Global Journal of Computer Science and Technology: E Network, Web & Security
Volume 15 Issue 5 Version 1.0 Year 2015
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

The Security of Elliptic Curve Cryptosystems - A Survey

By Koffka Khan
University of the West Indies, Trinidad and Tobago, W.I, India

Abstract- Elliptic curve cryptography or ECC is a public-key cryptosystem. This paper introduces ECC and describes its present applications. A mathematical background is given initially. Then its' major cryptographic uses are given. These include its' use in encryption, key sharing and digital signatures. The security of these ECC-based cryptosystems are discussed. It was found that ECC was well suited for low-power and resource constrained devices because of its' small key size.

Keywords: *elliptic curve cryptography; public-key; cryptosystem; security; rsa; el gamal; curve; key size.*

GJCST-E Classification : *C.2.0 D.4.6*

Strictly as per the compliance and regulations of:

The Security of Elliptic Curve Cryptosystems - A Survey

Koffka Khan

Abstract- **Elliptic curve cryptography or ECC is a public-key cryptosystem. This paper introduces ECC and describes its present applications. A mathematical background is given initially. Then its' major cryptographic uses are given. These include its' use in encryption, key sharing and digital signatures. The security of these ECC-based cryptosystems are discussed. It was found that ECC was well suited for low-power and resource constrained devices because of its' small key size.**

Index Terms: elliptic curve cryptography; public-key; cryptosystem; security; rsa; el gamal; curve; key size.

I. Introduction

Over the years, with the increase in processing power of computers, there has been a reduction in the work factor required to solve Integer Factorization (IFP) [17], [21], [3] and Discrete Logarithm (DLP) problems [6], [9], [18]. As a result, key sizes grew to more than 1000-bits so as to attain a reasonable level of security. However, in constrained environments carrying out thousand-bit operations is impractical. Therefore, a matter of growing importance in cryptography is the need for algorithms with low resource requirements [24], [14] that can be deployed on resource-constrained ubiquitous devices. This explains why other public-key methods would be welcomed, Elliptic Curve Cryptosystem (ECC) [12] being a probable candidate.

Elliptic curves are the basis for a relatively new class of public-key schemes. It is predicted that Elliptic Curve Cryptosystems (ECC) Elliptic curves were proposed for use as the basis for discrete logarithm-based cryptosystems in 1985, independently by Victor Miller and Neal Koblitz. Elliptic curve are not ellipse, but cubic curves. Properties of ECC made it stronger against various attacks in wireless sensor networks [7], RFID [8], smart card [20] and many others. It will replace many existing schemes in the near future. However, the complicated mathematical background of ECC results in more sophisticated algorithms. Mathematical basis for security of elliptic curve cryptography is computational intractability of elliptic curve discrete logarithm problem (ECDLP) [11].

Elliptic Curve Cryptography (ECC) can be applied to data encryption and decryption, digital signatures, and key exchange procedures. Every user has a public and private key. The public key is used for encryption or signature verification, while the private key is used for decryption or signature generation. ECC is used as an extension to current cryptosystems, for example, ECC Diffie-Hellman Key Exchange (EC-DH) [16], ECC Digital Signature Algorithm (ECDSA) [10] Elliptic Curve Integrated Encryption Scheme (ECIES) [23].

A motivation is given in Section II. In Section III a mathematical background is given. The major uses of ECC in present day cryptosystems are presented in Section III. The underlying theory of elliptic curve cryptosystems is discussed in section IV. Three ECC cryptosystems are given in section V. These are EC-DH, ECDSA and ECIES. The security of these cryptosystems are outlined in Section V with the advantages of using ECC. Finally the conclusion is given in section

II. Motivation

In order to understand the principle of asymmetric cryptography, the basic symmetric encryption scheme has to be recalled.

Figure 1 : Symmetric key encryption

Two properties are essential for symmetric key cryptosystems:

i. The same secret key is used for encryption and decryption.
ii. The encryption and decryption function are very similar (in the case of DES [5] they are essentially identical).

There is a simple real-world analogy for symmetric cryptography. Assume there is a safe with a strong lock. Only Alice and Bob have a copy of the key for the lock. The action of encrypting of a message can be viewed as Alice putting the message in the safe. In order to read, i.e., decrypt, the message, Bob uses his key and opens the safe.

However, there are several shortcomings associated with symmetric-key crypto-schemes.

Author: Department of Computing and Information Technology, The University of the West Indies, Trinidad and Tobago, W.I.
e-mail: koffka_@hotmail.com

- Key Distribution Problem. The key must be established between Alice and Bob using a secure channel. The communication link for the message is not secure, so sending the key over the channel directly can't be done.
- Number of Keys Even. Each user has to potentially deal with a very large number of keys. If each pair of users' needs a separate pair of keys in a network with n users, there are (n· (n−1)) / 2 key pairs. Thus, each user has to store n 1 keys securely. The number of keys that must be generated and transported via secure channels will become exorbitant.
- No Protection against cheating by Alice or Bob. Alice and Bob have the same capabilities, since they possess the same key. As a consequence, symmetric cryptography cannot be used for applications where we would like to prevent cheating by either Alice or Bob.

In order to overcome these drawbacks, Diffie, Hellman and Merkle made the following proposal. It is not necessary that the key possessed by the person who *encrypts* the message (that's Alice in our example) is secret. The crucial part is that Bob, the receiver, can only *decrypt* using a secret key. In order to realize such a system, Bob publishes a public encryption key which is known to everyone. Bob also has a matching secret key, which is used for decryption. Thus, Bob's key k consists of two parts, a public part, k_{pub}, and a private one, k_{pr} .

This systems works quite similarly to the good old mailbox system. Everyone can put a letter in the box, i.e., encrypt, but only a person with a private (secret) key can retrieve letters, i.e., decrypt (see Figure 1).

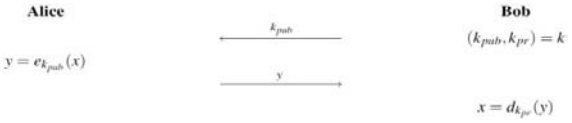

Figure 2 : Basic protocol for public-key encryption

By looking at that protocol the exchange of an encrypted key still remains a problem. This can be done by *encrypting a symmetric key*, e.g., an AES key, using the public-key algorithm. Once the symmetric key has been decrypted by Bob, both parties can use it to encrypt and decrypt messages using symmetric ciphers. But this still poses a grave problem for the public key sharing at the start of the protocol can be intercepted by Oscar. It is these security concerns that resulted in the need for the development of asymmetric cryptosystems.

Public key schemes are all built from one common principle, the one-way function.

Definition 1

A function f (x) is a one-way function if:

- y = f (x) is computationally easy, and
- x = f^{-1}(y) is computationally infeasible.

A function is easy to compute if it can be evaluated in polynomial time, i.e., its running time is a polynomial expression. In order to be useful in practical crypto schemes, the computation $y = f(x)$ should be sufficiently fast that it does not lead to unacceptably slow execution times in an application. The inverse computation $x = f^{-1}(y)$ should be so computationally intensive that it is not feasible to evaluate it in any reasonable time period, say, thousands of years, when using the best known algorithm.

Recently the key sizes of public key cryptosystems, for example, RSA prohibits their use in low-power, resource constrained computing devices. Due to this requirement ECC shows an advantage as much smaller key sizes (see Table 1) are needed for the same amount of security.

Table 1 Key sizes of ECC and RSA []

ECC(in bits)	RSA(in bits)
106	512
112	768
132	1024
160	2048
210	3072
283	7680
409	15360
571	21000

III. Mathemitical Background

In Section A modular arithmetic is described. Then, in section B integer rings is defined. Further, in section C finite fields is illustrated. In section D cyclic rings is explained. Section E portrays the concept of subgroups. In Section F the Discrete Logarithm in Prime Fields is depicted. Finally, in section G the Generalized Discrete Logarithm Problem is given.

a) Modular Arithmetic

Symmetric and asymetric ciphers are usually based on arithmetic with a finite number of elements. The sets of real and natural numbers are infinite. Consider a finite set of integers. The octal set of integer numerals are: {0, 1, 2, 3, 4, 5, 6, 7}. It is possible to do arithmetic in this set so long as≤0result ≤ 7. For instance: 2 x 2 = 4 or 3 + 4 = 7 is fine, but 7 + 5 gives 12. This result is not a subset of the octal set. To validate this operation an additional operator is used.

This is the modulus operation and is defined as follows:

Definition 2

Let p, r, q ∈ Z (where Z is a set of all integers) and q > 0. We write p ≡ r mod q, if q divides p− r. q is called the modulus and r is called the remainder.

Thus 7 + 5 = 12, which when divided by 8 (12/8) gives a remainder of 4. So 7 + 5 = 4 mod 8. In practice the integers involved have a length of 130–4096 bits so that efficient modular computations are a crucial aspect in modern cryptography.

b) Integer Rings

Consider the set of integers from zero to m-1 with two operators: addition and multiplication. A ring on this set is defined as follows:

Definition 3

A ring is the set of integers $Z_m = \{0, 1, 2, ..., m-1\}$ with the "+" and "×" operations $\forall\, e, f, g, h \in Z_m : e + f \equiv g \bmod m \wedge e \times f \equiv h \bmod m$

The following properties of rings are important:

- Closed: addition and multiplication of two numbers has a result in the ring.
- Ring operations are associative: $a + (b + c) = (a + b) + c$, and $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ for all a, b, c ∈ Z_m.
- A neutral element 0 with respect to addition, i.e., for every element a ∈ Z_m it holds that $a + 0 \equiv a \bmod m$.
- The additive inverse always exists for any element a in the ring, there is always the negative element –a such that $a + (-a) \equiv 0 \bmod m$.
- The neutral element 1 with respect to multiplication, i.e., for every element a ∈ Z_m it holds that $a \times 1 \equiv a \bmod m$.
- The multiplicative inverse exists only for some, but not for all, elements. Let

a ∈ Z, the inverse a^{-1} is defined such that $a \cdot a^{-1} \equiv 1 \bmod m$. If an inverse exists for a, we can divide by this element since $b/a \equiv b \cdot a^{-1} \bmod m$. Finding the inverse is difficult, usually employing the Euclidean algorithm []. An easier method is as follows. An element a ∈ Z has a multiplicative inverse a^{-1} if and only if GCD (a, m) = 1, where GCD is the greatest common divisor. If this holds, then a and m are relatively prime or coprime.

- The distributive law is followed: $a \times (b + c) = (a \times b) + (a \times c)$ for all a, b, c ∈ Z_m. Thus, the ring Z_m is the set of integers {0, 1, 2, ... , m– 1} in which we can add, subtract, multiply, and sometimes divide.

c) Finite Fields

The concept of a simpler algebraic structure, a group is illustrated.

Definition 4

A group is a set of elements G together with an operation ◦ which combines two elements of G. A group is set with one operation and the corresponding inverse operation. If the operation is called addition, the inverse operation is subtraction; if the operation is multiplication, the inverse operation is division (or multiplication with the inverse element).

A group has the following properties:

- The group operation ◦ is closed. That is, for all a, b, ∈ G, it holds that a ◦ b = c ∈ G.
- The group operation is associative. That is, a ◦ (b ◦ c) = (a ◦ b) ◦ c for all a, b, c ∈ G.
- There is an element 1 ∈ G, called the neutral element (or identity element), such that a ◦ 1 = 1 ◦ a = a for all a ∈ G.
- For each a ∈ G there exists an element a^{-1} ∈ G, called the inverse of a, such that $a \circ a^{-1} = a^{-1} \circ a = 1$.
- A group G is abelian (or commutative) if, furthermore, a ◦ b = b ◦ a for all a, b ∈ G.

Cryptography uses both multiplicative groups, i.e., the multiplication, and additive groups. Consider the set of integers $Z_m = \{0, 1, ... , m-1\}$ and the operation addition modulo m. Every element a has an inverse–a such that $a + (-a) = 0 \bmod m$. However, this set does not form a group with the multiplication operation because most elements do not have an inverse where a $a^{-1} = 1 \bmod m$.

Theorem 1

The set Z_n^* which consists of all integers a = 0, 1, ... , n– 1 for which GCD (a, n)= 1 forms an abelian group under multiplication modulo n. The identity element is e = 1. In Table 1 n = 9, so Z_n^* consists of the elements {1, 2, 4, 5, 7, 8}.

Table 1 : Multiplication table for Z_9^*

× mod 9	1	2	4	5	7	8
1	1	2	4	5	7	8
2	2	4	8	1	5	7
4	4	8	7	2	1	5
5	5	1	2	7	8	4
7	7	5	1	8	4	2
8	8	7	5	4	2	1

The following properties are satisfied:

- Closure: integers which are elements of Z_9^* are used.
- Group identity and inverses: each row and column is a permutation of the elements of Z_9^*.
- Commutativity: symmetry along the main diagonal.
- Associativity: Multiplication in Z_9^*.

In order to have all four basic arithmetic operations (i.e., addition, subtraction, multiplication, division) in one structure, a set which contains an additive and a multiplicative group is needed. This is called a field. A *finite field*, sometimes also called *Galois field*, is a set with a finite number of elements.

Definition 5

A field F is a set of elements with the following properties:

- All elements of F form an additive group with the group operation "+" and the neutral element 0.
- All elements of F except 0 form a multiplicative group with the group operation "×" and the neutral element 1.
- When the two group operations are mixed, the distributive law holds, i.e., for all a, b, c ∈ F: a(b + c)= (ab)+ (ac).

The set R of real numbers is a field with the neutral element 0 for the additive group and the neutral element 1 for the multiplicative group. Thus every real number a has an additive inverse, namely –a, and every nonzero element *a* has a multiplicative inverse 1/*a*. Also note that the number of elements in the field is called the

order or *cardinality* of the field. The following theorem explains the characteristic of a finite field:

Theorem 2

A field with order r only exists if r is a prime power, i.e., $r = c^n$, for some positive integer n and prime integer c. c is called the characteristic of the finite field.

This theorem implies that there are, for instance, finite fields with 243 elements (since $243 = 3^5$) or with 1024 elements (since $1024 = 2^{10}$, and 2 is a prime). However, there is no finite field with 24 elements since $24 = 2^3 \cdot 3$. Hence 24 is thus not a prime power.

The most native examples of finite fields are fields of prime order, i.e., fields with $n = 1$. Elements of the field $GF(c)$ can be represented by integers $0, 1, \ldots, c-1$. The two operations of the field are modular integer addition and integer multiplication modulo c.

Theorem 3

Let c be a prime. The integer ring Z_c^* is denoted as GF(c) and is referred to as a prime field, or as a Galois field with a prime number of elements. All nonzero elements of GF(c) have an inverse. Arithmetic in GF(c) is done modulo c.

This means that the integer ring Z_m^* with modular addition and multiplication, and m happens to be a prime, Z_m^* is not only a ring but also a finite field. In order to do arithmetic in a prime field, the rules for integer rings hold: Addition and multiplication are done modulo c, the additive inverse of any element a is given by $a + (-a) = 0 \bmod c$, and the multiplicative inverse of any nonzero element a is defined as $a \cdot a^{-1} = 1$.

d) Cyclic Groups

Definition of a finite group:

Definition 6

A group (G, ◦) is finite if it has a finite number of elements. We denote the cardinality or order of the group G by $|G|$.

The following are some examples of finite groups:

- $(Z_n^*, +)$: the cardinality of Z_n^* is $|Z_n^*| = n$ since $Z_n^* = \{0, 1, 2, \ldots, n-1\}$.
- $(Z_n^*, \cdot)$: remember that Z_n^* is defined as the set of positive integers smaller than n which are relatively prime to n. Thus, the cardinality of Z_n^* equals Euler's phi function [] evaluated for n, i.e., $|Z_n^*| = \Phi(n)$. For instance, the group Z_9^* has a cardinality of $\Phi(9) = 32 - 31 = 6$. Thus the group consists of the six elements {1, 2, 4, 5, 7, 8}.

Cyclic groups are the basis for discrete logarithm-based cryptosystems. The order of an element is defined as follows:

Definition 7

The order ord(b) of an element b of a group (G, ◦) is the smallest positive integer n such that: $b^n = b \circ b \circ \ldots \circ b = 1$, occurs n times and 1 is the identity element of G.

By using Definition 6, determine the order of $b = 4$ in the group Z_7^*. To do this, compute the powers of b until we obtain the identity element 1.

$$b^1 = b^1 \cdot b^0 = 4 \cdot 1 = 4 \equiv 4 \bmod 7$$
$$b^2 = b^1 \cdot b^1 = 4 \cdot 4 = 16 \equiv 2 \bmod 7$$
$$b^3 = b^2 \cdot b^1 = 2 \cdot 4 = 8 \equiv 1 \bmod 7$$

Shown from the last line: ord(4) = 3. Keep multiplying the result by *b*:

$$b^4 = b^3 \cdot b^1 = 1 \cdot 4 = 4 \equiv 4 \bmod 7$$
$$b^5 = b^4 \cdot b^1 = 4 \cdot 4 = 16 \equiv 2 \bmod 7$$
$$b^6 = b^3 \cdot b^3 = 1 \cdot 1 = 1 \equiv 1 \bmod 7$$

$$b^7 = b^3 \cdot b^4 = 1 \cdot 4 = 4 \equiv 4 \bmod 7$$
$$b^8 = b^3 \cdot b^5 = 1 \cdot 2 = 2 \equiv 2 \bmod 7$$
$$b^9 = b^3 \cdot b^6 = 1 \cdot 1 = 1 \equiv 1 \bmod 7$$

$$b^{10} = b^3 \cdot b^7 = 1 \cdot 4 = 4 \equiv 4 \bmod 7$$
$$b^{11} = b^3 \cdot b^8 = 1 \cdot 2 = 2 \equiv 2 \bmod 7$$
$$b^{12} = b^3 \cdot b^9 = 1 \cdot 1 = 1 \equiv 1 \bmod 7$$

The powers of b run through the sequence {1, 4, 2} indefinitely. This implies that $b = 4$ is a primitive element and $|Z_7^*|$ is cyclic. It follows that ord(b)= 4 = $|Z_7^*|$. The group Z_7^* has the element 4 as a generator.

This cyclic behavior gives rise to following definition:

Definition 8

A group G which contains an element c with maximum order ord(c) = $|G|$ is said to be cyclic. Elements with maximum order are called primitive elements or generators.

An element c of a group *G* with maximum order is called a generator since every element *b* of *G* can be written as a power $c^n = b$ of this element for some *n*, i.e., c *generates* the entire group.

The theorem below states that the multiplicative group of every prime field is cyclic. Thus these groups are the most useful for building discrete logarithm (DL) cryptosystems.

Theorem 4

For every prime p, $(Z_p^*, \cdot)$ is an abelian finite cyclic group.

Theorem 5 first shows Fermat's Little Theorem for all cyclic groups. Secondly it shows that only element orders which divide the group cardinality exist in a cyclic group.

Theorem 5

Let G be a finite group. Then for every $a \in G$ it holds that:

- $a|G| = 1$
- ord(a) divides $|G|$

e) Subgroups

Subgroups are subsets of cyclic groups which are groups themselves.

Theorem 6

Let (G, ◦) be a cyclic group. Then every element b ∈ G with ord(s) = t is the primitive element of a cyclic subgroup with t elements.

Consider a subgroup of G $= Z_{11}^*$. Now ord(3) = 5, and the powers of 3 generate the subset J = {1, 3, 4, 5, 9}. To verify whether this set is actually a group its multiplication table has to be explored:

Table 1 : Multiplication table for the subgroup J = {1, 3, 4, 5, 9}

× mod 11	**1**	**3**	**4**	**5**	**9**
1	1	3	4	5	9
3	3	9	1	4	5
4	4	1	5	9	3
5	5	4	9	3	1
9	9	5	3	1	4

J is a subgroup of Z_{11}^**:*

- J is closed under multiplication modulo 11 since the table only consists of integers which are elements of J.
- The group operation is obviously associative and commutative since it follows regular multiplication rules.
- The neutral element is 1.
- For every element b ∈ J there exists an inverse b−1 ∈ J which is also an element of J. Every row and every column of the table contain the identity element.
- J is a subgroup of prime order 5.
- The elements 3, 4, 5 and 9 are generators of J.
- Each element b ∈ G of a group G generates some subgroup J.

Subgroups of prime order are of enormous interest in cryptography. The following theorem follows.

Theorem 7

Let J be a subgroup of G. Then |J| divides |G|. Thus the cyclic group Z_{11}^* has cardinality $|Z_{11}^*| = 10 = 1 \cdot 2 \cdot 5$.

Thus, it follows that the subgroups of Z_{11}^* have cardinalities 1, 2, 5 and 10 since these are all possible divisors of 10. All subgroups J of Z_{11}^* and their generators g are given below.

Subgroup	Elements	Primitive Elements
H_1	{1}	g = 1
H_2	{1, 10}	g = 10
H_3	{1, 3, 4, 5, 9}	g = 3, 4, 5, 9

The following theorem gives us immediately a construction method for a subgroup from a given finite cyclic group. The only thing we need is a primitive element and the group cardinality c. One can now simple compute $g^{c/n}$ and obtains a generator of the subgroup with n elements.

Theorem 8

Let G be a finite cyclic group of order c and let g be a generator of G. Then for every integer n that divides c there exists exactly one cyclic subgroup J of G of order n. This subgroup is generated by $g^{c/n}$. J consists exactly of the elements b ∈ G which satisfy the condition $b^n = 1$. There are no other subgroups.

Consider the cyclic group Z_{11}^*. Now g = 8 is a primitive element in the group. To get a generator g for the subgroup of order 2 compute: $q = g^{c/n} = 8^{10/2} = 8^5 = 32768 \equiv 10 \bmod 11$. The element 10 generates the subgroup with two elements:

$$q^1 = 10,$$
$$q^2 = 100 \equiv 1 \bmod 11,$$
$$q^3 \equiv 10 \bmod 11 \ldots$$

f) The Discrete Logarithm in Prime Fields

The discrete logarithm problem (DLP), can directly be explained using cyclic groups. Two important areas are the DLP over Prime fields and the generalized DLP problem. Consider the DLP over Z_p^*, where p is a prime.

Definition 9

Given is the finite cyclic group Z_{11}^* of order p − 1 and a primitive element g ∈ Z_{11}^* and another element q ∈ Z_{11}^*. The DLP is the problem of determining the integer $1 \le x \le p - 1$ such that: $g^x \equiv q \bmod p$.

Such an integer x must exist since g is a primitive element and each group element can be expressed as a power of any primitive element. This integer x is called the discrete logarithm of q to the base g, and we can formally write: $x = \log_g q \bmod p$. Computing discrete logarithms modulo a prime is a very hard problem if the parameters are sufficiently large. Since exponentiation $g^x \equiv q \bmod p$ is computationally easy, this forms a one-way function.

Consider the group Z_{47}^* which has order 46. The subgroups in Z_{11}^* have thus a cardinality of 23, 2 and 1. Now g = 2 is an element in the subgroup with 23 elements, and since 23 is a prime, g = 2 is a primitive element in the subgroup. A possible discrete logarithm problem is given for q = 36 (which is also in the subgroup): Find the positive integer x, $1 \le x \le 23$, such that $2^x \equiv 36 \bmod 47$. By using a brute-force attack, a solution is x = 17.

g) The Generalized Discrete Logarithm Problem

The generalized discrete logarithm problem (GDLP) is used in cryptography and is not restricted to the multiplicative group Z_p^*, p prime, but can be defined over any cyclic groups.

Definition 10

Given is a finite cyclic group G with the group operation ◦ and cardinality k. We consider a primitive

element g ∈ G and another element q ∈ G. The discrete logarithm problem is finding the integer n, where 1≤ n ≤ k, such that: $q = g \circ g \circ \ldots \circ g = g^n$, n times.

Such an integer n must exist since g is a primitive element as in the case of the DLP in Z_p^*. Thus each element of the group G can be generated by repeated application of the group operation on g. Consider the additive group of integers modulo a prime. For instance, choose the prime p = 11, G = (Z_{11}^*, +) is a finite cyclic group with the primitive element g = 2. Here is how g generates the group:

We try now to solve the DLP for the element q = 3, i.e., we have to compute the integer 1≤ n ≤ 11 such that: $n \cdot 2 = 2 + 2 + \ldots + 2$ (n times) ≡ 3 mod 11. Even though the group operation is addition, we can express the relationship between g, q and the discrete logarithm n in terms of multiplication: n· 2 ≡ 3 mod 11. In order to solve for n, invert the primitive element g: n ≡ 2^{-1} 3 mod 11. Using, e.g., the extended Euclidean algorithm, compute $2^{-1} \equiv 6$ mod 11 to get the discrete logarithm: n ≡ 2^{-1} 3 ≡ 7 mod 11.

The DLP can be solved easily here as there are mathematical operations which are not in the additive group. They are multiplication and inversion. However, often it was found that the underlying DL problem is not difficult enough.

IV. Elliptic Curve Theory

a) Basic Properties

ECC is based on the generalized discrete logarithm problem. A cyclic group where the DL problem is computationally hard is required. This means that it must have good one-way properties. Polynomials functions with sums of exponents of x and y can be chosen. For example, the polynomial equation $a \cdot x^2 + b \cdot y^2 = c$ over the real numbers turns out to be an ellipse.

An elliptic curve is a special type of polynomial equation. In ECC the curve is not over the real numbers but over a finite field. The most popular choice is prime fields GF(p), where all arithmetic is performed modulo a prime p. The curve is nonsingular so that it has no self-intersections or vertices, and is achieved if the discriminant of the curve $-16*(4a^3 + 27b^2)$ is nonzero.

Definition 11

The elliptic curve over Z_p^*, p > 3, is the set of all pairs (x, y) ∈ Z_p^* which fulfill $y^2 \equiv x^3 + a \cdot x + b$ mod p together with an imaginary point of infinity O, where a, b ∈ Z_p^* and the condition $4 \cdot a^3 + 27 \cdot b^2 \neq 0$ mod p.

b) Group Operations on Elliptic Curves

"Addition" means that given two points and their coordinates, say A = (x_1, y_1) and B = (x_2, y_2), we have to compute the coordinates of a third point C such that: $A + B = C$ *or* $(x1, y1) + (x2, y2) = (x3, y3)$. Two cases are considered:

- the addition of two distinct points (point addition)
- the addition of one point to itself (point doubling)

Point Addition P + Q : This is the case where we compute R = P + Q and P ≠ Q. The construction works as follows: A line through P and Q intersects a third point between the elliptic curve and the line. Mirror this third intersection point along the x-axis. This mirrored point is, by definition, the point R. Figure 1 shows the point addition on an elliptic curve over the real numbers.

i	1	2	3	4	5	6	7	8	9	10	11
ig	2	4	6	8	10	1	3	5	7	9	0

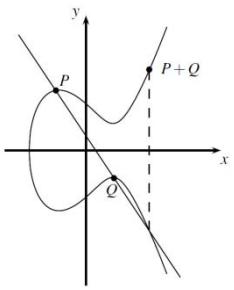

Figure 1 : Point addition on an elliptic curve over the real numbers

Point Doubling P + Q : This is the case where we compute P + Q but P = Q. Hence, R = P + P = 2P. First draw the tangent line through P and obtain a second point of intersection between this line and the elliptic curve. Then mirror the second point of intersection along the x-axis. This mirrored point is the result R of the doubling as shown in Figure 2.

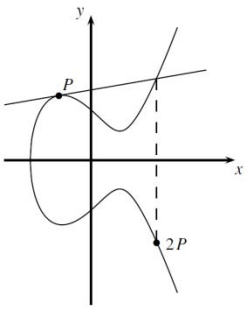

Figure 2 : Point doubling on an elliptic curve over the real numbers

With these operations the points on the elliptic curve fulfill the group conditions: closure, associativity, existence of an identity element and existence of an inverse. Consider the add, subtract, multiply and divide operations over prime fields GF(p) rather than over the real numbers. The following analytical expressions become relevant. The elliptic curve point addition and doubling formulae are shown:

if P ≠ Q (point addition), $s = \frac{y_2 - y_1}{x_2 - x_1} \ mod \ p$

if P = Q (point doubling), $s = \frac{3x_1^2 + a}{2y_1} \ mod \ p$

then

$x_3 = s^2 - x_1 - x_2 \ \text{mod} \ p$

$y_3 = s*(x_1 - x_3) - y_1 \bmod p$

The parameter s is the slope of the line through P and Q in the case of point addition, or the slope of the tangent through P in the case of point doubling. An identity (or neutral) element O such that: $P + O = P$ is compulsory. An abstract point at infinity is used as the neutral element O. This point at infinity is located towards "plus" infinity along the y-axis or towards "minus" infinity along the y-axis. Hence, the inverse $-P$ of any group element P is: $P + (-P) = O$.

- Finding the inverse of a point $P = (x_p, y_p)$ is the negative of its y coordinate. In the case of elliptic curves over a prime field GF(p) as $-y_p \equiv p - y_p \bmod p$, hence $-P = (x_p, p - y_p)$. An example for the group operation is now given. Consider a curve over the small field Z_{29}^*, $E : y^2 \equiv x^3 + 2x + 2 \bmod 17$. To double the point $A = (3, 1)$:

- $2P = P + P = (3, 1) + (3, 1) = (x_3, y_3)$.
- Now $s = (2 \cdot 1) - 1 * (3 \cdot 32 + 2) = 2 - 1 \cdot 29 \equiv 9 \cdot 12 = 63 \equiv 6 \bmod 1$.
- Also $x3 = s2 - x1 - x2 = 62 - 3 - 3 = 30 \equiv 13 \bmod 17$.
- And $y3 = s(x1 - x3) - y1 = 6 * (3 - 13) - 1 = -61 \equiv 7 \bmod 17$.
- Thus, $2P = (3, 1) + (3, 1) = (13, 7)$.

Inserting the coordinates into the curve equation: $y^2 \equiv x^3 + 2 \cdot x + 2 \bmod 17 = 7^2 \equiv 13^3 + 2 \cdot 13 + 2 \bmod 17$. So $15 = 2225 \equiv 15 \bmod 17$ which proves that the point is actually on the curve.

c) Building a Discrete Logarithm Problem with Elliptic Curves

Setting up the discrete logarithm problem is now discussed.

Definition 12

Given an elliptic curve E, consider a primitive element P and another element R. The DL problem is finding the integer d, where $1 \leq d \leq \#E$, such that: $P + P + \cdots + P = d * P = U$. P is repeated d times. In cryptosystems, d is the private key which is an integer, while the public key U is a point on the curve with coordinates $U = (x_u, y_u)$.

The operation in Definition 12 is called point multiplication. Thus, formally $U = d * P$. Note d*P is a notation for this repeated group operation. If a multiplicative notation is chosen, the ECDLP would have had the form $P^d = U$, which would have been more consistent with the conventional DL problem in Z_{29}^*.

Given a starting point P for the ECDLP elliptic curves over the real numbers, the computation becomes 2P, 3P, .. ., d*P = U . This is effectively hopping back and forth on the elliptic curve. The starting point P (a public parameter) and the final point U (the public key) is put in the public domain. To break the cryptosystem, an attacker has to figure out how often we "jumped" on the elliptic curve. Thus, the number of hops is the secret d, the private key.

V. Elliptic Curve Cryptosystems

a) Elliptic Curve Diffie–Hellman

As with the conventional Diffie–Hellman key exchange (DHKE) [] a key exchange using elliptic curves can be realized. This elliptic curve Diffie–Hellman key exchange (ECDH) requires agreed upon domain parameters on an elliptic curve and a primitive element on this curve:

- Choose a prime p and the elliptic curve: $E : y^2 \equiv x^3 + a \cdot x + b \bmod p$
- Choose a primitive element $P = (x_P, y_P)$. The prime p, the curve given by its coefficients a, b, and the primitive element P are the domain parameters.

The actual key exchange is the same as for the conventional Diffie–Hellman protocol. Alice and Bob choose the private keys a and b, respectively, which are two large integers. With the private keys both generate their respective public keys A and B, which are points on the curve. The public keys are computed by point multiplication. The two parties exchange these public parameters with each other. The joint secret T_{AB} is then computed by both Alice and Bob by performing a second point multiplication involving the public key they received and their own secret parameter. The joint secret T_{AB} can be used to derive a session key, e.g., as input for the AES algorithm []. Note that the two coordinates (x_{AB}, y_{AB}) are not independent of each other: Given x_{AB}, the other coordinate can be computed by simply inserting the x value in the elliptic curve equation.

Thus, only one of the two coordinates should be used for the derivation of a session key. EC-DH Key Exchange is now shown.

Alice	**Bob**
choose k_{prA} $= a \in \{2, 3, ..., \#E - 1\}$	choose k_{prB} $= b \in \{2, 3, ..., \#E - 1\}$
compute k_{pubA} $= a*P = A = (x_A, y_A)$	compute k_{pubB} $= b*P = B = (x_B, y_B)$
	$A = (x_A, y_A)$ →
	← $B = (x_B, y_B)$
compute $a_B = T_{AB}$	compute $b_A = T_{AB}$

Joint secret between Alice and Bob: $T_{AB} = (x_{AB}, y_{AB})$.

Proof. Alice computes aB = a (b P) while Bob computes bA = b (a P). Since point addition is associative, both parties compute the same result, namely the point $T_{AB} = ab P$.

Let's look at an example with small numbers.

We consider the ECDH with the following domain parameters. The elliptic curve is $y^2 \equiv x^3 + 2x + 2 \bmod 17$, which forms a cyclic group of order $\#E = 19$. The base point is $P = (5, 1)$. The protocol proceeds as follows:

Alice	**Bob**
choose $k_{prA} = a = 3$	choose $k_{prB} = b = 10$
compute k_{pubA} $= a^* P$ $= 3P$ $= (10, 6) = A$	compute k_{pubB} $= b^* P$ $= 10P$ $= (7, 11) = B$
$A = (10, 6)$ →	
← $B = (7, 11)$	
compute a^*B $= T_{AB}$ $= 3(7, 11)$ $= (13, 10)$	compute b^*A $= T_{AB}$ $= 10(10, 6)$ $= (13, 10)$

Joint secret between Alice and Bob: $T_{AB} = (13, 10)$.

b) The Elliptic Curve Digital Signature Algorithm (ECDSA)

The ECDSA standard is defined for elliptic curves over prime fields Z_p and Galois fields $GF(2^m)$. The former is often preferred in practice, and is used in what follows. The keys for the ECDSA are computed as follows:

i. *Key Generation for ECDSA*

Use an elliptic curve E with modulus p, coefficients a and b and a point A which generates a cyclic group of prime order q. Then choose a random integer d with $0 < d < q$. Finally compute $B = d A$. The keys are now: $k_{pub} = (p, a, b, q, A, B)$ and $k_{pr} = (d)$.

Note that we have set up a discrete logarithm problem where the integer d is the private key and the result of the scalar multiplication, point B, is the public key. Similar to DSA, the cyclic group has an order q which should have a size of at least 160 bit or more for higher security levels.

ii. *Signature and Verification*

The ECDSA signature consists of a pair of integers (r, s). Each value has the same bit length as q, which makes for fairly compact signatures. Using the public and private key, the signature for a message x is computed as follows.

iii. *ECDSA Signature Generation*

- Choose an integer as random ephemeral key k_E with $0 < k_E < q$.
- Compute $R = k_E A$.
- Let $r = x_R$.

Compute $s \equiv (h(x) + d \cdot r)\, k_E^{-1} \bmod q$

In step 3 the x-coordinate of the point R is assigned to the variable r. The message x has to be hashed using the function h in order to compute s. The hash function output length must be at least as long as q. The hash function compresses x and computes a fingerprint which can be viewed as a representative of x. The signature verification process is as follows.

iv. *ECDSA Signature Verification*

- Compute auxiliary value $w \equiv s^{-1} \bmod q$.
- Compute auxiliary value $u_1 \equiv w \cdot h(x) \bmod q$.
- Compute auxiliary value $u_2 \equiv w \cdot r \bmod q$.
- Compute $P = u_1 A + u_2 B$.

The verification $ver_{k_{pub}}(x, (r, s))$ follows from: $x_P \equiv r \bmod q \Rightarrow$ valid signature and $x_P \not\equiv r \bmod q \Rightarrow$ invalid signature.

In the last step, the notation x_P indicates the x-coordinate of the point P. The verifier accepts a signature (r, s) only if the x_P has the same value as the signature parameter r modulo q. Otherwise, the signature should be considered invalid.

Proof. We show that a signature (r, s) satisfies the verification condition $r \equiv x_P \bmod q$.

We'll start with the signature parameter s.

$s \equiv (h(x) + d\, r)\, k_E^{-1} \bmod q$

$= k_E \equiv s^{-1} h(x) + d\, s^{-1} r \bmod q$

Use the auxiliary values u_1 and u_2:

$= k_E \equiv u_1 + d\, u_2 \bmod q$

Multiply both sides of the equation with A as the point A generates a cyclic group of order q:

$= k_E A = (u_1 + d\, u_2) A$

Group operation is associative:

$= k_E A = u_1 A + d\, u_2 A$

Group operation is associative:

$= k_E A = u_1 A + u_2 B$

Thus the expression $u_1 A + u_2 B$ is equal to $k_E A$ if the correct signature and key (and message) have been used. But this is exactly the condition that we check in the verification process by comparing the x-coordinates of $P = u_1 A + u_2 B$ and $R = k_E A$.

Bob wants to send a message to Alice that is to be signed with the ECDSA algorithm. The signature and verification process is as follows. The elliptic curve E: $y^2 \equiv x^3 + 2x + 2 \bmod 17$. All points of the curve form a cyclic group of order 19, i.e., a prime, there are no subgroups and hence in this case $q = \#E = 19$.

Alice	**Bob**
	choose E with p = 17, a =2, b = 2, and A = (5, 1). with q = 19, choose d = 7. Compute B = d A = 7 (5, 1) = (0, 6)
← (p, a, b, q, A, B)	
(17, 2, 2, 19, (5, 1), (0, 6)) →	
sign: compute hash of message h(x) = 26 choose ephemeral key k*E = 10	

$R = 10\,(5, 1) = (7, 11)$
$r = x*R = 7$
$s = (26 + 7 \,.\, 7) \,.\, 2 \equiv 17 \bmod 1$

$(x, (r, s)) = (x, (7, 17))$ ←

verify:
$w = 17^{-1} \equiv 9 \bmod 19$
$u_1 = 9 \cdot 26 \equiv 6 \bmod 19$
$u_2 = 9 \cdot 7 \equiv 6 \bmod 19$
$P = 6 \cdot (5, 1) + 6 \cdot (0, 6) = (7, 11)$

$x_P \equiv r \bmod 19 \Rightarrow$ valid signature

c) Elliptic Curve Integrated Encryption Scheme (ECIES)

Elliptic curve cryptography can be used to encrypt plaintext messages, M, into ciphertexts. The elliptic group $E_p(a, b)$ and the generator point G are made public. Each user select a private key, $n_A < n$ and compute the public key P_A as: $P_A = n_A*G$. To encrypt the message point P_M for Bob (B), Alice (A) choses a random integer k and compute the ciphertext pair of points P_C using Bob's public key P_B:

$$P_C = [(k*G),(P_M + k*P_B)]$$

After receiving the ciphertext pair of points, P_C, Bob multiplies thefirst point, (k*G) with his private key, n_B, and then adds the result to the second point in the ciphertext pair of points, $(P_M + k*P_B)$:

$$(P_M + k*P_B) - [n_B(k*G)] = (P_M + k*n_BG) - [n_B(k*G)] = P_M$$

which is the plaintext point, corresponding to the plaintext message M. Only Bob, knowing the private key n_B, can remove $n_B(k*G)$ from the second point of the ciphertext pair of point, i.e. $(P_M + k*P_B)$, and hence retrieve the plaintext information P_M.

Consider the following elliptic curve: $y^2 = x^3 - x + 188 \bmod 751$ that is: $a = -1$, $b = 188$, and $p = 751$. The elliptic curve group generated by the above elliptic curve is $E_p(a,b) = E_{751}(-1,188)$. Let the generator point $G = (0,376)$. Then the multiples k*G of the generator point G are (for $1 \le k \le 751$):

$G = (0,376)$ $2G = (1,376)$ $3G = (750,375)$ $4G = (2,373)$ $5G = (188,657)$ $6G = (6,390)$ $7G = (667,571)$ $8G = (121,39)$ $9G = (582,736)$ $10G = (57,332)$ ... $761G = (565,312)$ $762G = (328,569)$ $763G = (677,185)$ $764G = (196,681)$ $765G = (417,320)$ $766G = (3,370)$ $767G = (1,377)$ $768G = (0,375)$ $769G = O$ (point at infinity)

If Alice wants to send to Bob the message M which is encoded as the plaintext point $P_M = (443,253) \in E_{751}(-1,188)$. She must use Bob public key to encrypt it. Suppose that Bob secret key is $n_B = 85$, then his public key will be: $P_B = n_B*G = 85(0,376) = (671,558)$. Alice selects a random number $k = 113$ and uses Bob's public key $P_B = (671,558)$ to encrypt the message point into the ciphertext pair of points:

$P_C = [(k*G),(P_M + k*P_B)]$
$= [113 \times (0,376),(443,253) + 113 \times (671,558)]$
$= [(34,633),(443,253) + (47,416)]$
$= [(34,633),(217,606)]$

Upon receiving the ciphertext pair of points, $P_C = [(34,633), (217,606)]$, Bob uses his private key, $n_B = 85$, to compute the plaintext point, P_M, as follows.

$(P_M + k*P_B) - [n_B(k*G)] = (217,606) - [85(34,633)]$
$= (217,606) - [(47,416)]$
$= (217,606) + [(47,-416)]$ (since $-P = (x_1, -y_1)$)
$= (217,606) + [(47,335)]$ (since $-416 \equiv 335 \pmod{751}$)
$= (443,253)$

and then maps the plaintext point $P_M = (443,253)$ back into the original plaintext message M.

VI. Security of ecc Cryptosystems

a) Security of EC-DH

Elliptic curves are used as the ECDLP has very good one-way characteristics. E, p, P, A, and B is available for an attacker who wants to break the ECDH. The attacker desires to compute the joint secret between Alice and Bob $T_{AB} = a * b * P$. This is known as the elliptic curve Diffie–Hellman problem (ECDHP). Presently, there seems to be only one way to compute T_{AB}, that is, to solve either $a = \log_P A$, or $b = \log_P B$. Each of which are discrete logarithm problems.

For carefully chosen elliptic curve the best known attacks against the ECDLP are considerably weaker than the best algorithms for solving the DL problem modulo p, and the best factoring algorithms which are used for RSA attacks. In particular, the index-calculus algorithms [22], which are powerful attacks against the DLP modulo p, are not applicable against elliptic curves. For carefully selected elliptic curves, the only remaining attacks are generic DL algorithms, that is, Shanks' baby-step giant-step method [19] and Pollard's rho method [1].

As the number of steps required for such an attack is approximately equal to the square root of the group cardinality, a group order of at least 2^{160} should be used. An attack with a group consisting of generic algorithms, will require about 2^{80} steps. Thus, a security level of 80 bits provide moderate security. Thus, in practice elliptic curve bit lengths of up to 256 bits are commonly used. This will provide security levels of up to 128 bits.

b) Security of ECDSA

Elliptic curves have several advantages over RSA and over DL schemes like Elgamal or DSA. In particular, the absence of strong attacks against elliptic curve cryptosystems (ECC), bit lengths in the range of 160–256 bit can be chosen which provide security equivalent to 1024–3072-bit RSA and DL schemes. The shorter bit length of ECC often results in shorter processing time and in shorter signatures.

Given that the elliptic curve parameters are chosen correctly, the main analytical attack against ECDSA attempts to solve the elliptic curve discrete logarithm problem. If an attacker were capable of doing this, he could compute the private key d and/or the ephemeral key. However, the best known ECC attacks have a complexity proportional to the square root of the size of the group in which the DL problem is defined, i.e., proportional to √q.

The security level of the hash function must also match that of the discrete logarithm problem. The cryptographic strength of a hash function is mainly determined by the length of its output. The security levels of 128, 192 and 256 were chosen so that they match the security offered by AES with its three respective key sizes. More subtle attacks against ECDSA are also possible. For instance, at the beginning of verification it must be checked whether r, s ∈ {1, 2,..., q}. Also, protocol-based weaknesses, e.g., reusing the ephemeral key, must be prevented.

c) Security of ECIES

The cryptographic strength of elliptic curve encryption lies in the difficulty for a cryptanalyst to determine the secret random number k from k*P and P itself. The fastest method to solve this problem (known as the elliptic curve logarithm problem) is the Pollard ρ factorization method [].

The computational complexity for breaking the elliptic curve cryptosystem, using the Pollard ρ method, is 3.8×1010 MIPS-years (i.e. millions of instructions per second times the required number of years) for an elliptic curve key size of only 150 bits []. Finally increasing the elliptic curve key length to only 234 bits will impose a computational complexity of 1.6 × 1028 MIPS-years (still with the Pollard ρ method).

VII. Conclusion

Public-key encryption can be used to eliminate problems involved with conventional encryption. However, it has not managed to be as widely accepted as conventional encryption because it introduces a lot of overheads. Therefore, it is very important to find ways to reduce the overheads yet not sacrificing on other aspects of security so that the desirability in public-key can be exploited.

ECC have been described, which is a promising candidate for the next generation public-key cryptosystem. Although ECC's security has not been completely evaluated, it is expected to come into widespread use in various fields in the future.

ECC has been shown to have many advantages due to its ability to provide the same level of security as other public key cryptosystems, yet using shorter keys. However, its disadvantage which may even hide its attractiveness is its lack of maturity, as mathematicians believed that enough research has not yet been done in ECDLP. Finally, the future of ECC looks brighter than that of other public key cryptosystems as today's applications (smart cards, pagers, and cellular telephones etc) cannot afford the associated overheads.

References Références Referencias

1. Bach, E. (1991). Toward a theory of Pollard's rho method. Information and Computation, 90(2), 139-155.
2. Banavar, G., & Bernstein, A. (2002). Software infrastructure and design challenges for ubiquitous computing applications. *Communications of the ACM*, *45*(12), 92-96.
3. Brent, R. P. (2010). Some integer factorization algorithms using elliptic curves. *arXiv preprint arXiv:1004.3366*.
4. Davis, V. M., Cutino, S. C., Berg, M. J., Conklin, F. S., & Pringle, S. J. (2001). U.S. Patent No. 6,282,522. Washington, DC: U.S. Patent and Trademark Office.
5. Denning D. E. R., Denning P. J. Internet besieged: Countering cyberspace scofflaws. ACM Press, 1998.
6. ElGamal, T. (1985, January). A public key cryptosystem and a signature scheme based on discrete logarithms. In Advances in Cryptology (pp. 10-18). Springer Berlin Heidelberg.
7. Ferri, R., Kim, M., & Yee, E. (2004). U.S. Patent Application 10/856,684.
8. Finkenzeller, K. (1999). RFID handbook: radio-frequency identification fundamentals and applications (pp. 151-158). New York: Wiley.
9. Gordon, D. (2011). Discrete logarithm problem. In Encyclopedia of Cryptography and Security (pp. 352-353). Springer US.
10. Johnson, D., Menezes, A., & Vanstone, S. (2001). The elliptic curve digital signature algorithm (ECDSA). International Journal of Information Security, 1(1), 36-63.
11. Kanayama, N., Kobayashi, T., Saito, T., & Uchiyama, S. (2000). Remarks on elliptic curve discrete logarithm problems. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, *83*(1), 17-23.
12. Koblitz, N. (1987). Elliptic curve cryptosystems. Mathematics of computation, 48(177), 203-209.
13. Lahiri, S. (2005). RFID sourcebook. IBM press.
14. McLoone, M., & Robshaw, M. J. (2006). Public key cryptography and RFID tags. In Topics in Cryptology–CT-RSA 2007 (pp. 372-384). Springer Berlin Heidelberg.
15. Messer, A., Greenberg, I., Bernadat, P., Milojicic, D., Chen, D., Giuli, T. J., & Gu, X. (2002). Towards a distributed platform for resource-constrained devices. In Distributed Computing Systems, 2002.

Proceedings. 22nd International Conference on (pp. 43-51). IEEE.

16. Miller, V. S. (1986, January). Use of elliptic curves in cryptography. In Advances in Cryptology—CRYPTO'85 Proceedings (pp. 417-426). Springer Berlin Heidelberg.
17. Montgomery, P. L. (1994). A survey of modern integer factorization algorithms. CWI quarterly, 7(4), 337-366.
18. Peralta, R. (1986, January). Simultaneous security of bits in the discrete log. In *Advances in Cryptology—Eurocrypt'85* (pp. 62-72). Springer Berlin Heidelberg.
19. Pollard, J. M. (2000). Kangaroos, monopoly and discrete logarithms. Journal of cryptology, 13(4), 437-447.
20. Rankl, W., & Effing, W. (2010). *Smart card handbook*. John Wiley & Sons.
21. Shoup, V. (1995). A new polynomial factorization algorithm and its implementation. Journal of Symbolic Computation, 20(4), 363-397.
22. Silverman, J. H., & Suzuki, J. (1998, January). Elliptic curve discrete logarithms and the index calculus. In Advances in Cryptology—ASIACRYPT'98 (pp. 110-125). Springer Berlin Heidelberg.
23. Smart, N. P. (2001). The exact security of ECIES in the generic group model. In Cryptography and Coding (pp. 73-84). Springer Berlin Heidelberg.
24. Zhao, W., Ramamritham, K., & Stankovic, J. A. (1987). Scheduling tasks with resource requirements in hard real-time systems. *Software Engineering, IEEE Transactions on*, (5), 564-577.