $P = (x,y)$ is given by

$$(x^4 - 2a_4x^2 - 8a_6x + a_4^2)[4(x^3 + a_4x + a_6)]^{-1} \text{ and}$$

$$y_{2P} = [-x_{2P}(3x^2 + a_4) + x^3 - a_4x - 2a_6](2y)^{-1}$$

All such curves are non-supersingular if $a_4 \neq 0$.

If a characteristic 2 non-supersingular curve has the form

$$y^2 + xy = x^3 + a_2x^2 + a_6.$$

$$\text{then } x_3 = \lambda^2 + \lambda + a_2 + x_1 + x_2$$

$$y_3 = (\lambda + 1)x_3 + v$$

with λ and v as before where

$P_1 = (x_1,y_1)$ and $P_2 = (x_2,y_2)$ are distinct points, $P_1 + P_2 \neq \boldsymbol{O}$.

The doubling formula yielding x_{2P} is given by

$$(x^4 - a_6)(x^2)^{-1} \text{ and}$$

$$y_{2P} = [-x_{2P}(x^2 + x + y) + x^3](x^{-1}).$$

If a characteristic 2 supersingular curve has form

$$y^2 + a_3y = x^3 + a_4x + a_6$$

$$\text{then } x_3 = \lambda^2 + x_1 + x_2$$

$$y_3 = \lambda x_3 + v + a_3$$

with λ and v as above.

Doubling formula for x_{2P} is given by

$$x_3 = (x^4 + a_4^2)(a_3^2)^{-1} \text{ and}$$

$$y_{2P} = [-x_{2P}(x^2 + a_4) + x(x^2 - a_4) + a_3(y + a_3)]\ (a_3)^{-1}$$

As an alternate form of the formulas for addition and doubling points of characteristic 2 non-supersingular and supersingular curves respectively we have the following**[9].**

Non-Supersingular Curves (j Invariant $\neq 0$)

Addition Formulas $P_1 \neq P_2,\ P_1 + P_2 \neq \boldsymbol{O}$

$$x_3 = (y_1 + y_2)^2(x_1 + x_2)^{-2} + (y_1 + y_2)(x_1 + x_2)^{-1} + x_1 + x_2 + a_2$$

$$y_3 = (y_1 + y_2)(x_1 + x_2)^{-1}(x_1 + x_3) + y_1 + x_3$$

Doubling Formulas $P = (x,y),\ 2P \neq \boldsymbol{O}$

$$x_{2P} = x^2 + a_6x^{-2}$$

$$y_{2P} = x^2 + (x + yx^{-1})\ x_{2P} + x_{2P}$$

Supersingular (j Invariant = 0)

Addition Formulas

$$x_3 = (y_1 + y_2)^2(x_1 + x_2)^{-2} + x_1 + x_2$$

$$y_3 = (y_1 + y_2)(x_1 + x_2)^{-1}(x_1 + x_3) + y_1 + a_3$$

Doubling Formulas

$$x_{2P} = (x^4 + a_4^2)(a_3^{-2})$$

$$y_{2P} = (x^2 + a_4)(a_3^{-1})(x + x_{2P}) + y + a_3$$

Characteristic 3 Non-supersingular curves take the form

$$y^2 = x^3 + a_2x^2 + a_6$$

We have the following addition formulas given by

$$x_3 = \lambda^2 - a_2 - x_1 - x_2$$

$$y_3 = -\lambda x_3 - v$$

λ and v as defined previously.

Doubling formula to obtain x_{2P} is given by

$$(x^4 + a_6x - a_2a_6)(x^3 + a_2x^2 + a_6)^{-1} \text{ and}$$

$$y_{2P} = [-x_{2P}(2a_2x) + x^3 + a_6](-y)^{-1}.$$

Finally, we have the characteristic 3 supersingular curves with form

$$y^2 = x^3 + a_4x + a_6$$

where $a_1 = a_2 = a_3 = 0$. The addition formulas then are given by

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

$$y_3 = -\lambda x_3 - v$$

where λ and v are as before with the doubling formula for x_{2P} given by

$$(x^4 + a_4x^2 + a_6x + a_4^2)(x^3 + a_4x + a_6)^{-1} \text{ and}$$

$$y_{2P} =[-x_{2P}(a_4) + x^3 - a_4x + a_6](-y)^{-1}.$$

THE ELLIPTIC CURVE $y^2 = x^3 - 36x$ OVER Q

For this particular curve, $y^2 = f(x) = x^3 - 36x$ over Q we have the following formulas:

Addition: $P_1 \neq P_2$, $x_1 \neq x_2$

$$x_3 = - x_1 - x_2 + ((y_2 -y_1)/(x_2-x_1))^2$$

$$\lambda = (y_2-y_1)/(x_2-x_1)$$

Doubling: $P_1 = P_2$, $y_1 \neq 0$

$$x_3 = -2x_1 + (f'(x_1)/2y_1)^2$$

$$\lambda = f'(x_1)/2y_1$$

In both cases $y_3 = -y_1 + \lambda(x_1 - x_3)$.

We clearly have the points (0,0), (6,0) and (-6,0) on the curve for starters and note that (-3,9) is also on the curve. We can generate some additional points and see also how these fit into the curve group. Using any two(distinct) of the above given three points will generate the other using the addition formula. So suppose we wish to add the points P_1:(-3,9) and P_2:(0,0). Then

$$x_3 = -(-3) - (0) + [-9/3]^2 = 12.$$

The slope λ is given as -1. We have

$$y_3 = -9 + (-3)(-3-12) = 36.$$

Thus $(x_3,y_3) = (12,36) = P_1 + P_2$.

Next, let us illustrate the use of the doubling formula with the point $P_1 = (-3,9)$. After appropriate substitutions we obtain

$$x_3 = 6 + (-1/2)^2 = 25/4 \qquad \lambda = -9/18 = -1/2$$

Thus we have

$$2P_1 = 2(-3,9) = (25/4, -35/8) \quad \text{(See Fig. F1.)}$$

Note that there is no selection criterion for picking an intitial point. One can pick an arbitrary point $(x,y) \in Q^2$ whose coordinates satisfy the equation. Then, quite mechanically, we can derive multiples of a single point by doubling repeatedly. Thus one may obtain 2P, 4P, 8P, and so on.

Figure 1. Graph of the Elliptic Curve $y^2 = x^3 - 36x$ defined over R

AN ELLIPTIC CURVE OVER F_5: $y^2 = x^3 + x + 1$

Since we are working in F_5, we are interested in the 5 possibilities for the values of x in $x^3 + x + 1$. We check to see which ones are squares in F_5. We compute the number of points on an elliptic curve over a finite field F_p by

$$\#E(F_p) = 1 + \Sigma_{x \bmod p} ((x^3 + x + 1)/p) + 1)$$

where (a/p) denotes the Legendre symbol**[5]**. Thus one can exhaustively count the points by this computation for curves defined over small fields. Computationally this is fine for prime fields of say up to about 10^5 to 10^7elements. This writer had no problem using Mathematica for curves over fields of this size. For example, $\#E(F_{100043}) = 99804$.

By a simple computation we get the following list:

(0,1)	(0,4)	(2,1)	(2,4)	(3,1)
(3,4)	(4,2)	(4,3)	*O*, the identity	

Next, we write down the particular formulas for point addition and duplication. This particular step bears more weight in dealing with larger curve groups in that one does not generally obtain many points with which to work. However, if one can obtain a few points and determine their order, one is able to get some idea rather quickly of the underlying group structure. The formulas then serve in computing large orders for a given point. Of course, the larger the curve group, the less likely one is to obtain the order for a given point since a large number of duplications may be required to take a point to the identity. In this example the rational expressions have the following

formulas.

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{if } x_1 \neq x_2 , \\ (3x_1^2 + 1)/2y_1 & \text{if } P_1 = P_2,\ y_1 \neq 0 \end{cases}$$

Finally, one computes the new coordinates for the added/doubled point to be

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

where the x_i are non-distinct in the latter case. Then

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

In addition to these formulas there is a convenient duplication formula for doubling points without recomputing the x values each time. Referring back to the first section in this thesis, the coefficients of our given curve can be checked and the a_i identified. Then the doubling formula for elliptic curves of characteristic $p \neq 2$ for this example becomes

$$x_{2P} = (x^4 - 2x^2 - 3x + 1)(4x^3 + 4x + 4)^{-1}$$

where the appropriate reductions mod p have been made for this field. After some computing, if P = (0,1), one obtains the points 2P = (4,2), 3P = (2,1) and 4P = (3,4) and so on(see Table T2). Note that this group is isomorphic to Z_9. Table T3, obtained in similar manner by direct computation, gives a listing of some curves defined over various small finite fields. Included are group order and discriminant for each curve.

Table 2. Addition Table for Elements in the Elliptic Curve Group of $y^2 = x^3 + x + 1$ over F_5

Point	Order	Coordinate Pair
P	9	(0,1)
2P	9	(4,2)
3P	3	(2,1)
4P	9	(3,4)
5P	9	(3,1)
6P	3	(2.4)
7P	9	(4,3)
8P	9	(0,4)
9P	3	***O***

Table 3. Table of Elliptic Curves over Various Small Fields

Curve	p: Field F_p	Discriminant Δ	Δ(mod p)	Group Order
x^3+x+1	3	-1	2	4
	5	-2^4*31	4	9
x^3+x+2	3	-1	2	4
	5	-2^8*7	3	4
x^3+x+3	5	$-2^4*13*19$	3	4

	7	″	3	6	
x^3+x+4	5	-2^6*109	4	9	
	7	″	3	10	
x^3+2x+1	3	-2	1	7	
	5	-2^4*59	1	7	
x^3+2x+2	3	-2	1	1	
	11	-2^6*5*7	4	9	
x^3+2x+3	7	-2^4*5^2*11	3	6	
	13	″	7	18	
x^3+2x+4	5	-2^8*29	1	7	
	7	″	3	10	
x^3+3x+1	7	-2^4*3^3*5	3	12	
	11	″	7	18	
x^3+3x+2	5	-2^7*3^3	4	5	
	7	″	2	9	
x^3+3x+3	5	-2^4*3^3*13	4	5	
	7	″	5	6	
x^3+3x+4	7	-2^6*3^3*5	5	10	
	11	″	6	14	
	13	″	5	14	
x^3+4x+1	5	-2^4*283	2	8	
	7	″	1	5	
x^3+4x+2	5	-2^6*7*13	1	3	
	11	″	6	6	
x^3+4x+3	5	-2^4*499	1	3	
	7	″	3	6	
x^3+4x+4	5	-2^8*43	2	8	
	7	″	3	10	

A CHARACTERISTIC 2 SUPERSINGULAR CURVE: $y^2 + y = x^3 + x + 1$ over F_8

We continue with the first of two examples of curves defined over extensions of F_2. In this section we will be working with the model, $F_8 \cong \mathbb{Z}_2[x]/ (x^3 + x + 1)$ and in the next section with $F_8 \cong Z_2[x]/ (x^3 + x^2 + 1)$. Table T4 gives the two multiplication tables

for F_8^* for these two models. Considering first

$$E_1: y^2 + y = x^3 + x + 1$$

We collect all $(x,y) \in F_8^* \times F_8^*$ that satisfy the equation plus any having 0 as a coordinate and including the identity of the group (***O***, the point at infinity). In this curve we should have $q + 1 + (2q)^{1/2}$ points in the curve group, where q is the order of the field(of characteristic 2, $q = 2^m$, m odd)**[9]**. Since we do indeed have all 13 points we conclude that the curve group has order 13(see Table T5). Also (u, 1) is found to be the generator for this group.

In order to be able to compare and more clearly understand the structures of various elliptic curves we define the notion of isomorphism for elliptic curves as follows. Two elliptic curves are isomorphic if they are isomorphic as projective varieties. Briefly, two projective varieties V_1, V_2 over K are isomorphic over K if there exist morphisms $\tau : V_1 \rightarrow V_2$, $\psi : V_2 \rightarrow V_1$ (τ, ψ defined over K), such that $\psi \circ \tau$, $\tau \circ \psi$ are the identity maps on V_1, V_2 respectively. (For a definition of morphism of projective varieties see **[12, p.16]**). The following theorem relates the notion of isomorphism of elliptic curves to the coefficients of the Weierstrass equations that define the curves.

Table 4.

Multiplication Table for $F_8 \cong \mathbf{Z}_2(u) \cong \mathbf{Z}_2[x]/ (x^3 + x + 1)$

	u	$u+1$	u^2	u^2+1	u^2+u	u^2+u+1	1
u	u^2	u^2+u	$u+1$	1	u^2+u+1	u^2+1	u
$u+1$	u^2+u	u^2+1	u^2+u+1	u^2	1	u	$u+1$
u^2	$u+1$	u^2+u+1	u^2+u	u	u^2+1	1	u^2
u^2+1	1	u^2	u	u^2+u+1	$u+1$	u^2+u	u^2+1
u^2+u	u^2+u+1	1	u^2+1	$u+1$	u	u^2	u^2+u
u^2+u+1	u^2+1	u	1	u^2+u	u^2	$u+1$	u^2+u+1
1	u	$u+1$	u^2	u^2+1	u^2+u	u^2+u+1	1

Multiplication Table for $F_8 \cong \mathbf{Z}_2(v) \cong \mathbf{Z}_2[x]/ (x^3 + x^2 + 1)$

	v	$v+1$	v^2	v^2+1	v^2+v	v^2+v+1	1
v	v^2	v^2+v	v^2+1	v^2+v+1	1	$v+1$	v
$v+1$	v^2+v	v^2+1	1	v	v^2+v+1	v^2	$v+1$
v^2	v^2+1	1	v^2+v+1	$v+1$	v	v^2+v	v^2
v^2+1	v^2+v+1	v	$v+1$	v^2+v	v^2	1	v^2+1
v^2+v	1	v^2+v+1	v	v^2	$v+1$	v^2+1	v^2+v
v^2+v+1	$v+1$	v^2	v^2+v	1	v^2+1	v	v^2+v+1
1	v	$v+1$	v^2	v^2+1	v^2+v	v^2+v+1	1

Table 5. Addition Table for Elements in the Elliptic Curve Group of $y^2 + y = x^3 + x + 1$ over F_8

		P1	P2	P3	P4	P5	P6	P7
P1:(u,1)		P2	P3	P4	P5	P6	P7	P8
P2:(u^2+u+1,u^2+u+1)		P3	P4	P5	P6	P7	P8	P9
P3:(u^2+u,1)		P4	P5	P6	P7	P8	P9	P10
P4:(u^2,0)		P5	P6	P7	P8	P9	P10	P11
P5:(u+1,u+1)		P6	P7	P8	P9	P10	P11	P12
P6:(u^2+1,u^2+1)		P7	P8	P9	P10	P11	P12	*O*
P7:(u^2+1,u^2)	P8	P9	P10	P11	P12	*O*	P1	
P8:(u+1,u)		P9	P10	P11	P12	*O*	P1	P2
P9:(u^2,1)		P10	P11	P12	*O*	P1	P2	P3
P10:(u^2+u,0)	P11	P12	*O*	P1	P2	P3	P4	
P11:(u^2+u+1,u^2+u)	P12	*O*	P1	P2	P3	P4	P5	
P12:(u,0)		*O*	P1	P2	P3	P4	P5	P6
P13:*O*	P1	P2	P3	P4	P5	P6	P7	

		P8	P9	P10	P11	P12	P13
P1:(u,1)		P9	P10	P11	P12	*O*	P1
P2:(u^2+u+1,u^2+u+1)		P10	P11	P12	*O*	P1	P2
P3:(u^2+u,1)		P11	P12	*O*	P1	P2	P3
P4:(u^2,0)		P12	*O*	P1	P2	P3	P4
P5:(u+1,u+1)		*O*	P1	P2	P3	P4	P5
P6:(u^2+1,u^2+1)		P1	P2	P3	P4	P5	P6
P7:(u^2+1,u^2)	P2	P3	P4	P5	P6	P7	
P8:(u+1,u)		P3	P4	P5	P6	P7	P8
P9:(u^2,1)		P4	P5	P6	P7	P8	P9
P10:(u^2+u,0)	P5	P6	P7	P8	P9	P10	
P11:(u^2+u+1,u^2+u)	P6	P7	P8	P9	P10	P11	
P12:(u,0)		P7	P8	P9	P10	P11	P12
P13:*O*	P8	P9	P10	P11	P12	*O*	

Theorem(see **[9]).** Two elliptic curves $E_1(K)$ and $E_2(K)$ given by

$$E_1(K): y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

$$E_2(K): y^2 + A_1xy + A_3 y = x^3 + A_2 x^2 + A_4 x + A_6$$

are isomorphic over K, written ($E_1(K) \cong E_2(K)$) if and only if there exists $u,r,s, t \in K, u \neq 0$, such that the change of variables

$$\psi: (x,y) \rightarrow (u^2x + r, u^3y + u^2sx + t)$$

transforms equation E_1 into equation E_2. The relationship of isomorphism is an equivalence relation.

The change of variables ψ is referred to as an *admissible change of variables*. Note that if $E_1(K) \cong E_2(K)$ and if ψ transforms E_1 into E_2, then the change of variables

$$\tau: (x, y) \rightarrow (u^{-2}(x - r), u^{-3}(y - sx - t + rs))$$

transforms equation E_2 into equation E_1 and is also and admissible change of variables. Also,

ψ maps E_2 onto E_1 while τ maps E_1 onto E_2. Moreover, $\psi \circ \tau$ is the identity map on E_1, while $\tau \circ \psi$ is the identity map on E_2. It should be clear that ψ maps $\boldsymbol{O}$ such that $\boldsymbol{O}(E_1) = \boldsymbol{O}(E_2)$.

If $E_1(K) \cong E_2(K)$, then the change of variables transforming E_1 to E_2 given by ψ yields the following set of equations:

$$uA_1 = a_1 + 2s$$

$$u^2A_2 = a_2 - sa_1 + 3r - s^2$$

(*) $u^3A_3 = a_3 + ra_1 + 2t$

$u^4A_4 = a_4 - sa_3 + 2ra_2 - (t + rs)a_1 + 3r^2 - 2st$

$u^5A_5 = a_6 + ra_4 + r^2a_2 + r^3 - ta_3 - t^2 - rta_1.$

The following is a restatement of the above theorem.

Theorem. Two elliptic curves $E_1(K)$ and $E_2(K)$ are isomorphic over K if and only if there exists u, r, s, $t \in K$, $u \neq 0$ satisfying the above relations (*).

It is worth mentioning that there is a nice theorem giving the isomorphism classes for curves of a general form with j-invariant 0, the supersingular curves, defined over any extension of F_2. An isomorphism class, then, is just the collection of elliptic curves up to isomorphism over a given field.

Theorem(see**[9]**). There are 3 isomorphism classes of elliptic curves over F_2^m with j-invariant equal to 0, where m is odd.

A representative from each class is given by

$$y^2 + y = x^3$$

$$y^2 + y = x^3 + x$$

$$y^2 + y = x^3 + x + 1$$

A CHARACTERISTIC 2 NON-SUPERSINGULAR CURVE: $y^2 + xy = x^3 + x^2 + 1$ OVER F_8

The second of the two examples is a bit more interesting in its structure. Using the model $F_8 \cong Z_2[x]/(x^3 + x^2 + 1)$, we compute points checking which elements from the appropriate table (see Table T4) satisfy this curve equation as before. Note that checking

to see which pairs of the elements in F_8* are valid is a bit more tedious, since there is an xy term on the left-hand side, so that new y values have to be computed for each x value and checked for equality. After collecting all possible solutions for the curve equation we employ the addition formulas given below. The appropriate formulas **[9]** for this curve are given below.

Addition of Distinct Points

$$P_1 = (x_1 , y_1) \text{ and } P_2 = (x_2 , y_2), \; P_1 + P_2 \neq \boldsymbol{O}:$$

$$x_3 = (y_1 + y_2)^2 ((x_1 + x_2)^2)^{-1} + (y_1 + y_2)(x_1 + x_2)^{-1} + x_1 + x_2 + 1 \text{ and}$$

$$y_3 = (y_1 + y_2) (x_1 + x_2)^{-1} (x_1 + x_3) + x_3 + y_1.$$

Doubling Formula for a Point $P_1 = (x_1 , y_1)$, $2P_1 \neq \boldsymbol{O}$:

$$x_3 = x_1^2 + (x_1^2)^{-1} \qquad y_3 = x_3x_1 + (x_3y_1)x_1^{-1} + x_1^2 + x_3$$

Note that if $x_1 = 0$, we have immediately that (x_3,y_3) is taken to the identity, $\boldsymbol{O}$. Observing the results of addition and doubling the pairs of solutions to the equation given above, we see that this curve contains 14 points in the group(see Table T6).

Table 6. Addition Table for Elements in the Elliptic Curve Group of $y^2 + xy = x^3 + x^2 + 1$ over F_8

	P1	P2	P3	P4	P5	P6	P7
P1:$(v+1,v^2)$	P2	P3	P4	P5	P6	P7	P8
P2:(v,v)	P3	P4	P5	P6	P7	P8	P9
P3:(v^2+1,v)	P4	P5	P6	P7	P8	P9	P10
P4:(v^2+v+1,v^2+v+1)	P5	P6	P7	P8	P9	P10	P11
P5:(v^2+v,v^2) P6	P7	P8	P9	P10	P11	P12	
P6:$(v^2,0)$	P7	P8	P9	P10	P11	P12	P13
P7:$(0,1)$	P8	P9	P10	P11	P12	P13	*O*
P8:(v^2,v^2)	P9	P10	P11	P12	P13	*O*	P1
P9:(v^2+v,v)	P10	P11	P12	P13	*O*	P1	P2
P10:$(v^2+v+1,0)$	P11	P12	P13	*O*	P1	P2	P3
P11:(v^2+1,v^2+v+1) P12	P13	*O*	P1	P2	P3	P4	
P12:$(v,0)$	P13	*O*	P1	P2	P3	P4	P5
P13:$(v+1,v^2+v+1)$ *O*	P1	P2	P3	P4	P5	P6	
P14:*O* P1	P2	P3	P4	P5	P6	P7	

	P8	P9	P10	P11	P12	P13	*O*
P1:$(v+1,v^2)$	P9	P10	P11	P12	P13	*O*	P1
P2:(v,v)	P10	P11	P12	P13	*O*	P1	P2
P3:(v^2+1,v)	P11	P12	P13	*O*	P1	P2	P3
P4:(v^2+v+1,v^2+v+1)	P12	P13	*O*	P1	P2	P3	P4
P5:(v^2+v,v^2) P13	*O*	P1	P2	P3	P4	P5	
P6:$(v^2,0)$	*O*	P1	P2	P3	P4	P5	P6
P7:$(0,1)$	P1	P2	P3	P4	P5	P6	P7
P8:(v^2,v^2)	P2	P3	P4	P5	P6	P7	P8
P9:(v^2+v,v)	P3	P4	P5	P6	P7	P8	P9
P10:$(v^2+v+1,0)$	P4	P5	P6	P7	P8	P9	P10
P11:(v^2+1,v^2+v+1) P5	P6	P7	P8	P9	P10	P11	
P12:$(v,0)$	P6	P7	P8	P9	P10	P11	P12
P13:$(v+1,v^2+v+1)$ P7	P8	P9	P10	P11	P12	P13	
P14:*O* P8	P9	P10	P11	P12	P13	*O*	

Since the only abelian groups of order 14 are cyclic, we see that this curve group is isomorphic to Z_{14}. Further, by the table we see that P_1 is a generator for the group.

We close this section with a remark about the isomorphism classes for this curve. There is a theorem that enumerates all possible isomorphism classes for a non-supersingular characteristic 2 curve. The following is given to clarify the meaning of Tr(γ), the trace of γ in the theorem.

Multiplication by γ in F_2^m is a linear transformation in F_2^m . A linear transformation can be represented by a matrix so Tr(γ) is simply the trace of this matrix.

Theorem([9, p.143]). There are 2(q-1) isomorphism classes of elliptic curves with j-invariant $j(E) \neq 0$ over F_2^m where $q = 2^m$. Let γ be an element of F_2^m satisfying Tr(γ) = 1 (for m odd, we can take γ = 1). A set of representatives of the isomorphism classes is given by

$$\{ y^2 + xy = x^3 + a_2 x^2 + a_6 \mid a_6 \in (F_2^m)^*, a_2 \in \{ 0, \gamma \} \}.$$

A complete tabulation of these isomorphism classes is included in Table T7.

Table 7. Isomorphism Classes of Curves
with Non-zero j-invariant over F_8

$a_2 = 0$	$a_2 = 1$
$y^2 + xy = x^3 + 1$	$y^2 + xy = x^3 + x^2 + 1$
$y^2 + xy = x^3 + \alpha$	$y^2 + xy = x^3 + x^2 + \alpha$
$y^2 + xy = x^3 + \alpha + 1$	$y^2 + xy = x^3 + x^2 + \alpha + 1$
$y^2 + xy = x^3 + \alpha^2$	$y^2 + xy = x^3 + x^2 + \alpha^2$
$y^2 + xy = x^3 + \alpha^2 + 1$	$y^2 + xy = x^3 + x^2 + \alpha^2 + 1$
$y^2 + xy = x^3 + \alpha^2 + \alpha$	$y^2 + xy = x^3 + x^2 + \alpha^2 + \alpha$
$y^2 + xy = x^3 + \alpha^2 + \alpha + 1$	$y^2 + xy = x^3 + x^2 + \alpha^2 + \alpha + 1$

AN ELLIPTIC CURVE OVER F_{27}: $y^2 = x^3 + 2x^2 + 1$

A final example of elliptic curves, $y^2 = x^3 + 2x^2 + 1$ defined over F_{27}, completes the survey. Once again, a multiplication table was tabulated (see Table T8) for use in finding points of the curve group and adding and doubling formulas obtained.

Addition ($P_1 + P_2 \neq \boldsymbol{O}$):

$$\lambda = (y_2 - y_1)/(x_2 - x_1)^{-1} \qquad x_3 = \lambda^2 - 2 - x_1 - x_2$$

Doubling ($2P_1 \neq \boldsymbol{O}$):

$$\lambda = x\,(\,2y\,)^{-1} \qquad x_3 = \lambda^2 - 2 - 2x_1$$

Again, if $y = 0$, then doubling $P=(x,y)$ immediately obtains the identity and for both cases the new y coordinate is given by

$$y_3 = -y_1 + \lambda(x_1 - x_3).$$

After computing by substituting the field elements and 0 in the curve equation, we find that we have precisely 20 points for the group (Table T9). Thus, the Hasse estimate shows that the bounds $|q + 1 - n| \leq 2\sqrt{q}$ are satisfied where $q = 3^3$ and

$$q + 1 - 2\sqrt{q} \approx 17.60 \leq 20 \leq q + 1 + 2\sqrt{q} \approx 38.39.$$

Table T10 gives the group table. Some consideration and a few computations should suffice to convince the reader that the group has structure isomorphic to $Z_2 \times Z_2 \times Z_5$.

Table 8. Field Elements of $F_{27}\backslash\{0\}$

$$F_{27} \cong \mathbf{Z}_3[x]/(x^3 + 2x^2 + 1) \cong \mathbf{Z}_3(\alpha)$$

j	α^j
0	1
1	α
2	α^2
3	$\alpha^2 + 2$
4	$\alpha^2 + 2\alpha + 2$
5	$2\alpha + 2$
6	$2\alpha^2 + 2\alpha$
7	$\alpha^2 + 1$
8	$\alpha^2 + \alpha + 2$
9	$2\alpha^2 + 2\alpha + 2$
10	$\alpha^2 + 2\alpha + 1$
11	$\alpha + 2$
12	$\alpha^2 + 2\alpha$
13	2
14	2α
15	$2\alpha^2$
16	$2\alpha^2 + 1$
17	$2\alpha^2 + \alpha + 1$
18	$\alpha + 1$
19	$\alpha^2 + \alpha$
20	$2\alpha^2 + 2$
21	$2\alpha^2 + 2\alpha + 1$
22	$\alpha^2 + \alpha + 1$
23	$2\alpha^2 + \alpha + 2$
24	$2\alpha + 1$
25	$2\alpha^2 + \alpha$

Table 9. Points of the Curve E: $y^2 = x^3 + 2x^2 + 1$ over F_{27}

A_1	$(\alpha^2 + \alpha + 2, \alpha + 1)$	A_3	$(\alpha^2 + 1, 2\alpha^2 + \alpha + 2)$
A_2	$(\alpha^2 + 1, \alpha^2 + 2\alpha + 1)$	A_4	$(\alpha^2 + \alpha + 2, 2\alpha + 2)$
B_1	$(2\alpha + 1, \alpha^2)$	B_2	$(2\alpha^2 + 2\alpha + 1, \alpha^2 + 2\alpha + 2)$
B_3	$(2\alpha^2 + 2\alpha + 1, 2\alpha^2 + \alpha + 1)$	B_4	$(2\alpha + 1, 2\alpha^2)$
C_1	$(2\alpha^2 + 2, 2\alpha^2 + 2\alpha)$	C_2	$(\alpha + 2, \alpha^2 + 2\alpha)$
C_3	$(\alpha + 2, 2\alpha^2 + \alpha)$	C_4	$(2\alpha^2 + 2, \alpha^2 + \alpha)$
D_1	(1,1)	D_2	(0,1)
D_3	(0,2)	D_4	(1,2)
$\mathbf{O}_1$	$(\alpha , 0)$	$\mathbf{O}_2$	$(\alpha^2 + 2, 0)$
$\mathbf{O}_3$	$(2\alpha^2 + 2\alpha + 2, 0)$	$\mathbf{O}$	

Table 10. Group Table for Elements of E: $y^2 = x^3 + 2x^2 + 1$

	A_1	B_1	C_1	D_1	A_2	B_2	C_2	D_2	A_3	B_3	C_3	D_3	A_4	B_4	C_4	D_4	O_1	O_2	O_3	O_4
A_1	D_2	B_2	A_2	C_2	B_3	A_3	D_3	C_3	C_4	B_4	D_4	A_4	O_4	O_3	O_2	O_1	D_1	C_1	B_1	A_1
B_1	B_2	D_2	C_2	A_2	D_3	C_3	B_3	A_3	D_4	A_4	C_4	B_4	O_3	O_4	O_1	O_2	C_1	D_1	A_1	B_1
C_1	A_2	C_2	D_2	B_2	C_3	D_3	A_3	B_3	A_4	D_4	B_4	C_4	O_2	O_1	O_4	O_3	B_1	A_1	D_1	C_1
D_1	C_2	A_2	B_2	D_2	A_3	B_3	C_3	D_3	B_4	C_4	A_4	D_4	O_1	O_2	O_3	O_4	A_1	B_1	C_1	D_1
A_2	B_3	D_3	C_3	A_3	D_4	A_4	C_4	B_4	O_4	O_1	O_3	O_2	C_1	D_1	A_1	B_1	B_2	D_2	C_2	A_2
B_2	A_3	C_3	D_3	B_3	A_4	D_4	B_4	C_4	O_1	O_4	O_2	O_3	B_1	A_1	D_1	C_1	A_2	C_2	D_2	B_2
C_2	D_3	B_3	A_3	C_3	C_4	B_4	D_4	A_4	O_3	O_2	O_4	O_1	D_1	C_1	B_1	A_1	D_2	B_2	A_2	C_2
D_2	C_3	A_3	B_3	D_3	B_4	C_4	A_4	D_4	O_2	O_3	O_1	O_4	A_1	B_1	C_1	D_1	C_2	A_2	B_2	D_2
A_3	C_4	D_4	A_4	B_4	O_4	O_1	O_3	O_2	D_1	A_1	C_1	B_1	B_2	D_2	C_2	A_2	B_3	D_3	C_3	A_3
B_3	B_4	A_4	D_4	C_4	O_1	O_4	O_2	O_3	A_1	D_1	B_1	C_1	A_2	C_2	D_2	B_2	A_3	C_3	D_3	B_3
C_3	D_4	C_4	B_4	A_4	O_3	O_2	O_4	O_1	C_1	B_1	D_1	A_1	D_2	B_2	A_2	C_2	D_3	B_3	A_3	C_3
D_3	A_4	B_4	C_4	D_4	O_2	O_3	O_1	O_4	B_1	C_1	A_1	B_1	C_2	A_2	B_2	D_2	C_3	A_3	B_3	D_3
A_4	O_4	O_3	O_2	O_1	C_1	B_1	D_1	A_1	B_2	A_2	D_2	C_2	D_3	B_3	A_3	C_3	D_4	C_4	B_4	A_4
B_4	O_3	O_4	O_1	O_2	D_1	A_1	C_1	B_1	D_2	C_2	B_2	A_2	B_3	D_3	C_3	A_3	C_4	D_4	A_4	B_4
C_4	O_2	O_1	O_4	O_3	A_1	D_1	B_1	C_1	C_2	D_2	A_2	B_2	A_3	C_3	D_3	B_3	B_4	A_4	D_4	C_4
D_4	O_1	O_2	O_3	O_4	B_1	C_1	A_1	D_1	A_2	B_2	C_2	D_2	C_3	A_3	B_3	D_3	A_4	B_4	C_4	D_4
O_1	D_1	C_1	B_1	A_1	B_2	A_2	D_2	C_2	B_3	A_3	D_3	C_3	D_4	C_4	B_4	A_4	O_4	O_3	O_2	O_1
O_2	C_1	D_1	A_1	B_1	D_2	C_2	B_2	A_2	D_3	C_3	B_3	A_3	C_4	D_4	A_4	B_4	O_3	O_4	O_1	O_2
O_3	B_1	A_1	D_1	C_1	C_2	D_2	A_2	B_2	C_3	D_3	A_3	B_3	B_4	A_4	D_4	C_4	O_2	O_1	O_4	O_3
O_4	A_1	B_1	C_1	D_1	A_2	B_2	C_2	D_2	A_3	B_3	C_3	D_3	A_4	B_4	C_4	D_4	O_1	O_2	O_3	O_4

CLASSIFICATION OF ELLIPTIC CURVES OVER F_3 UP TO ISOMORPHISM

One final area of interest regarding the elliptic curves of characteristic 3 would be to assess the actual number and structure of all the elliptic curves up to isomorphism for both supersingular and non-supersingular curves. In the case where these curves are defined over F_3, we examine the supersingular case (j-invariant 0) and then the non-supersingular and present a complete tabulation below.

Checking first through the list of curves(see Table T1), and making the substitution $x \rightarrow x + 1$ and $y \rightarrow y$ gives

$$C_1 \cong C_2 \cong C_3 .$$

In fact, all 3 of these curves have isomorphic curve groups, each being isomorphic to $\mathbf{Z_4}$. On the other hand, C_4, C_5 and C_6 each have distinct structures as can be seen from the tabulation.

Characteristic 3 curves with j(C) = 0

Curve Group Isomorphic to $\mathbf{Z_4}$	Distinct Curves	Curve Groups
$C_1: y^2 = x^3 + x$	$C_4: y^2 = x^3 + 2x$	$\cong$ $\mathbf{Z_2 \times Z_2}$
$C_2: y^2 = x^3 + x + 1$	$C_5: y^2 = x^3 + 2x + 1$	$\cong$ $\mathbf{Z_7}$
$C_3: y^2 = x^3 + x + 2$	$C_6: y^2 = x^3 + 2x + 2$	$\cong$ $\{\boldsymbol{O}\}$(trivial)

Thus we have a

Proposition. There are exactly 4 elliptic curves defined over $\mathbf{F_3}$ up to isomorphism with j-invariant j(E) = 0 and these are represented by C_1, C_4, C_5 and C_6.

A table of nonsupersingular curves with their respective j-invariants follows.

E_1: $y^2 = x^3 + x^2 + 1$ $\quad$ $j(E_1) = 2$

E_2: $y^2 = x^3 + x^2 + 2$ $\quad$ $j(E_2) = 1$

E_3: $y^2 = x^3 + 2x^2 + 1$ $\quad$ $j(E_3) = 1$

E_4: $y^2 = x^3 + 2x^2 + 2$ $\quad$ $j(E_4) = 2$

A computation shows that the remaining curves each have distinct group structures as shown below.

Curve		Group	Curve		Group
E_1: $y^2 = x^3 + x^2 + 1$	$\cong$	$\mathbf{Z_6}$	E_3: $y^2 = x^3 + 2x^2 + 1$	$\cong$	$\mathbf{Z_5}$
E_2: $y^2 = x^3 + x^2 + 2$	$\cong$	$\mathbf{Z_3}$	E_4: $y^2 = x^3 + 2x^2 + 2$	$\cong$	$\mathbf{Z_2}$

Proposition. There are 4 isomorphism classes of elliptic curves defined over $\mathbf{F_3}$ with non-zero j-invariant and they are represented by the curves E_1, E_2, E_3 and E_4. (Notice that $j(E_2) = j(E_3)$ and $j(E_1) = j(E_4)$ but the curves E_1, E_2, E_3 and E_4 are all in distinct isomorphism classes.)

THE IMPLEMENTATION

IMBEDDING

In this final section we will discuss some imbedding algorithms, the encryption algorithm used in this implementation and a few remarks pertaining to the computational work involved.

We begin by imbedding plaintexts as points on some elliptic curve E defined over a finite field F_p. We want to do this systematically so that we can retrieve a response plaintext from the knowledge of the coordinates of the corresponding embedded point. Let the plaintext be denoted by m and the corresponding point P_m.

There is no polynomial time deterministic algorithm known by which we can write down a large number of points on E over our field but probabilistic ones do exist **[5]**. Also, in order to encode a large number of possible messages m, we need some systematic way to generate points that are related to m in some way. We might use, for example, the x coordinate of P_m.

To begin, let us consider elements of a prime field, F_p, such that $p \equiv 3 \bmod 4$ and let $y^2 = f(x)$ be an elliptic curve over F_p. Next, we pick our plaintexts to be integers m in the range $0 \leq m \leq (p/1000) - 1$. We go about imbedding by trying to append three digits to each m in turn, until we obtain an x, $1000m \leq x \leq 1000(m + 1) < p$, such that f(x) is a quadratic residue in F_p. When a y is found such that $y^2 = f(x)$, we set up a 1-1 correspondence between the point found and some plaintext unit letting $P_m = (x,y)$. Since f(x) is square for approximately half of all x, there is only about a 2^{-1000} probability that

the method will fail to produce a point $P_m = (x,f(x))$ satisfying the above criteria.

COMPUTATION OF SQUARE ROOTS IN THE FIELD

Some programming instructions were applied to generate a few large primes of approximately 40 digits and also to test the prime p for congruence to 3 modulo 4. Since $\alpha^{p-1} = 1$ in the multiplicative group F_p^*, $\alpha^p = \alpha$ and $\alpha^{(p+1)} = \alpha^2$. Therefore, when f(x) is a square in F_p, then $(x, f(x)^{(p+1)/4})$ is a point on the curve and hence suitable for representing some plaintext message unit. Working in a larger field for this implementation presented a computational challenge to compute the points. The exponentiation to find the $f(x)^{(p+1)/4}$ required a combination of modular reduction and a special algorithm to permit rapid computation.

ENCRYPTION AND DECRYPTION

We now come to the encryption portion of the algorithm. This particular implementation will use the method of ElGamal cryptosystems for transmitting messages**[3]**. Two variations of the method and aspects of point computations are briefly discussed. For a general flow diagram of the imbedding/encryption algorithm see Fig F2.

Begin by fixing a finite field F_p, an elliptic curve E defined over it and a base point $B \in E$. These must be known to any user (transmitter or receiver) but one does not need to know the number N of points of E in this method. A receiver chooses a secret random integer a, and computes and publishes his key, aB. To send a message P_m, choose a random integer k and send the pair of points $(kB, P_m + k(aB))$ corresponding to

the encrypted plaintext and the multiplied base point for each message unit. To read the message, the receiver multiplies the first point in the pair by his own secret a and subtracts the result from the second point. Thus,

$$P_m + k(aB) - a(kB) = P_m.$$

More specifically, the pair of points a user would send to a receiver are given as follows. Let $B = (x_B, y_B)$, $P_m = (x_{Pm}, y_{Pm})$. Then the user sends the pair of points

$$\{(x_{kB}, y_{kB}), ((x_{Pm}, y_{Pm}) + (x_{kaB}, y_{kaB}))\}$$

where (x_{kB}, y_{kB}) and (x_{kaB}, y_{kaB}) are multiples k and ka of the point (x_B, y_B). Then the decryption of the message is performed by simply taking the additive inverse of the point (x_{akB}, y_{akB}) and by using an appropriate addition formula for the curve in use, adding this to the point $((x_{Pm}, y_{Pm}) + (x_{kaB}, y_{kaB}))$ to obtain the plaintext message point. Thus we have $(x_{Pm}, y_{Pm}) + (x_{kaB}, y_{kaB}) - (x_{akB}, y_{akB}) = (x_{Pm}, y_{Pm})$.

In this implementation the value of k was not fixed throughout encryption. One may let k vary since only the knowledge of the base point B and the curve equation is needed by a receiver to decrypt a message. The receivers knowledge of their key is sufficient for decryption.

Figure 2. Flow Diagram for Imbedding/Encryption (See file flowii.sam)

Figure 3. Flow Diagram for Decryption
(see file flowiii.sam)

Two ways of choosing a curve and a base point are mentioned here. First, one can simply choose a random curve E over a large field F_q not of characteristic 2, i.e., where E has an equation of the form $y^2 = x^3 + ax^2 + bx + c$ and a base point B = (x,y) on E. How this is done is illustrated using the following particular example. Given the equation

$$y^2 = x^3 + bx + c$$

we can choose random elements x, y and b from F_q, $q = p^n$. Then, any c satisfying

$$c = y^2 - (x^3 + bx)$$

gives the appropriate curve with a point on it. In the case of a characteristic 2 curve, $g(y) = y^2$ is replaced by $y^2 + y$. Since the discriminant Δ satisfies the requirement that $\Delta \neq 0$ (since $\Delta = 1$), this curve is an elliptic curve, and we may set B = (x,y).

A second method involves choosing a "global" elliptic curve E defined over the rationals, or more generally, a number field. Then pick B to be a point of infinite order on E. Next, choose a large prime p (or a prime ideal of the ring of integers of K if our curve is defined over an extension field K of Q) and consider the reduction of E and B mod p. That is, for all p not dividing the discriminant Δ_E of E and such that the coefficients in the equation for E have no p in their denominators, we may consider the coefficients in this equation mod p.

In both of these methods one wishes to choose the point B such that B generates a large subgroup. To do so, one should know beforehand whether B is a point of infinite order in the same curve defined over the rationals. For if this is the case, then it is likely that B will generate a large subgroup **[5]**. The question then comes down to knowing if B has infinite order in the same curve defined over the rationals. One can use the following two theorems to determine the answer. These are given without proof here**[13]**.

Theorem(Nagell-Lutz). Let E be a non-singular cubic curve defined by E: $y^2 = x^3 + ax^2 + bx + c$ with integer coefficients a, b, c, and let D be the discriminant given by $D = -4a^3c + a^2b^2 + 18abc - 4b^3 - 27c^2$. Then a rational point of finite order must have integer coordinates and either y = 0 or y divides D.

(A stronger form of the Nagell-Lutz Theorem handy for computational purposes states:

Let P = (x,y) be a rational point of finite order with $y \neq 0$. Then $y^2 \mid D$.)

Reduction Modulo p Theorem. Let E be as defined above with discriminant D as given above. Let $\Phi \subseteq E(Q)$ be the subgroup consisting of all points of finite order. For any prime p, let P P′ be the reduction modulo p map where $\Phi \Rightarrow E'(F_p)$,

$$P \Rightarrow P' = (x', y') \text{ if } P = (x,y),$$

$$\boldsymbol{O'} \text{ if } P = \boldsymbol{O} \text{ (identity element).}$$

If p does not divide 2D, then the reduction modulo p map is an isomorphism of Φ onto a subgroup of $E'(F_p)$.

Now for any curve, and in particular the given curve $y^2 = x^3 + x + 1$ of this implementation, we may apply the theorems as follows. We notice that the discriminant D = -31. First, by the Nagell-Lutz Theorem, any rational point of finite order must have

integer coordinates. At this point, we would like to apply these theorems to (0,1) and (72,611) observing that (72,611) is a point of infinite order. Next, we can use the reduction theorem and determine the order of Φ as follows. We first check $\#E(F_p)$ for several primes $p > 2D$ (here $2D = 2(-31)$ to obtain the following list of the orders of each curve group. For example, $\#E(F_{67}) = 56$ and $\#E(F_{71}) = 59$. Now since $\Phi \subseteq E(F_p)$ for p_ 2D, then #Φ must divide $\#E(F_p)$ since by the reduction theorem the map $\Phi \Rightarrow E'(F_p)$ is an isomorphism of Φ onto a subgroup of $E(F_p)$. But for the groups listed above, it is clear that if $\#\Phi \mid \#E(F_p)$ then #Φ must equal 1 since (56,59) = 1. Hence, the only point of finite order of E(Q) is ***O***, the identity itself. It follows that the points (0,1) and (72,611) are both good candidates for a base point B in the encryption algorithm since they will probably generate large subgroups.

THE SECURITY OF ELLIPTIC CURVE CRYPTOSYSTEMS

Breaking a cryptosystem of the El-Gamal type requires solution of the elliptic curve analog of the discrete log problem which is stated thus:

Elliptic Curve Discrete Logarithm Problem: Given an elliptic curve E defined over F_q and points $P, Q \in E$, find $x \in Z$ such that $Q = xP$, if such x exists.

It is believed that this problem will prove to be more intractable than the classical discrete log problem and thus far, the strongest techniques known do not seem applicable to the present encryption methods using elliptic curves **[5]**.

THE ORDER OF THE CURVE GROUP

At this point, the question arises, how large a field should one choose? Clearly, even in a relatively small prime field, there would be enough points available in the curve group for imbedding, so considerations for those aspects are not primary. What is important concerns the issue of security. For the purposes of this implementation, a prime field of $3^{83} + 356$ elements was chosen. This 39 digit integer, is close to the lower bound for a secure curve which has been shown to be a 40 digit integer **[6]**. One should seek a curve to be defined over a large enough field, such that the subgroups of the curve group have large or small indices. Such a choice ensures security since it is known that such curves (for example, nonsupersingular) are not vulnerable to certain types of attack such as those utilizing index calculus methods **[11]**.

We cannot directly determine the curve's order or structure since simply counting points works fine only for curves of considerably smaller order. The reason is that the method necessary to find the order of the curve group for this implementation would require excessive computational capabilities **[6]**. Now it is known that we have good security for a curve whose order is at least as large as a 40 digit prime when dealing with supersingular curves**[6, p.157]**. The same criterion was applied though this curve has non-zero j-invariant, hence is not supersingular.

COMPUTATIONS OF POINTS IN THE CURVE GROUP

To add and double points, separate algorithms were developed based on the point computation formulas as used over F_5. The doubling algorithm takes the following approach. One takes the number which will serve as the multiplier of a point P for some

multiple nP of P. Then n is expanded in binary code. For any n we can write n in binary representation where $n = \sum_{j=0}^{K} a_j 2^j$, $a_j \in \{0,1\}$. For each binary evaluation where 1 is a value in the j-th binary digit, take 2^j and sum for each nonzero value in the binary representation. The result is a sum of various powers of 2 that adds up to n. For small n this is hardly worth mentioning. But say one wishes to take large multiples say of the order of 10^{100}, then there is clearly no substitute for this approach. For example, to raise a value say a^{1000} only 15 total operations are needed to get the solution via multiplications and additions versus 1000 by brute exponentiation.

A Closing Remark

The output generated in this implementation includes a list of alphanumeric characters typically useful in everyday written communication. This list consists of a plaintext unit, i.e., the alphanumeric character and an assigned coordinate pair whose coordinates are those of a point on the curve. Since the imbedding algorithm calls for each range of x values to be computed in multiples of 1000, incremented one at a time the programming instructions were entered to reduce these computations but still obtain some points for each character. Thus the program is made to increment through a value of 100 at a time between each multiple of 1000 to save time and output. Also, note that in each case, the only digits we are concerned with are the initial three digits of the first coordinate in the pair. This is because, when encrypting, each plaintext is actually associated to those digits.

The program output for the encryption of the word “math” is also included. First, a list of points of the curve in which the plaintext characters are imbedded is generated. Next, an odd multiple of the base point B is obtained. The program then takes the message character by character and picks out the corresponding point in the character imbedding list. The program then multiplies the public key point labeled aB an odd number times and adds the result to the point representing the imbedded character. Finally, the pair of points representing the multiplied base point and the encrypted plaintext are sent .

The program code for this implementation reveals that when one has reached a multiple of the identity, further computations breakdown. Indeed, Silverman and Tate use this fact to aid in their prime factorization scheme adopted from Lenstra**[13]**. We can use this event to indicate that we have computed the order for some subgroup. We can with a little luck obtain several such orders.

BIBLIOGRAPHY

1. Agnew, G.B., Mullin, R.C., Vanstone, S.A. "An Implementation of Elliptic Curve Cryptosystems Over F_2^{155}", I.E.E.E. Journal on Selected Areas of Communications, Vol. 11 No. 5, June 1993, pp. 804-813

2. Diffie Whitfield and Hellman, Martin E. " New Directions in Cryptography", I.E.E.E. Transactions on Information Theory, Vol. IT-22, No. 6 November, 1976 pp. 644-654

3. ElGamal T., "A Public Key Cryptosystem and a Signature Scheme based on Discrete Logarithms," I.E.E.E. Transactions on Information Theory, Vol. 31, 1985 pp. 469-472

4. Husemoller, Dale Elliptic Curves Springer-Verlag, 1987

5. Koblitz, Neal"Elliptic Curve Cryptosystems", Mathematics of Computation Vol. 48 #177 Jan '87 203-209

6. Koblitz, Neal "Constructing Elliptic Curve Cryptosystems in Characteristic 2", Advances in Cryptology: Proceedings of Crypto '90 Springer-Verlag 157-167

7. Koblitz, Neal A Course in Number Theory and Cryptography Springer-Verlag, 1994

8. Lidl, Rudolf and Niederreiter,Harald Encyclopedia of Mathematics and Its Applications: Vol. 20, Finite Fields Harald Addison-Wesley Publishing Company, 1983

9. Menezes, Alfred and Vanstone, Scott "Isomorphism Classes of Elliptic Curves over Finite Fields of Characteristic 2" Utilitas Mathematica 38 (1990), pp. 135-153

10. Menezes, Alfred J., Okamoto, Tatsuaki, and Vanstone, Scott A. "Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field" I.E.E.E. Transactions on Information Theory, Vol. 39 No. 5, Sept. 1993 pp. 1639-1645.

11. Miller, V. "Uses of Elliptic Curves in Cryptography" Advances in Cryptology: Proceedings of Crypto '85 Lecture Notes in Computer Science 218 (1986) Springer-Verlag 417-426

12. Silverman, Joseph The Arithmetic of Elliptic Curves Springer-Verlag, 1985

13. Silverman, Joseph H. and Tate, John Rational Points on Elliptic Curves Springer-Verlag, 1992

APPENDIX

PROGRAMS

```
BeginPackage["CongruentPrime"]

c:=(3^83)-2;

Print["These are the primes from 3^83-2 to "];
Print["3^83+400 that are Congruent to 3 Mod 4"];

For[t=0,t<402,t++;
        If[PrimeQ[c+t]&&TrueQ[Mod[c+t-3,4]==0],
            Print["3^83+",t-2," is ",c+t]]];
            Unprotect[In,Out];
            Clear[In,Out];
                End[]
```

These are the primes from 3^83-2 to
3^83+400 that are Congruent to 3 Mod 4

3^83+236 is 39908383941873399295342466757234903546 3
3^83+356 is 39908383941873399295342466757234903558 3

(This package was used to verify that F_p was indeed a finite field of prime order where $p = 3^{83} + 356$ elements.)

```
Dupl[x_,y_]:= (          (*This duplicates points for y^2=x^3+x+1*)

p=(3^83)+356;

Newx =     Mod[Mod[PowerMod[x,4,p]-2PowerMod[x,2,p]-8x+1,p]*
        PowerMod[Mod[4PowerMod[x,3,p]+4x+4,p],-1,p],p];
Lambda =  Mod[Mod[3PowerMod[x,2,p]+1,p]*
                PowerMod[Mod[2y,p],-1,p],p];
Nu =    Mod[Mod[PowerMod[-x,3,p]+x+2,p]*PowerMod[2y,-1,p],p];
Newpty = Mod[-Lambda*Newx -Nu,p]; )


Addpts[x1_,y1_,x2_,y2_]:= ( (*This program adds points of   *)
                                    (*the curve group y^2=x^3+x+1*)
```

```
p=(3^83)+356;

Lambda = Mod[(y2-y1)*PowerMod[(x2-x1),-1,p],p];
Nu =    Mod[(Mod[y1*x2,p]-Mod[x1*y2,p])*
                PowerMod[x2-x1,-1,p],p];
Nextptx =  Mod[PowerMod[Lambda,2,p]-x1-x2,p];
Nexty=Mod[-Lambda*Nextptx-Nu,p];

PmKab1=Nextptx;
PmKab2=Nexty;

        If[x1==x2&&y1==y2,
            Do[Dupl[x1,y1];
            Nextptx=Newx;
            Nexty=Newpty;
                    ]];

        If[x1==x2&&y1!=y2,
            Do[Print["Identity has been reached,
                further computations may fail"];
            Break];
                    ] )

Oddmult[n_,x_,y_]:= (

(* This program finds new points on a curve E        *)
(* using the duplication and addition routines       *)
(* for a multiple nP of some point P on E where      *)
(* n is odd.                                         *)

(* It should be noted that for n generating a point  *)
(* value of one greater than the identity, i.e.,     *)
(* O + 1, a point value may not be computed.         *)

i=0;a=x;b=y;j=0;m=n;h=0;ssu=0;

p=(3^83)+356;

While[Floor[m] > 0,
        j=Mod[m,2];
        m=Floor[m/2];
Dupl[x,y];
```

```
If[j!=0,
        For[h=1,h<i,h++;
            Dupl[Newx,Newpty]];
            ssu=ssu+2^i;

        If[h==i&&i>=1,
            Do[Addpts[a,b,Newx,Newpty];

        If[x==Nextptx&&y==Nexty&&OddQ[n]==True,
            Print[ssu-1, " is a multiple of Order of Subgroup"]];

        If[x==Nextptx&&y==Nexty&&EvenQ[n]==True,
            Print[ssu," is a multiple of Order of Subgroup"]];
            a=Nextptx;
            b=Nexty];
                    ]];
                    Newx=0;
                    Newpty=0;
                        i=i+1];
        Kmult1=Nextptx;
        Kmult2=Nexty; )

Binary[n_,x_,y_]:= (

(* This program finds new points on a curve E    *)
(* using the duplication and addition routines    *)
(* for a multiple nP of some point P on E         *)
(* where n is 2 raised to any power.              *)

(* It should be noted that for n generating a     *)
(* point value of one greater than the identity,  *)
(* i.e.,O + 1,a point value may not be computed. *)

i=0;a=0;b=0;j=0;m=n;h=0;ct=0;chk=0;

p=(3^83)+356;

While[Floor[m] > 0,

Newx=x;
Newpty=y;
        j=Mod[m,2];
        m=Floor[m/2];
```

```
        i=i+1;
            If[j!=0,
                For[h=1,h<i,h++;
                    Dupl[Newx,Newpty]];
                    ct=ct+1;
            If[ct==1,
                Do[a=Newx;
                b=Newpty]];
            If[ct==2,
                Do[Addpts[a,b,Newx,Newpty];
                ct=0;
                chk=chk+1;
                a=0;
                b=0]];
If[chk==0&&ct==1,
        key1=Newx;
        key2=Nexpty;
                            ]]] )

SuprFast[x_,n_]:=(

            (*This is the modular exponentiation*)
            (*routine used to compute the square*)
            (*root of a point on a curve.       *)

v=1;t=x;u=n;p=(3^83)+356;

While[u!=0,
        If[OddQ[u],v=Mod[t*v,p]];
            t=Mod[t*t,p];
            u=Floor[u/2]] )

f[x_]:=Mod[(PowerMod[x,3,p]+Mod[x,p]+1),p];

chr[m_]:=FromCharacterCode[m];


            (*This program segment obtains a list of  *)
            (*characters and their corresponding curve*)
            (*points to be used in encryption.   *)

For[m=95,m<126,m++;
        scrib=0;
```

```
For[i=(1000m)-1,i<(1000(m+1))-1,i+=100;

    If[JacobiSymbol[f[i],p]==1&&Less[scrib,1],

        SuprFast[f[i],(p+1)/4];
        FirstNum[i]=i;
        SecNum[i]=v;

Print[chr[m]," ","(",FirstNum[i],",",SecNum[i],")"];
    scrib=scrib+1;
                        ]]]
```

```
` (96199,6389800374378885040457014286113997347 83)
a (97099,4074008914622103331508536743876887846 52)
b (98099,1183512579460980263085960569438504931187)
c (99299,1845537342405968058653005330058291591346)
d (100199,8853240022901672122512110509399456245 88)
e (101099,1327002817076673990734642367778354924 73)
f (102099,3742113120867548216510054180811595439545)
g (103499,3884326284861313328860998042543784769 58)
h (104099,3531704759303083846482021623608961735 20)
i (105099,3168184333516993997296042856190352511925)
j (106199,3025236684521334585758113882831062869655)
k (107199,9879604237601428588608146121938286914 1)
l (108199,3831308348650974884886213345764012612 19)
m (109199,3934820829704801630116070451435801664349)
n (110199,3840372591736730901350634246188652819640)
o (111199,1242914571737917737493365544851399135812)
p (112099,6893729421468828124978288470043356645 04)
q (113199,3109027603757293144338648065480016215496)
r (114099,3535847087175373911814302697966882322638)
s (115099,2299184535944242086339667580998578220257)
t (116299,3266063780054608413745513507237682464899)
u (117399,2283604676327804864911807583375038605791)
v (118099,1656163260722712624963123918140864967034)
w (119199,1490273550928298529029074023746874530766)
x (120099,3664192411868623775050125327083132214 66)
y (121099,1332787559491678113642211944499761412561)
z (122199,2073795046384097820917895255023833658213)
{ (123199,2635348364288049068951772678786870659878)
```

```
| (124199,20676082954785291815105834654690725222 62)
} (125599,44424722310591236658102956205324554606 6)
~ (126499,33935431858128763094888822179364913008 89)


Finder[T_]:= (

(* Finder takes a short message  and obtains the corresponding  *)
(* point in the imbedding list.   Next, it takes that point and  add*)
(*an odd multiple of the key point aB.  It also finds the same od*)
(* multiple of the base point B.  It then prints the pair (KB, P_m + KaB) *)
(* which is ready for transmission.    K is based on the position of        *)
(* the plaintext unit in the message.*)

B1=0;B2=1;        (* Base Point B is fixed here. *)


SuprFast[x,n];      (* Call to SuprFast to provide a list of characters *)
                    (* and the  associated points of the curve.          *)

Binary[2,B1,B2];                (*This instruction provides the key aB*)
            aB1=key1;
            aB2=key2;
                                        (* Begin the Main Routine     *)

For[j=0,j<StringLength[T],j++;      (* This subroutine matches the*)
        For[m=95,m<126,m++;         (* character string entered in   *)
                                    (* Finder[“                    ”].*)

If[TrueQ[{m}==ToCharacterCode[Part[Characters[T],j]]],
            For[i=(1000m)-1,i<1000(m+1)-1,i+=100;
                If[FirstNum[i]!=0,
                    abscissa[j]=FirstNum[i];
                    ordinate[j]=SecNum[i];
                                                ]]]]];
For[k=0,k<StringLength[T],k++;
        Oddmult[(2*k)+1,B1,B2];
            KB1=Kmult1;KB2=Kmult2;

        Oddmult[(2*k)+1,aB1,aB2];
            KaB1=Kmult1;KaB2=Kmult2;

        Addpts[abscissa[k],ordinate[k],KaB1,KaB2];
```

```
Print["(",KB1,",",KB2,")","(",PmKab1,",",PmKab2,")"];
                                                    ] )
```

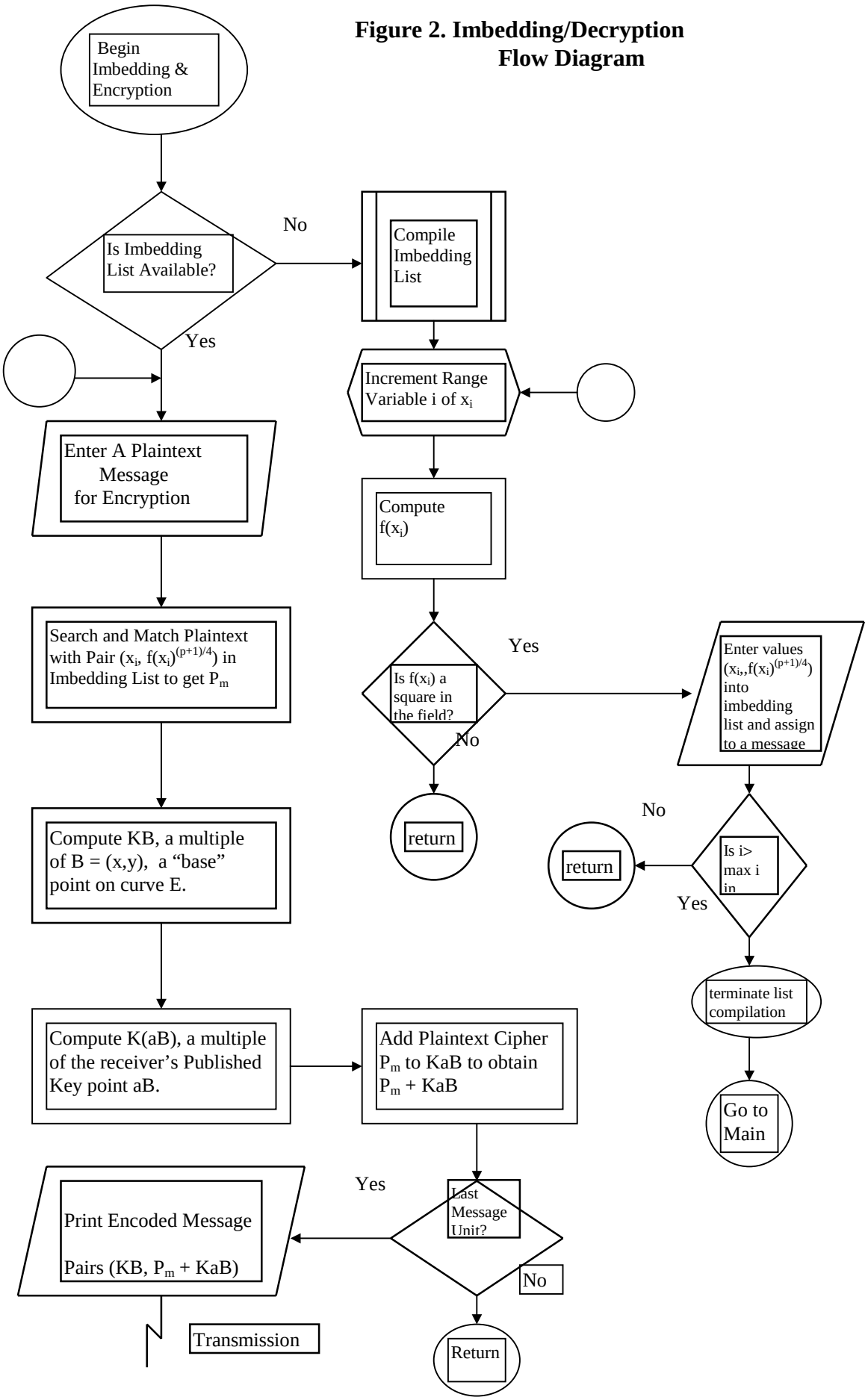
Note: The output below consists of a pair of points, namely 1 pair of coordinates representing the point KB, multiple K of a Base Point B on the curve and a second pair of co-ordinates for the point on the curve representing the encrypted plaintext character. The second pair is for the point P_m + Kab. Also, values of K are odd, K > 1 with K varying based on the value of the position of the character being read in the message. Finally, a is fixed(a = 2) and base point B = (0,1).

Finder["math"]	Value of K	Plaintext Character
(72,611),	3	
(212018253827337461342390973207479008873 7, 209733507127477784386576243919535888225 8)		m
(313679733050634051303167694484460198936 2, 358035230791637972155284058065107186303 2)	5	
(187102924319423140667247010547972547172 3, 205360697406759328196090669488892056467 8)		a
(213293794009560119678510143926190377158 0, 316040787727541328223552079617081739974 7)	7	
(728936147976064104144939706891487478589, 378584419561726981684967734748347250807 7)		t
(271999817055564929614913205557834267832 6, 363944929444958336667248349860327375 27)	9	
(317209502045799489728093740131937998765 5, 178760486142421866750648366186805018808 0)		h

BIOGRAPHY OF THE AUTHOR

Samuel T. Arslanian was born in New York, New York and attended high school at Heidelberg Americanische OberSchule, Heidelberg, Germany. He graduated from the University of Louisville, Louisville, Kentucky in the fall of 1992 with a bachelor's degree in Mathematics. After working for three years, during which time he took some additional coursework in data processing, he enrolled at the University of Maine to pursue a master's degree in Mathematics. During that period he served as AGS representative of the Mathematics Department and was initiated as member of Pi Mu Epsilon Honorary. He is a candidate for the Master of Arts degree in Mathematics from the University of Maine in August, 1998.

Figure 2. Imbedding/Decryption Flow Diagram



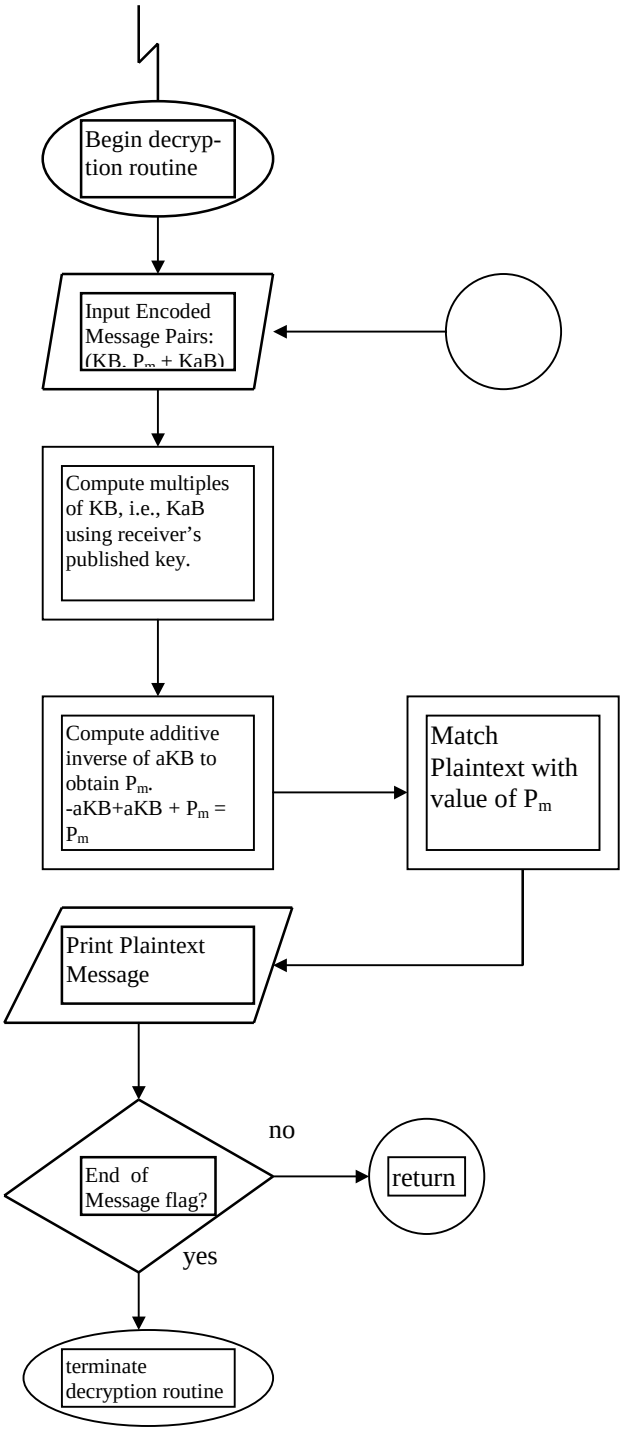


Figure3. Flow Diagram for Decryption