ISSN 2347-4289

An Efficient Blind Digital Signature Protocol Based on Elliptic Curve

Sumati Baral

Department of Computer Science and Engineering,
Trident Academy of Creative Technology, Bhubaneswar, Odisha, India
Email: sumatibbsr2010@gmail.com

ABSTRACT: Blind Signature is an addendum of Digital Signature. It is a two party protocol in which a requester sends a message to a signer to get the signature without revealing the contents of the message to the signer. The signer puts the signature using his/her private keys and the generated signature can be verified by anyone using signer's public keys. Blind signature has a major property called as untraceability or unlinkability i.e after the generation of the signature the signer cannot link the message-signature pair. This is known as blindness property. We have proposed blind signature scheme and its variation based on Elliptic Curve Cryptography (ECC) in which major emphasis is given on the untraceability property. We have cryptanalyzed Carmenisch et al.'s blind signature scheme and Lee et al.'s blind signature scheme and proposed an improvement over it. It is found that, the proposed scheme has less computational complexity and they can withstand active attacks.

Keywords: Digital Signature; ECC; DLP; IFP; BDS

1 INTRODUCTION

Now a day's transaction over the internet has been increased vastly. When data are transmitted through the Internet, it is better that the data are protected by a cryptosystem to prevent them from tampering by an illegal third party. Basically, an encrypted document is sent, and it is impossible for an unlawful party to get the contents of the message, except he gets the sender's private key to decrypt the message. Under a mutual agreement between the senders and receivers, each sender holds a private key to encrypt his messages to send out, and a public key used by the receiver to decrypt his sent-out messages. When the two message digests are verified to be identical, the recipient can have the true text message. Thus, the security of data transmission can be made sure. User need to be authenticated for many of the application they use. This service can be achieved by the cryptography protocol called Digital Signature. 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 is a special form of digital signature, which was first introduced by David Chaum 1982. In a blind signature scheme, a signer signs a message without knowing the contents of the message. The message is blinded by a requester. After receiving the signed message from the signer, the requester can derives 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. Blind signatures are widely used in many important cryptographic services, especially in those services that emphasize the privacy of users such as electronic voting over Internet and untraceable payment services. Basically a bind signature scheme is a protocol for a group of requesters and a signer. Each requester sends an encrypted message to the signer and obtains a valid signature from him. Note that the signer only signs the message and does not decrypt it. Later, the signer can verify the genuineness of the signature whenever he receives the message-signature pair; however, he cannot link the message-signature pair to the particular phase of the signing protocol that has led to this pair.

1.2 Digital Signature:

The digital signatures are used in private communication, where customer privacy is main object. All messages are capable of being encrypted and decrypted so as to ensure the integrity and non-repudiation of them. The concept of digital signatures originally comes from cryptography, and is defined to be a method that a sender's text messages are encrypted or decrypted through a hash function number in keeping the messages secured when transmitted. Especially, 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 makes a message of any length as input, but of a fixed length as output. Because its one-way property, it is impossible for the third party to decrypt the encrypted messages. Two phases of the digital signature process is described in the following.

1.2.1 Signing Phase: A sender firstly makes his message or data as the input of a one-way hashing function and then produces its corresponding message digest as the output. Secondly, the message digest will be encrypted by the private key of the sender. Thus, the digital signature of the message is done. Finally, the sender sends his message or data along with its related digital signature to a receiver.

1.2.2. Verification Phase: Once the receiver has the message as well as the digital signature, he repeats the same process of the sender does, 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.

1.3 Properties of Blind Signature:

The signer signs the requester's message and knows nothing about it; moreover, no one knows about the correspon-

dence of the message-signature pair except the requester. Blind signature scheme should satisfy following properties.

- **Correctness:** The correctness of the signature of a message signed through the signature scheme can be checked by anyone using the signer's public key.
- **Unforgeability**: only the signer can give a valid signature for the associated message.
- **Blindness**: The content of the message should be blind to the signer; the signer of the blind signature does not see the content of the message.
- **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.

1.3.1 Phases of blind signature scheme

- **Blinding phase:** A sender firstly chooses a random number called a blind factor to mess his message such that the signer will be blind to the message.
- **Signing phase:** When the signer gets the blinded message, he directly encrypts the blinded message by his private key and then sends the blind signature back to the sender.
- **Unblinding phase:** The sender uses his blind factor to recover the signer's digital signature from the blinded signature.
- **Signature verification phase:** Any one uses the signer's public key to verify whether the signature is genuine.

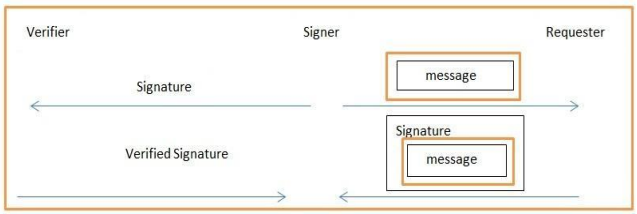


Figure 1.1: Operation of a Blind Signature Scheme

1.3.2 Applications of Blind Signature

E-cash:
E-cash was introduced by David Chaum as an anonymous cash system [10].It is interesting to know that e-coins are blind signatures. e-cash is a three party protocol, in which a customer or the requester requests for money withdrawal to his/her bank or the signer for buying products from the merchant [11, 12].The signer verifies the authenticity of the requester and then sends signed tokens to the requester. The requester sends the tokens to the merchant and the merchant give the token to the bank for verification of the tokens. So we can see one transaction can give one valid token packet or one valid sig-nature. For multiple transactions the corresponding signatures or the e-coins will be different. But, nowadays many requesters become malicious and spend the e-coins for multiple times. This is known as the double spending problem. Though blind signature provides untraceability or unlinkability but sometimes it is necessary to reveal the identity of the requester. To do so one requester should not blind all the internal structure of the message. It should blind the outer part of the message so that by using the public parameters the signer can able to trace the identity of the malicious requester. This is kind of blind signature is known as restrictive blind signature.

E-voting:
In e-voting system [1, 8, 13–15], a voter first registers himself/herself in a voting system and then sends the blinded vote to the voting system. The voting system then sends the vote to the ballot system. There it is verified whether the voter is a registered or valid voter or not. If yes then the ballot center gives its signature on the vote envelope and sends it to the counting system. So the ballot system here gives his signature on the vote envelope without knowing the contents of the envelope. This shows the blindness property. And when the vote is being disclosed the ballot system will unable to link the signature and the vote to a particular instance. This shows the untraceability or unlinkability property of the blind signature.

1.3.2 Variations of Blind Signature

Restrictive Blind Signature:
Restrictive blind signature means that a requester can blind the documents but with some restrictions. It is a protocol which says that any user can request for a blind signature on a document form a valid signer. But it has certain limitations as compared to the normal blind signature. Like normal blind signature the user can blind the message in any way but the choice of the message is restricted and must follow certain rules so that the original message and the blinded message are isomorphic. [4,5,16,17] The blind signature ensures that the signature generated by the signer for one transaction can only be used once. But if the requester becomes malicious and tries to replay the signature again after some time duration then the identity of the requester should be revealed. This can be done by applying restrictive blindness to the normal blind signature scheme. Revocable Anonymity: In any communication, protecting the contents is not enough. Sometimes it is required to keep the identity of the recipient as private. In the context of electronic commerce, If no anonymity is provided then the users preferences can be known. With this information anyone can know the profile of users and send them targeted advertisements or can sell the profiles to other commercial units. The buyer will get problem by this as they want to do the transactions anonymously. Blind signature allows a user to do any transactions anonymously. But in case of any legal disputes e.g money laundering, the identity of the malicious user needs to be revealed. This is known as revocable anonymity i.e to revoke the anonymity when needed [18,19,21,22].

Fair Blind Siganture:
Though it is another variation of blind signature, it can be obtained from the restrictive blind signature also.In a fair blind signature protocol a single trustee or multiple trustees may get involved in the system.It is also used to revoke the anonymity of malicious users and the trustte used to do that. To do so,the trustee view all the parts of the blinding process [13,23]. For this reason the trustee need to be remain online all the time, which compromises the efficiency of the system. Later many fair blind signatures [14, 15] are

developed in which the trustee need to keep a public-private key pair. The trustee can only involved in the tracing protocol and by using the key pairs he can trace the identity of the malicious user.

Partial Blind Signature:
To achieve revocable anonymity, another variation of blind signature called as par-tial blind signature is also used [5,17].To trace the identity of the malicious user, the signer need to keep some data in the database during the transaction. This will increase the space of the database. When the requester tries to use the signature twice, the signer checks the database to identify that requester. But to search the database each time is not so feasible. Partial blind signature overcomes this problem. In a partial blind signature protocol, the signer and the requester have some common agreed information. The requester can blind the message but the common agreed information need to be remaining unblind. By using the common information the signer can trace the identity of the requester when needed. The concept of partial blind signature was developed by Abe and Okamato [17].

1.4 Motivation

The motivation for this project came from the growing need for a full proof signature verification scheme which can assure untraceability property, conditional anonymity, and maximum possible security from the existing schemes. The idea be-hind the project is also to confirm that the proposed scheme can provide comparable results and if possible better performance than already proposed signature verification schemes.

1.5 Related Work

1.5.1. "Blind Signature Scheme Based on Elliptic Curve Cryptography" [2]

This article, propose a new blind signature which is based on ECC and consists of following parameters.
Xs: private key of the signer
Qs: public key of the signer
k: randomly chosen number by the signer
u, v: randomly chosen number by the requester
m: message which the requester wants to blind
H(.): a collision-free hash function
P: a generator point in ECC

Working procedure:
Here given an e-Payment application to indicate the effective of the proposed scheme. If a user (U) wants to withdraw a coin (E-cash) from the bank (B). The procedures of u proposed scheme works as follows:
1. U sends a request to B for withdrawing of E-coin, m.
2. B chooses a random number k, computes R' (=kP), and sends R' to U. After receiving R', 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.
3. B uses his/her private key to generate a blind signature S' (=Xbe'+k) for e' and sends it to U. Here Xb is B's private key.
4. U un-blinds B's signature S' by using u and v (i.e., S= S'u+ v), and verifies S by checking the equations: SP= eQb + R, where Qb is a public-key of the bank. If the equation holds, U obtains a valid E-cash.
U stores the E-cash S to a diskette or smart card. When the user U wants to purchase merchandise over Internet, he/she sends the E-cash to the merchant. The merchant verifies the E-cash whether legal one or not by checking the equations: SP= eQb + R. If the equation holds, the merchant obtains a valid E-cash. Security Analysis This section shows that this scheme preserves all the characteristic of a blind signature.

Blindness:
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.

Unforgeability:
No one can forge (m, R, S) because the elliptic curve discrete logarithm problem is difficult to solve. Assume three situations as follows. Situation 1: If someone tried to fake R1, m1, he/she cannot obtain S1. Because S1P= e1Qs+ R1 and S1 is unknown. It is an elliptic curve discrete logarithm problem and difficult to solve. Situation 2: If someone gets S1, m1, he/she cannot obtain R1. Because S1P = e1Qs + R1, R1 is unknown, and e1 = H(R1||m1). It is also an elliptic curve discrete logarithm problem and difficult to solve. Situation 3: If someone tries to fake R1 and S1, he/she cannot obtain m1. Because S1P = e1Qs + R1, he/she cannot get e1 without m1. It is an elliptic curve discrete logarithm problem and is difficult to solve.

Untraceability
If anyone obtains the valid signature, he/she cannot link this signature to the message. In this scheme, if the signer keep a record set (ki, R'i, e'i, S'i), where i= 1, 2, …, n, he/she cannot trace the blind signature. Expand this as follows. When the requester reveals n records (mi, Ri, Si) to the public, the signer will compute the values ei and u', and obtain Si and Ri, where ei = H(Ri || mi), and u'=et / et' However, the signer cannot trace the blind signature by detecting whether each Ri and Ri+1 have the same relation. Therefore, the signer cannot trace the blind signature. Merit: It satisfies Blindness, Unforgeability, Untraceability-Demerit: Unable to find correctness.

1.5.2 "A Novel Untraceable Blind Signature Based on Elliptic Curve" [3].

The initialization phase:
Let (q,FR,a,b,G,n,h) are the curve parameter, dB (a randomly selected in the interval [1,n-1]) and Q are private and public key of signer, respectively. Where Q = dB G which is made public. The signer randomly chooses k1, k2, l1 and l2 and calculates r1 and r2 as follows:
R 1 = k 1 G
R 2 = k2 G
R 1 = (xr1, y r1)
R 2 = (x r2 , y r 2)
r 1 = Xr1 mod n and r1 != 0
r 2 = Xr2 mod n and r2 != 0

The signer sends (R1, R2, l1, l2) to requester.

The requesting phase:
After receiving, the (R1,R2,l1,l2) requester randomly select four integers a,b,w and z such that w is relatively prime to z i.e. gcd(w,z)=1. According to Extended Euclid's algorithm there exist two integer e and d such that dz+ ew =1. Signer secret values are (b, a, z, d, w, e) .The requester computes R1 and R2 as,
R1 = R1w a l1, R1= (xr1,yr1)
R2 = R2 z b l2 , R2 = (xr2,yr2)
r1 = xr1 mod n r2 = xr2 mod n
After calculating r1 and r2 the requester blinds the message m as follows
m 1= e m r1 r1-1 r2-1 a-1 mod n , m2 = e m r2 r1-1 r2-1 b-1 mod n
The requester sends the blind messages m1 and m2 to signer for signature

The Signing Phase:
In this phase, the signer computes blind signature S1 and S2 by using received blinds messages m1 and m2 as follows
S1=dB m1 - r1k1l1 mod n
S2=dB m2 - r2k2l2 mod n
Then the signer sends the blind signatures s1 and s2 to the requesters.

The extraction phase:
After receiving the blind signatures S1and S2 the requester extracts the actual signature as follows,
S1=S1 r1-1 r1 r2 w a mod n
S2=S2 r2-1 r1 r2 w a mod n
S= S1+S2
R=R1+R2
r = (r1*r2) mod n
The pair (r, s) are the valid digital signature of message m.

The Verifying phase:
Anyone can verify the legitimacy of the digital signature (R, r, s) of message m by using
mQ = sG + rR
Merits: secure, robust and untraceable.
Demerits: Can be implemented for off-line applications

1.5.3. "An ECC-Based Blind Signature Scheme" by Fuh-Gwo Jeng, Tzer-Long Chen, Tzer-Shyong Chen [4]
In this scheme, an elliptic group Ep(a, b) is formed as y2 = x3+ax+b (mod p), where 4a3+27b2 ≠ 0 mod p such that the elliptic group Ep(a, b) is proper for cryptography. And then a base point G = (x, y) on Ep(a, b) is determined whose order is a very large value n such that n× G = O. Two parties, namely a group of requesters, {Ri | 1≤ i ≤ n, n € N}, and a signer, are the participants in this blind signature scheme. For requester Ri, he randomly selects a secret key ni € Zp, and generates his corresponding public key Pi ≡ ni×G (mod p). its working procedure is described in the following .Requester Ri holds message m, forms α ≡ m × (ni × Pi) (mod p), and sends α to the signer. Signer signs α by randomly selecting a number nv, and checks whether (α, nv) in his database. If yes, signer selects a distinct number nv for the in-coming identical blinded message. Then he computes r ≡ nv × α (mod p) and s ≡ (nv + ns) × α (mod p) and returns the message-signature pair (α, (r, s)) to Requester Ri. In case, he keeps (α, nv) in his database. Requester Ri strips s in (r, s) by applying his own secret key ni and the public key Ps of the signer to yield s′≡ s − m× ni × Ps (mod p). And then requester Ri computes m′ = ni (ni −1) · m. Anyone can use the signer's public key Ps to verify the authentication of the signature (m′, s′, r) by checking whether the formula r ≡ s′− m′× Ps (mod p) holds.

1.5.4 THEOREM:
The triple (m′, s′, r) is a valid signature of message m for the above protocol and the protocol is a blind signature scheme.

PROOF:
Prove that the triple (m′, s′, r) is a valid signature of message m for the above protocol. The validity of the signature (m′, s′, r) can be easily be shown as follows, since s′ ≡ s − m× ni × Ps (mod p), and m′ = ni (ni −1) m. it has s′− m′× Ps = s − m × ni × Ps − m′× Ps = s − m× ni × Ps − ni (ni −1)· m× Ps = s − m× ni × Ps − ni ×ni ×m× Ps+ ni ×m× Ps = s − ni ×ni ×m× Ps = (nv + ns)× α− ni ×ni ×m× Ps = (nv + ns)× m× (ni × Pi)− ni ×ni ×m× Ps =nv × m × ni × Pi + ns × m ×ni × Pi − ni × ni × m × Ps = nv × m × ni × Pi + ns × m × ni × ni × G − ni × ni × m × ns× G = nv × m × ni × Pi = nv × α ≡ r (mod p) Two distinct message-signature pairs (m, (r1, s1)) and (m, (r2, s2)) yielded by giving two identical messages because the identical messages have a unique blinding factor nv each. In order to prove the blindness of the protocol, for given two identical message-signatures, the blinding factors applied on each are identical. Let n1 and n2 be the blinding factors respectively for two identical messages, and (r1, s1) and (r2, s2) be the corresponding signatures yielded from the above protocol. Suppose (r1,s1) and (r2,s2) are identical, it is obtained r1 ≡ r2 (mod p); s1 ≡ s2 (mod p). Thus, n1× α ≡ n2× α (mod p), and n1× α+ ns× α ≡ n2× α+ ns× α (mod p). Both of the equations show that n1 is equal to n2. Therefore, n1 and n2 being two distinct blinding factors implies that the corresponding signatures are distinct. Thus the blindness of our protocol holds. As to the blindness defined by Chaum, the signer knows nothing about the relationship between the signed matter s and the stripped signed matter s′. It is obvious that the signer in this protocol is unable to trace s′ from s because s′ is generated by applying the secret key ni of requester Ri. So this protocol satisfies the blindness property.

Example:
Step 1: Let the elliptic curve equation be y2 = x3 + x + 1 (mod 5). The base point G = (4, 2) on the elliptic curve is determined. Make the elliptic curve equation and the base point public.
Step 2: A signer selects a random element 7 as his secret key and then generates his corresponding public key Ps, where Ps 7× (4, 2) (mod 5).
Step 3: A requester selects a random element 11 as his secret key and then generates his corresponding public key Pi, where Pi ≡ 11× (4, 2) (mod 5).
Step 4: The requester sets his blinding factor as (11 × Pi) so as to transform his message m into a blinded message α, where α ≡ m× (11× Pi) (mod 5). Then he sends the blinded message α to the signer.
Step 5: Signer randomly selects number 21 as the second

blinding factor. Signer should check whether (α, 21) in his database. If yes, signer selects a distinct number for the in-coming identical blinded message. Then he generates a pair of blind signatures (r, s) where r ≡ 21 × α (mod 5) and s ≡ (21 + 7) × α (mod 5). At last, he sends the message-signature pair (α, (r, s)) back to the requester.
Step 6: Requester strips s from (r, s) by applying his own secret key 11 and the public key Ps of the signer to yield s: s − m× 11 × Ps (mod 5). And then he also computes m= 11(11 −1) · m. At last, he publishes the triple (m′, s′, r).

Verification phase:
Anyone can use the signer's public key Ps to verify the authentication of the signature (m′, s′, r) by checking whether the formula r ≡ s′− m′× Ps (mod 5) holds.

PROOF:
The validity of the signature (m′, s′, r) can be easily be shown as follows, since s′ ≡ s − m× 11 × Ps (mod 5), and m′ = 11 (11−1) · m. We have s′− m′× Ps = s − m× 11 × Ps − m′× Ps = s − m×11 × Ps − 11 (11 −1)· m× Ps = s − m × 11 × Ps − 11 × 11 × m × Ps + 11 × m × Ps = s − 11 × 11 × m × Ps = (21 + 7) × α − 11 × 11 × m × Ps = (21 + 7) × m × (11 × Pi) − 11 × 11 × m × Ps = 21 × m × 11 × Pi + 7 × m × 11 × Pi − 11 × 11 × m × Ps = 21× m × 11 × Pi + 7 × m × 11 × 11× (4, 2) − 11 ×11 × m × 7× (4, 2) = 21× m× 11 × Pi = 21 × α ≡ r (mod 5) Advantage: It satisfies all the properties of blind signature scheme.

1.5.4. "Implementation of Blind Digital Signature Using ECC" by MS.Dhanashree M.Kuthe, Prof. Avinash J. Agrawal [5].
This scheme proposed is based on elliptic curve cryptographic algorithm named "The Electronic Voting". The selection of this algorithm is difficult to solve. The algorithm is used in the combination with a hashing function as the blinding factor to scramble the contents of the message to be signed by the signer. This article uses the concept of "zero knowledge proof".
Zero knowledge proof: A zero-knowledge proof is a way that a "prover" can prove possession of a certain piece of information to a "verifier" without revealing it. This is done by manipulating data provided by the verifier in a way that would be impossible without the secret information in question. Zero-knowledge proofs are proofs that yield nothing beyond the validity of the assertion. That is, a verifier obtaining such a proof only gains conviction in the validity of the assertion. This is formulated by saying that anything that is feasibly computable from a zero-knowledge proof is also feasibly computable from the (valid) assertion itself (by a so-called simulator) because it enables to force parties to behave according to a predetermined protocol. This scheme consists of six phases.

Phase I: Key Generation:
In this phase, the private keys and public keys are generated using elliptic curve cryptographic algorithm. In this phase, a number 'k' is chosed randomly between 1 to (n-1) to be served as the private key. This private key is then treated with the base point of the formed elliptic curve and computes the public key.

Phase II: Blinding:
Here, the voter elects the vote (message). As the votes of the individuals should be kept confidential the votes (message) are blinded. A blinding factor is selected and the vote (message) is then treated with this blinding factor to blind the vote that is to hide the vote from others. The blinding factor chooses should possess an existing inverse of itself so that the message blinded could also be unblinded when required.

Phase III: Requester Phase:
In this phase, the voter generates a digital signature using his private key using the scheme of ECC. The voter then sends the entire four entities to the signer as a request for authentication. The entities comprise of identification details, blinded message, digital signature and a proving factor that proves the voter to be a valid citizen. Here, the factor that proves the voter to be a valid citizen uses the concept of zero knowledge .A valid citizen possesses a private key to oneself but to prove oneself to be a valid citizen one cannot reveal the private key as it is to be kept confidential or intruder may misuse it. The zero knowledge concepts work best in this situation. Here the voter will prove to possess a private key without revealing the private key.

Phase IV: Signing Phase:
In this phase, the signer initially will have the incoming request from the voter with four entities. After receiving the request message the signer verifies for two matters. First, whether requester is a valid voter or not and this is done by cross verifying the proving factor. Secondly, signer notes the identification details and checks whether requester has already voted or not. If the requester through both the matters the signer generates blind signature for the particular requester and authenticates the voter. The signer then replies the requester with message-signature pair. The signer displays the identification details and the public keys of the voters those whose have voted.

Phase V: Unblinding:
Voter after receiving the message - signature pair, the message is unblinded and the unblinded message - signature pair is sent to the voting centre acting as a verifier and the counter of the votes. Here, the message is unblinded as when the message-signature pair is sent to the voting centre the counter must know to whom the voter has voted to be able to count the number of vote for individual elective.

Phase: VI: Verification:
Verifier after receiving the unblinded message - signature verifies the signer's blind digital signature using the public key of the signer. As the signature is verified the count is incremented for elective that is voted. Verifier now displays all the digital signatures and blind digital signatures pairs. Hence the voter is ensured that his/her vote is counted. And no would come to know who voted to whom because only voter know about his own digital signature and blind digital signature received from signer.

Advantages:
The voter after choosing the vote blinds it as the signer should not be able to know to whom the voter has voted so the voter's vote remains confidential. Signer signs the

blinded message and hence the blind digital signature.
When the blind digital signature - message pair is received by the voter, the message is unblinded. This unblinded message along with blind digital signature is sent to the verifier so that the verifier would see to whom the voter has voted for and update the counters.

Problem Statement
The objectives of this thesis are: To propose a ECC base blind signature scheme, resistant against universal forgery attack. To propose some methodologies to prevent the attack proposed on Lee et al.'s blind signature scheme by Lin et al.

Thesis Organization
The rest of the thesis is organized as follows.

Chapter 2 describes the mathematics of cryptography. It describes the methods required to generate the prime numbers, the methods to test the primality of a number,the cryptographic hash functions to generate the message digest and the basic building blocks of discrete logarithm problem.

Chapter 3 describes the proposed normal blind signature scheme and the Car-menish et al.'s scheme [25] .We have proposed an universal attack [31] on the Carmenish et al.'s scheme.

Chapter 4 describes the Lee et al.'s blind signature scheme [10] and the pro-posed improvement over it to prevent the attack proposed by Lin et al. [27]

Chapter 2

2. Mathematical Background

2.1 Mathematics of Cryptography
The mathematics of cryptography is a vast area of concern now a day. It can be differentiated into two parts:-

1. Mathematics of Symmetry-key cryptography
It describes Integer arithmetic, Modular Arithmetic, Matrices and Linear Congruence, Different Algebraic Structures .

2. Mathematics of Asymmetry-Key cryptography
It defines the concept of Primes, Primality Testing , Factorization , Chinese Remainder Theorem, Quadratic Congruence ,Exponentiation and Logarithm etc. However in this article the algebraic structures are of key concern.

2.1.1 Basic algebraic structures
The combination of set and the operations that are applied to the elements of set is called algebraic structure.

Groups: Definition 3.1: A Group (G) is a set of elements with a binary operator "•" that satisfies the following four properties:
1. **Closure**: If x and y are the elements of G, then z = x •y is also an element of G.
2. **Associativity**: If x, y and z are elements of G, then (x •y) •z = x •(y •z).
3. **Existence of identity**: For all x in G, there exist an element e, called the identity element, such that e •a = a •e = a.
4. **Existence of inverse**: For each x in G, there exists an element x0, called the inverse of x, such that x •x0 =x0 •x = e. Along with those properties if it also satisfies the commutative property then it is called as commutative group or abelian group. Commutative property means for all x and y in G, we have x •y = y •x.

A commutative group also called an abelian group satisfies the above properties along with the commutative property.
Ring: A Ring is an abelian group structure with two operations. It is denoted as R =<{…},•, □ >.The first operation must satisfy all the five properties that are required for an abelian group. The second operation must satisfy only the first two. In addition, it should also satisfy **the Distributive property which states that for all x, y and z elements of R, we have x¤(y . z)** = (x¤y) . (x¤z) and (x . y) ¤ z = (x¤z) . (y ¤ z). A Ring is said to be commutative ring if the second operation also satisfy the commutative property.
Field: A Field, denoted by F =<{…},•, □ >, is a commutative ring in which the second operation satisfies all the five properties defined for the first operation except that the identity of the first operation.
Finite Fields: Only finite fields are extensively used in cryptography. Galois showed that for a field to be finite the number of elements should be Pn, where p is a prime and n is a positive integer. The finite fields are usually called as Galois fields and denoted as GF (Pn).
GF (P): When n=1, we have GF (P) field [2, 4]. This field consists of the elements 0, 1, ..., P-1, with two arithmetic operations addition and multiplication.

2.2 Secure hash algorithm (SHA-1):
The Merkle-Damgard scheme is the basic for many cryptographic hash functions today. We should use a compression function that is collision resistant. There are two different approaches in designing a hash function: it can be made from scratch like MD,MD2, MD4, MD5, SHA, SHA1. Second approach is that it can also be designed by using symmetric key block cipher. SHA-1 hash function is being used in our schemes. A hash function is a function hash () which should satisfy the following properties:
• Compression hash (): takes input m of arbitrary length and produce a fixed length string output hash(m).
• Non-invertible :Given hash(m) and hash() it is difficult to get m.

Two types of hash function are discussed: keyed or non-keyed hash function. Modification detection code (MDC)is a non-keyed hash function which is further divided into one way hash function(OWHF) and collision resistance hash function (CRHF). Both of them supports random oracle model. In our thesis work we have used the SHA-1, one way hash function, for the message digest. This compression function will give a fixed length output of 160 bits. Maximum message size that it takes is 264-1 and the block size is 512 bits.. Total 80 numbers of rounds has been used with a word size of 32 bits.

2.3 Elliptic curve Cryptosystem

2.3.1 Introduction

Since the invention of public key cryptography in 1976 by Whitefield Diffe and Martin Hellman numerous public key cryptographic systems have been proposed. All of these systems are based on the difficulty of solving a mathematical problem. Over the years, many of the public key cryptography systems have been broken and some are proved to be impractical. Today only three types of system are considered to be safe, secure and efficient. They are,

1. Integer factorization problem (IFP)
2. Discrete Logarithm Problem (DLP)
3. Elliptic Curve Discrete Logarithm Problem (ECDLP)

2.3.2 Integer factorization problem

The integer factorization problem (IFP) is the following: given a composite number n that is the product of two large prime numbers p and q, find p and q. While finding large prime numbers is a relatively easy task, the problem of factoring the product of two such numbers is considered computationally intractable if the primes are carefully selected. Based on the difficulty of this problem, Rivest, Shamir and Adleman developed the RSA public-key cryptosystem.

2.3.3 Discrete Logarithm Problem

If p is a prime number, then Zp denotes the set of integers {0,1,2,…….,p – 1}, where addition and multiplication are performed modulo p. It is well-known that there exists a non-zero element α Є Zp such that each non-zero element in Zp can be written as a power of α such an element α is called a generator of Zp. The discrete logarithm problem (DLP) is the following: given a prime p, a generator α of Zp, and a non-zero element β Є Zp, find the unique integer k, 0…k…p-Є, such that βα = k(mod p). The integer k is called the discrete logarithm of β to the base α.

2.3.4 Elliptic Curve Discrete Logarithm Problem

If q is a prime power, then Fp denotes the finite field containing q elements. In applications, q is typically a power of 2 (2m) or an odd prime number (p). The elliptic curve discrete logarithm problem (ECDLP) is the following: given an elliptic curve E defined over Fq, a point PЄ(Fq) of order n, and a point QЄ(Fq), determine the integer k,0…k…p-1, such that Q = kP, provided that such an integer exists.

2.3.5 Comparison

Figure 3.1 compares, the time required to solve an instance of a problem based on ECC with the time required to solve the problem based on IFP or DLP. Here the time is measured in MIPS. As a benchmark, it is generally accepted that 1012 MIPS years represents reasonable security at this time. In the Figure 3.1 the times of RSA and DSA are grouped together because the asymptotic running time for both is same. As we can see that to achieve reasonable security, RSA and DSA should employ 1024-bit modulo, while a 160-bit modulus should be sufficient for ECC. Moreover, the security gap between the systems increases dramatically as the modulo sizes increases.

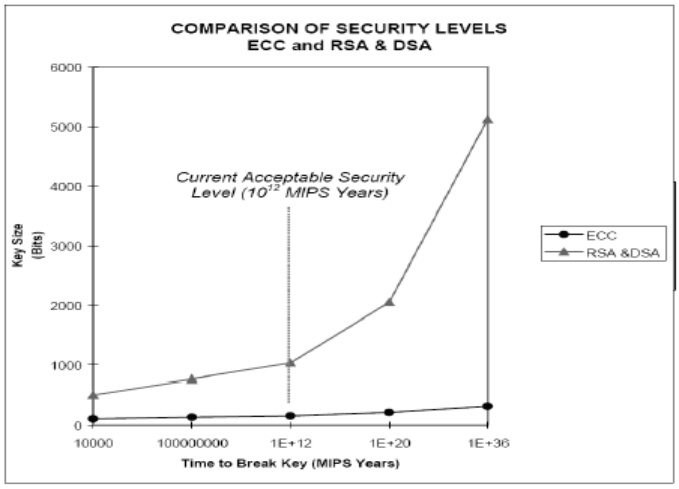


Figure 2.1 : Comparison of security level

2.4 Elliptic curves over real numbers

An elliptic curve over real numbers may be defined as the set of points (x,y) which satisfy an elliptic curve equation of the form:

y2 = x3 + ax + b,

where x, y, a and b are real numbers. Each choice of the numbers a and b yields a different elliptic curve. For example, a = -4 and b = 0.67 gives the elliptic curve with equation y2 = x3 - 4x + 0.67; the graph of this curve is shown below:

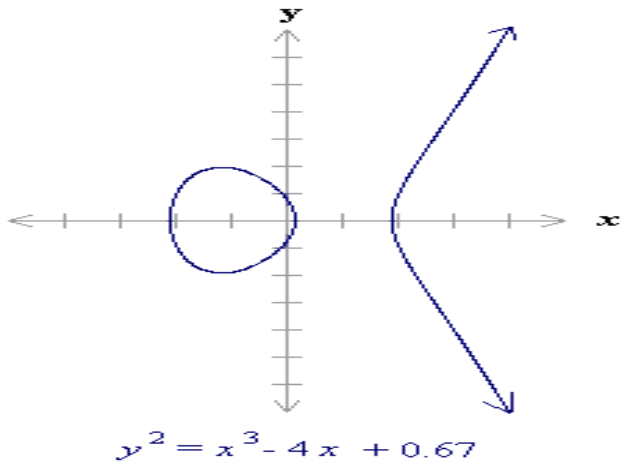


Figure 2.2 : An Elliptic Curve over real numbers

If x3 + ax + b contains no repeated factors, or equivalently if 4a3 + 27b2 is not 0, then the elliptic curve y2 = x3 + ax + b can be used to form a group. An elliptic curve group over real numbers consists of the points on the corresponding elliptic curve, together with a special point O called the point at infinity.

2.4.1 : Adding distinct points P and Q

Suppose that P and Q are two distinct points on an elliptic curve, and the P is not -Q. To add the points P and Q, a line is drawn through the two points. This line will intersect the

elliptic curve in exactly one more point, call -R. The point -R is reflected in the x-axis to the point R. The law for addition in an elliptic curve group is P + Q = R. For example:

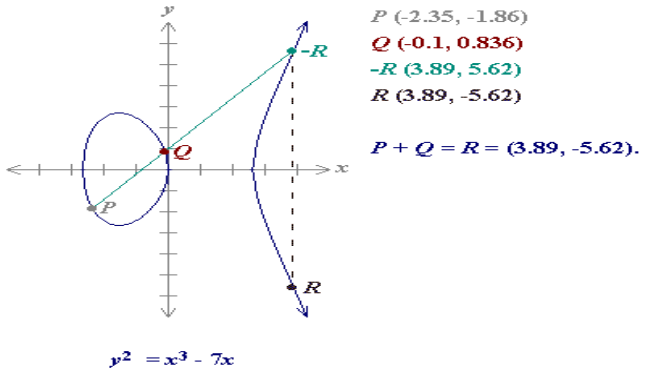


Figure 2.3 : Adding distinct points P and Q of an elliptic curve over real numbers

The line through P and -P is a vertical line which does not intersect the elliptic curve at a third point; thus the points P and -P cannot be added as previously. It is for this reason that the elliptic curve group includes the point at infinity O. By definition, P + (-P) = O. As a result of this equation, P + O = P in the elliptic curve group . O is called the additive identity of the elliptic curve group elliptic curves have an additive identity.

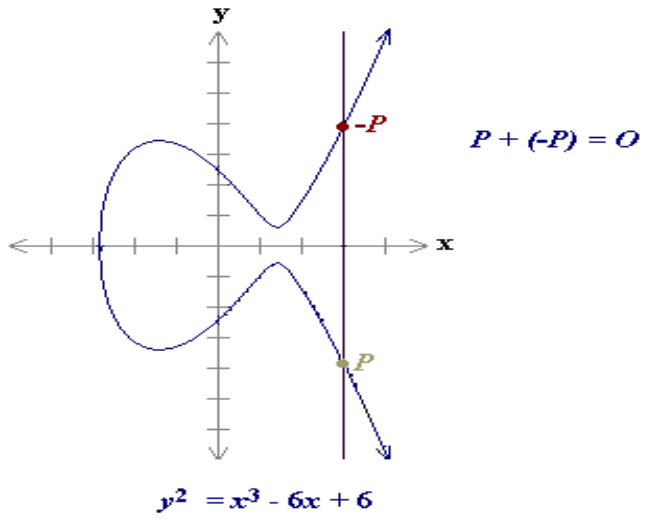


Figure 2.4 : Showing the additive identity of a elliptic curve over real numbers

To add a point P to itself, a tangent line to the curve is drawn at the point P. If yP is not 0, then the tangent line intersects the elliptic curve at exactly one other point, -R. -R is reflected in the x-axis to R. This operation is called doubling the point P; the law for doubling a point on an elliptic curve group is defined by: P + P = 2P = R

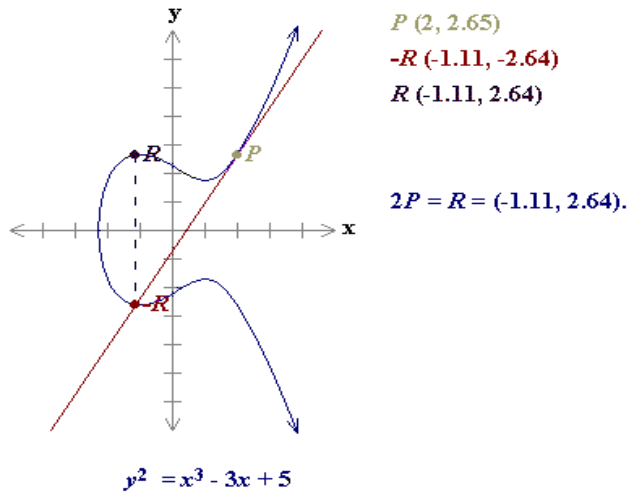


Figure 2.5 : Adding a point to itself in an elliptic curve over real numbers

P + P = 2P = R.

2.4.2 Adding distinct points P and Q Mathematically

When P = (xP,yP) and Q = (xQ,yQ) are not negative of each other,
P + Q = R where
s = (yP - yQ) / (xP - xQ)
xR = s2 - xP - xQ and yR = -yP + s(xP - xR)
Note that s is the slope of the line through P and Q.
2.4.3 Doubling the point P Mathematically
When yP is not 0,
2P = R where
s = (3xP2 + a) / (2yP)
xR = s2 - 2xP and yR = -yP + s(xP - xR)

2.5 Elliptic curves over Fp

Recall that the field Fp uses the numbers from 0 to p - 1, and computations end by taking the remainder on division by p. For example, in F23 the field is composed of integers from 0 to 22, and any operation within this field will result in an integer also between 0 and 22. An elliptic curve with the underlying field of Fp can formed by choosing the variables a and b within the field of Fp. The elliptic curve includes all points (x,y) which satisfy the elliptic curve equation modulo p(where x and y are numbers in Fp). For example: y2 mod p = x3 + ax + b mod p has an underlying field of Fp if a and b are in Fp. If x3 + ax + b contains no repeating factors (or, equivalently, if 4a3 + 27b2 mod p is not 0), then the elliptic curve can be used to form a group. An elliptic curve group over Fp consists of the points on the corresponding elliptic curve, together with a special point O called the point at infinity. There are finitely many points on such an elliptic curve. Recall that a is one of the parameters chosen with the elliptic curve and that s is the tangent on the point P. There are several major differences between elliptic curve groups over Fp and over real numbers. Elliptic curve groups over Fp have a finite number of points, which is a desirable property for cryptographic purposes. Since these curves consist of a few discrete points, it is not clear how to "connect the dots" to make their graph look like a curve. It is not clear how geometric relationships can be applied. As a result, the geometry used in elliptic curve groups over real

numbers cannot be used for elliptic curve groups over Fp. However, the algebraic rules for the arithmetic can be adapted for elliptic curves over Fp. Unlike elliptic curves over real numbers, computations over the field of Fp involve no round off error - an essential property required for a cryptosystem.

2.5.1 Adding distinct points P and Q
The negative of the point P = (xP, yP) is the point -P = (xP, -yP mod p). If P and Q are distinct points such that P is not -Q, then
P + Q = R where
s = (yP - yQ) / (xP - xQ) mod p
xR = s2 - xP - xQ mod p and yR = -yP + s(xP - xR) mod p
Note that s is the slope of the line through P and Q.
2.5.2 Doubling the point P Provided that yP is not 0,
2P = R where
s = (3xP2 + a) / (2yP) mod p
xR = s2 - 2xP mod p and yR = -yP + s(xP - xR) mod p

2.6 Elliptic curve over F(2m)
Recall that a is one of the parameters chosen with the elliptic curve and that s is the slope of the line through P and Q. An elliptic curve with the underlying field F(2m) is formed by choosing the elements a and b within F(2m) (the only condition is that b is not 0). As a result of the field F(2m) having a characteristic 2, the elliptic curve equation is slightly adjusted for binary representation:

y2 + xy = x3 + ax2 + b

Elliptic curve groups over F(2m) have a finite number of points, and their arithmetic involves no round off error. This combined with the binary nature of the field, F(2m) arithmetic can be performed very efficiently by a computer. The following algebraic rules are applied for arithmetic over F2m :

2.6.1 Adding distinct points P and Q
The negative of the point P = (xp, yp) is the point -P = (xp, xp + yp). If P and Q are distinct points such that P is not -Q, then
P + Q = R where
s = (yp – yq) / (xp+ xq)
xr = s2 + s + xp + xq + a and yr = s(xp+ xr) + xr+ yp
As with elliptic curve groups over real numbers, P + (-P) = O, the point at infinity. Furthermore, P + O = P for all points P in the elliptic curve group.

2.6.2 Doubling the point P
If xp = 0, then 2P = O
Provided that xp is not 0,
2P = R where
s = xp + yp / xp
xr = s2+ s + a and yr= xp 2 + (s + 1) * xr
Recall that a is one of the parameters chosen with the elliptic curve and that s is the slope of the line through P and Q .

2.7 ECC Domain Parameters:
Elliptic curve cryptography (ECC) domain parameters over GF(P), can be represented by a six tuple: E = (q, a, b, G, n, h), where q = P or q = 2m, where m is a natural number. a and b are the co-efficient of x3 and x respectively used in the equation.

Y2= x3 + ax + b (mod P) for q = P ¸ 3
Y2 + xy = x3 + ax2 + b for q = 2m ¸ 1
G is a base point on the elliptic curve. n is prime number which is of the order of G. The order of a point on an elliptic curve is the smallest positive integer r such that rp = 1.Finally h = (E/n). where |E| represents the total number of points on elliptic curve and it is called the curve order.

2.8 ECC key generation
1. Receiver chooses E(a,b) with an elliptic curve over GF(p) or GF(2n).
2. Receiver chooses a point on the curve e1(x1,y1) .
3. Receiver chooses an integer d.
4. Receiver calculates e2(x2,y2)=d*e1(x1,y1).Here multiplication means multiple addition of points.
5. Receiver announces E (a,b), e2(x2,y2),e1(x1,y1) as his public keys and keeps d as private key.

3. Proposed BDS scheme
The proposed BDS scheme was derived from a variation of the DSA. Our BDS scheme, meanwhile, is derived from a variation of the ECDSA, and again it has 5 phases:
• Initialization,
• Blinding,
• Signing,
• Unblinding,
• Verifying.

In our proposed scheme, we used the elliptic curves over the Fp prime field, which has been suggested by the National Institute of Science and Technology (NIST) and is called Federal Information Processing Standard 186-2 [18,19]. According to the Standards for Efficient Cryptography Group [20], elliptic curve domain parameters over Fp are defined as a six tuple: T = (p, FpabGnh), (1) where p is an integer specifying theFp finite field and ab Fp are integers specifying the elliptic curve E(Fp) defined by Eq. (2):

E (Fp) : y2 ≡ x3 + ax + b (mod p), (2)

Where G = (xGyG)is a base point onE(Fp), n is a prime number defining the order of G, and h is an integer defining the cofactor: h =# E(Fp)n .

3.1. Initialization and key pair generation for ECDSA
The signer defines the elliptic curve domain parameters T , defined as in Eq. (1). Next, for each request, an integer k is randomly selected by the user and the elliptic curve point ´R is calculated accordingly (refer to Table 1 for all abbreviations used in this section):

´R= kG = (x´1,y´1), (3)

´r= ´x1 (mod n). (4)

In addition, the signer checks whether Eq. (5) holds.

´r = 0 (5)

If the result is true, the signer sends the elliptic curve point ´R to the requester. If the result is false, then the signer selects another k randomly and repeats Eqs. (3) and (4) until he finds a ´r fulfilling Eq. (5).

Table 3.1. Interpretation of abbreviations used throughout Section 3.

Abbrev-iation	Interpretation
T	Elliptic curve domain parameters
p	Order of the finite field Fp, integer
Fp	Finite field
a, b	Coefficients defining the elliptic curve
G	Generator point
n	Order of G, a prime number
h	Cofactor, integer
ECC	Elliptic curve cryptography
ECDSA	Elliptic curve digital signature algorithm
H (:)	Hash value
d	Private key of the signer
Q	Public key of the signer, a point on the elliptic curve
m	Message
m´	Blinded message
s	Signature
s´	Blind signature
r	x coordinate of R
r´	´ x coordinate of R’
´ R, R’	Points on the elliptic curve
A, B, k	Random integer numbers
(x,y)	Coordinates for the Cartesian system

To generate the private and public key of the signer, the following steps are followed:
Integer d is chosen randomly in the range (1, n-1).
The elliptic curve point of Q is calculated as in Eq. (6):

$$Q = dG = (xQ, yQ) . \quad (6)$$

With these calculations, the public key of the signer is assigned as point Q and the private key of the signer is assigned as integer d.

3.2. Blinding phase

In order to blind the message m, the owner of message m needs the elliptic curve domain parameters T of the signer; refer to Eq. (1). Blinding is achieved through the following steps, which are shown in Figure 1: Signer sends the elliptic point ´R (refer to Eq. (3)) to the requester, which will be used as the blinding coefficient. Requester calculates ´r from the elliptic point ´R , as shown in Eq. (4).
3. Requester randomly chooses integers A and B, which are in the range of (1, n-1).
4. Requester calculates the elliptic point R:

$$R = A´R + BG = (x1y1). \quad (7)$$

5. Requester calculates r from the elliptic point R, which was given in Eq. (7):

$$r = x1 \pmod n. \quad (8)$$

6. Requester generates the blinded message ´m and sends it back to the signer for the signing operation:

$$´m = AH (m) ´r r{-}1 \pmod n \quad (9)$$

Where H is the “Hash” function, and in our scheme, we use the SHA-1 [21] algorithm as the hash function.

3.3. Signing phase

After the signer receives the blinded message ´m from the requester, he generates the blind signature ´s by following these steps, which are also shown in Figure 1:
1. Signer calculates ´r from the elliptic point ´R , as shown in Eq. (4).
2. The private key of the signer, d, was generated in the initialization phase.
3. k is a random integer that was generated in the initialization phase.
4. ´s is calculated as shown in Eq. (10):

$$´s = d´r + k ´m \pmod n. \quad (10)$$

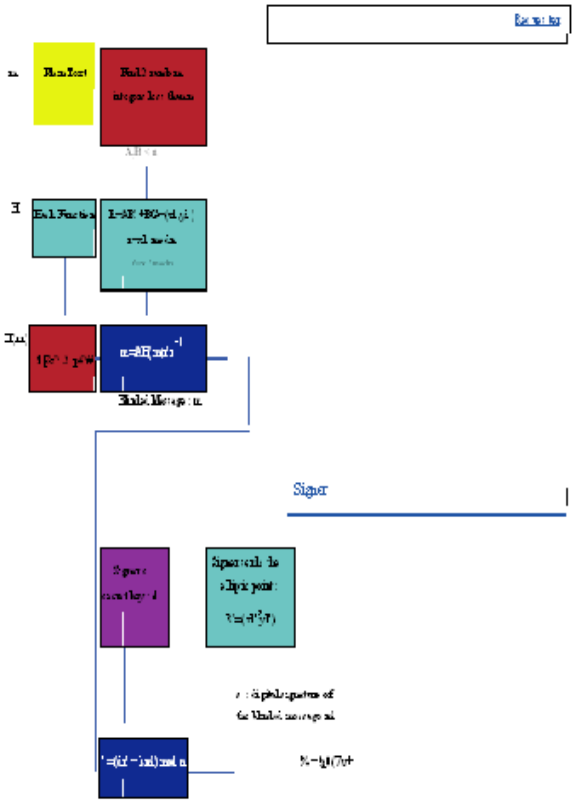


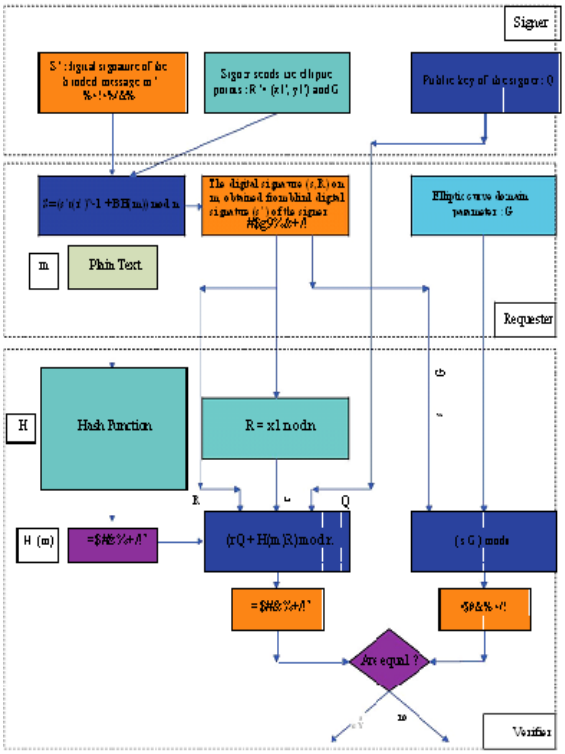
Figure 3.3. “Blinding” and “Signing” phases of the proposed BDS scheme.

3.4. Unblinding phase

When the requester receives the blind digital signature ´s from the signer, the unblinding operation is needed to obtain the digital signature (s, R) on message m, as shown in Figure 2.

1. Requester calculates ´r from the elliptic point ´R , as shown in Eq. (4).
2. Requester verifies whether ´r and ´s are in the range of (1, n-1). If so, the requester generates the digital signature (s, R) of the signer on message m, as shown in Eq. (11):

s = ´s r ´r−1 + BH (m) (mod n). (11)



Figurfigure 3.4 "Unblinding" and "Verifying" phases of the proposed Blind Signature scheme.

3.5. Verifying phase

Any party that has the elliptic domain parameters T of the signer (refer to Eq. (1)) can verify the digital signature of (s, R) on message m by following these steps, which are also shown in Figure 2:

u1 = sG (mod n). (12)

u2 is calculated using the public key of the signer, Q:

u2 = rQ + H (m)R (mod n). (13)

If the statement of u1 = u2 is met, then the signature is verified as valid; otherwise, it is considered invalid.

3.6. Correctness proof of the proposed scheme

We begin by expanding u2 , defined in Eq. (13), by substituting Q with dG according to Eq. (6):

u2 = rdG + H (m)R (mod n). (14)

Since from Eq. (7) we know that R = A´R + BG, then we can expand Eq. (14) as follows:

u2 = rdG+ H (m)A´R + H (m)BG (mod n) . (15)

Using Eq. (3), we substitute ´R with kG and we get:

u2 = rdG + H (m) AkG + H (m)BG (mod n) . (16)

Now, by expanding u1 , defined in Eq. (12), we need to achieve the same expression shown in Eq. (16). Since from Eq. (11) we know that s = ´s r ´r−1 + BH (m) (mod n), then Eq. (12) becomes:

u1 = ´s r ´r−1G + BH (m)G(mod n). (17)

By substituting ´s with d´r + k ´m (mod n) from Eq. (10), Eq. (17) results in:

u1 = d´r r ´r−1G + k ´mr ´r−1G + BH (m)G(mod n). (18)

By rearranging Eq. (18) we get:

u1 = rdG´r´r−1 + k ´mr ´r−1G + H (m)BG(mod n). (19)

From Eq. (9), substituting ´m with AH (m) ´r r−1 (mod n) in Eq. (19) results in:

u1 = rdG´r´r−1 + kAH(m) ´r r−1r ´r−1G + H (m)BG(mod n). (20)

From modular arithmetic, we know that ´r´r−1 = 1(mod n) and rr−1 = 1(mod n). By substituting these into Eq. (20) we get:

u1 = rdG + kAH(m)G + H (m)BG(mod n). (21)

Eq. (21) is the same expression shown in Eq. (16). Therefore, we have proven that u1 = u2 by showing that Eqs. (16) and (21) are equal to the same expression.

4. Discussions

In the applications, the key length of the algorithm is determined according to the desired security level. Today, it is most practical to use a key length of between 160 and 192 bits for ECC systems. In the case of RSA, the key length is 1024 bits for commercial applications and 2048 bits for more critical applications (where more security is needed). These key lengths correspond to the 192-bit and 224-bit ECC key lengths, respectively [22]. While the security of Chaum's BDS scheme [1] is based on the difficulty of the factorization problem [23] the security of Camenisch et al.'s BDS scheme [17] is based on the difficulty of the discrete logarithm problem [24]. On the other hand, the security of our BDS scheme relies on the elliptic curve discrete logarithm problem, which is considered to be much more difficult than either of the other problems [12]. In our work, to provide comparisons for the reader, the implementation and simulation of both the Blind Signature schemes of [1] and [17] have been accomplished. A 1024-bit RSA key length is chosen for the implementation of [1] and a 1024-bit DSA key length is chosen for the implementation of [17]. To provide further comparison for the reader, we have issued our scheme with a variety of NIST-suggested elliptic curves (NIST192, NIST224,NIST256, NIST384, and NIST521). This means that the key length of our scheme changes depending on the curve (192 bits, 224 bits, 256 bits, 384 bits,

and 521 bits, respectively). For example, if the NIST192 elliptic curve is chosen for our scheme, then the key length is apparently 192 bits.The test-bed system consists of a 1733-MHz processor with 512 MB of DDR-2 533-MHz RAM. Implementation is based upon the C programming language. For elliptic curve arithmetic operations, Miracle Library is used [25]. In order to compare the time consumptions of the algorithms, the clock command of the C programming language has been used. It gives the time that is spent on the processor between 2 events. For the same plain text message (m consists of 431 bytes), the time (in seconds) spent on the processor for the relevant algorithms is given in Figure 3.

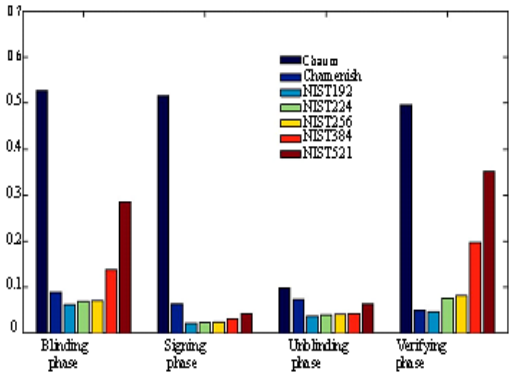


Figure 4.1. Comparisons of processing time for various BDS schemes classified according to phases.

Figure 4.1 is sorted according to the phases (blinding, signing, unblinding, and verifying) of the BDS schemes, while Figure 4 is sorted according to the types of the BDS schemes (Chaum's [1], Camenisch et al.'s [17], and our scheme with the following elliptic curves: NIST192, NIST224, NIST256, NIST384, and NIST521).

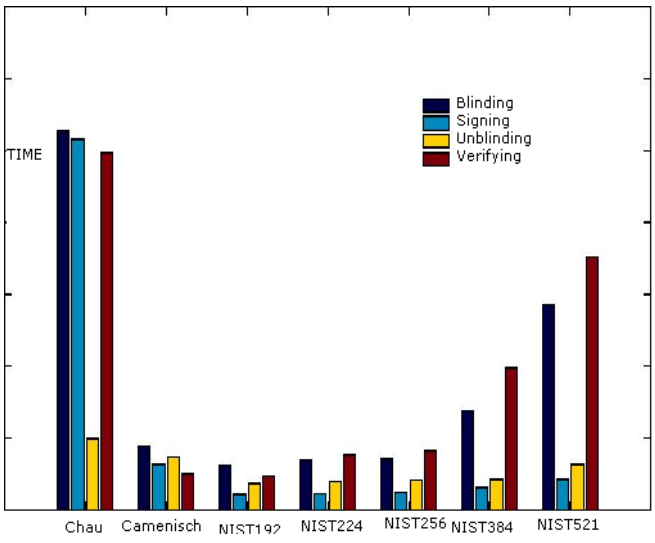


Figure 4.2. Comparisons of processing times for various BDS schemes classified according to schemes .

Table 4.1 gives the processing time (s) of our scheme compared to other schemes, when the NIST192 elliptic curve is used for our scheme. Table 3 gives the performance improvement (%) of our scheme compared to other schemes, when the NIST192 elliptic curve is used for our scheme. In this case, it is clear that in terms of the processing time, our scheme outperforms Chaum's scheme [1] by about 96% and Camenisch et al.'s scheme [17] by about 66%.

Table 4.1. Processing time (s) of Blind Signature schemes.

	Our scheme	Chaum's scheme	Camenisch et al.'s scheme
Blinding phase	0.0624	0.5267	0.0892
Signing phase	0.0218	0.5156	0.0641
Unblinding phase	0.0374	0.0984	0.0732
Verifying phase	0.0470	0.4971	0.0499

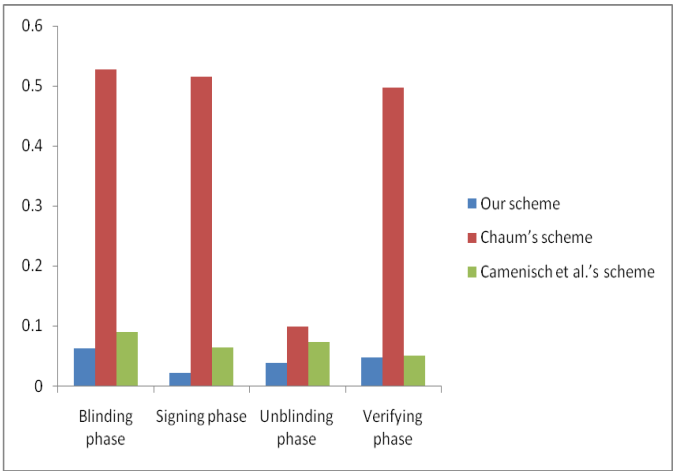


Figure 4.3. Comparisons of processing times for various BDS schemes

Table 4.2. Relative performance improvement (%) of our scheme compared to other schemes.

	Chaum's scheme	Camenisch et al.'s scheme
Blinding phase	88.15	30.04
Signing phase	95.77	65.99
Unblinding phase	61.99	48.90
Verifying phase	90.55	5.81

For all of the phases (blinding, signing, unblinding, and verifying), the fastest scheme is the one proposed in this study, which uses the NIST192 elliptic curve (in other words, the scheme that has a key length of 192 bits), and the slowest of all is Chaum's [1] scheme, which uses a 1024-bit RSA key length. It is important to mention that the key lengths for

the considered schemes are selected to provide equal security levels. For example, it has been proven that the security levels of the 1024-bit key length RSA algorithm, 1024-bit key length DSA algorithm, and 160-bit key length ECC algorithm are the same [26,27]. The computational effort needed to factor a 1024-bit size integer using the general number field sieve method is 3 × 1011 million instructions per second years, whereas the same effort is needed to compute elliptic curve logarithms of the 160-bit size elliptic point with the Pollard ρ-method [12]. In [12], it is suggested that a 192-bit size NIST elliptic curve is comparable to 1024-bit size RSA and DSA key lengths in terms of the intended cryptanalysis strength. Hence, we issued a 192-bit key length ECC algorithm, and, in this case, our scheme is not only faster but also more secure. Table 4 gives the comparable key sizes of the ECDSA and RSA/DSA algorithms in terms of the computational effort for cryptanalysis [28].

Table 4.3. Comparable key sizes in terms of the computational effort for cryptanalysis [28].

ECDSA (size of the prime field in bits)	RSA/DSA (modulus size in bits)
112	512
160	1024
224	2048
256	3072
384	7680
512	15,360

5. Conclusions and future remarks

In this thesis, we briefly introduced the concept of BDS, and later on, our contribution to the field was presented. Our proposed BDS scheme has lower complexity (i.e. in terms of computational load) and provides better security compared to those in [1] and [17]. Our proposed scheme uses ECC (ECDSA), providing all of its advantages over the other PKC algorithms. It offers smaller key lengths for desired security levels, along with high-speed cryptographic processes, leading to low-complexity hardware and software requirements [12]. These advantages are indispensable for applications where resource shortage is of prime importance, especially in mobile platforms. Our proposed scheme can be used in the applications where not only user anonymity but also processing time is critical under certain hardware constraints. According to the results; our proposed scheme outperforms that in [1] by 96% and that in [17] by 66% in terms of processing time. Thus, our proposed scheme leads to an apparent improvement in BDS systems. Eventually, this enhancement will drastically reduce the total cost of the commercial systems that are using BDS. The application of our scheme to smart cards, e-commerce, and e-voting is left as future work for us to consider.

REFERENCES

1. D. Chaum. Blind signatures for untraceable payments, volume 82, pages 199– 203. Plenum Publishing, 1983.

2. J. X. Zhao H. Q.Wang and L. J. Zhang. Blind signature scheme based on elga-mal signature equation. Nanjing University of Posts and Telecommunication, 25(4):65–69, 2005.

3. B.A.Farouzan. Cryptography & Network Security. Tata McGraw-Hill Pub-lishing Company Limited, Inc.,New York, 2007.

4. E. Mohammed, A.E. Emarah, and K. El-Shennawy. A blind signature scheme based on elgamal signature. In EUROCOMM 2000. Information Systems for Enhanced Public Safety and Security. IEEE/AFCEA, pages 51 –53, 2000.

5. S.Mohanty and B.Majhi. A secure multi authority electronic voting protocol based on blind signature. In Proceedings of the 2010 International Confer-ence on Advances in Computer Engineering, pages 271–273. IEEE Computer Society, 2010.

6. S.Wang, F.Hong, and G.Cui. Secure efficient proxy blind signature schemes based dlp. In Proceedings of the Seventh IEEE International Conference on E-Commerce Technology, pages 452–455. IEEE Computer Society, 2005.

7. G.Qadah and R.Taha. Electronic voting systems: Requirements, design, and implementation. Computer Standards Interfaces, 29(3):376–386, 2009.

8. D.Chaum. Blind signature system. In CRYPTO, page 153, 1983.

9. C.C.Lee, M.S.Hwang , and W.P.Yang. A new blind signature based on the discrete logarithm problem for untraceability. Applied Mathematics and Computation, 164(3):837–841, 2008.

10. H.F.Huang and C.C.Chang. An untraceable electronic cash system using fair blind signatures. E-Business Engineering, IEEE International Conference on, 0:39–46, 2006.

11. K.Chen W.Qiu and D.Gu. A new offline privacy protecting e-cash system with revokable anonymity. In Proceedings of the 5th International Conference on Information Security, pages 177–190. Springer-Verlag, 2008.

12. J.M.Piveteau J.Camenisch and M.Stadler. An efficient fair payment system. In ACM Conference on Computer and Communications Security, pages 88– 94, 1996.

13. Y.Tsiounis Y.Frankel and M.Yung. Indirect Discourse Proofs.

14. Jan Camenisch, Ueli Maurer, and Markus Stadler. Digital payment systems with passive anonymity-revoking trustees. J. Comput. Secur., 5(1):69–89, 1997.

15. Weidong and Qiu. Converting normal dlp-based signatures into blind. Applied Mathematics and Computation, 170(1):657–665, 2009.

16. Masayuki Abe and Tatsuaki Okamoto. Provably secure partially blind signa-tures, 2000.

17. D. Chaum, M. Jakobsson, R. L. Rivest, P. Y. A. Ryan, J. Benaloh, M. Kuty-lowski, and Ben Adida. Towards trustworthy elections, new directions in electronic voting. In Towards Trustworthy Elections, volume 6000 of Lecture Notes in Computer Science. Springer, 2010.

18. D.Alessio and M.Joye. A simple construction for public-key encryption with revocable anonymity. In Proceedings of the ninth ACM workshop on Digital rights management, pages 11–16. ACM, 2010.

19. M.Michels P.Hoster and H.Petersen. Comment:cryptanalysis of blind signa-tures bades on discrete logarithm problem. Electronic Letters, 31(21):1827, 1995.

20. R.Wendolsky S.Kopsell and H.Federrath. Revocable anonymity. In In Gnter Mller (Ed.): ETRICS 2006, Lecture Notes in Computer Science, pages 208– 222. Springer Verlag, 2006.

21. L.Harn. Cryptanalysis of blind signatures bades on discrete logarithm prob-lem. Electronic Letters, 31(14):1136–1137, 1995.

22. Ernie Brickell, Peter Gemmell, and David Kravitz. Trustee-based tracing extensions to anonymous cash and the making of anonymous change. In In Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algo-rithms, pages 457–466, 1995.

23. D. Chaum and T. P. Pedersen. Wallet databases with observers. In Ernest F. Brickell, editor, Advances in Cryptology - CRYPTO 92, 12th Annual Interna-tional Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings, volume 740 of Lecture Notes in Computer Science, pages 89–105. Springer, 1992.

24. J.M Piveteau J.Camenisch and M. Stadler. Blind signatures based on the discrete logarithm problem. In Advances in Cryptology - EUROCRYPT '94, pages 428–432, 1994.

25. T.Wu and J.R Wang. Comment: A new blind signature based on the discrete logarithm problem for untraceability. Applied Mathematics and Computation, 170(1):999–1005, 2009.

26. C.I Wang C.I Fan, D. J. Guan and D.R.Lin. Cryptanalysis of lee-hwang-yang blind signature scheme. Computer Standards & Interfaces, 31(2):319–320, 2009.

27. A.N.Oo and N.L.Thein. DLP based proxy blind signature scheme with Low-Computation. Networked Computing and Advanced Information Manage-ment, International Conference on, 0:285–288, 2009.

28. X.F.Chen S.L Liu and F.G Zhang. Forgeability of wang-tang-li's id-based restrictive partially blind signature scheme. J. Comput. Sci. Technol., 23(2):265–269, 2010.

29. S.Brands. Untraceable off-line cash in wallets with observers. In Advances in Cryptology - CRYPTO'93, pages 302–318. Springer-Verlag, 1993.

30. B. Yu and C. Xu. Security analysis on a blind signature scheme based on elgamal signature equation. In Proceedings of the 2007 International Confer-ence on Computational Intelligence and Security Workshops, pages 741–744. IEEE Computer Society, 2007.

31. J.Camenisch and T. Gro. Efficient attributes for anonymous credentials ex-tended version. IACR Cryptology ePrint Archive, 2010:496, 2010.

32. D. Chaum, R. L. Rivest, B. Preneel, A. D. Rubin, D. G. Saari, and P. L. Vora. Guest editorial: special issue on electronic voting. IEEE Transactions on Information Forensics and Security, 4(4):593–596, 2009.

33. I.C.Lin, M.S.Hwang , and C.C.Chang. Security enhancement for anonymous secure e-voting over a network. Comput. Stand. Interfaces, 25(2):131–139, 2003.

34. X.Hu and S.Huang. An efficient id-based restrictive partially blind sig-nature scheme. In Proceedings of the Eighth ACIS International Confer-ence on Software Engineering, Arti cial Intelligence, Networking, and Par-allel/Distributed Computing - Volume 03, pages 205–209. IEEE Computer Society, 2010.