IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 20, Issue 1, Ver. II (Jan.- Feb. 2018), PP 15-25
www.iosrjournals.org

Partially Blind Signature Scheme Based On ECDLP for Untraceable Electronic Payment System

[1]Mostary Begum [2]Md. Tareq Hasan
[1,2] *Department of Computer Science and Engineering, University of Development Alternative, Bangladesh*

Abstract: *Now a day's, most of the people are depended on the transaction over internet through which they carry out their daily task such as, banking transition, shopping etc. So, it is most important to ensure the security of the data that are transmitted over the internet. In this regard, we proposed a partial blind signature scheme based on Elliptic Curve Discrete Logarithm Problem (ECDLP) which is an extension of blind signature. In partial blind signature scheme, both user and signer have some common agreed information. The user only blind the message but the common agreed information remain unblind to the signer. Using common agreed information signer can find out the user but he cannot trace message signature pair. The proposed scheme satisfies all the security properties of blind signature. As, the scheme is based on ECDLP so it can provide same level of security using smaller key size compared to compared to other public cryptosystem such as, RSA. Thus, it requires less storage space. The proposed scheme also reduces the computational cost compared to other existing scheme. The scheme uses discrete logarithm problem which is difficult to solve. So, we can effectively use the proposed scheme in electronic payment system.*

Keywords: *Partial blind signature, ECC, Discrete logarithm problem, RSA, ECDLP*

Date of Submission: 26-01-2018 Date of acceptance: 09-02-2018

I. Introduction

Now a day's transaction over the internet has enlarged immensely where people can carry out their daily tasks, such as banking transactions, shopping without leaving their homes by using internet, which has increased the growing rate of the e-commerce. So, it is necessary to protect data that are transmitted through internet by using secure and reliable cryptosystem from an unauthorized or illegal third party. In this purpose, we proposed a partial blind scheme based on Elliptic Curve Cryptography based on discrete logarithms problem. In a partial blind signature protocol, the signer and the requester have some common agreed information such as, the value of the bill and the expiration date are visible by the signer but other data such as serial numbers are still invisible and blindly signed.. 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[1]. Recently, Cao, Lin and Xue[2] proposed such a partially blind signature protocol.

1.1 Background

The concept of blind signature was first introduced by Chaum in 1982[3]. Any blind signature must satisfying two properties: Blindness and untraceability Blindness means the content of a message should be blind to the signer. Untraceability is satisfied if, whenever a blind signature is revealed to the public, the signer will be unable to know who the owner of the signature is. In the literature, several applications of blind signature schemes have been developed through the e-commerce and e-voting fields. In 1995, Camenisch[4] and al. proposed a novel blind signature scheme based on the Discrete Logarithm Problem (DLP). But it fails the untraceability. Blind signature scheme suggested by Camenisch and al. has been proved by Lee and al. that it does not satisfy correctness property. In 2005, Wu and Wang proved the untraceability of the Camenisch and al.'s scheme. They corrected the proof of Lee and al. untraceability and concluded that Camenisch and al.'s scheme is still more efficient than Lee and al. The definitions of security and partial blind signature can be found in Juels [5] , Pointcheval[6] , and Pointcheval and Stern[7]. Abe and Fujisaki were the pioneers of the concept of partial blind signatures. The partial blind signature provides a signer with common agreed information which is clearly evident in spite of the blinding process.

Later, Fan and Lei [8] have proposed the partially blind based on quadratic residue problem in which there is no need for modular exponentiation and inverse computations to be performed by the signature requesters.

Furthermore, it needs several modular additions and multiplications for receivers to get and verify a signature in their protocol. They minimizes the number of computations for the signature requesters or users by nearly 98 under a 1024-bit modulus, however it does not reduce the load of computation for the signer.

However later, Zhang and Chen [9] have proved that the scheme proposed by Huang and Chang is not secured, where any malevolent requester can confiscate the embedded public common information from the signer's signature and get a partially blind signature with special public information.

II. Methodology

The purpose of this chapter is to introduce our proposed partially blind signature scheme. But, before describing our scheme we would like to mention about blind signature, hash function and elliptic curve cryptography as follows.

2.1 Analysis on Blind Signature

2.1.1 Properties of Blind Signature

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

2.1.2 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.

2.1.3 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. [10, 11, 12, 13] 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 [14, 15, 16, 17].

Fair Blind Signature

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 trustee used to do that. To do so, the trustee views all the parts of the blinding process [18, 19]. 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 [20, 21] are developed in

which the trustees need to keep a public-private key pair. The trustee can only involve 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 partial blind signature is also used. To trace the identity of the malicious user, the signer needs 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.

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 Analysis on Elliptic Curve Cryptosystem

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,

- Integer factorization problem (IFP)
- Discrete Logarithm Problem (DLP)
- Elliptic Curve Discrete Logarithm Problem (ECDLP)

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.

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.

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.

2.3.1 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:

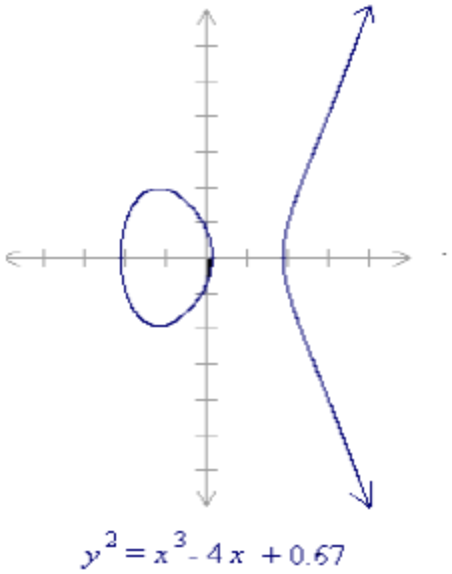

Figure 1: An Elliptic Curve over real numbers

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.

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

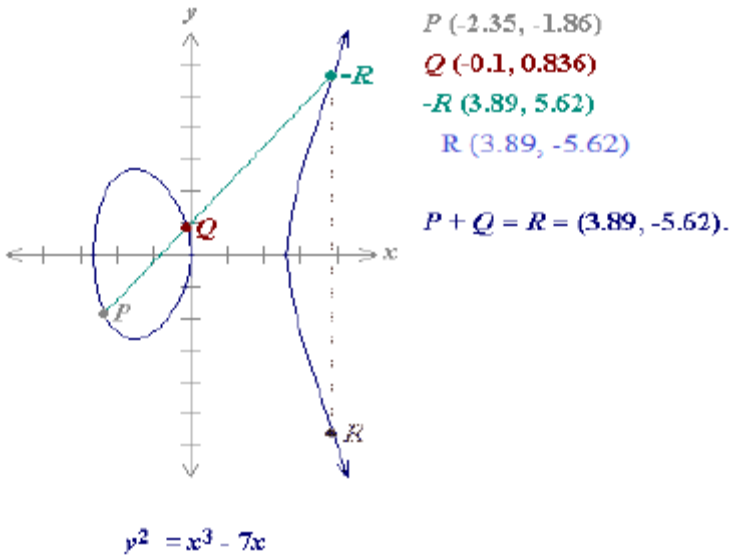

Figure 2: Adding distinct points P and Q of an elliptic curve over real numbers

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.

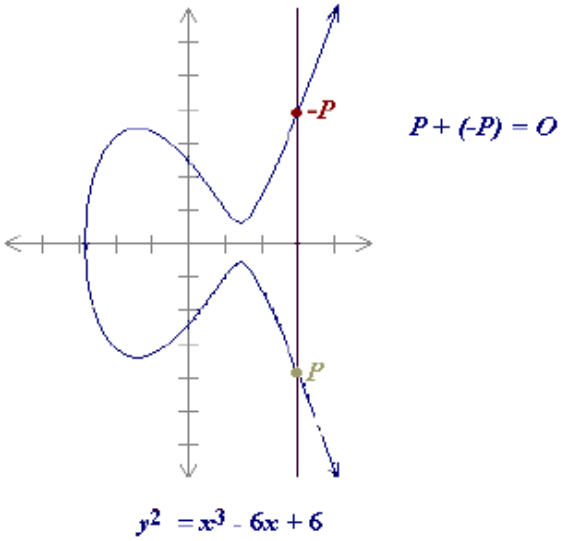

Figure 3: Showing the additive identity of an elliptic curve over real numbers

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

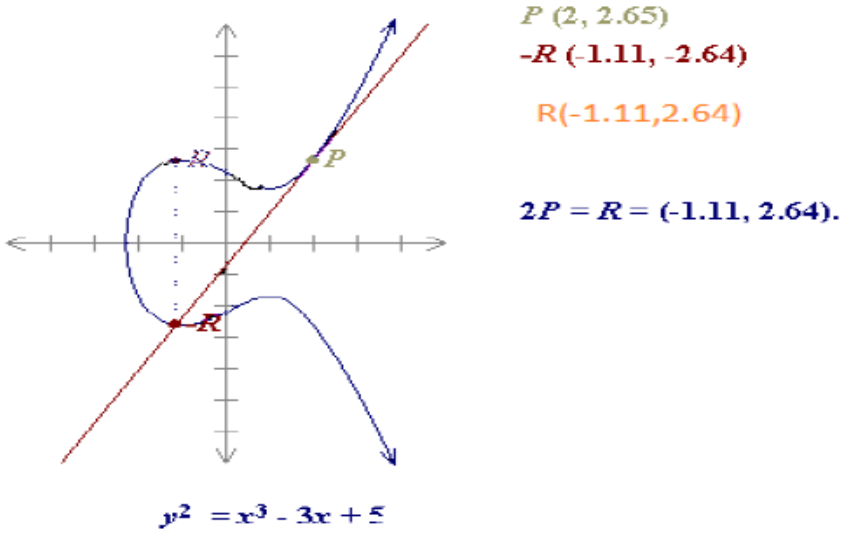

Figure 4: Adding a point to itself in an elliptic curve over real numbers

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)

2.3.2 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.

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. 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

2.3.3 Elliptic curve over F (2m)

Recall that 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:

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.

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.

2.3.4 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.

2.3.5 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.

2.4 Proposed Partially Blind Signature Scheme Based on ECDLP for Untraceable Electronic Payment System

This section explains about our proposed Partial Blind Signature (PBS) scheme based on the elliptic curve discrete logarithm problems. The scheme can be divided into five phases:

- Initialization,
- Blinding,
- Signing,
- Unblinding,
- Verifying.

The signer publishes the necessary information in the initialization. In the blind phase, a user sends the blinded data and the common information to the signer. In the signing phase, the signer signs the blinded data with this common information imposed on it and then sends the result back to the requester. Finally, the user unblinds the signature from the signed data in the unblinding phase. In the verifying phase anyone can verify the signature of the signer using signer's public key. The details of the proposed partially blind signature scheme are described as follows.

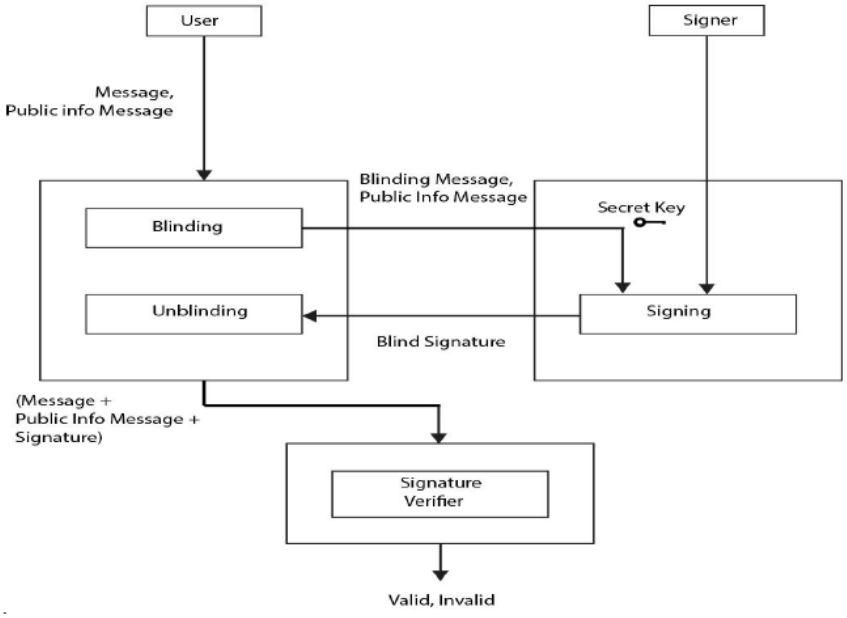

Figure 5: Flow diagrams for partial blind signature

Initialization Phase:
Actors: Customer (User)
Bank (Signer)
Merchant (Verifier)
All actors are agree on elliptic curve parameters
Curve:
Signer generates his private key by choosing random value x in [1, n-1] range, then he computes and generates his public key Q = x*P where p is a base point on the elliptic curve.

Blinding Phase:
Signer holds record for User's and Verifier's account balances; User wants to make a payment to Verifier. For that purpose:

- User generates a token message M = "This is 10 dollar Serial No. 000111222"
- User generates public info message C = "Nominal: 10, Currency: USD, Expiry date: 2020-01-01 2:00:00C",
- User hashes his token message which results with m = hash(M)
- User hashes his public info message c = hash(C)
- User generates random blinding factor v = a random element
- User blinds his token message u = (m - v) mod n
- User sends (u, C) to Signer

Signing Phase:
Signer charges 10 USD from User's account and blindly signs u by generating pair (s', R')

- For that purpose Signer generates random value r that will be used for protecting her private key x
- Then Signer computes s' = (x - r + u + hash(C)) mod n = (x - r + u + c) mod n
- And R' = r * P
- Signer sends the resulting (s', R') pair to Signer

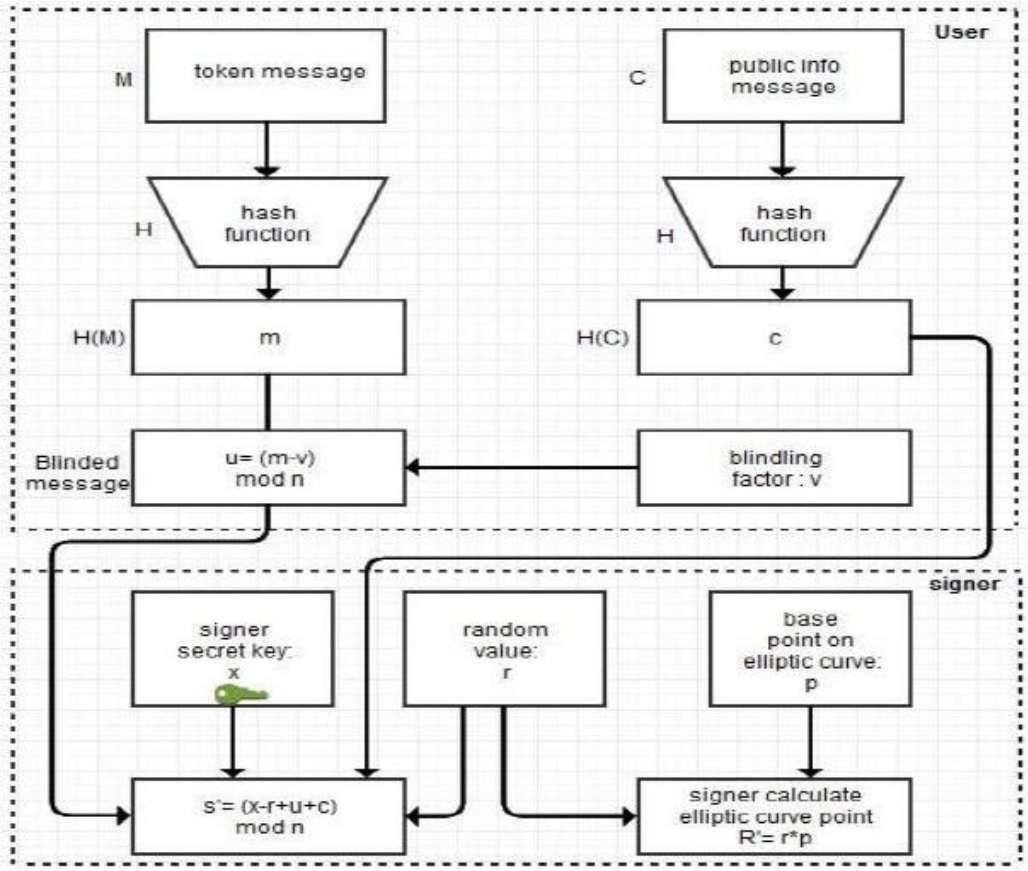

Figure 6: "Blinding" and "Signing" phase of the proposed scheme

Unbinding Phase:
By receiving (s', R') User unblinds the message into (s, R), this is done according to following steps:

- User generates random unblinding factor w = a random element
- User then computes s = (s' + w - m) mod n
- User then computes R = R' + (m - w - v)*P
- User then pays for services from Merchant by sending him tuple (s, R, C, M)

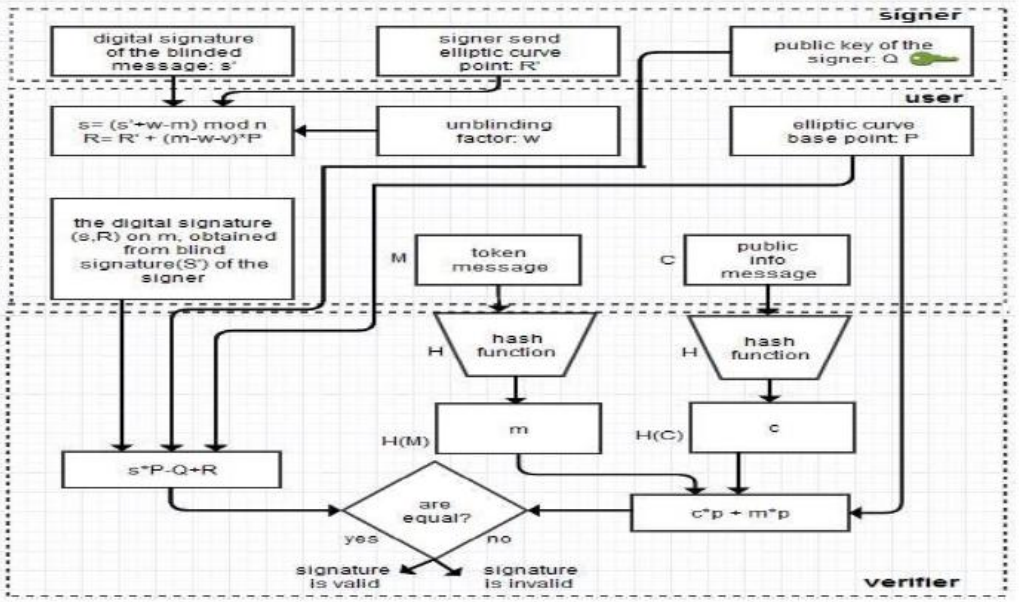

Figure 7: "Unblinding" and "Verifying" phase of the proposed scheme

Verifying Phase:
By receiving User's tuple (s, R, C, M) Verifier validates:

- Token message M = "This is 10 dollar Serial No. 000111222"
- Public info C = "Nominal: 10, Currency: USD, Expiry date: 2020-01-01 12:00:00C"
- Verifier then verify the signature (s, R) by checking equation s*P - Q + R = c*P + m*P

Table 1: Interpretation of abbreviations used throughout the Section 2.4

Abbreviation	Interpretation
P	Base point
N	Order of base point
V	Blinding factor
W	Unblinding factor
M	Message
C	Pubic info message
M	Blinded message
X	Private key of the signer
Q	Public key of the signer
S	Signature
S'	Blinded signature
R	Random value
R,R'	Point on elliptic curve
H(:)	Hash function

III. Results And Discussion

3.1 Security Analysis

Proof of Partial Blindness:
We used w (unblinding factor) in the unblinding phase and v (blinding factor) in the blind phase. The signer can never find w and v so blind property is correctly achieved. But, the public info message remains unblind to the signer through which she can trace the identity of the user when needed which satisfy the partial blindness property. Blindness is the first important property in a blind signature.

Proof of Untraceability:
After performing validation checks Merchant asks Signer to add 10 USD to his account by sending her the tuple he received from Customer (s, R, C, M)
When Signer receives (s, R, C, M)

- She validates C = "Nominal: 10, Currency: USD, Expiry date: 2020-01-01 12:00:00C"
- She validates M = "This is 10 dollar Serial No. 000111222"
- She adds (000111222, 2020-01-01 12:00:00C) to her database of expired tokens
- The token will be deleted after 2020-01-01 12:00:00C preventing the database to grow indefinitely

Signer is able to verify her own signature (s, R, C, M) by repeating steps made by Merchant though she is not able to track who made a payment to Merchant. It's true because even if Signer has recorded the s' component in her database during signature phase, she will not be able to match it because during repayment phase she will receive (s, R) instead of (s', R') and so by subtracting recorded s' from s she will get:
s - s' = ((x - r + v + c+ w) - (x - r + v + m + c) + m + c) mod n =
= (x - r + v + c + w - x + r - v - m - c + m + c) mod n =
= w mod n
Therefore, it is impossible for signer to discover w that satisfies the untraceable property of the algorithm.

Proof of Unforgebility:
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 R, m, he/she cannot obtain s. Because s*P - Q + R = c*P + m*P and s is unknown. It is an elliptic curve discrete logarithm problem and difficult to solve. Situation 2: If someone gets s, m, he/she cannot obtain R. Because s*P - Q + R = c*P + m*P and R is unknown, .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. It is an elliptic curve discrete logarithm problem and difficult to solve.
Proof of Correctness:
Anyone can proof the correctness of the algorithm by verifying the following equation:
s*P - Q + R = c*P + m*P
• Since s = s' + w - m then s is also equal to x - r + v + c + w
• Since R = R' + (m - w - v)P then R is also equal to r*P + (m - w - v)*P which in turn is equal to r*P + m*P - w*P - v*P
• Then the initial equation
s*P - Q + R = c*P + m*P
=>(x - r + v + c + w)*P - Q + (r*P + m*P - w*P - v*P) = c*P + m*P
=> x*P - r*P + v*P + c*P + w*P - Q + r*P + m*P - w*P - v*P = c*P + m*P
By applying simplification we get that:
⇨ c*P + m*P = c*P + m*P

3.2 Performance
In this chapter, we investigate the performance of the proposed scheme with other scheme to show the efficiency of our propose scheme.
3.2.1 Comparison between Chaum's Scheme[3] and Proposed Scheme
The proposed scheme uses ECC discrete logarithm problem that provide same level of security by using small key sizes as compared to RSA based Chaum's Scheme[1]. Thus, the proposed scheme requires less storage space. It is also secure along with high-speed cryptographic processes. According to [22], the comparison is given below.

Table 2: Comparison of key sizes and computational time for our scheme compared to chaum's scheme

Blind Signature schemes	Average computational time in ms	Key sizes	Security level
Blind Signature based on RSA	220ms	160 bits	73
Blind Signature based on ECC	83ms	1024 bits	73

3.2.2 Comparison between Carmenisch[2] et al.'s Scheme and Proposed Scheme
The computational complexity of any cryptographic algorithm mainly depends upon on four major operations, namely, number of inverse operation, number of hash function, number of exponential and number of multiplication operation. Ignoring the time for performing addition and subtraction operation in the analysis process, the following notations are used to analyze the performance of the proposed scheme with comparison to the existing scheme.
• TE is the time complexity of modular exponentiation
• TM is the time complexity of multiplication
• TH is the time complexity of hash function
• TI is the time complexity of inverse function

Table 3: Computational complexity analysis of our scheme to Carmenisch *et al.*'s Scheme

Phases	Carmenisch *et al.*'s Scheme	Proposed Scheme
Initialization Phase	$2T_E$	$1T_M$
Blinding Phase	$T_I + 4T_M + 2\ T_E$	$2\ T_H$
Singing Phase	$2\ T_M$	$1\ T_M + 1\ T_H$
Unblinding Phase	$3T_E + T_M$	$1\ T_M$
Verifying Phase	$3\ T_M + T_I$	$3T_M$

It is clear that the proposed scheme consists of minimum no. of operations. So, the computation cost of the proposed scheme is much less than Carmenisch et al.'s Scheme.

IV. Conclusion

The proposed scheme recommends a secure and untraceable partial blind scheme based on ECC Discrete logarithm problem. The scheme also satisfies all the security properties of blind signature. The scheme provide the same security level using small key sizes as compared to RSA because the scheme is based on ECDLP. The scheme has lower computational complexity. So, it can reduce the computational cost along with lower storage space as compared to other scheme. Thus, the proposed scheme can be effectively applied in electronic payment system. There are different variations of blind signature. Here, we use partial blind signature for our proposed scheme. In future, we want to implement our scheme in other variation of blind signature such as, proxy blind signature, restrictive blind signature and fair blind signature.

References

[1]. D. Chaum, "Blind signature for untraceable payments," Advances in Cryptology, Proceedings of CRYPTO '82, pp. 199–203, 2000.
[2]. A. Juels, M. Luby, and R. Ostrovsky, "Security of blind signatures," Advances in Cryptology - Crypto 1997, LNCS 1294. Springer-Verlag, pp. 150–164., 1997.
[3]. D. Pointcheval, "Strengthened security for blind signatures," Advances in Cryptology -Eurocrypt 1998, LNCS 1403. Springer-Verlag,, pp. 391– 403, 1998.
[4]. D. Pointcheval and J. Stern, "Provably secure blind signature schemes," Advances in Cryptology - Asiacrypt 1996, LNCS 1163. Springer- Verlag, pp. 252–265, 1996.
[5]. T. Abe, M. Okamoto, "Provably secure partially blind signatures," Advances In Cryptology-Crypto,LNCS 1880.Springer-Verlag, vol. 5, pp. 271–286, 2000.
[6]. J. Carmenisch, J. Piveteau, and M. Stadler, \Blind signatures based on discrete logarithm problem," in Advances in Cryptology, EUROCRYPT' 94, pp. 428{432, Lecture Notes in Computer Science, 950, 1994.
[7]. C. I. Fan and C. Lei, "Low-computation partially blind signatures for electronic cash," IEICE Transaction on Fundamentals, vol. E81-A, pp. 199–203, 1998.
[8]. F. Zhang and X. Chen, "Cryptanalysis of huangchang partially blind signature scheme," The Journal of Systems and Software, vol. 76, pp. 76: 323–325, 2005.
[9]. B.A.Farouzan, "Cryptography & Network Security. Tata McGraw-Hill Publishing Company Limited", Inc.,New York, 2007.
[10]. Sumita Bari, "An efficient blind digital signature protocol based on elliptic curve", International Journal of Technology Enhancement and Engineering Research, vol 2, Issue 9 ISSN 2347-4289, 2014.
[11]. Fuwen Liu, "A tutorial on elliptic curve cryotography (ECC)", Brandenburg Technical University of Cottbus Computer Networking Group.
[12]. Masayuki Abe and Tatsuaki Okamoto. Provably secure partially blind signatures, 2000.
[13]. D. Chaum, M. Jakobsson, R. L. Rivest, P. Y. A. Ryan, J. Be-naloh, 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.
[14]. S.Mohanty and B.Majhi, " A secure multi authority electronic voting protocol based on blind signature", In Proceedings of the 2010 International Conference on Advances in Computer Engineering, pages 271–273. IEEE Computer Society, 2010.
[15]. Y.Tsiounis Y.Frankel and M.Yung. Indirect Discourse Proofs.
[16]. D. Chaum and T. P. Pedersen. Wallet databases with observers. In Ernest F. Brickell, editor, Advances in Cryptology - CRYPTO 92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings, volume 740 of Lecture Notes in Computer Science, pages 89–105. Springer, 1992
[17]. Jan Camenisch, Ueli Maurer, and Markus Stadler, "Digital payment systems with passive anonymity-revoking trustees", J. Comput. Secur., 5(1):69–89, 1997.
[18]. Weidong and Qiu, " Converting normal dlp-based signatures into blind. Applied Mathematics and Computation", 170(1):657–665, 2009.
[19]. 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.
[20]. M.Michels P.Hoster and H.Petersen, "Comment: cryptanalysis of blind signatures based on discrete logarithm problem", Electronic Letters, 31(21):1827, 1995.
[21]. L.Harn, "Cryptanalysis of blind signatures based on discrete logarithm problem", Electronic Letters, 31(14):1136–1137, 1995.
[22]. Nitu Singh, Sumanjit Das, "Cryptanalysis of Blind Signature Schemes", International Journal of Computer Applications (0975 – 8887) Volume 71– No.19, June 2013.

Mostary Begum, Md. Tareq Hasan, "Partially Blind Signature Scheme Based On ECDLP for Untraceable Electronic Payment System." IOSR Journal of Computer Engineering (IOSR-JCE) 20.1 (2018): PP 15-25.