ISSN 2250-3153

Implementation of Blind Digital Signature Using ECC and Zero Knowledge Protocol

Ms. Dhanashree M. Kuthe , Prof. Avinash J. Agrawal

Abstract- An Elliptic Curve Cryptography scheme for blind digital signature is proposed where a sender proves his/her identity using zero knowledge protocol. In this scheme, the ECC is used to implement blind digital signature and the zero knowledge is used at time of signing where the requester needs to prove his/her identity.

The ECC is used because of its difficulty of solving discrete logarithmic problem. The proposed scheme uses zero knowledge protocol to hide the identity of the requester. Hence we maintain the untraceability of the requester and also anonymity of the requester is achieved. Here, in the proposed scheme verifier is not supplied with any of the secret facts of the requester even then the sender is verified using a verifying factor. Hence, the identity of the sender is kept in secrecy.

Zero knowledge protocol is used because allow identification without leaking any secret information during the conversation and with smaller computational requirements.

Index Terms- Blind digital signature, elliptic curve cryptography, zero knowledge protocol.

I. INTRODUCTION

Now days, online communication is at its hike, many a times data travelling over the communication links is secret and the entire users ought to be authenticated for many of application they use. This is best served by implementing Blind Digital Signature. This blind digital signature is best implemented in the application where secrecy of the user's data is to be conserved. Blind Digital Signature was first introduced by David Chaum in with the help of a carbon lined envelop which finely explained the concept. The scheme goes as the sender requests for a digital signature as an authentication to his message. The signing authority in return provides with a digital signature but without gaining knowledge about any of the message contents. And hence, the innovation of digital signatures as Blind Digital Signature.

Now, why would one sign a document unless he does not know the contents of the document? The answer is that Blind Digital Signature seems to mean that the authority signs the document blindly but, that's not the case. Basically, the concept is that the user is authenticated for his identity from the signing authority and not for the message that too without any knowledge of message contents. Then obtained Blind Digital Signature can be verified as the traditional Digital Signature for the same unblinded message. Blind Signatures are very useful in applications that guarantee the anonymity of the participants . The important application of blind digital signature is electronic voting and electronic cash.

1) Basics of Elliptic Curve Cryptography

In 1985, Elliptic Curve Cryptography (ECC) was proposed by Neal Koblitz and Victor Miller.ECC is capable of improving the existed cryptogram systems in terms of having smaller system parameter, smaller public-key certificates, lower bandwidth usage, faster implementations, lower power requirements, and smaller hardware processor requirements. Therefore, using ECC to build a cryptosystem is commendable by the reasons of high security and efficiency [14]. The mathematic settings of ECC are depicted below.

The elliptic curves can be categorized into two classes non prime and prime elliptic curves .The elliptic curve cryptography is based on the elliptic curve equation which is given as:

$$y^2 = x^3 + ax + b$$

To plot an elliptic curve one needs to compute:

$$y = \text{sqrt}(x^3 + ax + b)$$

So, value of y is calculated for each value of x, symmetric about y = 0 where values of a and b will be given. Groups are defined based on the set E (a, b) for values of a and b such that:

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

Non - Prime Curves:

Here, is a point of infinity called as the "Zero Point" which is the third point of intersection of a straight line across the elliptic curve. One point that is to be noted is when three point on elliptic curve lie on a straight line they sum up to zero. There are some rules for operation addition '+'for elliptic curve points to follow. Those all are listed down as:

1) If point is O then
O = -O

2) If point P on the curve then
P + O = P

3) If two are P and negative of then
i.e. P ≡ (x,y) and -P ≡ (x,-y)
P+ (-P) = P – P = O

4) If P and Q are two distinct points the addition is as follows :
a) Draw a straight line between P and Q

b) Extend the line and find the third point of intersection with the elliptic curve 'R'
c) To form the Group adds these three points as:

$$P + Q = -R$$

Thus, P + Q are the mirror image of the point R.

5) If both the points are the same point P then the steps are as follows :
a) Draw a tangent through point P
b) P + P = 2P = -R

Prime Curves:

In case of these curve the cubic is applied. For prime curves a large prime number p is assumed, and values of all of the variables and coefficients are selected within the range of 0 to p-1 such that the following condition is satisfied.
T
he condition is:

$$y^2 \bmod p = (x^3 + ax + b) \bmod p$$

Example: a = 1, b = 1, x = 9, y = 7, p = 23

$$7^2 \bmod 23 = (9^3 + 9 + 1) \bmod 23$$

$$3 = 3$$

II. THE DISCRETE LOGARITHMIC PROBLEM

Consider and elliptic curve E over a finite field Fq . Let points P and Q are the point belonging to the curve. The equation is given as:

$$Q = kP$$

Now, calculating Q is very straight forward but calculating the value of k is very difficult. This particular difficulty is called discrete logarithmic problem. In words one can say that 'k' is the logarithm of P to the base Q which is very difficult to calculate. And the ECC is very is secure of the entire available cryptographic algorithm.

III. WHAT IS A 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 (i.e., the protocol requires parties to provide zero-knowledge proofs of the correctness of their secret-based actions, without revealing these secrets).

IV. PROPOSED SCHEME

The proposed scheme involves the Six phase as follows:

1. Initialization and Key Generation Phase
2. Blinding Phase
3. Proving and Signing Phase
4. Unblinding Phase
5. Verification Phase

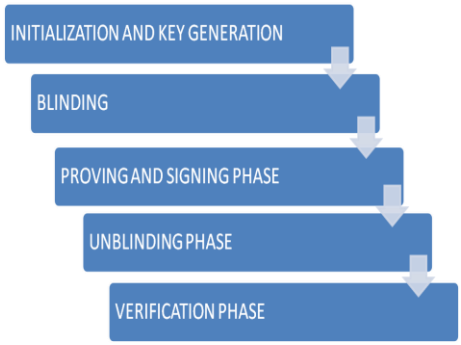


Fig.5.1. Phases of the proposed scheme.

Phase I: Initialization:

In this phase, the elliptic curve is formed by providing the value of coefficients a and b of the equation of the elliptic curve i.e

$$y^2 = x + ax^2 + b$$

Generation of ECC :$y^2 = x^3 + ax + b$

Condition: $4a^3 + 27b^2 \neq 0$

Compute: $y = \text{sqrt}(x^3 + ax + b)$

Example: $y^2 \bmod p = (x^3 + ax + b) \bmod p$

$a = 1, b = 1, x = 9, y = 7, p = 23$

$7^2 \bmod 23 = (9^3 + 9 + 1) \bmod 23$

$3 = 3$

Also all the points on elliptic curve are generated and displayed using the field arithmetic. The calculation of the points is done using "Adding and Doubling" method. For example, consider the following:

Point Addition: $J + K = L \quad K \neq -J$

Point Doubling: $2J = L \quad K = J$

Example: Let k = 11

So, KP=11 x P = 2(2(2P) +P) +P)

The above calculations can be analytically expressed as follows:
Adding Points

Let, J (XJ, YJ) K (XK, YK) L (XL, YL) be the points
J + K = L

XL = (s – XJ - XK) mod P

YL = (YJ + s*(XJ - XL)) mod P

Where, s = Slope of the line passing through the points J and K.

Doubling Points

Let, J (XJ, YJ) be the point.

2J = L (XL, YL)

Where,
$s = ((3\ XJ^2 + a) / 2\ YJ) \bmod P$

XL = (S2 – 2XJ) mod p

YL = (-YJ + (s*(XJ - XL)) mod P

Key Generation:

After the points of the elliptic curve are generated a base point is selected out of them of order n. The private keys and public keys are generated using this base point.

It goes as follows: a number 'k' is chosen randomly between 1 to (P-1) to be served as the private key where P is the large prime number. This private key is then treated with the base point (generating point) of the formed elliptic curve and computes the public key. All of the public keys and private keys in the proposed scheme are generated using the above criteria. The private key is used to encrypt the message and public key is used to decrypt the message and vice –versa.

Phase II: Blinding Phase:

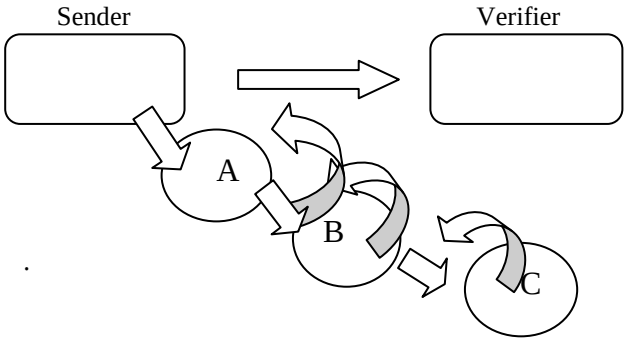
In this phase, after the private key and public key is generated the sender when wants to send the message uses his/her private key to encrypt the message. The sender generates an encrypted message using his private key using the scheme of ECC. The sender then sends the encrypted message to the blinders to blind the message.

We have the three blinders and each of message encountering should get blinded by all the three signers. The criteria of signing go as follows:

After the encrypted message encounters one of blinder is selected arbitrarily for blinding the encrypted the message. This blinded message is again scrambled by arbitrarily selected blinders but this time the blinder is from the remaining two signers. This double blinded message is finally blinded by the last remaining blinder. Now the last blinder sends the encrypted message back to his sender and the process goes on in the reverse direction. The encrypted package will finally reach the original sender from there the package will be moved to the signer for blind digital signature. The Sender encryption for message M using his/ her private key 'k':

$$M' = k\ (M)$$

This completes the blinding and message is passed on to the Signers



.

Fig.5.2. The Blinding Phase

Phase III: Proving and Signing

Signer after receiving the decrypted message calculates for the actual message value. The signer uses the zero knowledge concepts for the calculations. In this phase, the signer uses zero knowledge concepts for the verification. The signer asks for the value of e from the sender whose value is either 1 or 0. On the basis of this value the signer verifies the message to be the same as the sender's message

1) Stage 1: The sender chooses a random number r calculates r^2modP and transmits to the verifier V.

2) Stage 2: The signer S now chooses one of two questions to ask the sender. The signer S can ask either for the value of the product (rk) mod P, or for the value of r that the sender has just chosen. This is generally performed by S, sending a bit e to sender, indicating its choice of question, referred to as the challenge, such that the sender has to provide the answer,

$y = rk^e$modP, where e (0, 1). Sender can answer both correctly if it knows the secret k.

Stage 3: The sender provides $y = rk^e \bmod P$ as requested and the signer checks the result as follows. If the challenge is for e=1, the signer expects to have received rk mod P. The signer cannot deduce any information about k from this, because r is a random number not known to signer. Therefore, the verifier checks y^2modP, which should be $((rk \bmod P)^2 \bmod P)$ is the same as

$r^2 \times k^2$modP. The verifier received r^2 from sender in stage 1 of this round, and selects vk. If the challenge is for e = 0, the verifier expects to have received r, and checks that its square matches the value of r^2 mod P provided in stage 1.

After verification of the sender the signer uses his/her private key and signs the message and sends it back to the requester.

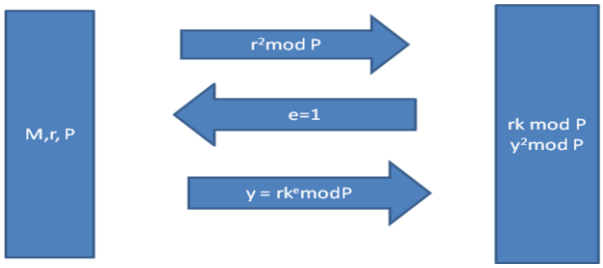


Fig.5.3. Proving for e=1

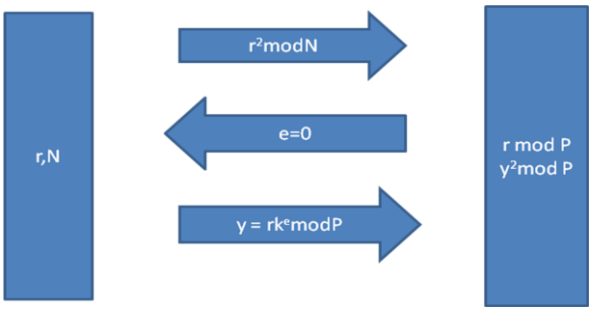


Fig.5.4. Proving for e=1

Here, now the sender is proved be a valid requester and hence he/she is authenticated and blind digital signature is done by the signer as he encrypts the message with his private key and sends the message signature pair back to the requester.

Phase IV: Unblinding

This phase is optional, it is on user's choice whether he/she wants to unblind the message or not. If the user wishes to unblind the so he will do that using his own public key and the public key of the blinders. The message needs to be unblinded in the application like the electronic voting as to the counter

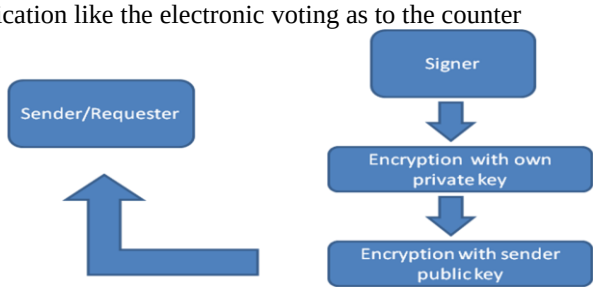
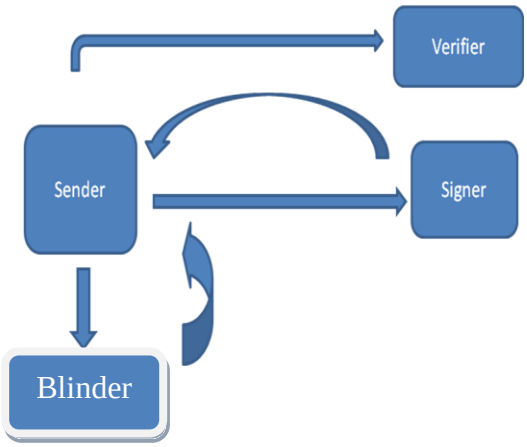


Fig.5.5. Blind Digital Signature passed on to the sender

Phase V: Verification:

Here, the blind digital signature with requester/sender can be verified by any verifying authority using signer's public key which can be treated as the simple digital signature.



5.6. The flow of the system

V. RESULT

The above proposed scheme shows you that the sender of the message proves his identity without revealing any of his/her facts to the verifier using the zero knowledge protocol. Also the concept of three blinder who are selected randomly for the blinding purpose add on three layers complexity to recognize as to who is the sender in actual. The above proposed scheme can also be served as a skeleton to implement application where the anonymity of the user is to be maintained.