ISTANBUL UNIVERSITY –
JOURNAL OF ELECTRICAL & ELECTRONICS ENGINEERING

YEAR : 2006 (211 - 221)
VOLUME : 6
NUMBER : 2

A SURVEY OF ELLIPTIC CURVE CRYPTOGRAPHY

M.Ali AYDIN[1] **G.Zeynep AYDIN[2]**

[1,2] Istanbul University Engineering Faculty, Computer Engineering Dept.
34320 Avcilar, Istanbul-Turkey

[1] E-mail: aydinali@istanbul.edu.tr [2] E-mail : zeynepg@istanbul.edu.tr

ABSTRACT

As the importance of information increases, many various methods are being used for keeping this information and trasfering confidently. One of these methods is cryptographic algorithms. These algorithms have to be adequately powerful.Thus far,various algorithms have been proposed and used. In this paper,a survey of elliptic curve cryptography,which is thought as the best method for future applicatons, has been studied

Keywords: *Elliptic Curve, Elliptic Curve Cryptography, Security*

1. INTRODUCTION

The vast majority of the products and standarts that use public-key cryptography for encryption and digital signatures use RSA. The bit length for secure RSA usage has increased over recent years, and this has put a heavier precessing load on applications using RSA. This burden has ramifications, especially for electronic commerce sites that conduct large numbers of secure transactions. Recently, a competing system has begun to challenge RSA: elliptic curve cryptography(ECC). In the mid-1980s, Miller and Koblitz introduced elliptic curves into cryptography[1] , and Lenstra showed how to use elliptic curves to factor integers. Since that time, elliptic curves have played an increasingly important role in many cryptographic situations. One of the advantages is that they seem to offer a level of security comparable to classical cryptosystems that use much larger key sizes. Already, ECC is showing up in standardization efforts, including the IEEE P1363 Standard for Public -Key Cryptography[1,2].

2. ELLIPTIC CURVES

An Elliptic Curve E is the graph of an equation $E: y^2 = x^3 + ax + b$, where a, b are in whatever is the appropriate set(rational numbers, complex numbers, integers mod n, etc.), together with a special point O called the ***point at infinity***. Elliptic curves are not ellipses. They are so named because they are described by cubic equations, similar to those used for calculating the circumference of an ellipse. In general, cubic equations for elliptic curves take the form $E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ where a_1, a_3, a_2, a_4 and a_6 are real numbers that satisfy some simple conditions. If we are working *mod p*, where p>3 is prime, or if we are working with real, rational, or complex numbers, then simple changes of variables transform the present equation into the form $y^2 = x^3 + ax + b$. However, if we are working mod 2 or mod 3, or with a finite field of characteristic 2 or 3(that is 1+1=0 or 1+1+1=0), then we need to use the more general form. We begin by looking briefly at elliptic curves defined over the real numbers,

Received Date : *13.03.2004*
Accepted Date: *26.06.2006*

because some of the basic concepts are easier to motivate in this setting[3].

2.1 Elliptic Curves Groups over the Reals

Let a, $b \in \Re$ be constants such that $4a^3 + 27b^2 \neq 0$. A non-singular elliptic curve is the set *E* of solutions $(x,y) \in \Re x \Re$ to the equation; $y^2 = x^3 + ax + b$, together with a special point *O* called the ***point at infinity***. It can be shown that the condition $4a^3 + 27b^2 \neq 0$ is necessary and sufficient to ensure that the equation $x^3 + ax + b=0$ has three distinct roots(which may be real or complex numbers). If $4a^3 + 27b^2 = 0$, then the corresponding elliptic curve is called a singular elliptic curve[3].

2.1.1 Elliptic Curve Addition

Elliptic curve groups are additive groups; that is, their basic function is addition. Suppose E is a non-singular elliptic curve and $P,Q \in E$, where $P=(x_1,y_1)$ and $Q=(x_2,y_2)$. The negative of a point $P=(x_1,y_1)$ is its reflection in the x-axis: the point -P is $(x_1,-y_1)$. Notice that for each point P on an elliptic curve, the point -P is also on the curve[3].

i. Adding distinct points P and Q

Suppose that P and Q are two distinct points on an elliptic curve, and the P is not -Q. To add the points P and Q, a line(L) is drawn through the two points. This line will intersect the elliptic curve in exactly one more point, call -R. The point -R is reflected in the x-axis to the point R. The law for addition in an elliptic curve group is P + Q = R where

$\lambda = (y_2 - y_1) / (x_2 - x_1)$, $x_3 = \lambda^2 - x_1 - x_2$

and $y_3 = \lambda(x_1 - x_3) - y_1$ Note that λ is the slope of the line through P and Q[3].

ii. Adding the points P and –P

The line through P and -P is a vertical line which does not intersect the elliptic curve at a third point; thus the points P and -P cannot be added as previously. It is for this reason that the elliptic curve group includes the point at infinity *O*. By definition, P + (-P) = *O*. As a result of this equation, P + *O* = P in the elliptic curve group . *O* is called the additive identity of the elliptic curve group; all elliptic curves have an additive identity[3].

iii. Doubling the point P

To add a point P to itself, a tangent line to the curve is drawn at the point P. If y_1 is not 0, then the tangent line intersects the elliptic curve at exactly one other point, -R. -R is reflected in the x-axis to R. This operation is called doubling the point P; the law for doubling a point on an elliptic curve group is defined by P + P = 2P = R. The slope of L can be computed using implicit differentiation of the equation of

E: $2dy\frac{dx}{dy} = 3x^2 + a$. Substituting $x=x_1$, $y=y_1$, we see that the slope of the tangent is $\lambda=(3x_1^2+a)/(2y_1)$, $x_3 = \lambda^2 - x_1 - x_2$ and $y_3=\lambda(x_1 - x_3) - y$[3]

iv. Doubling the point P if y_1 = 0

If a point P is such that $y_1=0$, then the tangent line to the elliptic curve at P is vertical and does not intersect the elliptic curve at any other point. By definition, 2P = *O* for such a point P. If one wanted to find 3P in this situation, one can add 2P + P. This becomes P + *O* = P Thus 3P = P. 3P = P, 4P = *O*, 5P = P, 6P = *O*, 7P = P, etc.

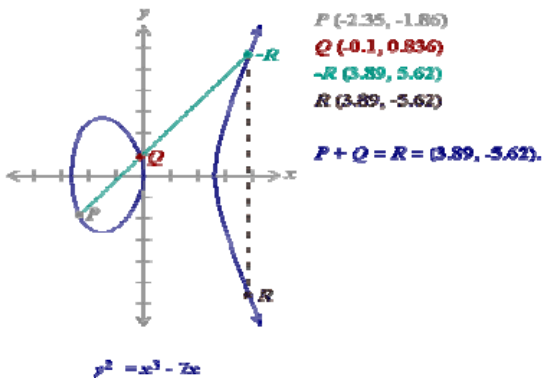

Fig.1. *Example of EC Addition*

At this point the following properties of the addition operation, as defined above, should be clear:

- addition is closed on the set E,
- addition is associative,

- addition is commutative
- *O* is an identity with respect to addition, and
- Evey point on E has an inverse with respect to addition.

In order to show that (E,+) is an abelian group.

A disadvantage of using the real numbers for cryptography, is that it is very hard to store them precisely in computer memory, and to predict how much storage we will need for them. This problem can be solved by using finete fields. i.e. fields with a finite number of elements. Since the number of elements is finite, we can find a unique representation for each of them, which allows us to store and handle the elements in a manageable way. Two types of finite fields are popular for use in Elliptic Curve Cryptography: fields of the form GF(p), with p prime, and fields of the form GF(2^n), with n a positive integer. GF(p) is the same things as Z_p [3,4].

2.2. Elliptic Curves Groups over Finite Fields

Let p>3 be prime. Elliptic curves over Z_p can be defined exactly as they were over the reals (and the addition operation is also defined in an identical fashion) provided that all operatşons over $\Re$ are replaced by analogous operations in Z_p. The elliptic curve $y^2 \equiv x^3 + ax + b(mod\ p)$, where $a,b \in Z_p$ are constants such that $4a^3 + 27b^2 \pmod{p} \neq 0$, together with a special point *O* called the point at infinity. The addition operation on E is defined as follows (where all arithmetic operations are performed in Z_p).

Suppose $P=(x_1,y_1)$ and $Q=(x_2,y_2)$ are points on E. If $x_2 = x_1$ and $y_2 = -y_1$, then P+Q=*O;* otherwise $P+Q=(x_3,y_3)$, where $x_3 = \lambda^2 - x_1 - x_2$, $y_3 = \lambda(x_1 - x_3) - y_1$ and $P \neq Q \Rightarrow \lambda = (y_2 - y_1) / (x_2 - x_1)$, $P=Q \Rightarrow \lambda = (3x_1^2 + a)/(2y_1)$.

Finally define P+*O*=*O*+P=P for all $P \in E$. Note that the addition of points on an elliptic curve over Z_p does not have the nice geometric interpretation that it does on an elliptic curve over the reals. However; the same formulas can be used to define addition, and the resulting pair (E,+) still forms an abelian group.

For example, let p=23 and consider the elliptic curve E: $y^2 \equiv x^3 + x + 4$ defined over Z_{23} (a=1 and b=4). Note that $4a^3 + 27b^2 \pmod{23}$= 4+432=436(mod 23)= $22 \neq 0$,which satisfies the condition for an elliptic group mod 23. For the elliptic group, we are only interested in the nonnegative integers in the quadrant form (0,0) to (p,p) that satisfy the equation mod p. Table 1 lists the points(othet than *O*) that are part of Z_{23}(1,4). In general, the list is created in the following manner:

- For each x such that $0 \leq x < p$, calculate $x^3 + ax + b(mod\ p)$
- For each result from the previous step, determine if it has a square root mod p. If not, there are no points in Z_p(a,b) with this value of x. If so, there will be two values of y that satisfy the square root operation(unless the value is the single y value of 0). These (x,y) values are points in Z_p(a,b).

For a given x, we can test to see if $z= x^3 + x + 4\ mod 23$ is an quadratic residue by applying Euler's criterion. There is an explicit formula to compute square roots of quadratic residues modulo p for primes $p \equiv 3 \pmod 4$. Applying this formula, we have that the square roots of a quadratic residue z are $\pm z^{(p+1)/4}\ \mathbf{mod}\ p = \pm z^{(23+1)/4}\ \mathbf{mod}\ 23 = \pm z^6\ \mathbf{mod}\ 23$ [3]. The results of these computations are tabulated in Table 1.

Table 1. *Points on the elliptic curve* $y^2 \equiv x^3 + x + 4$

x	$x^3+x+4 \mod 23$	Quadratic residue?	y
0	4	yes	2,21
1	6	yes	11,12
2	14	no	
3	11	no	
4	3	yes	7,16
5	19	no	
6	19	no	
7	9	yes	3,20
8	18	yes	8,15
9	6	yes	11,12
10	2	yes	5,18
11	12	yes	9,14
12	19	no	
13	6	yes	11,12
14	2	yes	5,18
15	13	yes	6,17
16	22	no	
17	12	yes	9,14
18	12	yes	9,14
19	5	No	
20	20	No	
21	17	No	
22	2	Yes	5,18

Example 1:

1-) Let P=(4,7) and Q=(13,11). Then P+Q=(x_3, y_3) is computed as follows:

$\lambda \equiv (11-7)/(13-4) \equiv 3 \pmod{23}$

$x_3 = 3^2 - 4 - 13 = -8 \equiv 15 \pmod{23}$, and

$y_3 = 3(4-15) - 7 = -40 \equiv 6 \pmod{23}$.

P+Q=(15,6).

2-) Let P=(4,7). Then 2P=P+P=(x_3, y_3) is computed as follows:

$\lambda \equiv (3(4^2)+1)/14 \equiv 15 \pmod{23}$

$x_3 = 15^2 - 8 = 217 \equiv 10 \pmod{23}$, and

$y_3 = 15(4-10) - 7 = -97 \equiv 18 \pmod{23}$.

Hence 2P=(10,18).

E has 29 points on it. Since any group of prime order is cyclic, it follows that E is isomorphic to Z_{29}, and any point other than the point at infinity is a generator of E. Suppose we take the generator α=(4,7). Then we can compute the "powers" of α (which we will write as multiples of α, since the group operation is additive). 2α=(10,18). The next multiple would be $3\alpha = 2\alpha + \alpha$=(10,18)+(4,7)=(13,11).

Continuing in this fashion, the remaining multiples can be computed to be the following:

α=(4,7)	2α=(10,18)	3α=(13,11)
4α=(15,6)	5α=(8,8)	6α=(1,11)
7α=(7,20)	8α=(18,9)	9α=(9,12)
10α=(11,9)	11α=(17,9)	12α=(14,18)
13α=(0,2)	14α=(22,5)	15α=(22,18)
16α=(0,21)	17α=(14,5)	18α=(17,14)
19α=(11,14)	20α=(9,11)	21α=(18,14)
22α=(7,3)	23α=(1,12)	24α=(8,15)
25α=(15,17)	26α=(13,12)	27α=(10,5)
28α=(4,16)	29α=*O*	

Hence, as we already knew, α=(4,7) is indeed a primitive element.

2.2.1 Basic Facts

GROUP ORDER: Let E be an elliptic curve over a finete field *Zp.* More precisely, a well-known theorem due to Hasse asserts that the number of

points on E, which we denote by #E, satisfies the following inequality :

$p+1-2\sqrt{p} \le \#E \le p+1+2\sqrt{p}$.

In other words, the order of an elliptic curve E(*Zp)* is roughly equal to the size p of the underlying field. If p is large , say around 10^{20}, it is infeasible to count the points on an elliptic curve by listing them.

More sophisticated algorithms have been developed by Schoof, Atkin, Elkies, and others to deal with this problem. Now, given that we can compute #E, we further want to find a cyclic subgroup of E in which the discrete log problem is intractible. So we would like to know something about the structure of the group E. The following definition gives a considerable amount of information on the group structure of E[3].

GROUP STRUCTURE: Let E be an elliptic curve defined over *Zp*, where p is prime, p > 3. Then there exist integers n1 and n2 such that E is isomorphic to $Z_{n1} x Z_{n2}$. Further; $n_2 \mid n_1$ and $n_2 \mid (p - 1)$. Hence, if the integers n_1 and n_2 can be computed, then we know that E has a cyclic subgroup isomorphic to Z_n, that can potentially be used as a setting for an ElGamal Cryptosystem. Note that if n_2=1, then E is a cyclic group. Also, if #E is a prime, or the product of distinct primes, then E must be a cyclic groupindexcyclic group. For example; consider the elliptic curve $E(Z_{23})$ defined above. Since #E(Z_{23})=29, which is prime, $E(Z_{23})$ is cyclic and any point other than *O is a generator of* $E(Z_{23})$.

For example, α=(4,7) is a generator as shown in example 1. The Shanks and Pohlig-Hellman algorithms apply to the elliptic curve logarithm problem, but there is no known adaptation of the index calculus method to elliptic curves. However, there is a method of exploiting an explicit isomorphism between elliptic curves and finite fields that leads to efficient algorithms for certain classes of elliptic curves. This technique, due to Menezes, Okamoto and Vanstone, can be applied to some particular examples within a special class of elliptic curves called supersingular curves that were suggested for use in cryptosystems[5,6]. If the supersingular curves are avoided, however, then it appears that an elliptic curve having a cyclic subgroup of size 2^{160} will provide a secure setting for a cryptosystem, provided that the order of the subgroup is divisible by at least one large prime factor (again, to guard against a Pohlig-Hellman attack).

2.3. Elliptic Curve Groups over GF(2^n)

Elliptic Curve groups over F_{2^n} have a finite number of points, and their arithmetic involves no round off error. This combined with the binary nature of the field, F_{2^n} arithmetic can be performed very efficiently by a computer. An elliptic curve E over F_{2^n} is defined by an equation of the form $y^2 + xy = x^3 + ax^2 + b$, where $a,b \in F_{2^n}$, and $b \neq 0$. The set $E(F_{2^n})$ consists of all points (x,y), $x \in F_{2^n}$, $y \in F_{2^n}$, together with a special point *O* called the point at infinity[7,8].

Example 2 : As a very small example, consider the field F_24, defined by using polynomial representation with the irreducible polynomial $f(x) = x^4 + x + 1$.

The element g = (0010) is a generator for the field.

The powers of g are:

g^0=(0001)	g^1 = (0010)	g^2 = (0100)
g^3 = (1000)	g^4 = (0011)	g^5 = (0110)
g^6 = (1100)	g^7 = (1011)	g^8 = (0101)
g^9 = (1010)	g^{10} = (0111)	g^{11} = (1110)
g^{12} = (1111)	g^{13} = (1101)	g^{14} = (1001)
g^{15}=(0001)		

In a true cryptographic application, the parameter m must be large enough to preclude the efficient generation of such a table otherwise the cryptosystem can be broken. In today's practice, n = 160 is a suitable choice. The use of generator notation (g^e) rather than bit string notation, as used in the following example. Also, using generator notation allows multiplication without reference to the irreducible polynomial $f(x) = x^4 + x + 1$.

Consider the elliptic curve

$y^2 + xy = x^3 + g^4x^2 + 1$.

Here $a = g^4$ and $b = g^0$ =1.

The point (g^5, g^3) satisfies this equation over F_{2^n} : $y^2+xy=x^3+g^4x^2+1$, $(g^3)^2 + g^5g^3 = (g^5)^3 + g^4g^{10} + 1$ $g^6 + g^8 = g^{15} + g^{14} + 1 \Rightarrow$ $(1100)+(0101)=(0001)+(1001)+(0001)$ $\Rightarrow (1001)=(1001)$

The fifteen points which satisfy this equation are:

$(1,g^{13})$	(g^3,g^{13})	(g^5,g^{11})	(g^6,g^{14})	(g^9,g^{13})
(g^{10},g^8)	(g^{12},g^{12})	$(1,g^6)$	(g^3,g^8)	(g^5,g^3)
(g^6,g^8)	(g^9,g^{10})	(g^{10},g)	$(g^{12},0)$	$(0,1)$

ADDITION FORMULA: As with elliptic curves over Z_p , there is a chord-and-tanget rule for adding points on an elliptic curve $E(F_{2^n})$ to give a third elliptic curve point. Together with this addition operation, the set of points $E(F_{2^n})$ forms a group with O serving as its identity. The algebraic formula for the sum of two points and the double of a point are the following[4].

- $P+O=O+P=P$ for all $P \in E(F_{2^n})$
- If $P=(x,y) \in E(F_{2^n})$, then $(x,y)+(x,x+y)=O$.(The point $(x,x+y)$ is denoted by -P, and is called the negative of P; observe that -P is indeed a point on the curve.)
- (Point addition) Let $P=(x_1,y_1) \in E(F_{2^n})$ and $Q=(x_2,y_2) \in E(F_{2^n})$, where $P \neq \pm Q$. Then $P+Q=(x_3,y_3)$, where $x_3=((y_1+y_2)/(x_1+x_2))^2+(y_1+y_2)/(x_1+x_2)+x_1+x_2+a$ and $y_3=((y_1+y_2)/(x_1+x_2))(x_1+x_2)+x_3+y_1$
- (Point doubling) Let $P=(x_1,y_1) \in E(F_{2^n})$, where $P \neq -P$. Then $2P=(x_3,y_3)$, where $x_3=x_1^2+b/x_1^2$ and $y_3= x_1^2+(x_1+y_1/x_1)x_3+x_3$

Example 3: Consider the elliptic curve defined in before example

1-) Let $P=(g^6,g^8)$ and $Q=(g^3,g^{13})$. Then $P+Q=(x_3,y_3)$ is computed as follows:

$\mathbf{x}_3=((g^8+g^{13})/(g^6+g^3))^2+(g^8+g^{13})/(g^6+g^3)+g^6+g^3+g^4=(g^3/g^2)^2+g^3/g^2+g^6+g^3+g^4=1$

$\mathbf{y}_3=(g^8+g^{13})/(g^6+g^3)(g^6+1)+1+g^8=(g^3/g^2)\,g^{13}+g^2=g^{13}$. Hence $P+Q=(1, g^{13})$

2-) Let $P=(g^6,g^8)$. Then $2P=P+P=(x_3,y_3)$ is computed as follows:

$x_3=(g^6)^2+(1/(g^6)^2)=g^{12}+g^3=g^{10}$ and $y_3=(g^6)^2+(g^6+g^8/g^6)\,g^{10}+g^{10}=g^{12}+g^{13}+g^{10}=g^8$. Hence $2P=(g^{10}, g^8)$.

3. ELLIPTIC CURVE CRYPTOSYSTEMS

Elliptic curves versions exist for many cryptosystems, in particular those involving discrete logarithms. An advantage of elliptic curves over working with integers mod p is the following. In the integers, it is possible to use the factorization into primes(especially small primes) to attack the discrete logarithm problem. More specifically, the ECC relies upon the difficulty of the Elliptic Curve Discrete Logarithm Problem (ECDLP). Recall that we examined two geometrically defined operations over certain elliptic curve groups. These two operations were point addition and point doubling. By selecting a point in a elliptic curve group, one can double it to obtain the point 2P. After that, one can add the point P to the point 2P to obtain the point 3P. The determination of a point nP in this manner is referred to as Scalar Multiplication of a point. The ECDLP is based upon the interactability of scalar multiplication products. Specifically, consider the operation called "scalar multiplication" under additive notation: that is, computing kP by adding together k copies of the point P. Using multiplicative notation, this operation consists of multiplying together k copies of the point P, yielding the point P*P*P*P&.*P = Pk[3,9].

In the multiplicative group Zp*, the discrete logarithm problem is: given elements r and q of the group, and a prime p, find a number k such that r = qk mod p.

If the elliptic curve groups is described using multiplicative notation, then the elliptic curve discrete logarithm problem is: given points P and Q in the group, find a number that Pk = Q; k is called the discrete logarithm of Q to the base P. When the elliptic curve group is described using additive notation, the elliptic curve discrete logarithm problem is: given points P and Q in the group, find a number k such that Pk = Q

Example 4: In the elliptic curve group defined by y2=x3+9x+17 over F23, What is the discrete logarithm k of Q=(4,5) to the base P=(16,5) ?

One way to find k is to compute multiples of P until Q is found.
The first few multiples of P are: P = (16,5) 2P = (20,20) 3P = (14,14) 4P = (19,20) 5P = (13,10) 6P = (7,3) 7P = (8,7) 8P = (12,17) 9P = (4,5) Since 9P = (4,5) = Q, the discrete logarithm of Q to the base P is k=9.
In a real application, k would be large enough such that it would be infeasible to determine k in this manner.

In the following subsections, three elliptic curve versions of classical algorithms are described[3,4].

3.1. An Elliptic Curve ElGamal Cryptosystem

ElGamal Cryptosystem : Alice wants to send a message x to Bob, so Bob chooses a large prime p and an integer α mod p. He also chooses a secret integer a and computes $\beta = \alpha^a$ ***(mod p).***

Bob makes p, α, β public and keeps a secret. Alice chooses a random k and computes y_1 and y_2, where $y_1 \equiv \alpha^k$ and $y_2 \equiv x\beta^k$ (mod p). She sends (y_1, y_2) to Bob, who then decrypts by calculating $x \equiv y_2 y_1^{-a}$ (mod p).

The elliptic curve version : Bob chooses an elliptic curve E (mod p), where p is a large prime. He chooses a point α on E and a secret integer a. He computes $\beta = a\alpha$ $(= \alpha + \alpha + ... + \alpha)$. The points α and β are made public, while a is kept secret. Alice expresses her message as a point x on E. She chooses a random integer k, computes $y_1 = k\alpha$ and $y_2 = x + k\beta$, and sends the pair y_1, y_2 to Bob. Bob decrypts by calculating $x = y_2 - ay_1$

Example 5: E: $y^2 \equiv x^3 + x + 4$ *over* Z_{23}.
Suppose that $\alpha = (4,7)$ and Bob's private key is 3, so $\beta = 3\alpha = (13,11)$. Thus the encryption operation is $e_k(x,k) = (k(4,7), x+k(13,11))$, where $x \in E$ and $0 \le k \le 28$, and the decryption operation is $d_k(y_1, y_2) = y_2 - 3y_1$. Suppose that Alice wishes to encrypt the plaintext x=(1,11)(which is a point on E). If she chooses the random value k=5, then she will compute $y_1 = 5(4,7) = (8,8)$ and $y_2 = (1,11) + 5(13,11) = (1,11) + (22,18) = (18,14)$.

Hence, y=((8,8),(18,14)). Now, if Bob receives the ciphertext y, he decrypts it as follows:

x = (18,14) - 3(8,8) = (18,14) - (22,18) = (18,14) + (22,5) = (1,11).

Hence, the decryption yields the correct plaintext. There are some practical difficulties in implementing an ElGamal Cryptosystem on an elliptic curve. This system, when implemented in Z_p (or in $GF(p^n)$ with n > 1) has a message expansion factor of two, An elliptic curve implementation has a message expansion factor of (about) four. This happens since there are approximately p plaintexts, but each ciphertext consists of four field elements. A more serious problem is that the plaintext space consists of the points on the curve E, and there is no convenient method known of deterministically generating points on E.

3.2. Elliptic Curve Menezes-Vanstone Cryptosystem

A more efficient variation has been found by Menezes and Vanstone. In this variation, the elliptic curve is used for "masking," and plaintexts and ciphertexts are allowed to be arbitrary ordered pairs of (nonzero) field elements (i.e., they are not required to be points on E). This yields a message expansion factor of two, the same as in the original ElGamal Cryptosystem.

The Menezes-Vanstone Cryptosystem: Let E be an elliptic curve defined over iz, (p > 3 prime)

such that E contains a cyclic subgroup H in which the discrete log problem is intractible. $P=Z_p^* \times Z_p^*$, $C=E \times Z_p^* \times Z_p^*$, and define $K=\{(E, \alpha, a, \beta): \beta = a\alpha\}$, where $\alpha \in E$. The values α and p are public, and a is secret. For $K=(E, \alpha, a, \beta)$ for a (secret) random number $k \in Z_{|H|}$, and for $x=(x_1, x_2) \in Z_p^* \times Z_p^*$, define $e_k(x,k)=(y_0, y_1, y_2)$, where $y_0 = k\alpha$, $(c_1, c_2) = k\beta$,

$y_1 = c_1 x_1 \bmod p$, and $y_2 = c_2 x_2 \bmod p$.

For a ciphertext $y=(y_0, y_1, y_2)$, define $d_k(y)=(y_1 c_1^{-1} \bmod p, y_2 c_2^{-1} \bmod p)$, where a $y_0=(c_1, c_2)$.

If we return to the curve $y^2 \equiv x^3 + x + 4$ *over* Z_{23}, we see that the Menezes - Vanstone Cryptosystem allows 20x20=400 plaintexts, as compared 29 in the original system.

Example 6: As in the previous example, suppose that $\alpha=(4,7)$ and Bob's secret "exponent" is 3, so $\beta = 3\alpha = (13,11)$.

Suppose Alice wants to encrypt the plaintext x=(9,1) (note that x is not a point on E), and she chooses the random value k=5. First, she computes $y_0 = k\alpha = 5(4,7) = (8,8)$ and $k\beta = 5(13,11) = (22,18)$, so $c_1=22$ and $c_2=18$. Next, she calculates $y_1 = c_1 x_1 \bmod p = 22x9 \bmod 23 = 14$, and $y_2 = c_2 x_2 \bmod p = 18x1 \bmod 23 = 18$.

The ciphertext she sends to Bob is $y = (y_0, y_1, y_2) = ((8,8),14,18)$. When Bob receives the ciphertext y, he first computes $(c_1, c_2) = a y_0 = 3(8,8) = (22,18)$, and then $x = (y_1 c_1^{-1} \bmod p, y_2 c_2^{-1} \bmod p) = (14x22^{-1} \bmod 23, 18x18^{-1} \bmod 23) = (14x22 \bmod 23, 18x9 \bmod 23) = (9,1)$.

Hence, the decryption yields the correct plaintext.

3.3. Elliptic Curve Diffie-Hellman Key Exchange

Diffie-Hellman Key Exchange: Alice and Bob want to establish a key for communicating. The Diffie-Hellman scheme for accomplishing this is as follows:

- Either Alice or Bob selects a large, secure prime number p and a primitive root α (mod p). Both p and α can be made public.
- Alice chooses a secret random x with $1 \le x \le p-2$, and Bob selects a secret random y with $1 \le y \le p-2$.
- Alice sends α^x (mod p) to Bob, and Bob sends α^y (mod p) to Alice.
- Using the messages that they each have received, they can each calculate the session key K. Alice calculates K by $K \equiv (\alpha^y)^x \pmod p$, and Bob calculates K by $K \equiv (\alpha^x)^y \pmod p$.

Elliptic Curve Diffie-Hellman Key Exchange: Alice and Bob want to exchange a key. In order to do so, they agree on a public basepoint α on an elliptic curve $y^2 \equiv x^3 + ax + b \pmod p$. Let's choose p=23 and a=1 an $\alpha=(4,7)$. This gives us b=4. Alice chooses N_A randomly and Bob chooses N_B randomly. Let's suppose $N_A=12$ and $N_B=5$. They keep these private to themselves but publish $N_A \alpha$ and $N_B \alpha$.

In our case, we have $N_A \alpha = (14,18)$ and $N_B \alpha = (8,8)$. Alice now takes $N_B \alpha$ and multiplies by N_A to get the key: $N_A(N_B \alpha) = 12(8,8) = (10,18)$ Similarly, Bob takes $N_A \alpha$ and multiplies by N_B to get the key: $N_B(N_A \alpha) = 5(14,18) = (10,18)$. Notice that they have the same key.

3.4. Elliptic Curve Digital Signature Algorithm

In 2000, the Elliptic Curve Digital Signature Algorithm(ECDSA) was approved as FIPS 186-2. Let p be a prime or a power of two, and let E be an elliptic curve defined over Z_p.

Let α be a point on E having prime order q, such that the Discrete Logarithm problem in

$<\alpha>$ is infeasible. Let P=$\{0,1\}^*$, $\alpha = Z_q^* \times Z_q^*$, and define K={(p, q, E, α, m, β): $\beta = m\alpha$}, where $0 \le m \le q-1$.

The values p, q, E, α and β are the public key, and m is the private key. For K=(p, q, E, α, m, β), and for a (secret) random number k, $1 \le k \le$ q-1, define $sig_K(x,k) = (r,s)$, where $k\alpha = (u,v)$, r = u mod q and $s = k^{-1}(SHA-1(x) + mr) \bmod q$. (If either r = 0 or s = 0 , a new random value of k should be chosen.)

For $x \in \{0,1\}^*$ and r, $s \in Z_q^*$, verification is done by performing the following computations:

$w = s^{-1} \bmod q$, $i = wSHA-1(x) \bmod q$,
$j = wr \bmod q$, $(u,v) = i\alpha + j\beta$,
$ver_K(x,(r,s)) = true \Leftrightarrow u \bmod q = r$.

Example 7: We will base our example on the same elliptic curve that was used previous example, namely, E: $y^2 \equiv x^3 + x + 4$ *over* Z_{23}. The parameters of the signature scheme are p=23, q=29, Suppose that α=(4,7), and Bob's private key is m=3, so $\beta = 3\alpha$=(13,11).

Suppose we have a message x with SHA-1(x)=4, and Alice wants to sign the message x using the random value k = 5. She will compute (u,v) = 5(4,7) = (8,8), r = u ***mod*** 29 = 8, and s = $5^{-1}(4+3x8) \bmod 29 = 23$. Therefore (8, 23) is the signature.

Bob verifies the signature by performing the following computations:

w = $23^{-1} \bmod 29 = 24$,
$i = 24x4 \bmod 29 = 9$,
$j = 24x8 \bmod 29 = 18$,
$(u,v) = 9\alpha + 18\beta = (8,8)$,
$u \bmod 29 = 8 = r$.

Hence, the signature is verified.

4. SECURITY OF ELLIPTIC CURVE CRYPTOGRAPHY

Because of the apparent difficulty of the ECDLP, highly secure systems can be designed that require much smaller key sizes than RSA or DSA in order to achieve comparable levels of security. ECC demands less resources. On the server, no particular performance need for switching to ECC. In the client, there are good reasons. Table 2 gives approximate parameter sizes for comparable strength elliptic curve systems and RSA. Table 3 gives the key size estimate values of RSA and ECC. This is based on current best techniques for solving the ECDLP and factorising large integers. Consequently, using elliptic curves, we can define highly secure systems that use much smaller keys compared with equivalent "traditional" systems, such as RSA or DSA. In particular, such systems require relatively modest computing capability and memory - ideal, for example, for a smart card or mobile phone[10,11].

Table 2. *Comparative Bit-Lengths*

Elliptic curve system (order of base point P)	RSA (length of modulus n)
106 bits	512 bits
132 bits	768 bits
160 bits	1024 bits
224 bits	2048 bits

Table 3. *Key Size-Estimates*

Year	RSA	ECC
2002	1028	135
2005	1149	139
2010	1369	146
2015	1613	154
2020	1882	160

The security of ECC depends on how difficult it is to determine k given kP and P. This referred to as the elliptic curve logarithm problem. The fastest known technique for taking the elliptic curve logarithm is known as the Pollard rho method. Table 4 compares the efficiency of this method with factoring a number into two primes using the general number field sieve. As can be seen, a considerably smaller key size can be used for ECC compared to RSA is. Futhermore, for equal key lengths, the computational effort required for ECC and RSA is comparable. Thus,

there is a computational advantage to using ECC with a shorter key length than a comparably secure RSA[10,11].

Table 4. *Computational Effort for Cryptanalysis of Elliptic Curve Cryptography Compared to RSA*

Key Size	MIPS-Years
150	3.8×10^{10}
205	7.1×10^{18}
234	1.6×10^{28}

(a) Elliptic Curve Logarithms Using the Pollard rho Method

Key Size	MIPS-Years
512	3×10^{4}
768	2×10^{8}
1024	3×10^{11}
1280	1×10^{14}
1536	3×10^{16}
2048	3×10^{20}

(b) Integer Factorization Using the General Number Field Sieve

5. SOME PROBLEMS AND ISSUES WITH ELLIPTIC CURVE SYSTEMS

5.1. Security

The main issue is that the true difficulty of the ECDLP is not fully understood. Recent research has shown that some elliptic curves that were believed suitable for elliptic curve cryptography are in fact not appropriate. For example, if the order of the base point P is equal to the prime p then it turns out that the ECDLP can be solved efficiently[10,11].

5.2. Curve Generation

When defining an elliptic curve system, a curve and a base point (P) are required. Note that these elements are not secret (and may be the same for all system users). For a given curve and base point, it is trivial to generate public and private keys for users (the private key is simply a random integer k and the public key is the point kP on the curve). The difficulty of the ECDLP means that it is infeasible to deduce the private key from the public key. However, it is an extremely difficult problem to generate a suitable curve and base point in the first place. The main problem is how to count the number of points on the curve. Having done this, it is then necessary to select a suitable base point P, which must have a large order to ensure the difficulty of the ECDLP. But the order of P must divide the number of points on the curve (remember that the points on the curve, together with the point at infinity form a finite group). So, having found the number of points on the curve, it is quite likely that a suitable base point cannot be found. There are a variety of other restrictions that must be satisfied when generating curves[12].

5.3. Incompatible Systems

The "odd" and "even" elliptic curve implementations are similar, but sufficiently different to ensure that an "odd" system will be incompatible with an "even" system. Furthermore, within the even case there are a number of ways to represent curves and base points and a user with a system appropriate for one representation may not be able to communicate successfully with a user with a different representation. This is different to the case of RSA, where (in theory) all implementations are compatible. Ignoring issues of compatibility, there are good reasons to use "even" elliptic curve systems, mainly to do with speed of processing, but here again users need to be wary. A number of experts in this area believe that the ECDLP may be easier to solve for the even case than the odd case, although it must be admitted that the evidence for such assertions is a little flimsy[11,12].

5.4. Royalties and Patents

The issue of royalties and patents relevant to elliptic curve cryptosystems is somewhat unclear. There are a number of patents in this area, mainly applicable to the even case[10].

5.5. Processing

We have already mentioned that because elliptic curve systems use small key sizes then less computing power is required than for RSA. How does this translate into speed of processing? Table 5 provides comparative figures for RSA and ECDSA (odd case) signature generation and verification, where both algorithms were implemented using two parallel Motorola 56303 Digital Signal Processors (66 MHz). Note that the RSA signature verify figures assume the use of a public exponent e = 65537[12].

Table 5. *Comparative Processing Times*

	Generate Signature	**Verify Signature**
RSA (1024 bits)	25 ms	2 ms
ECDSA (160 bits)	32 ms	33 ms
RSA (2048 bits)	120 ms	5 ms
ECDSA (216 bits)	68 ms	70 ms

Clearly, different implementations will yield different timings, but the pattern is clear. As key sizes increase, signature generation for ECDSA becomes significantly faster than comparable RSA systems. This difference would be magnified even further if only a single processor were available. On the other hand, signature verification using ECDSA is much slower than for RSA and again this difference would be even greater if only a single processor were available. Note that ECDSA processing times could be improved somewhat if the even case were implemented.The time taken for signature verification when using ECDSA may have an adverse impact on system performance. Many systems have a large number of remote devices communicating with a central server. The time taken by the remote device to generate a signature may not be important (several seconds may be acceptable), but the server must be able to validate signatures quickly. RSA based systems (even using large keys) may be more applicable in some circumstances than elliptic curve systems.

6. CONCLUSIONS

Elliptic curve systems are increasingly seen as an alternative to RSA, rather than a replacement. There are potential advantages, especially when used in devices with limited processing capability and/or memory. Typical applications include: - m-commerce (e.g. WAP mobile phone, hand-held devices); - smart card systems (e.g. EMV); - e-commerce and banking applications (e.g. SET); - internet based applications (e.g. SSL)[10,11,12].

There are, however, some problems and issues that are inhibiting the widespread adoption of elliptic curve systems. These include: - the real security of such systems is still not well understood; - difficulty of generating suitable curves; - incompatibility of implementations; - royalties and patents; - relatively slow signature verification[10,11,12].

REFERENCES

[1] Miller, Victor S. *"Elliptic Curves and their use in Cryptography"* DIMACS Workshop on Unusual Applications of Number Theory, 21 March 1997

[2] Win, Erik De and Preneel, *"Elliptic Curve Public Key Cryptosystems-an introduction"* State of the Art in Applied Cryptography 1997 pp.131-141

[3] Stinson, Douglas R. *Cryptography: Theory and Practice.* CRC Press, 1995, and 2002(second edition).

[4] Wade Trappe&Lawrence C. Washington, *Introduction to Cryptography with Coding Theory*. Prentice-Hall, 2002

[5] Juristic, A. and Menezes, A. *"Elliptic Curves and Cryptography."* Dr. Dobb's Journal, April 1997

[6] Johnson, D. and Menezes, A. *"The Elliptic Curve Digital Signature Algorithm"* Technical Report. CORR 99-34, Dep. Of C&O, University of Waterloo, Canada. Aug.23, 1999 Update: Feb. 24, 2000

[7] Michael Rosing, *Implementing Elliptic Curve Cryptography.* Manning Publications Co. 1999

[8] Dr. Michael J Ganley, *Elliptic Curve Cryptography,* Thales e-Security Limited, 2001

[9] William Stallings, *Cryptography and Network Security, Principles and Practice.* Prentice-Hall 1999 (Second Edition)

[10] M.Aydos, E.Savaş, and Ç.K.Koç, *"Implementing network security protocols based on elliptic curve cryptography",* Proceedings of the Fourth Symposium on Computer Networks, pp. 130-139, Istanbul, Turkey, May 20-21 1999.

[11] Çetin Kaya KOÇ, *Cryptography: State of the Art and Current Trends*, Istanbul, Turkey, SACIS 2003

[12]http://www.certicom.com/resources/ecc_tutorial/ecc_tutorial.html