A Novel Proficient Blind Signature Scheme using ECC

Nitu Singh
M.Tech Scholar
Dept. of Computer Science & Engineering
Centurion University of Technology and Management

Sumanjit Das
Assistant Professor
Dept. of Computer Science & Engineering
Centurion University of Technology and Management

ABSTRACT
Internet has become an Integral part and prime need of every human being in their day to day life. People now prefer to do transaction over the internet than doing it offline as it reduces time consumption. In such scenarios, securing customer data is one of the biggest challenges for software developers. For this reason Security protocols like Blind Signature Scheme came into picture where requester is allowed obtain a signature from the signer who signs a message without reading the content of the message, and cannot link the protocol with the resulting message signature pair. Blind Signature scheme is mostly used for E-commerce applications like Digital payment system, Electronic-voting system as it provides security goals like blindness, untraceability, correctness and nonforgeability. Neal Koblitz and Victor S. Miller has presented Elliptic curve cryptography (ECC) which is an approach to public key cryptography based on algebraic structure of elliptic curves in over finite fields. Elliptic curve cryptosystem is more efficient than other public key cryptosystem due to difficulty in solving elliptic curve discrete logarithm problem. Using the benefits of elliptic curve cryptography we have presented implementation of blind signature scheme using JAVA.

Keywords
Cryptography, Digital Signature, Blind Signature, Elliptic Curve Cryptosystem (ECC), E-cash system.

1. INTRODUCTION
Now a day's as the Internet and other forms of electronic communication become more prevalent. While transferring secure data through the internet for the applications like electronic voting system and E-cash, it is important to maintain the integrity of the data. To protect the information present in the electronic document, content of the document need to be hidden from an unauthorized person, protected from the unauthorized change and available to an authorized person when it is needed for him. The implementation of Blind Signature scheme is the most strongly accepted protocol to secure consumer's privacy from an illegal third party. Both sender and receiver hold a private key to encrypt their messages and a public key decrypt their messages. The recipient can receive the correct text message when the two message digests are identical. A Cryptography protocol called Digital Signature is used to provide authentication of the user for various applications they use. A digital signature scheme provides a way for signer to sign messages using his private key so that the signatures can later be verified by anyone else by using public key of signer.

Blind signature scheme was first introduced by David Chaum[1]. A blind signature scheme is a form of digital signature in which the content of a message is disguised before it is signed. This scheme is a two-party protocol between requester and signer. After receiving the signed message from the signer, the requester can derive the valid signature for the message from the signer. Anyone can verify the blind signature using the public key of the signer. If the message and its signature are published, the signer can verify the signature, but he/she cannot link the message-signature pair. This scheme provides Authentication and non-repudiation to the original sign request sent from a requester so as to prevent fraudulent action by the signer [2].

Digital Signature is a method where sender's text messages are encrypted or decrypted through a hash function number in keeping the messages secured when transmitted. When a one-way hashing function is performed to a message, its related digital signature is generated which is called a message digest. A one way hash function is a mathematical algorithm that takes a message of any length as input and generates fixed length of output.

Digital signature consists of two phases: signing phase and verification phase. In signing phase a sender enters his message or data as the input of a one-way hashing function and then produces its corresponding message digest as the output. The message digest will be encrypted by the private key of the sender to produce digital signature. Finally, the sender sends his message or data along with its related digital signature to a receiver. In verification phase the receiver repeats the same process of the sender, letting the message as an input into the one-way hashing function to get the first message digest as output. Then he decrypts the digital signature by the sender's public key so as to get the second message digest. Finally, verify whether these two message digests are identical or not.

In blind signature scheme the signer signs the requester's message and knows nothing about it .Blind signature scheme provides following four properties [1, 2].

Correctness: The correctness of the signature of a message signed through this scheme can be checked by anyone using signer's public key.

Blindness: The content of the message should be blind to the signer that means the signer can't read the content of the message.

Unforgeability: The signature is the proof of the signer, and no one can derive any forged signature and pass verification.

Untraceability: The signer is unable to link the message signature pair even when the signature has been revealed to the public.

A blind signature scheme consists of following phases.

Initialization Phase: All the system parameters of both requester and signer are initialized in this phase.

Blinding Phase: A requester uses his private key to blind the message, so that signer will be blind to the message.

Signing Phase: when the signer receives the blinded message, he generates the blind signature and sends it to the requester.

Unblinding Phase: The sender uses his private key to recover the signer's signature from blind signature.

Verification Phase: Anyone can use signer's public key to identify genuineness of the signature.

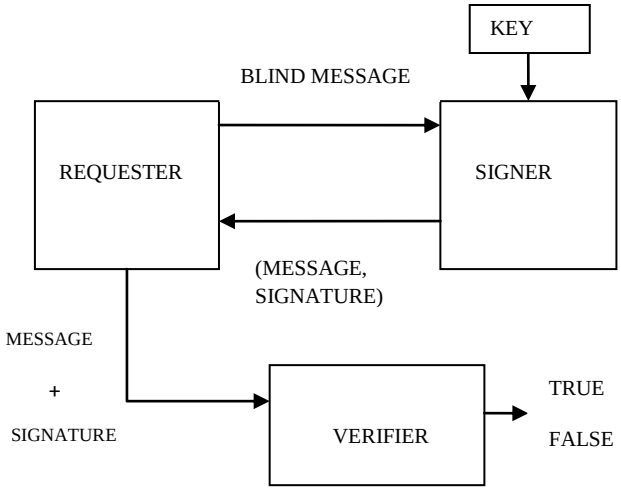

Fig 1: Flow of Blind Signature Scheme.

3. RELATED WORK

In Schnorr Blind Signature scheme Signer selects two points (P_1, P_2) € E (Fp) with q and two random numbers (d1, d2) € Zq to be the secret keys and computes Q_1 and Q_2 [6]. In signing phase signer chooses two numbers (r1, r2) € Zq* securely to compute U= $r_1P_1+r_2P_2$ and send U to the receiver. recipient selects two random numbers and computes R' = (r',Yk) =U +α(P_1+P_2) +YQ and e'= (m,r') and sends e'=e'+Y to the signer . Signer sends $Y_1=r_1+ed_1$ mod q and $Y_2= r_2+ed_2$ mod q to the recipient. On receiving the signature (r', Y1, Y2') the recipient validates it by checking e'=H (m,Xy) Where (Xv,Yv)= Y_1'P_1+Y_2'P_2+e'Q.

In this scheme attackers are infeasible to sign a valid signature (r', Y_1', Y_2') on behalf of the original signer. Both random number (r_1, r_2) and hash function is used which makes computationally infeasible get the duplicate signature. It proves the unlinkability property where signer can't find corresponding signature from the record that he ever signed. Given a valid signature (r', Y_1', Y_2') on message m, it is unable to get another message m' after verifying the signature since it is hard to find e' from hash function which proves untraceable property.

In the paper of Chwei-Shyong et al has proposed E- payment application using blinding signature scheme where (U) wants to withdraw a coin (E-cash) from the bank (B) [9]. In this scheme when a user sends a request to B for withdrawing of E-coin (m), B chooses a random number k, computes R (=kP), and sends R' to U. U computes R (=uR+vP) and e (=H (R‖m)), using secret random value u and v. Then, U calculates the blinded value e (e=e/u) and sends it to B. B uses his/her private key to generate a blind signature S' (=Xbe+k) for e' and sends it to U. U un-blinds B's signature S' by using u and v (i.e., S= S'u+ v), and verifies S by checking the (SP= eQb + R) where Qb is a public-key of the bank. If the equation holds, U obtains a valid E-cash.

Here e' is generated by the concatenation of R and m with a hash function H (.) which satisfies the blindness property. The signer signs a message without knowing its contents. Blindness is the first important property in a blind signature. In this scheme, the requester calculates R = uR' + vP, and generates e' which is a concatenation of R and m with a hash function H (.). Then, he/she sends them to the signer. Hence, the signer cannot know the message m. No one can forge (m, R, S) because the elliptic curve discrete logarithm problem is difficult to solve and also if anyone obtains the valid signature, he/she cannot link this signature to the message. This scheme satisfies both unforgeability and untraceability property but unable to satisfy correctness property.

Paper suggested by Debasish et. al. a blind signature scheme based on ECDLP [10]. In this scheme private key d_B (a randomly selected in the interval [1, n-1]) and Q is the public key of signer computed as $Q = d_B G$. The signer randomly chooses k_1, k_2, l_1 and l_2 and calculates r_1 and r_2 as R=kG= (xr_1, yr1) and $r_1=xr_1$ mod n and r2=xr_2 mod n.

After receiving, the (R_1,R_2,l_1,l_2) requester randomly select four integers (a,b,w and z), and computes R_1 = R_1w a $l_1$and R_2 = R_2 z b l_2 . Requester blinds the message m as m 1= e m r_1 r_1^{-1} r_2^{-1} a^{-1} mod n, m2 = e m r_2 r_1^{-1} r_2^{-1} b^{-1} mod n and send to signer.

The signer computes $S_1=d_B$ m_1 - $r_1k_1l_1$ mod n and $S_2=d_B$ m_2 - $r_2k_2l_2$ mod n and send to requester. The requester extracts the actual signature as $S_1=S_1$ r_1^{-1} r_1 r_2 w a mod n and S2=S_2 r_2^{-1} r_1 r_2 w a mod n. Any one can verify the legitimacy of the digital signature (R, r, s) of message m by using mQ = sG + rR .

As requester randomly select six blinding factors (b_x ,a_x ,z_x ,d_x ,w_x ,e_x) to compute the blind message so from the blind messages the signer can not compute the original message as it is based on ECDLP. Signer can't trace the blind signature without knowledge of six blind factors of requester. This scheme is secure, robust and untraceable but can't be implemented for online applications.

Fuh-Gwo et al proposed an ECC based blind signature scheme which hold two fundamental properties, blindness and intractableness [12]. In this scheme using the base point G = (x, y) on E_p(a, b) Requester R choose his private key n_i and computes public key $P_i \equiv n_i \times G \pmod p$. Similarly signer choose his private key n_s and computes its public key as Ps =n_s×G (mod p). Requester R, blinds message m as $\alpha \equiv m \times (n_i \times P_i) \pmod p$, and sends α to the signer.

Signer signs α by randomly selecting a number n_v, and checks whether (α, n_v) in his database. Then he computes $r \equiv n_v \times \alpha \pmod p$ and $s \equiv (n_v + n_s) \times \alpha \pmod p$ and send (α, (r, s)) to Requester R. Requester R strips s in (r, s) by applying his own secret key and the public key P_s of the signer to yield $s' \equiv s - m \times n_i \times Ps \pmod p$. And then requester R computes $m' = n_i (n_i - 1) \cdot m$. Anyone can use the signer's public key Ps to verify the authentication of the signature (m′, s′, r) by checking $r \equiv s' - m' \times P_s \pmod p$.

No one else can copy the signed matter (*r, s*) as the signer's so as to pass the verification phase because copying a signed matter is equivalent to solve the elliptic curve discrete logarithm problem which is very difficult. The signer is unable to derive the message *m* without the value *ni* from the

blind message which satisfies blindness property. The signer also has no ideas on the correspondence between the signed matter *s* and the stripped signed matter *s'* since *s* and *s'* are two distinct points on the elliptic group *Ep*(*a, b*). The signer also cant trace *s'* from *s* since *s'* is yielded by applying the secret key *ni* of requester *R*

Dhanashree and Avinash scheme is based on ECC using Zero knowledge protocol [15]. Here the points on Elliptic curve are generated; a base point is selected out of them of order n. The private key 'k' is chosen randomly between 1 to (p-1), where p is a large prime number. The private key 'k' is treated with the base point to compute the public key. This scheme uses three blinders where sender uses his private key to encrypt the message and send it to one of arbitrarily selected blinder. This blind message again blinded by remaining two blinders. The last blinder send the encrypted message back to the sender and process goes on reverse direction. Now the sender sends blinded message to the signer.

Signer uses zero knowledge concepts for both calculation and verification. Signer asks sender the value of e (=0, 1), to verify the message to be same as sender's message. The sender chooses a random number r and calculates r^2 mod p and send to the verifier V. The signer asks for either value of [(rk) mod p / r]. Sender calculates $Y=rk^e$ mod p where e= (0, 1) .If e=1, the signer receive rk mod p. The verifier checks Y^2 mod p which is same as r^2k^2 mod p. The verifier has received r^2 and selects Vk. If e=0, the verifier receive r and checks with r^2 mod p. After verification signer sign the message using his/ her private key and send it to sender. Sender unblinds the message using his own public key and the public key of the blinder. Blind digital signature can be verified by any verifying authority using public key of signer.

4. ELLIPTIC CURVE CRYPTOSYSTEM

In 1985, Elliptic Curve Cryptography (ECC) was proposed by Neal Koblitz and Victor Miller [3, 4]. ECC is capable of improving the existed cryptogram systems in terms of having smaller system parameter, smaller public-key certificates, lower bandwidth usage, faster implementations, lower power requirements, and smaller hardware processor requirements [5].Vanstone states in his paper "the elliptic curve discrete logarithm problem is believed to be harder than both the integer factorization problem and the discrete logarithm problem modulo p" [5]. Therefore, using ECC to build a cryptosystem is commendable by the reasons of high security and efficiency. The elliptic curves can be categorized into two classes: non prime and prime elliptic curves .The elliptic curve cryptography is based on the elliptic curve equation which is given as: $y^2 = x^3 + ax + b$. To plot an elliptic curve one needs to compute: y mod p= sqrt($x^3 + ax + b$) mod p. So, value of y is calculated for each value of x, symmetric about y = 0 where values of a and b will be given. Groups are defined based on the set E (a, b) for values of a and b such that: $4a^3 + 27b^2 \neq 0$.

Prime curves *(Zp)* are good to use in software application, because it does not require extended bit-fiddling operation, which binary curves require. Binary curves (*GF* (2^n)) are best for hardware application as it requires a few logic gates to build a powerful cryptosystems. Secondly, the variables and coefficients of the elliptic curves are limited to the elements of a finite field. Because of this limitation, it would increase the efficiency of ECC computing operation. There are some rules for operation addition '+'for elliptic curve points to follow. Those all are listed down as [11]:

1) $O + P = P$ and $P + O = P$, where O serves as the additive identity.

2) $-O = O$.

3) $P + (-P) = (-P) + P = O$, where $-P$ is the negative point of P.

4) $(P + Q) + R = P + (Q + R)$.

5) $P + Q = Q + P$.

Adding distinct point P and Q
For any two points P= (Xp, Yp) and Q= (Xq, Yq) over Ep (a,b), the elliptic curve addition operation, which is denoted as

P + Q = R = (Xr, Yr)

$Xr = (S^2 - Xp - Xq) \bmod p$

$Yr = (S(Xp-Xr) - Yp) \bmod p$

Where $S = (Yq-Yp) / (Xq-Xp) \bmod p$

Doubling distinct point P
If P=Q, then the point addition operation is known as point doubling in ECC. For points P= (Xp, Yp) the elliptic curve doubling operation, which is denoted as

2P=P+P=R= (Xr, Yr)

$Xr = (S2-2Xp) \bmod p$

$Yr = (-Yp+S*(Xp-Xr)) \bmod p$ where,

$S= ((3Xp^2 +a) / (2Yp)) \bmod p$

Matlab simulation
Fig 1 shows the simulated result for the elliptic curve. This figure represents all the points generated by elliptic curve. Here where a= 121, b= 151, p= 751. For both x and y axis we have taken the range between 5000 to 6000.

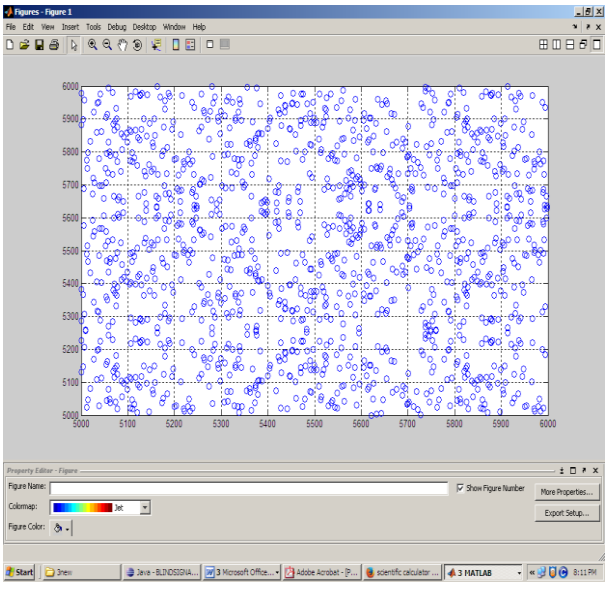

Fig 2: All possible points on ECC

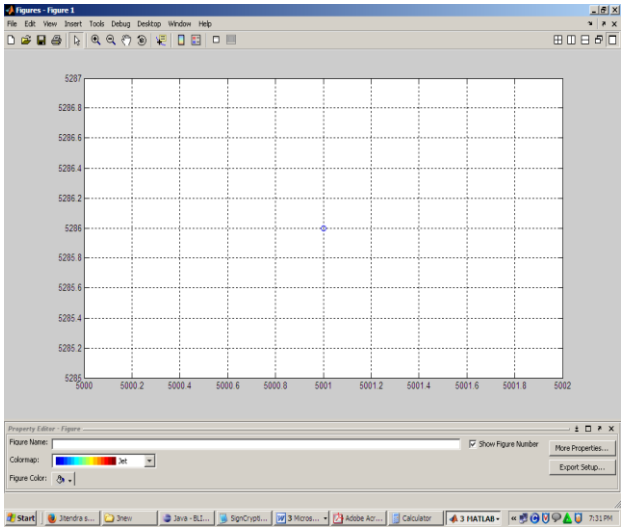

Fig 3: Randomly selected single point on ECC

5. PROPOSED ALGORITHM

Initialization and Key pair generation Phase:

Signer defines Elliptic curve domain parameters 'E'. User randomly select integer 'K' and calculate

$Q=K*G = (X_1,Y_1)$ ----------eq(1)

$r_1 = X_1 \pmod p$ ----------eq(2)

$r_1 \neq 0$ ----------eq(3)

If eq (3) holds then signer sends elliptic curve point 'Q' to the requester else signer selects another 'K' and repeat above process.

Signer generates his private key randomly selecting Integer 'A' between [1, p-1] and calculates Public key 'B' as

$B = A *G$ ----------eq(4)

Blinding Phase

Message m is blinded using following steps

1. Requester randomly select two integers 'M' and 'N' in the range [1, p-1].

2. Requester calculates elliptic curve point 'R'

$R = M* Q +N*G = (X_2, Y_2)$ ----------eq (5)

5- Requester calculates 'r_2' from 'R' given in eq (5)

$r_2 = X_2 \pmod p$ ----------eq (6)

6-requester blind message 'm' and send it to the signer.

$m' = M\ H(m)\ r_1\ r_2^{-1} \pmod p$ ----------eq(7)

Where 'H' is the hash function. In this scheme SHA-1 algorithm is used as the hash function.

Signing Phase

Signer generates blind signature S'

$S' = A*r_1 + K * m' \pmod p$ ----------eq (8)

Unblinding Phase

Requester checks whether r_1 and S' are in the range [1, p-1]. If so then requester retrieves (S, R) on message 'm'.

$S = S'r_2\ r_1^{-1} + NH(m) \pmod p$ ----------eq (9)

Verification Phase

Any party having elliptic domain parameters 'E' of the signer can verify digital signature (S, R) on message 'm' using following equation.

$SG = r_2*B + H(m)\ R$ ----------eq (11)

Correctness proof of proposed scheme

Expanding eq (11)

SG

$= S'\ r_2r_1^{-1}\ G + N\ H(m)\ G$

[Replacing value of S from eq (9)]

$= A\ r_1\ r_2r_1^{-1}\ G + K\ m\ r_2r_1^{-1}\ G + N\ H(m)\ G$

[Replacing value of S' from eq(8)]

$= A\ r_1\ r_2r_1^{-1}\ G + K\ M\ H(m)\ G\ r_1\ r_2^{-1}\ r_2r_1^{-1} + N\ H(m)\ G$

[Replacing value of m' from eq(7)]

$= Ar_2G + K\ M\ H(m)\ G + N\ H(m)\ G$

[We know $r_1r_1^{-1} = 1 \pmod p$ and $r_2\ r_2^{-1} = 1 \pmod p$]

$= Ar_2G + H(m)\ M\ Q + H(m)\ N\ G$

[Replacing value of KG with Q from eq(1)]

$= Ar_2G + H(m)\ [M\ Q + N\ G]$

[Replacing with R from eq (5)]

$= r_2 * B + H(m)\ R$ [Replacing AG with B from eq(4)]

Abbreviation	Interpretation
E	Elliptic curve domain parameters
G	Generator point
a , b	Coefficient defining Elliptic curve
p	Order of G, a prime number
A	private key of signer
B	public key of signer
Q , R	Points on the Elliptic curve
K , M ,N	Random integer numbers
r_1	X –coordinate of Q
r_2	X –coordinate of R
X,Y	Coordinates for the Cartesian system
H(:)	Hash value
m'	Blind message
m	message
S	Signature
S'	Blind signature

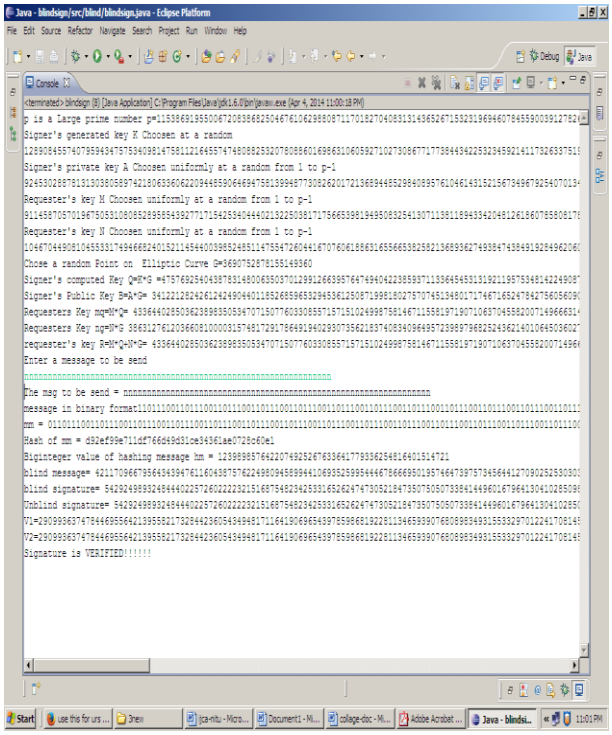

Fig 4. Out put of implemented scheme

6. SECURITY ANALYSIS

The security of our scheme is based both on the strength of the hash function and the difficulty of solving ECDLP. We used hash function on message m before signing. It is very easy to find hash (m) for any arbitrary length message. But it is very difficult to find two messages m and m' for which hash (m) = hash (m'). Our scheme has achieved all the four security properties of blind signature scheme.

Blindness: Our scheme satisfies the blindness property and signer knows nothing about the original message. m' is computed using a one-way hash function. As hash values are non-invertible, from value of m', the value of m can't be computed.

Correctness: In the verification phase any third party can check the correctness of the signature of a message signed through the signature scheme by using the signer's public key. Here B is signer's public key which is used in eq (11) of verification phase. Correctness of the signature can be verified from eq(12) and eq(13)as both gives same expression.

Unforgeability: Only the signer can give a valid signature for the associated message. In the proposed scheme it is computationally infeasible for an attacker to pretence an honest sender in creating an authentic blind signature for the requester.

Untraceability: The signer of the blind signature is unable to link the message-signature pair even when the signature has been revealed to the public. Requester use two randomly selected integers for computing R and from R computes r_2 to blind the message which signer can't know. So we can't trace blind message.

Table 1. Security level at Signer side using point on ECC

	Chang et al.	Tsai et al	Debasish et al.	Fuh Gwo et al.	Dhanshree & Avinash	Proposed scheme
Signer	2	2	3	1	1	2
Requester	0	0	0	1	1	1

Table 1 show numbers of time signer and requester have used elliptic curve point in both sides. In our proposed scheme signer has used elliptic curve point two times and requester has used it once only. As the signer use elliptic curve point two times which makes signer side calculation more secure.

Table 2. The storage requirement of other blind signature scheme with proposed scheme in bits.

	Schnorr Blind Scheme	Chang et al.	Proposed Scheme
System parameters	1024*3= 3072	160*5=800	160*4=640
Public key	1024*1= 1024	(160+1)*2= 332	160*2=320
Private key	160*1= 160	160*2=320	160*1=160
Total bits	4256	1442	1120

Chang et al have used only 800 bits for parameters a, b, p and q in ECCs. But our proposed scheme the storage space is only 640 bits for system parameters Q, B, R and V_1. The key pair take (1024+160) = 1184 bits in discrete logarithm cryptosystems. In Chang et al. scheme the key pair take ((160+1)*2 + 160*2) =642 bits in ECCs. In our proposed scheme key pair take only ((160*2) +160) = 480 bits. Totally the proposed scheme takes (1120/4256) 26% of Schnorr blind scheme in the storage requirement.

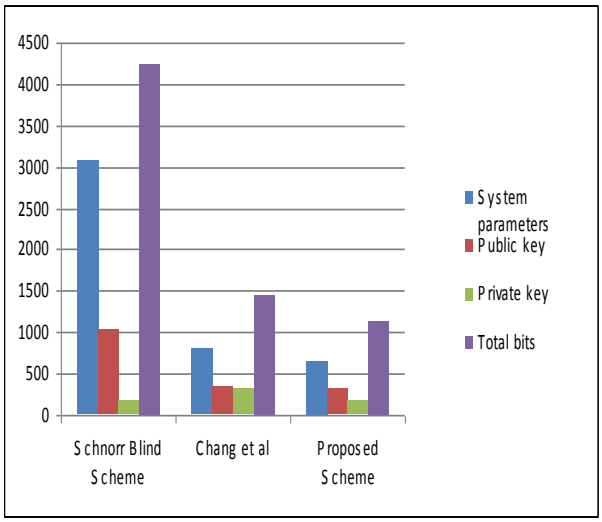

Fig 4: Storage requirement analysis

Table 3: The comparative performance of other schemes with proposed scheme based on modular multiplication

	Schnorr scheme	Chang et al	Debasish et al	Proposed scheme
Signature Initialization	1	2	2	1
Signing	5	7	9	4
Signature Verification	2	2	0	0

Table 3 shows comparison table of number of modular multiplication used in proposed scheme with other schemes. Schnorr scheme has used (1+5+2) =8 modular multiplications. Chang et al. have used (2+7+2) = 11 modular multiplications. Debasish et al. have used (2+9) = 11 modular multiplications. But our proposed scheme has only (1+4) = 5 modular multiplications.

Table 4: Speed up ration of proposed scheme with other schemes based on modular multiplication

	Schnorr scheme	Chang et al	Debasish et al	Proposed Scheme	Speed up ration of
Signature Initialization	**240*1=240**	**29*2=58**	**29*2=58**	**29*1=29**	**(240/29)==8**
Signing	**240*5=1200**	**29*7=203**	**29*9=261**	**29*4=116**	**(1200/116)==10**
Signature Verification	**240*2=480**	**29*2=58**	**0**	**0**	**N\A**

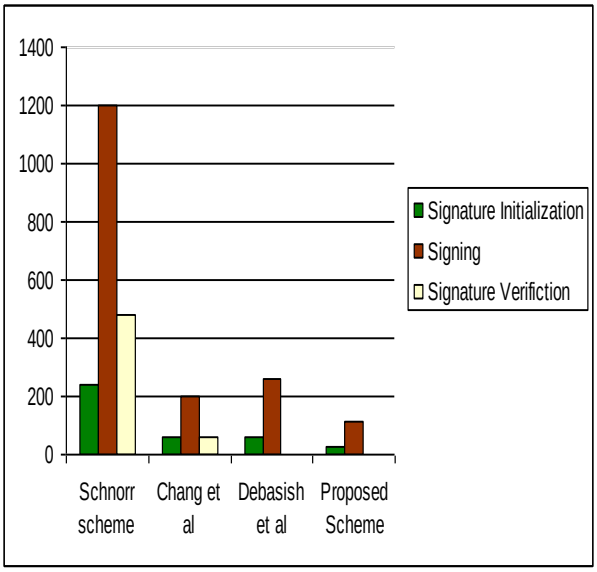

Fig 5: Comparative performance analysis based on modular multiplication.

6.1. E-cash System using Proposed Algorithm:

Step 1: When a user (U) wants to withdraw a coin (E-cash) from the bank (B), he sends request to bank.

Step 2: Bank choose random number K and computes (Q = KG =(X_1,Y_1)) and checks whether $r_1 = X_1 \bmod p \neq 0$. If not choose another K. Bank sends Q to the user.

Step 3: Bank randomly select Integer 'A' between [1, p-1] and computes its public key as B = A *G.

Step 4: User randomly select two integers 'M' and 'N' in the range [1, p-1] and computes R = M* Q +N*G = (X_2, Y_2), also checks $r_2 = X_2 \bmod p$. User calculates the blinded value as $m = M\,H(m)\,r_1\,r_2^{-1} \pmod p$

Step 5: Bank calculates the Blind Signature $S' = A*r_1 + K * m \pmod p$ and send to user.

Step 6: User unblinds the signature by computing $S = S' r_2 r_1^{-1} + NH(m) \pmod p$ and verifies the signature by computing $SG = r_2*B + H(m) R$. If the statement holds user can get valid E-cash.

7. CONCLUSION

In this paper we have shown implementation of blind signature scheme using advantageousness of elliptic curve cryptosystem. This scheme has proved itself to be more secure and computationally faster as compared to other schemes as the speed up ration of the proposed scheme shows high speed up ration. This scheme is also more efficient in terms lowering storage requirements and computational overhead, which is due to the use of ECC and it also, offers smaller key lengths for desired security levels. The high speed cryptographic process of this scheme, leads to low-complexity hardware and software requirements. Our scheme fulfils all the four security properties of blind signature scheme like correctness, blindness, untraceability and unforgeability . This scheme can be implemented in online process and it is more helpful for voting process and electronic payment system. This scheme can be improved using hyperbolic curve in future to make it more effective in terms of computational cost and communication overhead.

8. REFERENCES

[1].D.Chaum,"Blind signatures for untraceable payments," *Advances in Cryptology, CRYPTO'82*, pp. 199-203, 1982.

[2] D. Chaum, "Blind signatures system," *Advances in Cryptology, CRYPTO'83*, pp. 153-156, 1983.

[3] V.S.Miller, "Use of Elliptic Curves in ryptography," Advances in Cryptology: Proceedings of Crypto '85, vol. 218, pp. 417-426, 1986.

[4] N. Koblitz, "Elliptic Curve Cryptosystems," Mathematics of Computation, vol. 48, no. 177, pp. 203- 209, 1987.

[5] S.A.Vanstone, "Elliptic curve cryptosystem-the answer to strong, fast public-key cryptography for securing constrained environments," *Information Security Technical Report*, vol. 2, no. 2, pp. 78-87, 1997.

[6] Ming-Hsin Chang, "Schnorr blind Signature based on Elliptic curve", Asian Journal of Information Tecnology, 2(3), pages 130-134, 2003

[7] M. Hwang, C, Lee and Y.Lai, "An untraceable blind Signature scheme," in IEICE Trans. Fundamentals, Vol. E86-A, No 7, pp.1902-1906, 2003.

[8] Hwang Lai Su, An efficient signcryption scheme with forward secrecy based on elliptic curve, Journal of applied mathematics and computation, pages 870-881, 2005.

[9] Chwei-Shyong Tsai, Min-Shiang Hwang,Pei-Chen Sung, "Blind Signature Scheme Based on Elliptic Curve Cryptography"

[10] Debasish Jena, Sanjay Kumar Jena and Banshidhar Majhi "A Novel Untraceable Blind Signature Based on Elliptic Curve Discrete Logarithm Problem" IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.6, June 2007

[11] K. H. Huang, Y. F. Chung, C. H. Liu, F. Lai, and T. S. Chen (2007), "Efficient migration for mobile computing in distributed networks," Computer Standards & Interfaces, 2007.

[12] Fuh-Gwo Jeng, Tzer-Long Chen, Tzer-Shyong Chen "An ECC-Based Blind Signature Scheme" journal of Networks, VOL. 5, NO. 8, pp.921-928, August 2010.

[13] C. W. Shieh (2006), "An Efficient Design of Elliptic Curve Cryptography Processor," Master Thesis, Tatung University, Taipei, 2006.

[14] Y. F. Chung, H. H. Lee, F. Lai, and T. S. Chen (2008), "Access control in user hierarchy based on elliptic curve cryptosystem," *Information Sciences*, vol. 178, no. 1, pp. 230-243, 2008.

[15] MS.Dhanashree M.Kuthe, Prof. Avinash J. Agrawal "Implementation of Blind digital signature using ECC and Zero knowledge protocol" International Journal of Scientific and Research Publications, Volume 2, Issue 9, September 2012.

[16] MS.Dhanashree M.Kuthe, Prof. Avinash J. Agrawal "Implementation of Blind digital signature using ECC" International Journal of Computer Science and Network (IJCSN) Volume 1, Issue 5, October 2012.

[17] Sumanjit Das and Prasant Sahoo "Cryptoanalysis of signcryption protocols based on Elliptic curve" IJMER, Vol-3, No.2, Feb 2013.