ISSN:0975-9646

Elliptic Curve Diffie-Hellman Key Exchange Algorithm for Securing Hypertext Information on Wide Area Network

Ram Ratan Ahirwal, Manoj Ahke
Samrat Ashok Technological Institute
Vidisha (M. P.) 464001 India

Abstract— **The Elliptic curve cryptography (ECC) an emerging favorite because it requires less computational power, communication bandwidth, and memory when compared to other cryptosystems. In this paper we present Elliptic curve cryptography and Diffie–Hellman key agreement protocol, itself is an anonymous (non-authenticated) key-agreement protocol, it provides the basis for a variety of authenticated protocols, and is used to provide forward secrecy for web browsers application using HTTPS. In its popular deployment on the internet, HTTPS provides authentication of the web site and associated web server that one is communicating with, which protects against Man-in-the-middle attacks. Additionally, it provides bidirectional encryption of communications between a client and server, which protects against eavesdropping and tampering with and/or forging the contents of the communication.**

Keywords— **Elliptic curve, diffie Hellman, attacks, SSL, HTTPS.**

I. INTRODUCTION

Elliptic Curve Cryptography (ECC) was first proposed by victor Miller and independently by Neal Koblitz in the mid-1980s and has evolved into a mature public-key cryptosystem. Compared to its traditional counterparts, ECC offers the same level of security using much smaller keys.
This result in faster computations and savings in memory, power and bandwidth those are especially important in constrained environments. More significantly, the advantage of ECC over its competitors increases, as the security needs increase over time. Recently the National Institute of standards and Technology (NIST) approved ECC for use by the U.S government. Several standards organizations, such as Institute of Electrical & Electronics Engineers (IEEE), American National
Standards Institute (ANSI), Open Mobile Alliance (OMA) and Internet Engineering Task Force (IETF), have ongoing efforts to include ECC as a required or recommended security mechanism. Here we present our new algorithm using Diffie_hellman key exchange algorithm providing forward secrecy for web browsers application.

II. RELATED WORK

Standards for elliptic curve systems are currently being drafted by various accredited standards bodies around the world; some of this work is summarized below.
1. The Elliptic Curve Digital Signature Algorithm (ECDSA) was adopted in January 1999 as an official American National Standards Institute (ANSI) standard. The ANSI X9 (Financial Services) working group is also drafting a standard for elliptic curve key agreement and transport protocols.
2. Elliptic curves are in the draft IEEE P1363 standard (Standard Specifications for PublicKey Cryptography), which includes encryption, signature, and key agreement mechanisms. Elliptic curves over Fp and over F2m are both supported. For the
3. The OAKLEY Key Determination Protocol of the Internet Engineering Task Force (IETF) describes a key agreement protocol that is a variant of Diffie–Hellman. It allows for a variety of groups to be used, including elliptic curves over Fp and F2m . The document makes specific mention of elliptic curve groups over the fields F2 155 and F2 210 .
A draft is available from the web site http://www.ietf.cnri.reston.va.us/.
4. ECDSA is specified in the draft document ISO/IEC 14888: Digital signature with appendix – Part 3: Certificate-based mechanisms.
5. The ISO/IEC 15946 draft standard specifies various cryptographic techinques based on elliptic curves including signature schemes, public-key encyrption schemes, and key establishment protocols.
6. The ATM Forum Technical Committee's Phase I ATM Security Specification draft document aims to provide security mechanisms for Asynchronous Transfer Mode (ATM) networks. Security services provided include confidentiality, authentication, data integrity, and access control. A variety of systems are supported, including RSA, DSA, and elliptic curve systems.
As these drafts become officially adopted by the appropriate standards bodies, one can expect elliptic curve systems to be widely used by providers of information security An elliptic curve is the set of solutions of an equation of the form can be shown as below:

$$y^2 + axy + by = x^3 + cx^2 + dx + e \quad (1)$$

Where a, b, c, d, and e, are real numbers.
A special addition operation is defined over elliptic curves and this with the inclusion of a point $\mathcal{O}$,called point at infinity.If three points are on a line intersecting an elliptic curve, then their sumis equal to this point at infinity $\mathcal{O}$,which acts as the identity element for this addition operation.
Sometimes the general equation (1) can be referred as Weierstrass equation as shown in (2)

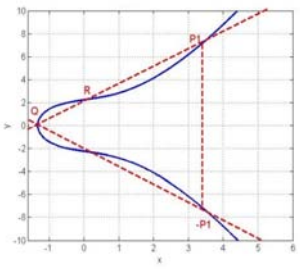

Fig.1: Elliptic curves

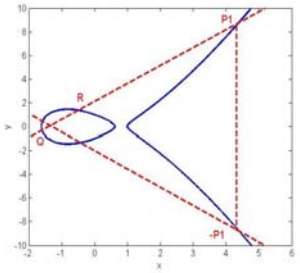

Fig.2: Elliptic curve

If we wanted use a elliptic curve to be used for cryptography the necessary condition is the curve is not singular, i.e. the discriminate of polynomial:

$$f(x) = x^3 + ax + b :$$

$$4a^3 + 27b^2 \neq 0 \qquad (3)$$

Figures 1 and 2 show the two elliptic curves are

$$y^2 = x^3 + 2x + 5 \qquad (4) \text{ and}$$

$$y^2 = x^3 - 2x + 1 \qquad (5)$$

We can see those two equations meet

An elliptic curve is the set of solutions of an equation of the form can be shown as below:

$$y^2 + axy + by = x^3 + cx^2 + dx + e \qquad (1)$$

Where a, b, c, d, and e, are real numbers.

An elliptic group over the Galois Field $E_p(a,b)$ is obtained by computing $x^3 + ax + b \bmod p$ for $0 \leq x < p$. The constants a and b are non negative integers smaller than the prime number p and as here we used "mod p", so equation (3) should be read as:

$4a^3+27b^2 \bmod p \neq 0$

For each value of x one needs to determine whether or not it is a quadratic residue. If it is the case, then there are two values in the elliptic group. If not, then the point is not in the elliptic $E_p(a,b)$ group. When we fixed a prime number, p and then we can have the Galois Field $E_p(a, b)$ group via the fixed constants a *and* b following the above conditions characteristic two finite fields, polynomial bases and normal bases of F_2m over an arbitrary subfield F2l are supported. P1363 also includes discrete log systems in subgroups of the multiplicative group of the integers modulo a prime, as well as RSA encryption and signatures.

The latest drafts are available from the web site http://stdsbbs.ieee.org/.

For example, let the points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ be in the elliptic group $E_p(a,b)$ group and $\mathcal{O}$ be the point at infinity. The rules for addition over the elliptic group $E_p(a,b)$ are :

(1) $P+\mathcal{O} = \mathcal{O} + P = P$

(2) If $x_2 = x_1$ and $y_2 = -y_1$,

that is $P(x_1, y_1)$ and $Q = (x_2, y_2) = (x_1 - y_1) = -P$,

that is the case: $P+Q = \mathcal{O}$.

3) If $Q \neq -P$, then their sum $P + Q = (x_3, y_3)$ is given by;

$$x_3 = \lambda - x_1 - x_2 \bmod p$$

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

III. Diffie-Hellman Key Exchange

Diffie–Hellman establishes a shared secret that can be used for secret communications by exchanging data over a public network. The following diagram illustrates the general idea of the key exchange by using colours instead of a very large number. The key part of the process is that Alice And Bob exchange their secret colours in a mix only. Finally this generates an identical key that is mathematically difficult (impossible for modern supercomputers to do in a reasonable amount of time) to reverse for another party that might have been listening in on them. Alice and Bob now use this common secret to encrypt and decrypt their sent and received data. Note that the yellow paint is already agreed by Alice and Bob:

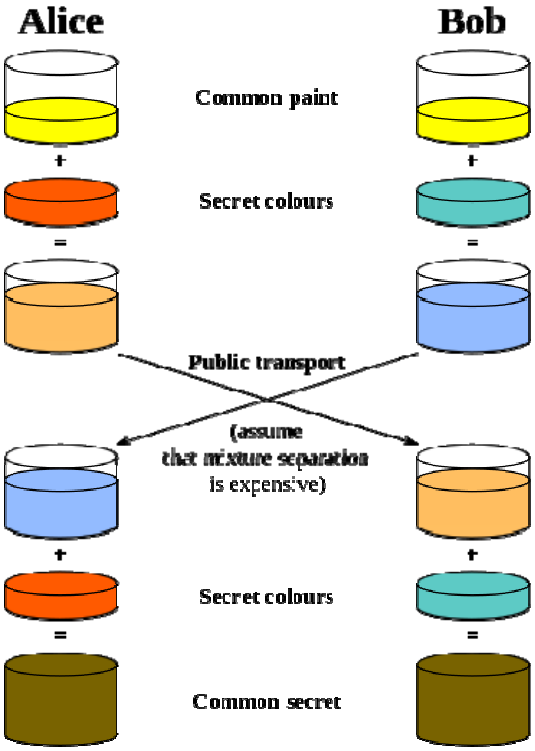

Fig 3: Diffie-hellman key exchange.

IV. IN PREVIOUS SYSTEM

RSA algorithm simply capitalizes on the fact that there is no efficient way to factor very large integers. The security of the whole algorithm relies on that fact. If someone comes up with an easy way of factoring a large number, then that's the end of the RSA algorithm. Then any message encrypted with the RSA algorithm is no more secure.

The security of RSA algorithm depends on the ability of the hacker to factorise numbers. Newer faster and better methods for factoring numbers are constantly being devised. The current best for long numbers of the number field sieve. Prime number of a length that was unimaginable a mere decade ago are now factored easily.Obiviously the larger the number is, the harder it is to fact and so the better the security of RSA.

As theory and computers improve large and larger keys will have to be used. The disadvantage in using extremely long keys is the computational overhead involved in encryption/ decryption. This will only become a problem if a new factoring technique emerges that requires keys of such lengths to be used that necessary key length increase much faster than the increasing average speed of computers utilizing the RSA algorithm.

V. OUR PROPOSED WORK AND PERFORMANCE

5.1 Diffie-Hellman key exchange using Elliptic Curve (DHECC)

Elliptic Curve Cryptography (ECC) is emerging as an attractive public-key cryptosystem for mobile/wireless environments. Compared to traditional cryptosystems like RSA, ECC offers equivalent security with smaller key sizes, which results in faster computation, lower power consumption, as well as memory and bandwidth savings. This is especially useful for mobile devices which are typically limited in terms of their CPU, power and network connectivity. However, the true impact of any public-key cryptosystem can only be evaluated in the context of a security protocol. This paper presents a first estimate of the performance improvements that can be expected in SSL (Secure Socket Layer), the dominant security protocol on the Web today, by adding ECC support.

An elliptic curve E which is over the finite field Fp is given through an equation. An equation will be of the form

$$Y^2 = X^3 + aX + b,$$
$$a, b \in Fp, \text{ and } -(4a^3 + 27b^2) \neq 0$$

Please note that as stated in the beginning of the section, the "=" should be replaced by a "≡" in the above definition. Another remark is that when we talk about partial derivatives we mean the "formal partial derivate" and this formal partial derivate can be defined (see beginning of this section) over an arbitrary field.

Suppose two communication parties, Alice and Bob, want to agree upon a key which will be later used for encrypted communication in conjunction with a private key cryptosystem.

They first fix a finite field F_q, an elliptic curve E defined over it and a base point B∈E (the base point will be with high order). To generate a key, first Alice chooses a random a∈F_q (this random is of high order) which she keeps secret. Next she calculates aB∈E which is public and sends it to Bob. Bob does the same steps, i.e. he chooses a random integer b (this random integer will be secret) and calculates bB which is sent to Alice. Their secret common key is then P = abB∈E.

Definition An elliptic curve E over the field F is a smooth curve in the so it is called"long transform"

$$Y^2 + a_1XY + a_3Y = X^3 + a_2X^2 + a_4X + a_6,\ a_i \in F$$

We let E(F) denote the set of points (x, y) ∈F2 these points satisfy this equation, they are also along with a "point at infinity" which is denoted by O.

Remember that smooth curve means that is the curve in which there is no point in E(F) where both partial derivatives vanish. (Remember that when we talk about partial derivatives we mean the "formal partial derivate" and this formal partial derivate can be defined (see beginning of this section) over an arbitrary field.) The definition given above is valid for any field. But in cryptography we are only interested in finite fields. Considering only finite fields we get an "easier" equation. Two finite fields are of particular interest.

The finite field Fp with p ∈ E elements, because of its structure, and the finite field F_qm with q = pr Elements, since setting p = 2 the arithmetic in this field will be well suited for implementations in hardware.

For generation of a shared secret key between A and B using ECDH, both have to agree up on EC domain parameters. Both end have a key pair consisting of a private key d(a randomly selected integer less then n, where n is the order of the curve) and another is a public key Q= d*G (G is the generator point). Let (d_A, Q_A) be the private-public key pair of A and (d_B, Q_B) be the private-public key of B.

1. The end A Computes $K_A = (X_A, Y_A) = d_A * Q_B$
2. The end B Computes $K_B = (X_B, Y_B) = d_B * Q_A$
3. Since $d_A * Q_B = d_A d_B G = d_B d_A G = d_B * Q_A$. Therefore $K_A = K_B$ and hence $X_A = X_B$
4. (Where G is generator point)
5. Hence the shared secret is K_A.

Since it is practically impossible to find the private key d_A or d_B from the public key K_A

5.2 Two level Encryption Decryption by Diffie – Hellman and Elliptic Curve method

Today, the scientific efforts are looking for a smaller and a very faster public key cryptosystem, at the same time the approach should be practical and very secure, even for the most constrained environments. For any cryptographic technique, there is an analogue for Elliptic Curve. One of

these systems is Diffie – Hellman key exchange system.
This paper proposed methods to encrypt and decrypt the message, and we will encrypt and decrypt the message by using the Diffie–Hellman Exchanging key. And this is a secrete point in the proposed methods (M1) and (M2).
In the first method (M1), the sender compute the multiplication between the coordinates of the key in the encryption algorithm, and the receiver compute the multiplication between the coordinates of the key in the decryption algorithm.
In the second method (M2), we support the system more security of the first method, because the sender compute the exponentiation function between the coordinates of the key in the encryption algorithm (use fast exponentiation method), and the receiver compute the inverse of the exponentiation function between the coordinates of the key in the decryption algorithm.
This system is merely a method for exchanging key and no massages are involved in the system. The following algorithm illustrates this system.
The Algorithm of Diffie–Hellman key exchange system Using ECC

- Alice and Bob first choose a finite field F_p and an elliptic curve E defined over it ($E(F_p)$).
- They publicly choose a random base point B belongs E.
- Alice chooses a secret random integer e. He then computes eB∈E. In addition, send it to Bob.
- Bob chooses a secret random integer d. She then computes dB∈E. And send it to Alice.
- Then eB and dB are public and e and d are secret.
- Alice computes the secret key edB = e(dB).
- Bob computes the secret key edB = d(eB).

There is no fast way to compute edB if only knows B, eB and dB.After these setups, Alice and Bob have the same point (only Alice and Bob know it). Then to start with (M1) and (M2), let us consider the following algorithms

Algorithm of (M1)
Alice and Bob Compute edB = S = (S_1, S_2). (Using Diffie – Hellman Scheme)

Alice sends a message M ∈ E to Bob as follows:
Compute (S_1 * S_2) mod N = K.
Compute K*M = C, and send C to Bob.

Bob receives C and decrypts it as follows:
Compute (S_1 * S_2)modN = K.
Compute (K^{-1})modN.
(where N = #E)
K^{-1}*C = K^{-1}*K*M = M.
In the first method (M1), the sender compute the multiplication between the coordinates of the key in the encryption algorithm, and the receiver compute the multiplication between the coordinates of the key in the decryption algorithm.
Algorithm of (M2)
Alice and Bob Compute edB = S = (S_1, S_2).
Using Diffie – Hellman Scheme)
Alice sends a message M to Bob as follows:
Compute (s_1^{S2}) mod N = K.
Compute K*M = C, and send C to Bob.
Bob receives C and decrypts it as follows:
Compute (s_1^{S2})mod N = K.
Compute (K–1) mod N.
K^{-1}*C = K^{-1}*K*M = M.
In the second method (M2), we support the system more security of the first method, because the sender compute the exponentiation function between the coordinates of the key in the encryption algorithm (use fast exponentiation method), and the receiver compute the inverse of the exponentiation function between the coordinates of the key in the decryption algorithm.

System Test:
Let E be an elliptic curve define over Fp
Where p = 3023 with parameters a = 1, b = 2547
Where ($4a^3$+27 b^2) mod p = 2027 0.
And #E = 3083.
Since #E is prime number then by theorem1, every point on E in base point, therefore let B = (2237, 2480).
To apply this system test using (M1), at first we must apply Diffie-Hellman Exchanging key

- Alice chooses a secret random integer e = 2313.
  eB = 2313 (2237 , 2480) = (934 , 29)
  And send (934, 29) to Bob .
- Bob chooses a secret random integer d = 1236.
  dB =1236 (2237 , 2480) = (1713, 1709)
  And send (1713 , 1709) to Alice
- Alice computes the secret key e (dB)
  = 2313 (1713, 1709).
  edB = (2537 , 1632) = S
- Bob computes the secret key d (eB) =1236 (934, 29).
  deB = (2537 , 1632) = S

Now, Alice and Bob have the same point S = (2537, 1632)
If Alice send a message M = (2284, 2430) to Bob

- Compute (S_1, S_2).mod p = (2537 * 1632) mod 3083 = 2998 = K.
- Compute K*M = 2998 (2284, 2430)
  = (2179, 1833)
  =C, and send it to Bob.
- Bob receives C and decrypts it as follows:
  - Compute (S_1, S_2).mod p = 2998 =K
  - Compute (K–1) mod N = (2998)–1

    mod 3083= 1342
  - K–1 C =1342 (2179, 1833)
    = (2284, 2430)

To apply this system test using the algorithm (M2), at first we must apply Diffie–Hellman Exchanging key.
By the same procedure to solve Diffie–Hellman scheme we have obtained
S = (2537, 1632)
If Alice sends a message M = (2284, 2430) to Bob using (M2), he does the following:

- Compute(s_1^{S2})mod N = (25371632) mod 3083= 323=K.
- Compute K* M = 323 (2284, 2430)
  = (2555, 1066)
  =C, and send it to Bob.

- Bob receives **C** and decrypts it as follows:
  - Compute (s_1^{S2})mod N = 323=K.
  - Compute (K–1) mod N = (323)–1 mod 3083= 1594.
  - K–1 C = 1594 (2555 , 1066)
    = (2284, 2430)
    = M.

Performance— Performance of Elliptical Cryptography with Diffie Hellman Key Exchange will depend on the hardware of system. The Performance of Elliptical Cryptography with Diffie Hellman Key Exchange will also depend on an another factor that is the quality of the JavaScript which is our execution environment.

The following table shows the times taken for various public-key operations on a cross-section of browsers and hardware both.

EC multiply = Elliptic curve point multiplication, bit size denotes both curve prime size and scalar multiplier size.

Browsers used:

Google Chrome version 10.0.648.151

Mozilla Firefox version 3.6.15

Microsoft Internet Explorer version 8.0.7601.17514

PC = Win7 64-bit, Intel Core i5 M520 (2.4GHz)

TABLE 1: CROSS SECTION OF BROWSERS AND HARDWARE

S.No.	Operation		Browser's (In which operations are performed)					
	RSA (with key size)	DHECC (with key size)	Chrome(ms)		Fire fox(ms)		IE(ms)	
			RSA	DHECC	RSA	DHECC	RSA	DHECC
1	public, 512 bit, e=3	multiply, 128 bit	0	25	1	200	4	500
2	public, 512 bit, e=F4	multiply, 160 bit	1	30	6	450	20	820
3	public, 1024 bit, e=3	multiply, 192 bit	1	35	3	750	10	1250
4	public, 1024 bit, e=F4	multiply, 224 bit	2	50	15	900	70	2500
5	private, 512 bit	multiply, 256 bit	5	65	75	1300	190	3100

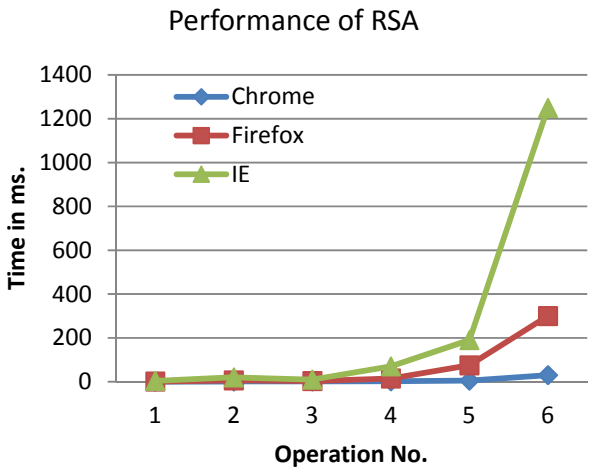

Fig 4: Performance of RSA

Figure 4 shows the performance of RSA algorithm with respect to time and number of operation performed by algorithm in Chorme and Firefox, internet explorer web browser.

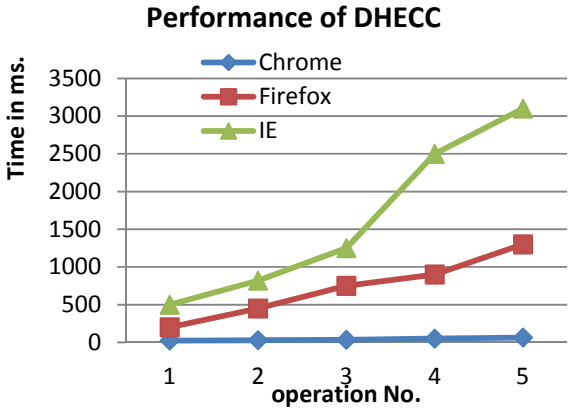

Fig 5: Performance of DHECC

Figure 5 shows the performance of Elliptic curve cryptography algorithm performance with respect to time and number of operation using chrome, Firefox and internet explorer web browser.

CONCLUSION

The Diffie–Hellman scheme is one of the exchanging key cryptosystem, no massages are involved in this scheme, in this report, and we try to benefit from this scheme by use the key (which exchange it) as a secret key. (That is, we know now the one of the advantages of the Diffie–Hellman key exchange system) and we are using Elliptic curve cryptography for encryption and Decryption. We proposed two different methods to encrypt and decrypt the message. In the second method, we support the system more security of the first method, because the sender compute the exponentiation function between the coordinates of the key in the encryption algorithm (use fast exponentiation method),

and the receiver compute the inverse of the exponentiation function between the coordinates of the key in the decryption algorithm. While in the first method, the sender compute the multiplication between the coordinates of the key in the encryption algorithm, and the receiver compute the multiplication between the coordinates of the key in the decryption algorithm and we can use our approach for forward secrecy in HTTPS protocol.

REFERENCES

1. Yufang Huang, "Algorihtm for elliptic curve diffie-Hellman key exchange based on DNA title self assembly In Proceedings of 46th IEEE Theories and Applications,pp.31-36, 2008.
2. A. M. Fiskiran and R. B. Lee. "Workload characterization of elliptic curve cryptography and other network security algorithms for constrained environments". IEEE International Workshop on WWC-5, 2009
3. B. Kaliski, M. Liskov and Y. L. Yin. "Efficient finite field basis conversion techniques". Proposal for Inclusion in IEEE P1363,2009
4. V.S. Miller, "uses of elliptic curves in cryptography," in Advances in Cryptology, CRYPTO'85, ser . Lecture Notes in Computer Science, vol. 218, Springer, 1986. pp. 417-428.
5. N. Koblitz,"Elliptic curve cryptosystems," Mathematics of Compution, vol. 48, no.177, pp.203-209, Jan 1987.
6. D. Hakerson, A. Menezes, and S. Vanston , "Guide to Elliptic Curve Cryptography," Springer-Verlag, NY (2004).
7. H. Cohen, A Miyaji and T. Ono, "Efficient elliptic curve exponentiation using mixed coordinates," Lectures Notes in Computer Science, 1514,51-65 (1998).
8. Dimitrov V., L. Imbert, and P. K. Mishra, "Efficient and secure elliptic curve point multiplication using double-base chains," Lectures Notes in Computer Science, 3788, 59-78 (2005).
9. M. Ciet, M. Joye, K. Lauter and P.L. Montgomery,"Trading inversions for multiplications in elliptic curve cryptography," Designs, Codes, and Cryptography, 39, 189-206 (2006).
10 . Y. Chen, J.S. Chou, C.H. Huang , "Comments on five smart card based password authentication protocols," International Journal of Computer Science and Information Security 8 (2) (2010) 129–132.
11 Ch. Suneetha, D. Sravana Kumar and A. Chandrasekhar, "Secure key transport in symmetric cryptographic protocols using elliptic curves over finite fields," International Journal of Computer Applications, Vol. 36, No. 1 November 2011.
12 Mohsen Machhout eat.al., "coupled FPGA/ASIC Implementation of elliptic curve crypto-processor," International Journal of Network Security & its Applications Vol. 2 No. 2 April 2010.
13. M. Kumar, "An enhanced remote user authentication scheme with smart card," International Journal of Network Security 10 (3) (2010) 175–184.