The University of Adelaide

AUTO-ID LABS

Elliptic Curve Cryptography

Raja Ghosal and Peter H. Cole

Auto-ID Labs White Paper WP-HARDWARE-026

Mr. Raja Ghosal
PhD Student, Auto-ID Lab, ADELAIDE
School of Electrical and Electronics
Engineering,
The University of Adelaide

Prof. Peter H. Cole
Research Director, Auto-ID Lab, ADELAIDE
School of Electrical and Electronics
Engineering,
The University of Adelaide

Contact:
rghosal@eleceng.adelaide.edu.au or cole@eleceng.adelaide.edu.au.

Internet: www.autoidlabs.org

Abstract

Public key cryptography systems are based on sound mathematical foundations that are designed to make the problem hard for an intruder to break into the system. The major approaches that since 1976 have withstood intruder attacks, are the discrete logarithm

problem (as in D-H), and the integer factorisation problem as in RSA. The growing area of lightweight devices, such as mobile cell phones, PDA's, palmtops, where memory, processing power, bandwidths are limited, are constrained in using public key cryptography systems, which are based on large key sizes. The larger key requires higher computation power.

Elliptic Curve Cryptography (ECC) is a newer approach, with a novelty of low key size for the user, and hard exponential time challenge for an intruder to break into the system. In ECC a 160 bits key, provides the same security as RSA 1024 bits key, thus lower computer power is required. The advantage of elliptic curve cryptosystems is the absence of subexponential time algorithms, for attack.

Elliptic Curves belong to a general class of curves, called Hyperelliptic curves, of which elliptic curves is a special case, with genus, g=1. Hyperelliptic curves were initially candidates, to the next progression, or generalizations, to more secure systems, as they appeared to require even shorter key lengths than elliptic curves for the same level of security. It is found, however, that hyperelliptic curves of genus g=4, or higher, do not have the same level of security, as genus 2 or 3 curves, where attacks of subexponential time algorithms have been. Hence elliptic curves are the optimal practical solution from this family of curves.

Elliptic curve systems have been fine tuned to have analogues of other systems, such as El-Gamal, Diffie-Hellman, and RSA. There is not much gain in the difficulty of the integer factorisation problem in RSA over the field of elliptic curves, as contrasted with the discrete logarithm problem in elliptic curves which is much harder than in El-Gamal, and Diffie-Hellman systems. Hence elliptic curves in cryptography usage are based on the hardness of the discrete logarithm problem.

Many of the definitions in this document have come from Wikipedia (http://en.wikipedia.org/), and are licensed under the GNU Free Documentation License.

1. Introduction

This section introduces the developments in Elliptic Curves, and why they have become a very useful applications, to Cryptography, the area of Elliptic Curve Cryptography (ECC).

1.1 Elliptic Curves

Elliptic curves in general are the two dimensional analogues of trigonometric functions or curves, in the complex domain [2] and [34]. Unlike the single period trigonometric functions, such as sine, cosine (y = cos(x), where both x, and y are real variables), elliptic curves have double periods. This is due to their being in the complex domain for the general case (eg: z = f(w), where w, z are complex numbers of the type x + iy, where x and y are real, and i = sqrt (-1). Each of the components x and y have a period) [32]. They are the inverse solutions to elliptic functions [33]. The inverse functions are studied to obtain the perimeter of curves. Hence the name elliptic curves, as they measure the perimeter of ellipses. They do not, however, represent the ellipses of conic sections. In general, conic sections, parabolas, ellipses, circles, hyperbolas and other planar figures as triangle, are known as genus, g=0 figures. Elliptic curves are genus g=1. Hyperelliptic curves are a generalization with genus, g>1. Genus is a term from topology and measures the number of torsions or twists in the figure.

Graphical images: To assist very basic understanding of elliptic curves, the graphical images of elliptic curves and operations thereon, is given in a simple tutorial form by Certicom [6] using a javascript based geometrical experimental “screen window”.

Elliptic curves over rational numbers: The domain (Q) of rational numbers is found to be the minimal covering domain in the practical usage of cryptography, for reasons below:

A typical representation of an elliptic curve is: $y^2 = x^3 + ax + b$, with a, b integers [2], [6]. This avoids the not used features such as, double periods, associated with elliptic curves in complex numbers domain, which in any event do not contribute anything extra to assist the encryption/decryption problem. With real numbers for the x, and y coordinates of the points, the inverse or discrete-log k (k to be explained later), may not always be integer. That the discrete log, k, is an integer makes the problem harder. This is exactly as restricting solutions to integer only to an equation in x and y, which is much more difficult (as in Diophantine equations), or an integer programming problem, than if rational or real solutions were permitted. Any discrete solution space makes the problem more complex, than a continuos solution space. We avoid curves where points (x,y), such that, x, and/or y is irrational, or transcendental. In cryptography, elliptic curves restricted over the domain of rational numbers (Q), is found to provide sufficient hardness in the discrete logarithm problem. We do we not restrict further only to points with x,y integers, because like Diffie-Hellman, or RSA factorisation, our solution in elliptic curves is always an integer, k, (the inverse or the discrete-log). In the elliptic curve scalar point multiplication [6], M = kN, where k is a representation of the plaintext or ciphertext message, as in Diffie-Hellman, RSA, and other systems.

For k to be an integer, we have to allow the coordinates of points (x,y) to be rational numbers. Thus points M, and P on the elliptic curves are allowed to take (x,y) values in rational numbers, such that M = kP where this operation is called scalar multiplication [2], [6], [19], [27] and [29].

Lightweight Devices using ECC: Algorithmic complexity theory tries to find optimal procedures, both in computational time required to solve the problem, and the memory space, required, to perform the intermediate steps. ECC addresses the space complexity (small key sizes for same security as other systems as RSA) very well, and hence the corresponding the mathematical operations on the keys take less computer power [30]. The developments are discussed in detail sec 12.2 on lightweight cryptography.

Prime factorisation over elliptic curves: The study of elliptic curve is an old branch of mathematics based on some of the elliptic functions of Weierstrass [32], [2]. The applications of Elliptic Curve to cryptography, was independently discovered by Koblitz and Miller (1985) [15] and [17]. Interests in Elliptic Curve Cryptography (ECC) arose from the results of Arjen Lenstra, (1984) [2], that the factorisation to primes of a composite number in the elliptic fields modulo EF(n)), is more difficult than the traditional methods over Galois Field GF(n) (or modulo (n) arithmetic). Breaking into an RSA system, requires the intruder to obtain the 2 large prime factors, of the public key, n (or the publicly known number, n, used as the modulo).

No sub-exponential time algorithm: Balasubramaniam [2] points out that RSA, Diffie-Hellman and various other cryptosystems require only the usage of finite abelian groups G. These are simpler than groups over elliptic curves. The advantage of elliptic curve cryptosystems is the absence of sub-exponential time algorithms that could find discrete-logs in these groups. This leads to the use of the abelian group of points of an elliptic curve, that is much smaller in size, at the same time maintains the same level of security. In turns out the discrete-logarithm problem is much harder over elliptic curves than the integer factorisation like RSA. Hence the discrete log approach taken in elliptic curve cryptography [2].

Preference for discrete log systems: Based on various Elliptic Curve (EC) analogues to exisiting systems, like EC-DHP (Diffie Hellman), EC-DLP (Discrete Logarithm), EC-ElGamal, researchers, Koyama, Maurer, Okamoto, and Vanstone [ref [10] in [2]] proposed EC-analogues of RSA. In such EC-RSA system one works with elliptic curves defined over the ring Zn (n is composite). The order of the abelian group of points of the elliptic curve serves as the trapdoor. The security of these EC-RSA variants depends on the difficulty of factoring n. Researches of many including Kurosawa, Okada, and Tsji [[11] in [2], Pinch [[16] in [2]], Kaliski, [[6] in [2]], and Bleichenbauer [[2] in [2]], show that they do not have significant advantages over their RSA counterparts. Hence the focus of Elliptic Curve cryptography has been the hardness of the discrete logarithms problems.

2. Group Theory and Finite Fields in Cryptography

Modern algebra, like various other branches of mathematics, offers conceptual models for design, analysis, and proof for wide range of problems. The most constrained structures of modern algebra are fields, and after them are rings. At the simplest end of spectrum is the subgroup structures monoids, semi-groups (subsets of group, eg: not having an inverse, such as operations on strings, or languages, such a concatenation of strings. Strings do not have inverses). Without an inverse a decryption is not possible for an encryption. Hence group is first of the simplest and most complete and robust algebraic structure, on which to base cryptography design. Groups which also obey commutative or symmetric property are known as Abelian groups. Abelian groups are extensively used in cryptography, as the order of the sender-receiver transmission should not confuse the common key.

Diffie-Hellman key exchange: In the Diffie-Hellman key exchange, user A, sends g^a mod(p), with its random number generated, a. User B, sends g^b mod(p) with its random number, b. User B, receives A's g^a mod(p), and the user A receives B's g^b mod (p).In absence of the commutative property the same or identical key cannot be constructed by both sides. To see this, user B, using its random number b, creates key $K_b = (g^a)^b$ (mod p) = g^{ab} (mod p), while user A, using its random number, a, generates key $K_a = (g^b)^a$ mod(p) = g^{ba} mod (p) . The commutative property in abelian group ensures a*b = b*a mod(p), so $K_b = g^{ab}$ (mod p) , and, $K_a = g^{ba}$ (mod p) are identical and equal to a common key, K, between users A and B.

The abelian group of points of an elliptic curve, due to the smaller key size (and hence lower number of members of the closed set), that is much smaller in size, at the same time maintains the same level of security. Closure, a fundamental property of groups, is used. The modulo (n) operation causes the domain to have finite number of members, (or solutions) [21],[27] and [29]. This ensures the problem is solvable for the valid receiver, as well as for the problem to be hard eg: discrete log (for Diffie-Hellman, or Elliptic Curves, and prime factorisation for RSA). We note that for a non-group say, $y = x^a$, which is not limited (not closed), but over infinite real numbers, or integers. It is easy for an intruder over time to map, or guess, the exponential pattern, from the random samples eavesdropped. If we modify this to $y = x^a$ mod(n), where a, x, y, n are integers, then the input-output relationship, (originally, an exponential relationship), between x, and y values now becomes more random, and hence it becomes much harder for an intruder to guess any pattern, [6]. At the same time, given y, and n, publicly known values in public key cryptography, it becomes very difficult to guess x. This is due to the hardness of the discrete log problem which is due to the group closure requirements, and is achieved via the trapdoor, modulo(n) function [12], [19] and [31].

3. Basics of Elliptic Curves

The mathematics of elliptic curves, used in cryptography, uses the fundamentals basic theory as above, and is also based on diophantine equations. The elliptic curve used as the underlying field, (EC) $y^2 = x^3 + ax + b$, all variables, x, y, and parameters, a, b must be integers. (EC is used in lieu of a Galois Field GF(p), where p is a prime number as typically modulo arithmetic, or characteristic 2 fields, such as irreducible polynomial fields, eg: GF(2^n) as the latter are easier to implement by shift, and xor circuits [2] and [6] . ECC is now an accepted standard, ANSI X9.42, Public Key Cryptography Systems pkcs#12, X.509 [IET98]. There is another approach to characteristic 2 operations, using optimal normal base representations (ONB) [6], also via polynomials in base 2. Such are more efficient to implement in circuitry, but the correspondence between the polynomial representation and the circuitry or xor, bit shift not so obvious as say, typically, in a CRC implementation of an irreducible polynomial: $x^3 + x^2 + 1$, GF(2^3), where, $x \varepsilon \{0,1\}$, being characteristic 2 field.

Prior to the Diffie-Hellman-Merkle* [19], [27] and [29] innovation of public key cryptography, secret channels, secret methods of secure key exchanges were practised eg: DES (Data Encryption Standards), which evolved from an earlier 1970's private key method, Lucifer, at IBM labs. [27]. (*Ralph Merkle had also conceived of the concept, of public key cryptography, independently, but did not publish).

4. Advantages of Public Key Cryptography over Private or Secret Channels

There are 3 major limitations of private key cryptosystems, such as DES (Data Encryption Standard). Menezes enumerates them [17]:

Key Distribution Problem: Two users have to select a key in secret mutually, before they can commence any communications over an insecure channel. A secret or dedicated channel, for selection of keys may not be possible

Key Management Problem: If there is a network of m users, every pair of users much share a secret key. This therefore requires, m(m-1)/2 keys, or a $O(n^2)$ problem. If the number of users is large, this becomes unmanageable.

No Signatures Possible: A digital signature is the electronic equivalent of a hand-written signature. The digital signature permits the receiver of a message to be sure that indeed the sender had sent the message.

Better scalability of the public key method: Key management becomes important with large users, as in a secret environment of n users, the number of keys requires is (n*(n-1))/2. This is $O(n^2)$ and hence as the number of users increase the number of keys rise sharply. Also there is the problem of generation and distribution of keys to the individual users, as these numbers increase. This method of key generation requires a central authority to store and generate the keys. Also keys must be generated prior to the communications protocol exchange, as opposed to the various public-key methods (below), where keys are generated mutually on demand, on-line etc. Key-escrow discusses the problems related to management of keys. e.g. how securely is a key placed, and are there designated restricted trusted management systems, whereby if key is lost, this can be recovered during the sessions. [27].

5. Fundamental Methods in Public Key Cryptography

There are 3 fundamental methods used, in public key cryptography. All such methods are computationally hard so that someone not knowing the private (decryption, or inversion key) is dealing with a computationally infeasible problem.

All existing public key cryptography systems, use one, or a mix of the following methods [2], [27] and [29]:

integer factorisation problem, as in RSA system,

discrete logarithm problem, as in Diffie-Hellman, El-Gamal systems, and

the discrete logarithm problem elliptic curve cryptography system.

The much smaller size keys, makes ECC very promising for the wireless, smaller size, smaller memory, bandwidth and power limited devices. Application to lightweight cryptography is helped via shorter keys. As indicated before, 160 bit keys in elliptic curves provide same levels of security as 1024 bit RSA. Likewise 224 bit key in elliptic curve provide same levels of security as 2048 bit key in RSA, [30] and [10].

Symmetric and asymmetric systems: There are two types of public key systems: symmetric and asymmetric. The inherent basics lie in number theory, such as the calculations of inverses using the Euler totient function, or the Extended Euclid Algorithm and/or the Chinese Remainder Theorem [12], [27], [29] (traditionally used in such number theoretic operations). RSA gives rise to asymmetric or different public and private keys, whereas Diffie-Hellman, [27], [29] gives rise to both users having identical keys.

The number theory, computation methods are such, that the various advances in mathematics and their applications, in particular algebraic number theory, algebraic geometry, elliptic curves, mutually are correlated with number theory, such as the Pythagoras numbers, and the right angled triangle).

6. Basics of Security Requirements of Public Key Algorithms

As per Schneier [27] cryptoanalysts (or intruders), have access to the public key. Hence they can always choose any message to encrypt. They can therefore, guess the value of the message, m, given, the ciphertext, C= $E_k(m)$, if some pattern, is inferred.

We discuss the basic fundamentals in the three major methods discussed in section 5.

Diffie-Hellman key exchange: The Diffie-Hellman method [27], [29], can be used for the exchange of the keys, but only in a symmetric key system. A symmetric key system is one where the sender and the receiver use the same keys, as calculated below. A group of more than 2 people can be in communication, but they must all share the same common key. The security of the channel lies in each party at the end, calculates the same key, via the group generator process, modulo arithmetic, as per number theory, below. The group generator, g, the prime number, p are public, mutually known to everyone.

The Diffie-Hellman protocol is shown below [27]. Most other cryptosystems use some variations based on the corresponding mathematical foundations:

Alice obtains a random integer x, and sends to Bob: $X = g^x \bmod (n)$

Bob obtains a random integer, y, and sends to Alice: $Y = g^y \bmod (n)$

Alice computes $k = Y^x \bmod (n)$

Bob computes $k' = X^y \bmod (n)$

Effectively both have computed $g^{xy} \bmod (n)$. The k, and k' are the same value. Anyone listening, can learn of n, g, X, Y, but cannot compute k, k', without knowing x, and y, the exponents used in the messages. The process, of obtaining x, and y, is known as the discrete logarithm problem. This is a hard process for an intruder. Alice and Bob above can calculate the discrete logarithm, using traditional number theory methods, based on Extended Euclid method, and the Chinese Remainder Theorem [27] and [29]. Schneier[27] says the Extended Euclid method is more efficient than the Euler's Totient function in calculating discrete-log. The discrete logarithm is obtained via modular exponentiation, by the concerned parties, Alice and Bob, to decrypt. This is on the same lines as multiplication for multiplicative inverse in modulo field, as earlier shown, in section 3, eg: 7*3 = 21 =1 (mod 5), or, 7^{-1} =3 mod (5). While this is not exactly elliptic curve arithmetic, the same principle, except an elliptic curve field replaces the modulo.

In Diffie-Hellman, if there are 3 participants, then mutually they communicate via 3 different keys x, y, z. Each eventually calculates the common key g^{xyz} mod (n). The modulus, n, should be prime. Also (n-1)/2 should be prime. g should be a primitive mod (n), i.e. g^{α} (mod n), for α = 0,…, n-2, should generate all the numbers in the group, G = {1,2,…, n-1}. For example in the modulo(7) finite set, 2 is primitive. $2^{\square}$ (mod 7), α= 0,…, n-2 = 5, generates all the numbers, of the group, G = {1,2,..,6}. This helps in obtaining the discrete log. n should be large, at least 512 bits, or better 1024 bits or more.

The key exchange in elliptic curves is analogous to the Diffie-Hellman key exchange protocol, but on the discrete-logarithm. Given an abelian group, G, with the known elliptic curve E, and a base point, P(x,y), user Alice sends the coordinates (x',y') of the point M = aP to Bob, where aP is a scalar multiplication of the point P a times over the elliptic curve. User Bob sends the coordinates (x", y") of the point N = bP, where nP is the scalar multiplication of point P b times over the elliptic curve. In both, each party can calculate the encrypting keys, a, and b respectively via the discrete logarithm over elliptic curves.

The Diffie-Hellman system is a symmetric public key cryptosystem. Here 2 or more parties, mutually communicating with one another, must via exchange of keys use exactly the same key.

The discrete logarithm problem (DLP) is the basis of security in the Diffie-Hellman scheme is based on the analogy of: $y = e^{x}$, $x = \log_{e}(y)$ in real and complex numbers. The most used operation is the discrete exponentiation, m^{a} (mod p), due to the useage of integers in modulo (p), or, Galois Field GF(p), where p in Diffie-Hellman systems is a prime number. The discrete logarithm problem is the inverse of the discrete exponentiation. Based on results in number theory, Euler's functions, Fermat's Little Theorem, Chinese Remainder theorem etc.., to obtain the discrete logarithm, we have to perform a discrete exponention (analogous to finding multiplicative inverse in GF(p)).

In the Diffie-Hellman system, we perform, encryption of a message m, as cipehrtext, $c = m^{a}$ (mod p).

The discrete logarithm problem or the DLP is: Given c, can we find m? This is the decryption process, which tries find the multiplicative inverse of a, i.e. $d = a^{-1}$, such that ad = 1 (mod p). Then c^{d} (mod p) = $(m^{a})^{d} = m^{ad} = m^{1}$ (mod p) = m.

Public-key systems are resistant from chosen-plaintext attacks. The security of the public key systems is based on good public-key protocols and are designed so that the different parties communicating do not decrypt arbitrary messages generated by other parties. The proof of identity protocols are a good example.[27]

Many algorithms have been designed. Some are suitable only for key exchange and not message exchange. Only a few algorithms are both secure and practical. They are based on computational complexity of hardness. Only two algorithms are suitable for both encryption and key distribution. They are RSA, and El-Gamal. Diffie-Hellman is used for key exchange and distribution, and not data encryption.[27] and [29].

All of the above, Diffie-Hellman, RSA, El-Gamal, have their elliptic curve variants. These are EC-DHP (Elliptic Curve Diffie-Hellman Problem), EC-DLP (Elliptic Curve Discrete Logarithm Problem). The DLP schema has been used in several other methods such as the El-Gamal scheme, for message exchanges as well as for signatures.

RSA: The integer factorisation process (IFP) is used in RSA, to obtain the 2 prime factors, of the modulus number, N, a public key. If p and q are two large random primes, then N= p*q. The factors, p, and q, are sufficiently dispersed, eg: they should not be twin primes, eg: (p, p+2), eg (5, 7), (11, 13), or other close (p, p+2n) combinations, eg: (7, 11), or (5, 13) which are sufficiently close so that guessing one, can easily lead to the other). Above are easy, to guess such as by first obtaining the square root of m=sqrt(N), and then scanning for primes around this region. While N is publicly known, the difficulty or security of the RSA lies in an intruder able to correctly guess the prime factors, p and q, by any intelligent or trial and error means.

The RSA system is based on a integer factorisation of a composite number, a product of 2 large primes. This is done over the ring of integers modulo n, or Galois Field GF(n). Recently AKS [2] “Primes in P” have shown a deterministic algorithm whereby testing of whether a number is prime can be done in P (polynomial) time.

Sieve of Erasthonese, is a very complex, (NP) problem. It not only gives a result whether a number is prime or not, it also for a composite number gives the prime factors. The factorisation is a very complex process, and forms the basis of RSA security systems (and in some ways the fundamental idea behind this, in Elliptic Curve Cyrptography, of Lenstra’s prime factorisation over elliptic curves [2]).

We present here some basics of the RSA prime factoring techniques:

Zn: n = p*q where p, q are primes. p, and q are important [27] to obtain ϕ(n)= (p-1)*(q-1), where ϕ is the Euler’s Totient Function, defined below.

ϕ (x), the Euler's Totient Function, counts the numbers less than x, not prime to x. For x = prime number p, this set is {1, 2, 3, …, p-1}. Hence for a prime number, p,

ϕ (p) = p - 1. For a composite number n=p*q, one can prove, the result above.

The RSA system is based, on finding 2 numbers a, b such that a*b = 1 mod (ϕ (n)) or,

a*b = 1 mod [(p-1)*(q-1)]

User A, selects a random number, a. A then finds b satisfying the above condition. The hardness of the RSA problem lies in the factorisation. n is public. If an intruder can obtain the factors, p and q, then can calculate (ϕ (n), and obtain a, and b. ϕ (n) should also not be made public, as this is the key to the hardness of the problem. n, being public, knowing ϕ (n), would make the factorisation very simple [2], [4], [22].

Thus a, b become the encryption and decryption keys, respectively for a user A. The encryption key, a, is published, placed in a public directory, so that anyone sending a message to A knows that exponent a, must be used for encryption. A user sending a message M, to user A, sends M^a mod (n), where b is a private or secret key for A, only known to A. It is possible for different authenticated users who know, n, and hence p,q, and thus ϕ(n), to calculate b from the public a. As the public and private keys, and b respectively, are different this is known as asymmetric public key cryptosystem. From another angle, each user X, has their own public key, a, and private key b. Each authenticated user can know q of n=p*q, and hence ϕ(n). This can be used in a more efficient computation procedure in decrypting the original message [27]:

a*b = 1 mod (ϕ (n)), is equivalent to: a*b = t* ϕ (n) + 1, for some t >= 1.

$(x^b)^a = x^{t*\phi(n) + 1}$ (mod (n))

The AKS algorithm [22], which showed for the first time, primality testing can be done in (P) polynomial time, with the simplest assumptions and without the earlier approaches of using the Extended Riemann Hypothesis. AKS uses a result due to Fouvry, to check for primes, closely related to the Fermat's Little Theorem (FLT). FLT is used to check for primes. However there Carmichael numbers, which are composite, and also obey FLT [22]. Fouvry's result shows, if a is co-prime to p, or gcd(a,p) =1, then p is prime if and only if [22]:

$(x – a)^p = (x^p – a)$ mod (p)

For computational efficiency (using, characteristic 2, polynomial fields),

$(x – a)^{p} = (x^{p} – a) \mod (x^{r} – 1)$

EC-DLP: The Elliptic Curve- Discrete Logarithm Problem (EC-DLP) is based on finding the scalar multiplying factor, a, given points P, and Q on the (x,y) plane, where Q = aP, in an Elliptic Curve group (G) [2],[6]. The actual process is more of a multiplicative inversion. But given this is a fairly complex operation on an Elliptic curve, given a point Q (x,y), to locate the original point P(x,y), where via repeated additions (or scalar multiplication over the elliptic curve field), Q is obtained. Given Q, how difficult is it to obtain a, and hence the original message P. Q, and G are public information. Elliptic Curves are group like structures in that they are closed. The points (representing the integer solutions), lie on the curves drawn out by the equations of the elliptic curves. The points are also representable by algebra, the arithmetic operations of the EC-DLP is done by algebra [2] and [6].

Primality testing: The AKS algorithm, “Primes in P” [22] is a deterministic algorithm that checks whether a given number is prime in O(P) time. This will not be of much help to intruders. The key issue is: Can the factorisation of a composite to two primes be done under a O(P) algorithm? Thus far this problem is kept intractable for any scheme. AKS has made a significant contribution to an ancient problem—given a number, it can now be checked in O(P) time whether this is prime or not, deterministically. Most other very heavily used algorithms, eg: Miller-Rabin are probabilistic (usually correct but to 99.9999..%, there is always the little open question in any such randomised algorithms). Miller’s original algorithm was deterministic, to check if a number is prime or not. This is based on the classic – Extended/General Riemann Hypthesis (ERH), being true. ERH, is one of the major unsolved problems in mathematics, since 1859 ref: Clay Instt’s 7 unsolved problems of the millennium, all the experimental results seem to indicate ERH is correct, but there is no formal proof [22].

Euler had shown the distribution of primes, the number of primes less than N, is $N/\log_{e}(N)$. Hadamard and Van Puissen had also independently proved this 1890’s. Van Puissen’s seems the most accurate $N/[\log_{e}(N) – 1)$. Legendre had obtained $N/[\mathrm{Log}_{e}(N)- 1.078]$ [22].

As is becoming clear, primes numbers, and factorisation of primes such as Lenstra’s factorisation over elliptic curves (1984) play a pivotal role in security of systems, whatever the underlying theory, eg: RSA, Diffie-Hellman, Elliptic curves.

7. Mathematical Operations on Elliptic Curves

Elliptic Curve theory is an extension of group theory and Galois Field Theory. Most modulo operations are done, mod (number) or a modulo (prime number). Elliptic curves are an extension to this theory. They originate from Weierstrass equations. [2].

Cryptography on elliptic curves is based on scalar multiplication of points on the elliptic curves, as the basic operation. The location of the multiplicative inverse over the elliptic curve is the challenging part (as the factorisation in RSA, discrete logarithm in Diffie-Hellman). Scalar multiplication is also known as point-multiplication. There is also a doubling operation, as shown below [2].

Given an integer k and a point P ε E($F_2{}^m$), the scalar multiplication is the process of adding P to itself k times. The result of the multiplication is denoted as k x P (or, k * P) or simply kP.

Scalar point multiplication, is given by Q = kP, where Q, P are points ε E($F_2{}^m$),

Suppose K is the public encryption key for a given user. The user using elliptic curves can perform the point multiplication, and point doubling operations. The base point, G, is fixed for each curve. The basic principles of ECC, is based on the random integer key, k acts as a private key. The result of multiplying k with the curve’s base point, G, acts as the public key, K.

Knowing G, K, and the ciphertext Q, the decryption process is to obtain the multiplicative inverse of Q, in the elliptic field E($F_2{}^m$), using the private k, to obtain the original message, P.

Modulo prime fields (mod (p)), Characteristic 2 fields ($F_2{}^m$): ECC operations involve arithmetic operations on an elliptic field, over a finite field. This is analogous to arithmetic operations over a ring of integers, or a modulo field, also known as Galois Field (GF). There are two types of finite fields. The field F_p where p is an odd prime number is called a prime field. The only even prime number case p=2, both the prime field, and the type below, degenerate to the same field. All operations in F_p are modulo (p) in this field, eg: if p=5, 4*7 mod (5) = 28 (mod 5) = 3. The division operation, n/m, such as 4/7 cannot be performed over integers. However over a prime field this is possible. eg: 4/7 mod (5) is treated as $4*7^{-1}$ (mod 5), where 7^{-1} in mod (5) field is the a number k < 5, such that 7*k = 1 mod (5). This is also known as the multiplicative inverse of a number in the prime field. We know 7*3 = 21 mod (5) = 1. Hence 7^{-1} =3 mod (5). The reason for p to be prime, is that by number theory, every number has a multiplicative inverse in the field. If the modulus is not prime this is not guaranteed. Another method of factorisation is based on the index calculus method, group generator and modulo arithmetic. [27] and [29].

Like the multiplicative inverse which is an integer, in the prime field, there is a notion of a discrete logarithm. The principle is the same, with the inverse of the operation of exponentiation. If $c = a^b$ mod (p), then b is called the discrete logarithm of c to the base a in the field (p). The discrete logarithm problem (DLP) plays a very important role in cryptography, and is the basis of Diffie-Hellman, El-Gamal systems. The following is also related to modulus being prime. For a given generator, which is a number, $g<p$, every number, n, in the field (1,2,.., p-1) can be generated as $g^a = m$, where $a < p-1$. This field is also known as Galois Field GF(p). [2] and [6]

The other type is called characteristic 2 finite field, $F_2{}^m$, where m >= 1. This is also known as $GF(2^m)$. This has very important applications, as realization of the circuits (or simulation via software) is very easy. The operations become bit-shift and xor in this type of field.

In the above field, $GF(2^m)$, there are two methods: the polynomial methods ([2] and [6]) as shown in this paper, and also the optimal normal basis representations (ONB) of Types I and II [6]. These are also based on polynomials, but are less obvious (or "insightful"), to see the correlation between the polynomials and the xor bit shift circuits, and hence may be thought of as more complex. The mathematical process, though, is much more efficient than the pure polynomial method.

While prime numbers are used as modulus, [22],[27],[29] in GF(p), irreducible polynomials GF(2m), eg: $x^3 + x + 1$ are used as the modulus. e.g.: $(x^7 + x^5 + x + 1)*(x^2 + 1)$ mod $(x^3 + x + 1)$. As the highest degree of the irreducible polynomial is, m=3, this is $GF(2^3)$. Each power of x, eg: x^n, represents the value of the nth bit, which can only be 0 or 1 (in binary fields, or 2^m fields).

Elliptic Curves over Fp: Let F_p, be a prime finite field, with p an odd prime number, and suppose a,b ε F_p, such that $4a^3 + 27b^2$ # 0 (mod p). This condition ensures there are no repeated factors in the elliptic curve polynomial. Repeated factors, or repeated roots, violate the group properties of an elliptic curve [2], [6], e.g. for a = -3, b= 2, $4a^3 + 27b^2 =0$, or the elliptic curve, $y^2 = x^3 -3x + 2 = (x-1)^2 *(x+2)$, has multiple factor (x-1).

We then define the elliptic curve, $E(F_p)$ with the parameters, a,b ε Fp, as the set of solutions, or the points P = (x,y), for x, y ε Fp, for the equation,

$$y^2 = x^3 + ax + b \bmod (p)$$

In addition there an extra point O, called the point at infinity. O is introduced to ensure the closure property, and hence all the group properties. Elliptic curves obey abelian group properties. Point O is the identify point for the group, for any point, P, P+O = P. The point O is shown on graphically by joining two points whose x co-ordinates are the same, but the y coordinates are different. P + Q = O, where P = (x, y), and Q = (x, y’), or a straight line parallel to the y axis.

The equation $y^2 = x^3 + ax + b$ mod (p) is called the defining equation for $E(F_p)$. For any solution, or point P, on the elliptic curve, P = (x_p, y_p), x_p is called the x-coordinate of P, and y_p is called the y-coordinate of P. The elliptic curves as used in cryptography, are the solutions, or points on the elliptic curve P = (x_p, y_p), where x_p, y_p are integers. This is because of the modulo (p) above, and also the application e.g. factorisation over an elliptic curve of a number. In general study of elliptic curves, x_p, y_p can be real numbers. In the most general theory of elliptic curves, and modular forms, such as the basis of Taniyama-Shimura theorems, x_p , y_p are complex numbers [4], [8].

Elliptic Curves over F_2^m : If F_2^m is a characteristic 2 finite field, and if a,b ε F_2^m , such that b # 0. Then there exists a (non-supersingular) elliptic curve $E(F_2^m)$ over F_2^m based on the parameters, a, b ε F_2^m . This elliptic curve is the locus or the set of solutions or points P = (x,y) for x,y ε F_2^m which satisfies the equation:

$$y^2 + xy = x^3 + ax^2 + b, \text{ in } F_2^m$$

along with the extra point at infinity, O. (As per [2] only the elliptic curves over F_2^m which are non-supersingular elliptic curves are of interest in cryptography).

The number of points over the elliptic field $E(F_2^m)$ is represented as # $E(F_2^m)$. The Hasse Theorem gives an estimate of the number of points [2], [28]:

$$2m + 1 – 2\text{*sqrt}(2^m) <= \# E(F_2^m) <= 2^m + 1 + 2\text{*sqrt}(2^m)$$

Addition Rules over Elliptic Curves: We may list a few of the rules obeyed by elliptic curves [2], [28]:

(a) The rule to add the point at infinity to itself: O + O = O.

(b) The rule to add any point P = (x,y) to the point at infinity (O) is:

$$(x,y) + O = O + (x,y) = (x,y), \text{ for all } (x,y)\ x, y\ \varepsilon\ E(Fp)$$

(c) The negative (inverse) of point (x,y) is –(x,y), which is given by (x, x+y)

This arises as a result of adding two points, with the same x-coordinates, when the points are either distinct, or have x-coordinate 0:

$$(x,y) + (x, x + y) = O \text{ for all } (x,y)\ \varepsilon\ E(Fp)$$

(d) The rule to add two points with different x-coordinates: If (x_1, y_1) ε $E(F_2{}^m)$, and (x_2, y_2) ε E(Fp), be two points such that $x_1 \# x_2$. Then $(x_1, y_1) + (x_2, y_2) = (x_3, y_3)$, where:

$$x_3 = \lambda^2 + \lambda + x1 + x2 + a \text{ in } (F_2{}^m), \text{ and,}$$

$$y_3 = \lambda (x1 + x3) + x3 + y1 \text{ in } (F_2{}^m), \text{ where}$$

$$\lambda = (y1 + y2)/(x1 + x2) \text{ in } (F_2{}^m)$$

(e) The rule to add a point to itself (double a point): If (x_1, y_1) ε $E(F_2{}^m)$, be a point such that $x_1 \# 0$. Then $(x_1, y_1) + (x_1, y_1) = (x_3, y_3)$, where:

$$x_3 = \lambda^2 + \lambda + a \text{ in } (F_2{}^m), \text{ and,}$$

$$y_3 = x_1{}^2 + (\lambda/x_3) \text{ in } (F_2{}^m), \text{ where}$$

$$\lambda = x_1 + y_1/x_1 \text{ in } (F_2{}^m)$$

Graphically, the doubling of a point is achieved by drawing a tangent, at the point P. Where the tangent, cuts the curve (x,y), we reflect (negate the y coordinate of the point) [6].

The set of points on $E(F_2{}^m)$ forms an abelian group under the addition rules above. Under the elliptic field $E(F_2{}^m)$, the addition rules can be very easily implemented via simple field arithmetic.

8. Possible Applications of ECC to Higher Level Protocols

Besides data encryption, there is also other higher level security protocols, such as Session Security [1]. Session Security ensures the observed transcript of the protocol, at the sender is the same as at the other end. Sequentiality of the message exchanges is proven. Finally, Safe Termination ensures; both sender and receiver are aware they both close the session gracefully [14], [27]. There is scope for study of applications of ECC to higher level security protocols (in multi-level or multi-layer security model).

9. Developments in Elliptic Curves leading to usage in Cryptography

Elliptic Curves combine different and very diverse areas of mathematics [15], [17], [19], [4]: algebraic number theory, algebraic geometry, complex analysis, representation theory. These have many applications. Those results based on number theory, have very useful applications to cryptography, especially, public key cryptography. Elliptic curves is a fairly diverse, long established branch of mathematics.

“Congruent Numbers”, whose proof was given by J. Tunnell (1983) [2] plays a very important role in the advancement of elliptic curves applications to number theoretic problems, especially cryptography. This is an ancient problem. A number, n, is a congruent number, if there exists a right angle triangle, with all three sides rational, and whose area is n. For example, the area of the right angle triangle, with sides of length, 3, 4, and 5 is 6. Hence 6 is a congruent number. 5 is the smallest congruent number (integer). This is the area of a triangle with lengths, 1.5, 6.333, 6.833.

Arjen Lenstra’s factorisation of primes, (1984), over an elliptic curve, played a very important role in the applications of elliptic curves fields to cryptography. This factorisation is computationally much more difficult than Galois Field (GF) factorisation done in RSA systems, modulo (N), where N= p*q, product of two primes, p and q. This followed considerable research of Lenstra’s supervisor, Carl Pomerance’s works on Elliptic Curves [2]

Van Lint (1988) had shown algebraic geometry of elliptic curves, give rise to a new source of error-correcting codes [4]. These performed better than most others. Error correcting codes, eg: Goppa codes, have been used in various cryptosystems, such as McEliece system [27] and [29].

10. HyperElliptic Curves

These are generalization of Elliptic curves [4]. Elliptic curves are a very special case of hyperelliptic curves, of genus 1. Outstanding contribution in this area has come from many in particular, Prof Gerhard Frey, (originator of Frey elliptic curve) [8], [17].

It has been observed that genus 4, 5 or higher curves are not considered very favourable. They give better results than Elliptic Curve (genus 1), in terms of much shorter keys, but the security cannot be guaranteed. In general the higher the genus (hyper-elliptic curve), the shorter is the key length required. It has been observed that up to genus 3 works well with hyper elliptic curves in cryptography. Above genus 3, the security is lower than elliptic curves (genus 1). Aldeman (of RSA) has shown sub-exponential time attacks.

What is genus? Genus, g= 0 applies to conic sections. There is no twisting or torsion or turning in a circle, ellipse, parabola or hyperbola, or in general geometrical figures like triangles or rectangles. Genus, g=1 includes elliptic curves. We can see one torsion in the graphical representation of these equations. g = 2 and above are many different types of curves, hyperelliptic curves being one such group.

In general genus is a term from algebraic geometry. It indicates how many twists or torsions, are there in the curve. For example for any conic section, such as the circle, the (actual, geometrical) ellipse, parabola, hyperbola, there are no twists, hence genus is 0. Likewise, the same say for other planar figures such as a triangle.

Mishra et al [18] have done considerable work in elliptic curve and hyperelliptic curve cryptography. Mishra has used base-3 arithmetic to overcome the complexity of the arithmetic in hyper-elliptic curves. Lange [16], Seppala [26] have considerable research in hyperelliptic curves. Seppala[26] has re-visted Myrberg’s 1920’s numerical uniformization.

11. Applications of Smaller Key Sizes in Elliptic Curve

Smaller key sizes have advantages on mobile, wireless, and PDA devices. It has been observed via experimentation that an ECC 160 bit key systems provides the same level of security as a 1024 bit key RSA system. An ECC 224 bit key system provides the same level of security as a 2048 bit key RSA system [30].

Its variants such as Elliptic Curve discrete logarithm problem (EC-DLP), EC-DSA (digital signature algorithm) are used. In Wireless LANs, 802.11b WEP (Wireless Encryption Protocol) is found to be somewhat insecure. Smaller and held wireless devices, mobile cellphones, RFID tags, have small memory and limited computation power, and bandwidth spectrums available for their operations [36].

Hyperelliptic Curves of genus, g>2 (g=1 are elliptic curves), require even smaller key sizes to provide the same levels of security. The complexity of the algorithm and the arithmetic circuit realization is much more complex. As stated earlier, genus, g=2 or g=3, are at most used in higher systems. Above genus g=3, the security is found to degrade.

12. Applications of Elliptic Curve Cryptography

General Applications: Elliptic Curve Cryptography is not only used in small wireless systems, but for example Sun's Java servers versions 7.0 upwards, use ECC [9] and [30]. Netscape's SSL- ensures the connection between client- and server is secure, while S-HTTP (Secure Hyper Text Transport Protocol), using “https” instead of “http”, ensures each message is transmitted securely. Hence SSL and S-HTTP complement one another. Both technologies have been approved by IETF as standards [13].

Lightweight Cryptography: Lightweight cryptography refers to cryptography applied to limited space, memory size, bandwidth, and power requirements. Mobile cellphones, PDA's, palmtops, smart cards, RFID tags are some examples of such devices. There is considerable challenge in fine-tuning the mathematics of cryptography, and applications via software and hardware to such devices.

It is possible for the more complex embedded systems, lightweight computation power, bandwidth, memory, devices such as mobile other wireless devices, to consider using ECC, compared with say GSM's current A3, A5, A8 algorithms.

Sun have developed a coin sized ECC based web browser “sizzle”.[10] and [21]. This uses 128kbytes flash memory. There has been considerable literature, in this growing area of lightweight cryptography.

It may be predicted in future the technology, speeds, algorithmic implementations (such as base 3 vs base 2 arithmetic [18] would make ECC on RFID feasible.

ECC in RFID: In RFID, the computational time complexity is still not feasible [36]. Wolkerstorfer,[35], [36] has developed the first ECC (Elliptic Curve Cryptography) arithmetic unit on a RFID tag. He has noted the 0.35 micro-milimetre CMOS technology takes up too much space on a RFID tag. He has designed the use of nanometre CMOS technology, which fits within the tiny space restrictions of RFID tags. Currently the processing speeds are slow, about 1 tag per second [36] .Wolkenstorfer has also analysed the feasibility, various hardware, software, logisitics issues, including limited physical space on RFID devices, with respect to ECC useage very well [35]. A recent design of ECC processor in RFID tags is reported in Battina et al [3].

Ranasinghe (following [23]) has suggested multiple usage of circuitry, such as the CRC logic could also be used for encryption at a different time to optimise the limited space on the RFID tags, in order to accommodate ECC and other non-lightweight compute methods on RFID.

Pipelined Processors in Elliptic Curve Systems: Mishra et al [18] have used pipelining, to the scalar multiplication arithmetic operations. These arithmetic operations are the most computationally dominant operations in Elliptic curve cryptography. The scalar multiplication as shown in Section 3 consists of a series of point addition and point doubling operations. The pipelining scheme is based on the key observation that to start the subsequent operation one need not wait till the current one exits. The next operation can begin while a part of the current operation is still being processed. This simple observation coupled with slightly more hardware support provides a significant speed-up in the implementation of ECC. While this is significant to general ECC, this is not yet realizable in lightweight cryptography, and in particular, in RFID system. The aim of the authors is to be able to implement ECC in smart cards.

Base-3 arithmetic used in Hyper-Elliptic Curve Cryptography: Mishra et al [18] have also been working on Hyper-Elliptic curves and their applications to cryptography. One of their designs uses a base 3 arithmetic, in preference to base 2. The exponentiation process may be seen as a series of multiplication operations. Likewise each multiplication may be seen as a series of point additions and point doublings in the elliptic curve and hyper elliptic curve fields. With the pipelining methods as above, and partial adders, the authors have been successful in arithmetic using hyper elliptic curves. Weil pairings, Tate pairings, Miller’s algorithm, Montgomery multiplication [17], and various advanced mathematical techniques have been used for fast arithmetic logic and hardware design, in elliptic and hyperelliptic curves. A Tate pairing based system, using base 3 (or characteristic 3) implementation, with methods to achieve fast computation of the Tate pairing is reported in [9] . The authors report that the Tate pairing is more efficient for computation. The Weil and Tate pairings have been used to various construct cryptosystems such as identity based key exchange and signature schemes. The Weil pairing (named after Andre Weil [8]) was first introduced by Menezes, Okamoto, and Vanstone [17] to attack the discrete logarithm problem on certain elliptic curves. The Tate pairing was introduced to cryptography by Frey and Ruck in their extension of the work of Menezes, Okamoto, and Vanstone [9].

13. Conclusions

Elliptic Curve Cryptography offers a promising approach in the areas of public-key or dedicated areas of cryptography, due to the hardness, or lack of sub-exponential time attack on the discrete-log problem in elliptic fields. Amongst the general class of hyper elliptic curves, they seem to be the practical choice, as higher genus (g>3) are shown to be less secure than genus 2 or 3 curves. The latter have more complex computations hence circuits. Elliptic curves are based on very sound mathematical foundations, of centuries, and distribute the potential ciphertext values fairly randomly to make any guess or attack difficult. They have all the abelian group properties as have other well known methods. The much shorter key sizes make them suitable for lightweight computing, bandwidth, power devices as mobiles, laptops, mobile web browsers etc. There have been some developments of ECC arithmetic units on RFID tags. In future perhaps some refining of algorithms towards lightweight computations on RFID or better technology may make ECC more readily feasible for RFID. There is a lot of potential of ECC in general cryptography, as well as the area of lightweight cryptography.

Acknowledgements

The authors are grateful to Mr Damith Ranasinghe, Mr, Manfred Janstcher, Dr Behnam Jamali, Mr. Alfio Grasso, Ms. Mun Leng Ng, and Mr. Kin Seong Leong for discussions of fundamental concepts, and their kind insights in this fairly complex conceptual area. ECC being a fascinating growing area, the discussions helped deeper understanding, as elliptic curves (perhaps commenced with Fermat proving his theorem for case, n=4), to have played pivotal roles in the proof of 350 year general Fermat's Last Theorem [4], [8]. Also, the valuable inputs to ECC applications to banking, the shorter keys permit faster processing, of cheques ("check-21"), digital postage stamps, signatures, current and future applications, including lightweight cryptosystems. These are based on the latest EC-signature schemes, variants of the complete complex, ECDSA , to useage of ECPVS (Elliptic Curve Pintsov Vanstone Signatures), which has partial message recovery, ECQV (Elliptic Curve Qu

Vanstone), which allows implicit certificate type, and ECNR (Elliptic Curve Nyberg Rueppel), method which allows full message recovery [37].

References:

[1] Avoine, Gildas (2006): “Bibliography on Security and Privacy in RFID Systems”, MIT; Cambridge, Massachusetts; Retrieved May, 17, 2006 from http://lasecwww.epfl.ch/~gavoine/rfid/

[2] Balasubramanian, R. (2003): “Elliptic Curves and Cryptography”, in Bhandari A.K., Nagraj D.S., Ramakrishnan, B., Venkataraman T.N., (editors), Elliptic Curves, Modular Forms, and Cryptography, Hindustan Book Agency, New Delhi, 2003. ISBN 81-85931-42-9, pp 325-345.

[3] Batina L., Guajardo J., Kerins T., Mentens N., Tuyls P., and Verbauwhede I (2006)., “An Elliptic Curve Processor Suitable For RFID-Tags”, Cryptology ePrint Archive: Report 2006/227 (4th July 2006).

http://eprint.iacr.org/2006/227.pdf#search=%22Is%20Elliptic%20Curve%20Suitable%20to%20RFID%22

[4] Bhandari, A.K, Nagraj, D.S., Ramakrishnan, B., Venkataraman, T.N., (editors) (2003): Elliptic Curves, Modular Forms, and Cryptography, Hindustan Book Agency, New Delhi, 2003. ISBN 81-85931-42-9

[5] Chang, S., Eberle, H., Gupta, V., and Gura, N. (2004): “Elliptic Curve Cryptography – How it Works”; Sun Microsystems Laboratories; 2004; Retrieved May, 9, 2006 from http://research.sun.com/projects/crypto/

[6] Certicom (2006): “ECC Tutorial” http://www.certicom.com/index.php?action=ecc,ecc_tutorial

[7] [Fermat01] http://mathworld.wolfram.com/FermatEllipticCurveTheorem.html

[8] [Fermat02] http://mathworld.wolfram.com/FermatsLastTheorem.html

[9] Galbraith, S.D., Harrison S., Soldera D., (2002): Implementing the Tate pairing, Lecture Notes In Computer Science; Vol. 2369 Proceedings of the 5th International Symposium on Algorithmic Number Theory, 2002, pp 324 – 337, 2002, Springer. ISBN:3-540-43863.

http://www.hpl.hp.com/techreports/2002/HPL-2002-23.html

[10] Gura N., Shantz S., Eberle H., et al (2002): "An End-to End Systems Approach to Elliptic Curve Cryptography", Sun Microsystems Laboratories; 2002; Retrieved May, 10, 2006 from http://research.sun.com/projects/crypto

[11] Gupta V., Wurm M., Zhu Y., Millard M., Fung S., Gura N., Eberle H., Sheuel Chang Shantz S. (2002): "Sizzle: A standards-based end-to-end security architecture for the embedded Internet", Elsevier, Pervasive and Mobile Computing vol 1., 2005, pp 425-445.

[12] Goldreich, Oded (2001): Foundations of Cryptography, Cambridge University Press, Cambridge 2001, ISBN 0-521-79172-3

[13] IET (1998): http://www.ietf.org/rfc/rfc2412.txt - X.509 standards showing various modulo groups, Elliptic Curve primes and parameter fields

[14] Jantscher, Manfred (2006): Use of Random and Varying Codes in Object Identification, Auto-ID Labs, School of Electrical and Electronics Engineering, University of Adelaide, Adelaide, Australia.

[15] Koblitz, Neal (1993): Introduction to Elliptic Curves and Modular Forms, 2nd ed., Springer-Verlag, GTM-97, Berlin, 1993 ISBN: 0-387-97966-2

[16] Lange, T. (2006): "Advanced Topics in Cryptology", Department of Mathematics, Technical University of Denmark,

http://www.hyperelliptic.org/tanja/teaching/AdvCrypto/AdvCrypto06.html

[17] Menezes, Alfred J. (1993): Elliptic Curve Public Key Cryptosystems, Kluwer Academic Publishers, Boston, 1993. ISBN: 0-7923-9368-6.

[18] Mishra, P. (2005): "Pipelined Computation of Scalar Multiplication in Elliptic Curve

Cryptosystems”

http://www.scs.carleton.ca/~kranakis/IT/IT-05-abstracts/mishra.txt

[19] Mollin, Richard (2005), Introduction to Cryptography, Chapman & Hall/CRC, Boca Raton, 2000, ISBN” 1-58488-127-5.

[20] Morgan, T. P. (2006), “Sun Creates World's Smallest SSL Web Server”

http://www.computerwire.com/industries/research/?pid=C55355B9-B6CD-42EC-80BC-ACFDA6F2CDD3

[21] Murphy, T. (2006), Course 373-Finite Fields, University of Dublin, Trinity College School of Mathematics

http://www.maths.tcd.ie/pub/Maths/Courseware/FiniteFields/FiniteFields.pdf

[22] Nongkynrih, A. (2003), “Primality and Factoring”, in in Bhandari A.K., Nagraj D.S., Ramakrishnan, B., Venkataraman T.N., (editors), Elliptic Curves, Modular Forms, and Cryptography, Hindustan Book Agency, New Delhi, 2003. ISBN 81-85931-42-9, pp 303-323.

[23] Ranasinghe, D., Engels, D. and Cole, P. (2005), “Low-Cost RFID Systems: Confronting Security and Privacy”, White Paper Series; AutoID Labs; Edition 1; 2005.

[24] [RSA01] http://www.rsasecurity.com/rsalabs/node.asp?id=2187 - P, NP

[25] [RSA02] http://www.rsasecurity.com/rsalabs/node.asp?id=2306 - ANSI X9 key mgt

[26] Seppala, M. (2004), “Myrberg’s Numerical Uniformization of Hyperelliptic Curves”, Annales, Academia Scienatiaum Fennicae, Mathematica, Vol 29 (2004), pp 3-20.

http://www.math.fsu.edu/~seppala/papers/UniformizationForSubmission/NumericalUniformization.pdf - wrt hyperelliptic curves

[27] Schneier, Bruce (1994), Applied Cryptography: Protocols, Algorithms and Source Code in C, John-Wiley and Sons, New York, 1994. ISBN: 0-471-5975602

[28] SEC 1 Elliptic Curve Cryptography (2000), in Standards for Efficient Cryptography, Certicom, 2000.
http://www.secg.org/collateral/sec1_final.pdf#search=%22SEC%20Elliptic%20Curves%20Hasse%20%22

[29] Stinson, Douglas R. (1995), Cryptography: Theory and Practice, CRC Press, Boca Raton, Florida, 1995, ISBN: 0-8493-8521-0.

[30] [SUN06] http://research.sun.com/projects/crypto/ECC-Whitepaper.pdf

[31] Welsh, Dominic (1990), Codes and Cryptography, Oxford University Press, Oxford. 1990, ISBN 0-19-853287-3.

[32] [WIK01] http://en.wikipedia.org/wiki/Elliptic_function

[33] [WIK02] http://en.wikipedia.org/wiki/Elliptic_integral

[34] [WIK03] http://en.wikipedia.org/wiki/Elliptic_curves

[35] Wolkenstorfer, Johannes (2003), Hardware Aspects of Elliptic Curve Cryptography, Abstract of PhD Thesis, IAIK Graz, 2003.
http://www.iaik.tugraz.at/research/publications/theses/wolkerstorfer.htm

[36] Wolkerstorfer, J. (2005) , Is Elliptic-Curve Cryptography Suitable to Secure RFID Tags?, in Handout of the Ecrypt Workshop on RFID and Lightweight Crypto, Graz (Austria), July 2005.

http://www.iaik.tugraz.at/research/krypto/events/RFID-SlidesandProceedings/Wolkerstorfer-ECC%20and%20RFID.pdf

[37] ECC-DS Certicom, (2006), An Introduction to the Uses of ECC-based Certificates, Code and Cipher, Vol. 2, no. 2, Certicom's Bulletin of Security and Cryptography.

http://www.certicom.com/index.php?action=res,cc&issue=2-2&&article=1