Mutual Authentication Protocol Using Hyperelliptic Curve Cryptosystem in Constrained Devices

Kakali Chatterjee[1], Asok De[2], and Daya Gupta[1]
(Corresponding author:Kakali Chatterjee)

Computer Engineering Department, Delhi Technological University, Delhi, India[1]
Ambedkar Institute of Advanced Communication Technologies & Research, Delhi, India[2]
(Email: kakali2008@gmail.com)

(*Received Aug. 10, 2011; revised and accepted Feb. 22, 2012*)

Abstract

Hyperelliptic Curve Cryptosystem (HECC) is well suited for secure communication in wireless sensor network as limited resources (storage, time or power) on sensor nodes restrict the use of conventional security techniques. We can construct genus 2 HECC on 80-bit finite fields in order to achieve the same security level as 160-bit ECC or 1024-bit RSA. This paper proposes a mutual authentication protocol based on Hyperelliptic Curve Digital Signature Algorithm for a secure access in constrained devices which allows both the entities to verify each other's authenticity. Our experimental result shows that performance of the proposed system is comparable with that of ECC.

Keywords: Hyperelliptic Curve Cryptosystem (HECC), Hyperelliptic Curve Digital Signature Algorithm (HECDSA), mutual authentication protocol

1 Introduction

With the rapid development of the information technology, wireless networks are now extensively used to transmit critical information relating to monitoring of real time data. The security mechanisms are essential to ensure integrity, confidentiality and authenticity of the data. Implementation of suitable cryptosystem in this environment is challenging as these networks consist of many tiny and smart devices which are constrained in terms of memory, computing power and energy supply. While considering the different security threats involved in wireless sensor network, different symmetric /asymmetric algorithms are proposed in [6, 10, 15, 21]. In general these approaches either need pre-distributed keys which mean a higher configuration effort before deployment or they produce much traffic which results in higher energy consumption [20].

In case of symmetric algorithms memory usage is reduced but an adversary can easily eavesdrop on communication or a stealing of node can be possible. To avoid these types of attacks, a mutual authentication protocol based on ECC has been proposed [22]. The principal attraction of ECC compared to RSA is that it offers equal security for a far smaller key size, thereby reducing processing overhead. ECC based mutual authenticated key agreement protocol was already established for Wireless LAN security [2, 14]. So far, several protocols have been proposed to provide robust mutual authentication and key establishment for Wireless LAN. The significant improved performance of some of the protocols in computational and communicational load over many other key agreement protocols were compared and discussed in [3]. These protocols utilize ECDSA signature technique that enhances the security of user authentication and key exchange. However, the security level can also be increased using hyperelliptic curves because it has some advantage over ECC. For HECC over a finite field one needs 40-bits to 80-bit long operands to compute the group operations for these curves. In the case of ECC we have to work with operand lengths of approximately 160-bit whereas in the case of RSA, the operands will be approximately 1024-bit in order to achieve the same security. HECC is, therefore, more suitable for implementation in the constrained platforms in wireless networks.

Hyperelliptic Curve Cryptosystem (HECC) was proposed by Koblitz [13] in 1989 based on the discrete logarithm problem on the Jacobian of hyperelliptic curves over finite fields. The main difference between ECC and HECC is in group operation because these consist of different sequences of operations. Unlike elliptic curves, the points on the hyperelliptic curve do not form a group. The additive group on which the cryptographic primitives are implemented is the divisor class group. Each element of this group is a reduced divisor. Divisor group operations of HECC are more complex compared to point operation of ECC for implementing the cryptographic primitives. Hence it is challenging to implement HECC in constrained environment.

Different aspects of Hyperelliptic curve cryptosystem are discussed in Avanzi [1], Menezes et al. [16], Pelzl et al. [17, 18]. We have discussed Evolution of Hyperelliptic Curve Cryptosystems in [9]. Current Research on HECC emphasize on finding efficient methods to select secure hyperelliptic curves, fast operations on the Jacobians and implementation of

HECC for use in practical applications to enhance network security.

Our contributions in this paper are as follows:

(1) We propose a mutual authentication protocol based on HECDSA for a secure communication in wireless network for constrained devices. The proposed protocol is a secure authenticated protocol as it can resist the possible attacks from both internal users and external hackers as discussed in the Security Analysis.

(2) We have implemented our proposed authentication protocol over the finite field $\mathbb{F}_p$ in affine co-ordinates in Netbeans IDE 6.8. Our experimental result shows the timings of basic operations of the proposed Hyperelliptic curve based scheme which enables us to exchange keys, sign & authenticate documents and we have compared these timings to study the relative performance of HECC with that of ECC.

The rest of the paper is organized as follows: Section 2 presents Mathematical Background; Section 3 provides Proposed Mutual Authentication Protocol; Section 4 discusses Security Analysis; Section 5 presents Implementation Results; Finally, we conclude the paper in Section 6.

2 Mathematical Background

2.1 Arithmetic of Hyperelliptic Curves

Let $\mathbb{F}$ be a finite field, and let $\overline{\mathbb{F}}$ be the algebraic closure of $\mathbb{F}$ [18]. A hyperelliptic curve C of genus $g \geq 1$ over $\mathbb{F}$ is the set of solutions $(u, v) \in \mathbb{F} \times \mathbb{F}$ to the equation $C: v^2 + h(u)v = f(u)$. The polynomial $h(u) \in \mathbb{F}[u]$ is of degree at most g and $f(u) \in \mathbb{F}[u]$ is a monic polynomial of degree $2g + 1$. For odd characteristic it suffices to suppose that $h(u) = 0$ and $f(u)$ is square free. If no point on the curve over the algebraic closure $\overline{\mathbb{F}}$ of $\mathbb{F}$ satisfies both partial derivatives $2v + h(u) = 0$ and $h'(u)v - f'(u) = 0$, then the curve is said to be non-singular.

The points of curve C generate a set called Jacobian. The Jacobian of the curve C is the quotient group $\mathbb{J} = \mathbb{D}^0/\mathbb{P}$, where $\mathbb{D}^0$ is the set of divisors of degree zero, and $\mathbb{P}$ is the set of divisors of rational functions. The element of Jacobian over $\mathbb{F}$, denoted by $\mathbb{J}_C(\mathbb{F})$, can be represented uniquely by a divisor $D = \sum m_i P_i$, $m_i \in \mathbb{Z}$, is a finite formal sum of $\overline{\mathbb{F}}$ points. Its degree is the sum of the coefficients $\sum m_i$. The set of all divisors form an Abelian group denoted by $\mathbb{D}(C)$. The set of degree zero divisors $\mathbb{D}^0$ forms a subgroup of $\mathbb{D}(C)$ [16].

Cantor shows that each element of the Jacobian can be represented in the form $D = \sum_{i=1} P_i - r.\infty$ such that for all $i \neq j$, P_i and P_j are not symmetric points [7]. Such a divisor is called a semi-reduced divisor. Each element of the Jacobian can be represented uniquely by such a divisor, subject to $r \leq g$. Such divisors are referred to as reduced divisors. We use the reduced divisor in addition of $\mathbb{J}_C$. Cantor's algorithm is used for doing arithmetic in general hyperelliptic curve which applies to any genus and characteristic. This transfer the group laws in a sequence of Composition and Reduction using only polynomial arithmetic. Group operations on a Jacobian are performed in two steps: addition of generic divisors and doubling of generic divisors. Addition of divisor classes means multiplication of ideal classes, which consists in a composition of the ideals and a first reduction to a basis of two polynomials. The output of this algorithm is called semi-reduced divisor. Then the second algorithm (reduction) is used to find the unique representative in the class.

The equivalence classes of the Jacobian are represented by a unique reduced divisor (which is represented using Mumford representation) upon which we perform the group law. Each unique reduced divisors can be represented via a unique pair of polynomials $u(x)$ and $v(x)$, $u, v \in \mathbb{F}_q[x]$, where

(1) u is monic

(2) $deg\, v < deg\, u \leq g$

(3) $u \mid v^2 + vh - f$

This is known as Mumford representation. Mumford proposed a convenient way to represent each reduced divisors as $D = (u(x), v(x))$, where $u(x)$ is a monic polynomial with $deg(u(x)) \leq g$ and $v(x)$ (which is not monic in general) with $deg(v(x)) < deg(u(x))$.

When developing formulas for implementing genus g arithmetic, we are largely concerned with the frequent case that arises where both reduced divisors $D_1 = [u_1, v_1]$, $D_2 = [u_2, v_2]$ are used to find the unique reduced divisor equivalent to $D_1 + D_2$. To find the unique reduced divisor equivalent to $D_1 + D_2$, one performs two steps (Cantor's algorithm):

(1) Determine a semi-reduced divisor D representing the sum of D_1 and D_2.

(2) Transform the semi-reduced divisor D into a reduced one, i.e compute D' such that D' is the unique reduced divisor equivalent to D.

2.2 Security of Hyperelliptic Curves

In hyperelliptic curve cryptography, finding a suitable hyperelliptic curve is an important fundamental problem [16]. Security of HEC is based on the difficulty of solving the discrete logarithm problem in Jacobian of HEC. The HCDLP in $J(C; \mathbb{F}_q{}^n)$ is: given two divisors D_1, D_2 defined on $J(C, \mathbb{F}_q{}^n)$ over $\mathbb{F}_q{}^n$, to determine integer m such that $D_2 = mD_1$, provided such an integer m exists.

To establish a secure hyperelliptic curve, its Jacobian should satisfy the following conditions:

(1) Adleman et al. [4] found a subexponential time algorithm to solve the DL in the Jacobian of HEC of a big genus over a finite field. Curves of higher genera (preferably $g \leq 4$) are, therefore, not suitable for cryptographic use $(2g+1 < log\, q^n)$.

(2) If the group order is large but divisible by only small primes, the DLP can be broken by Pohlig-Hellman attack. It is claimed that this largest prime factor should be at least 160 bits in length.

(3) To prevent the attack of Frey [11] which uses Tate pairing generation of MOV attacks, the large prime factor of $J(C; \mathbb{F}_q{}^n)$ should not divide $(q^n)^k$-1, here $k < (log\, q^n)^2$.

(4) To prevent the attack generated by Ruck [19], the Jacobian of a hyperelliptic curve over the large prime field should not have p-order subgroup.

(5) To harden a cryptographic primitive against simple side-channel attacks, we make the observable information independent of the secret scalar. This can be achieved by applying Montgomery's ladder for scalar multiplication.

3 Proposed Mutual Authentication Protocol

A mutual authentication protocol is necessary to resist the attacks when a malicious user pretends as an authorized one and duplicate, modify, insert or delete the data during transmission. We propose a mutual authentication protocol based on HECDSA [8] in wireless network which gives authentication and non-repudiation. The novelties of the protocol are as follows:

- Our proposed mutual authentication protocol based on HECDSA in wireless network is suitable for constrained devices as it uses genus 2 HEC on 80-bit finite fields which achieve the same security level as 160-bit ECC.
- Any remote user can obtain service from other users without registering each time with the KDC. They can transfer data after mutual authentication.
- New session key is established for each particular session to protect data which resists replay attack in wireless network.
- Encryption of transmitted message using asymmetric encryption process saves energy and storage, which is critical for constrained devices.

Notations used

C: A hyperelliptic curve of genus g defined over $\mathbb{F}_p$
p : A large prime number
q : A large prime divisor of p *-1.*
P: A Base point on the Hyperelliptic Curve
D: The semi reduced divisor of the HEC
D': The unique reduced divisor of the HEC
PR_A, PU_A: Private and Public key of A respectively
PR_B, PU_B: Private and Public key of B respectively
ID_A: Identity of A
ID_B: Identity of B
M: Input Message
(r,s): Signature pair
M': Received message
(r',s'): Received Signature pair
$H(.)$: One-way hash function with fixed length output
K: Common Secret key
K_s: Session key
K_a: Premaster key shared between Users and KDC
T_s: Session time
$\oplus$: Group addition between Jacobian elements

Our proposed protocol is described below:

During the initialization phase, Key Distribution Center (KDC) generates a random hyperelliptic curve C defined over $\mathbb{F}_p$. Then KDC computes semi reduced Divisor D and the unique reduced divisor D' of the selected curve using Cantor's algorithm. In this protocol we use the unique reduced divisor (D') and semi reduced divisor (D) distinctly.

KDC also computes a point $P = (x_1,y_1) \in C(\mathbb{F}_p)$ which is a base point on the curve, a large prime number p and a prime divisor q such that q divides p*-1*. $\mathbb{F}_p$ contains the representation of all field elements of order n. Finally the following system parameters ($\mathbb{F}_p$, C, D', p, q, D, n) are generated by KDC.

During the network deployment phase, all Users send request message to KDC for getting registered in the network. After registration KDC assigns a unique ID to each User and send ID with a Premaster key K_a to each User. KDC also makes a list of all Users with their ID.

After the network deployment phase, KDC publish this list encrypted by the Premaster key K_a to all Users of the network with the system parameters ($\mathbb{F}_p$, C, D', p, q, D, n). We also assume after deployment of the Users (nodes), they become static.

Now User A wants to communicate with User B. So User A sends a request message to User B containing ID_A and a nonce N_1.Once the message received, if User B wants to communicate with A, it first verify the ID from the list. If it matches, then B sends an accept message to User A containing ID_B and nonce N_1.

Now User A and User B will communicate after the mutual authentication.

Our proposed mutual authentication scheme is shown in Figure 1.

In this scheme the session key are generated using following three major steps:

Step 1

User B chooses a random challenge d_B, where $1 \leq d_B \leq n - 1$, then it calculates

$Q_B = d_B \times D' = [u_B, v_B]$ using scalar multiplication in genus 2 HECC described in [8].

Then User B generates private key $PR_B \in_R N$ [choose a positive prime at random in N] and public key $PU_B = [PR_B]D$. Here PU_B is represented using Mumford representation which is of the form $[u_B, v_B]$.

Finally User B sends (Q_B, PU_B) to User A.

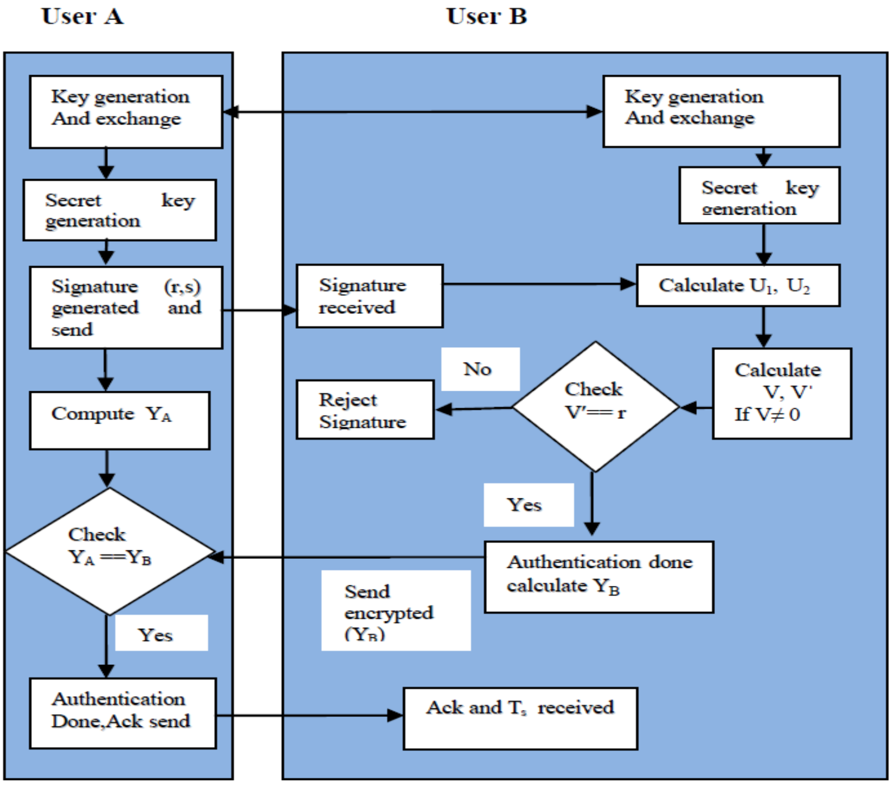

Figure 1: Mutual authentication between two users

Step 2

User A chooses a random challenge d_A, where $1\leq d_A \leq n-1$, then computes

$Q_A = d_A \times D' = [u_A, v_A]$ using scalar multiplication in genus 2 HECC.

Then A computes $z = Q_A \oplus Q_B$ for mutual authentication (by performing Divisor addition in affine coordinate).

The user A generate private key $PR_A \in_R N$ [choose a positive prime at random in N] and public key $PU_A = [PR_A]\, D$.

After that User A compute the secret key $K = d_A \times Q_B = [u_i, v_i]$.

In addition, User A calculates $r = (\sum_{i=0}^{e-1} L(u_i)q^{i}\ )\ mod\ p$ where

e is an integer with $e \leq g$ and assuming the finite field

elements are ordered such that $0 \leq L(\ u_i) < q$ [for mapping between Jacobian J($\mathbb{F}_p$) and finite field GF(p)].

User A also computes $s = [r^{-1}(H(M) - [PR_A]\, r)]$ mod p.

Finally $(r,\ s)$ becomes the signatures pair and A transfers signature pairs with $(Q_A,\ PU_A)$ to User B.

Step 3

User B computes $\beta = Q_A \oplus Q_B$ by performing Divisor addition in affine coordinate.

He also computes secret key $K = d_B \times Q_A = [u_i, v_i]$. If the protocol works correctly, both the Users generate the same value of K.

This can be proved by the simple mathematical calculation shown below:

$K = d_B \times Q_A = d_B \times d_A \times D' = d_A \times d_B \times D' = d_A \times Q_B = [u_i, v_i\ ]$

User B also calculates $w=(s')^{-1} mod\ p$ where

(r', s') is the received signature.

After that User B calculates $U_1 = (H(M')\ w)\ mod\ p$ and $U_2 = (r'w)\ mod\ p$.

In addition, B calculates $V = [U_1]\, D \oplus [U_2]\, PU_A = [u_f, v_f]$. If $V = [1,0]$, it implies that the signature is incorrect and User B rejects the signature with message.

Otherwise User B calculates

$V' = (\ \sum_{i=0}^{e-1} L(u_{f,i})q^{i}\ )\ mod\ p$

[for mapping between Jacobian J($\mathbb{F}_p$) and finite field GF(p)].

If $(V' == r)$, it implies that the signature is correct, so User B authenticates User A and User B can be confirmed that User A has actually established the same secret key.

Then User B computes $Y_B = H(\beta) + u_i$ and sends encrypted Y_B (encryption done by using secret key K) to User A. User A decrypt the packet and find Y_B.

In order to authenticate User B, User A will compute $Y_A = H(z) + u_i$ and then User A will verify the value of Y_A by checking that ($Y_A == Y_B$).

Table 1: Experimental Results of genus 2 HECC (Prime Field)

G-2 Prime Field Group order	Divisor Generation (ms)	Public Key PU_A (ms)	Public Key PU_B (ms)	User-A Secret key K (ms)	User-B Secret key K (ms)	Signature Generation (ms)	Signature Verification (ms)
(2^152)	10	16	15	11	15	84	28
(2^158)	11	16	15	16	15	72	30
(2^162)	10	15	14	10	9	60	31
(2^166)	11	16	16	15	16	56	32

Table 2: Comparison of our experimental result with existing literature

Reference	Curves	Group Order	Signature Generation (ms)	Signature Verification (ms)
N. Jansma et al. [12]	ECC (genus-1)	2^163	150	230
M. Aydos et al. [5]	ECC (genus-1)	2^160	46	92
Our results	HECC (genus-2)	(2^152)	84	28
		(2^158)	72	30
		(2^162)	60	31
		(2^166)	56	32

If they match, then User A authenticates User B and User A can be confirmed that User B has actually established the same secret key with him. User A then sends an acknowledgement with the session time T_s to User B.

Finally, A and B agree on the common session key K_s where $K_s = H(ID_A || ID_B || K)$

If all the above steps are executed correctly, both sides will agree on the session Key K_s. Once the protocol run completes successfully, both sides may use K_s to encrypt messages (using ElGamal method) with timestamp for subsequent session traffic in order to create a confidential communication channel. After each valid session a new session key will generate.

4 Security Analysis

In this section, we discuss the security of our proposed mutual authentication protocol based on HECDSA. The proposed protocol will be considered to be a secure authenticated protocol, if it satisfies the following properties:

Man in the middle attack: It can be considered as an active attack. In this protocol, no useful information about the secret key K is revealed during a successful run. If an attacker E intercepts the message packet containing (Q_B , PU_B), E then receives PU_B and Q_B from B. However, this means that E must calculate K but E cannot compute the value of K because E does not know the value of d_A or the value of d_B. This problem is called Computational Diffie-Hellman Problem (CDHP). So, E will not be able to compute K. Thus this protocol resists the man in-the-middle attack.

Small subgroup attack: If hyperelliptic curve C has enough prime factors, the attacker could determine the secret scalar modulo of all these primes and recover a large part of the secret by using Chinese remaindering. To avoid this attack, we check D has order l where l is prime. For checking this, we first check that $[l]D=0$ and computing $[h]D$ for $h=c/p_i$, for all prime divisors p_i of c and checking that the result is not 0 [8].

Known-key attack: In our proposed protocol, both users generate new PU_A and PU_B in every new session, and in addition the secret key K is generated with every new session also. Another important aspect of our protocol is that the session key is calculated independently on both sides and protected by the secure hash function. Thus our proposed protocol is secure against known key attacks assuming that the hyperelliptic curve discrete logarithm problem is intractable.

Perfect forward secrecy: In perfect forward secrecy, even if the user's ID is compromised, it never allows the adversary to determine the session key for past sessions and decrypt them. In our protocol, it is based upon the assumption that the discrete logarithm problem is intractable and on the value of the secret key K. Even if the attacker knew the correct Q_B, the attacker still cannot compute the previous session keys because K_s is derived from the secret key K which is generated from the value of d_A and d_B . Thus the property of perfect forward secrecy is satisfied by our proposed protocol.

Replay attack: Replay attack involves passive capture of data and its subsequent retransmission to show unauthorized effect. Any unauthorized malicious user can send duplicate data repeatedly to the receiver which is already sent. Our protocol protects replay attack as it depends upon timestamp values. In our protocol, after each valid session time T_s which is unknown to malicious user, a new session key will generate for encryption, so that replay attack is not possible.

5 Implementation Results

We have implemented our proposed protocol of HEC (genus 2) on different prime fields using *jdk1.6.2*. The timings of basic operations have been measured on a PC with Intel Core i3 CPU 540 @3.07 GHz and Windows 7 operating system having jdk1.6. For HECC implementation, we have considered the genus 2 hyperelliptic curve $C: y^2 = x^5 + x^3 + 1$ over the finite field $\mathbb{F}_p$. Considering $D = (u(x), v(x))$, $D' = (u'(x), v'(x)) \in J(C)$, we generate $D = (x^2 + 6x + 33, 22x + 47)$ and $D' = (x^2 + 13x + 12, 46x + 59)$. Next we have implemented our proposed scheme using Hyperelliptic curves of different group order and main operations like key Generation, Signature Generation / Verification timing of the proposed mutual authentication protocol are listed in the Table 1.

The proposed protocol is suitable for constrained devices as it requires less key size and has low storage requirement for user side. The protocol has also low computational load (4 point multiplication + 2 secret key encryption / decryption + 1 signature generation / verification + 3 SHA-1 operation) on each side.

It can be seen that the proposed protocol of HEC (genus 2) is efficient as the timings of our signature generation / verification compares favorably with the timings of ECC available in existing literature as shown in the Table 2.

As HECC (genus 2) of 80-bit operand lengths provide same security level with ECC of 160-bit, it can be stated that HECC is more suitable for implementation in the constrained platforms in wireless networks.

6 Conclusions

HECC is well suited for secure communication in wireless network for constrained devices as HEC operand size is only a fractional amount of the EC operand size and almost all the standard discrete logarithm based protocols such as the Diffie-Hellman and ElGamal can be planted to HEC. We have proposed in this paper a mutual authentication protocol based on HECDSA for a secure access in constrained devices which allows both the entities to verify each other's authenticity. It is seen that the proposed protocol of HECC is efficient as the timings of our signature generation / verification compares favorably with the timings of ECC available in existing literature. As HECC (genus 2) of 80-bit operand lengths provide same security level with ECC of 160-bit, in our view, HECC is more suitable for implementation in the constrained platforms in wireless networks.

References

[1] R. M. Avanzi, "Aspects of hyper-elliptic curves over large prime fields in software implementations," Cryptographic Hardware and Embedded Systems, LNCS vol. 3156, pp. 148-162, 2004.

[2] M. A. Azim and A. Jamalipour. "An efficient elliptic curve cryptography based authenticated key agreement protocol for wireless LAN security", International Workshop on High Performance Switching and Routing (HPSR'05), pp. 376-380, 2005.

[3] P. E. Abi-char, A. Mhamed, and B. E. Hassan, "A secure authenticated key agreement protocol based on elliptic curve cryptography", IEEE International Symposium on Information Assurance and Security, vol. 57, pp. 89-94, 2007.

[4] L. Adleman, J. DeMarrais, and M. Huang, "A subexponential algorithm for discrete. logarithms over the rational subgroup of the Jacobians of large genus hyperelliptic curves over finite fields", Algorithmic Number Theory (ANTS-1), LNCS 877, pp. 28-40, 1994.

[5] M. Aydos, T. Yanık, and C. K. Koc, "High-speed implementation of an ECC-based wireless authentication protocol on an ARM microprocessor", IEE Proceedings: Communications, vol.148, no. 5, pp. 273–279, 2001.

[6] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks", Proceedings of the IEEE Security and Privacy Symposium, pp. 197-213 , 2003.

[7] D. G. Cantor, "Computing in the Jacobian of a hyperelliptic curve", Mathematics of Computation, vol. 48, pp. 95-101, 1987.

[8] H. Cohen and G. Frey, "Handbook of Elliptic and Hyperelliptic Curve Cryptography", Chapman & Hall/CRC Press, 2006.

[9] K. Chatterjee and D. Gupta, "Evolution of Hyperelliptic Curve Cryptosystems", in proceedings of ICDCIT 2010, LNCS 5966, pp.206-211, Springer -Verlag Berlin Heidelberg 2010.

[10] L. Eschenauer and V. Gligor. "A key management scheme for distributed sensor networks", Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS'02), pp. 41-47, 2002.

[11] G. Frey and H. Ruck, "A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves", Mathematics of Computation, vol. 62, pp. 865-874, 1994.

[12] N. Jansma and B. Arrendondo, "Performance Comparison of Elliptic Curve and RSA Digital Signatures", Technical Report, University of Michigan, 2004. (http://www.nicj.net/files/498termpaper.pdf)

[13] N. Koblitz, "Hyperelliptic cryptosystems", Journal of Cryptology, vol. 1, no. 3, pp. 139–150, 1989.

[14] L. Law, A. Menezes, M. Qu, J. Solinas, and S.Vanstane, "An efficient protocol for authenticated key agreement", Designs, Codes and Cryptography, vol. 28, pp. 361-377, 1998.

[15] T. Landstra, S. Jagannathan, and M. Zawodniok, "Energy-efficient hybrid key management protocol for wireless sensor networks", International Journal of Network Security, vol. 9, no. 2, pp. 121-134, 2009.

[16] A. Menezes, Y. Wu, and R. Zuccherato, "An elementary introduction to hyperelliptic curves", Technical Report

CORR 96-19, Department of C&O, University of Waterloo, Ontario, Canada, November 1996. (http://www.cacr.math.uwaterloo.ca/techreports/1997/tech-reports97.html)

[17] J. Pelzl, T. Wollinger, J. Guajardo, and C. Paar, "Hyperelliptic curve cryptosystems: Closing the performance gap to elliptic curves", Cryptology ePrint Archieve, Report 026, pp. 351-365, 2003. (http://eprint.iacr.org/)

[18] J. Pelzl, T. Wollinger, and C. Paar, "Elliptic & hyperelliptic curves on embedded µP", ACM Transactions on Embedded Computing Systems, vol. 3, no. 3, pp. 509-533, 2004.

[19] H. G. Ruck "On the discrete logarithms in the divisor class group of curves", Mathematics Computation, vol. 68, pp. 805-806, 1999.

[20] L. Uhsadel, A. Poschmann, and C. Paar, "An efficient general purpose elliptic curve cryptography module for ubiquitous sensor networks", Workshop on Software Performance Enhancement for Encryption and Decryption (SPEED'07), pp. 95-104, 2007.

[21] R. Watro, D. Kong, S. F. Cuti, C. Gardiner, C. Lynn, and P. Kruus. "TinyPK: Securing sensor networks with public key technology", Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'04), pp. 59–64, New York, USA, 2004.

[22] L. Yongliang, W. Gao1, H. Yao, and X. Yu, "Elliptic curve cryptography based wireless authentication protocol", International Journal of Network Security, vol. 5, no. 3, pp. 327–337, 2007.

Kakali Chatterjee has done M.Tech from Centre for Development of Advanced Computing, a R&D and Academic Centre of Govt. of India. She is currently pursuing Ph.D. at Delhi College of Engineering (Faculty of Technology), University of Delhi, India. Her field of interest is Information Security and Cryptography.

Asok De received Ph.D. from IIT Kharagpur (India) and his field of interest is Microwave Antennas and Communication Systems. He is Professor in Delhi Technological University (formerly Delhi College of Engineering). Presently he is working as Principal, Ambedkar Institute of Advanced Communication Technologies & Research, Delhi. He has published many research papers in reputed International Journals.

Daya Gupta is a Professor and Head of Computer Engineering Department of Delhi Technological University, India. She has done Ph.D. in Computer Engineering from Delhi University. Her field of interest is Software Engineering, Information Security etc. She has published many research papers in reputed International Journals.