A Novel protocol for smart card using ECDLP

Bibhudendu Panda, B.Majhi and S.K Jena

*Abstract:***In this paper, we propose a novel protocol for smart card using Elliptic Curve Discreet Logarithmic Problem(ECDLP). It is believed that smart card technology application should benefit more from advantages of public key cryptography specifically in initiation and maintenance of secure channel. This paper introduces a public key cryptographic protocol for secure entity authentication, data integrity and data confidentiality. The proposed secure channel protocol uses a combination of secure public key system, secret key and a public encryption system to achieve the desired goal. Signature authentication along with sign encryption for smart card based on Digital Signature Authentication (DSA) using (ECDLP) has been proposed. Recently as all the application transaction is bound to happen in smart card, bandwidth utilization, power consumption and above all minimization of computation for better utilization of memory are major challenges. Sign encryption is useful in protocols that guarantee the anonymity of the participants and its message. The proposed scheme can be easily extended to M-Commerce, Financial transaction and Healthcare applications where the requester needs a signature on the message.**

Keywords: **Smart Card,Signature,Elliptic Curve, ElGamal, Digital Cash, RSA**.

I. INTRODUCTION

A Elliptic Curve over Finite Field

The use of Elliptic Curve Cryptography was initially suggested by Neal Koblitz [3] and Victor S. Miller [2] and after that many researchers have suggested different application of Elliptic Curve Cryptosystems. Elliptic curve cryptosystems over finite field have some advantages like the key size can be much smaller compared to other cryptosystems like RSA, Diffie-Hellman since only exponential-time attack is known so far if the curve is carefully chosen [3] and elliptic curve discrete logarithms might be still intractable even if factoring and multiplicative group discrete logarithm are broken. ECC is also computationally efficient than the first generation public key systems such as RSA and Diffie-Hellman.

B. Elliptic Curve Groups over F_q

A non-super singular Elliptic curve E over F_q can be written as:

$$E : y^2 \bmod q = (x^3 + ax + b) \bmod q \qquad \cdots\cdots(1)$$

$$where\ (4a^3 + 27b) \bmod q \neq 0$$

The point P in the Elliptic curves is described by the coordinates (x, y) $where\ x, y \in F_q$ that satisfy the equation (2) together with a “point of infinity” denoted by O form an abelian group $(E,+,O)$ whose identity element is O.

C. Addition of two distinct points *P* and *Q*

The negative of the point $P = (x_1, y_1)$ is the point $-P = (x_1, -y_1)$. If $P(x_p, y_p)$ and $Q(x_q, y_q)$ are two distinct points such that P is not $-Q$, then

$$P + Q = R \qquad \cdots\cdots(2)$$

$$where \qquad R = (x_r, y_r)$$

$$s = (y_p - y_q)/(x_p - x_q) \bmod q,$$

$$where \quad s\ is\ the\ slope\ of\ the\ line\ pass ing\ through\ P\ and\ Q.$$

$$x_r = (s^2 - x_p - x_q) \bmod q\ and\ y_r = (-y_p + s*(x_p - x_r)) \bmod q$$

D:Doubling the point P

Provided that y_p is not 0,

$$2P = R(x_r, y_r) \qquad \cdots\cdots(3)$$

$$where$$

$$s = ((3x_p{}^2 + a)/(2y_p)) \bmod q$$

$$x_r = (s^2 - 2x_p) \bmod q\ and$$

$$y_r = (-y_p + s*(x_p - x_r)) \bmod q$$

The Elliptic Curve Discrete Logarithm Problem *(ECDLP)* is defined as:

Definition 1. Let E be an elliptic curve over a finite field F_q and let $P \in E(F_q)$ be a point of order n. Given $Q \in E(F_q)$, the elliptic curve discrete logarithm problem is to find the integer $d \in [0, n-1]$, such that $Q = dP$.

II. PROPOSED PROTOCOL FOR SMART CARD

We propose a novel efficient and low computation protocol.

Initially the curve parameters [7, 8, 9] must be agreed upon by terminal and certifying authority. Signer must have a key pair suitable for elliptic curve cryptography[10,11], consisting of a private key d_u (a randomly selected in the interval [1,n-1]) and a public key Q where $Q = d_u G$ .When a signer wants to send a signed message *m* to receiver, he/she must generate a digital signature[4]. The overall protocol has been described in following steps.

Bibhudendu Panda is M Tech(Res) student at NIT,Rourkela.
Dr. B.Majhi, and Dr S.K Jena are professor at ,NIT,Rourkela

1. *Terminal* is a host defined as an off-card entity that requires establishing a secure channel with the smart card, application or smart card operating system (SCOS) as shown in Fig. 1, i.e. terminal initialization has to be made.
2. Card/User represents smart card. Typically a sufficient tamper resistant device which is relatively difficult to compromise; it has access to a variety of cryptographic algorithms and a good random number generator. A multi-application smart card platform will provide significant functionality that will strengthen the overall concept of dynamic application management as shown in Fig. 2, i.e. card initialization has to be made.
3. All entities share public values i.e. large prime multiplicative order modulo *p*.
4. Each card has a Diffie-Hellman key agreement key pair. More specifically, card *has* private key agreement key *y* with corresponding public key Q_s. The card's key pair can be either generated off-card by the issuer or the application provider and subsequently loaded onto the card, or it can be generated on-card (if the functionality is provided by the card). In either case the public key has to be certified by the corresponding off-card entity, i.e. the issuer or an application provider.
5. The terminal has an ECC public encryption key, which is certified by the corresponding certification authority as shown in Fig 1.
6. The card and the terminal share a symmetric cryptosystem and a key generation function (e.g. a one-way function).
7. The card is capable of generating random numbers.
8. Each card (e.g. through a security domain) has a trusted copy of its owner's (e.g. certification authority, issuer or application provider) public certification key whose corresponding private key is used by the off-card entity for issuing certificates (i.e. for the ECC keys) as shown in Fig 3.

Select k randomly between [1, n-1] and generate $R, r \text{ and } s$ as:

After receiving (r, s) from signer, the receiver can verify the correctness of the signature on the message.

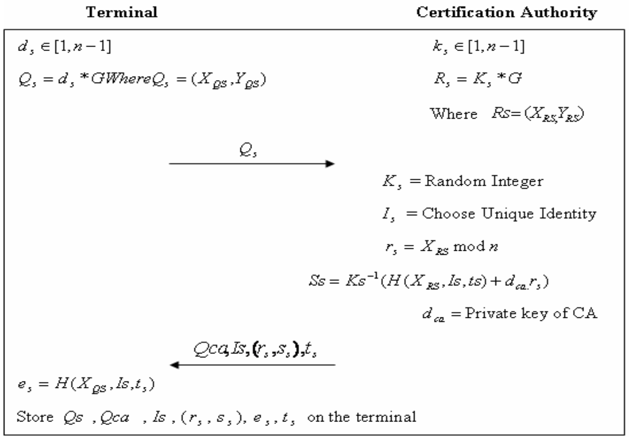

Terminal
Certification Authority
$d_s \in [1, n-1]$
$k_s \in [1, n-1]$
$Q_s = d_s * G Where Q_s = (X_{QS}, Y_{QS})$
$R_s = K_s * G$
Where $Rs = (X_{RS,} Y_{RS})$
Q_s
K_s = Random Integer
I_s = Choose Unique Identity
$r_s = X_{RS} \bmod n$
$Ss = Ks^{-1}(H(X_{RS}, Is, ts) + d_{ca.} r_s)$
d_{ca} = Private key of CA
$Qca, Is, (r_s, s_s), t_s$
$e_s = H(X_{QS}, Is, t_s)$
Store Qs, Qca, Is, (r_s, s_s), e_s, t_s on the terminal

Fig.1 Terminal Initialization

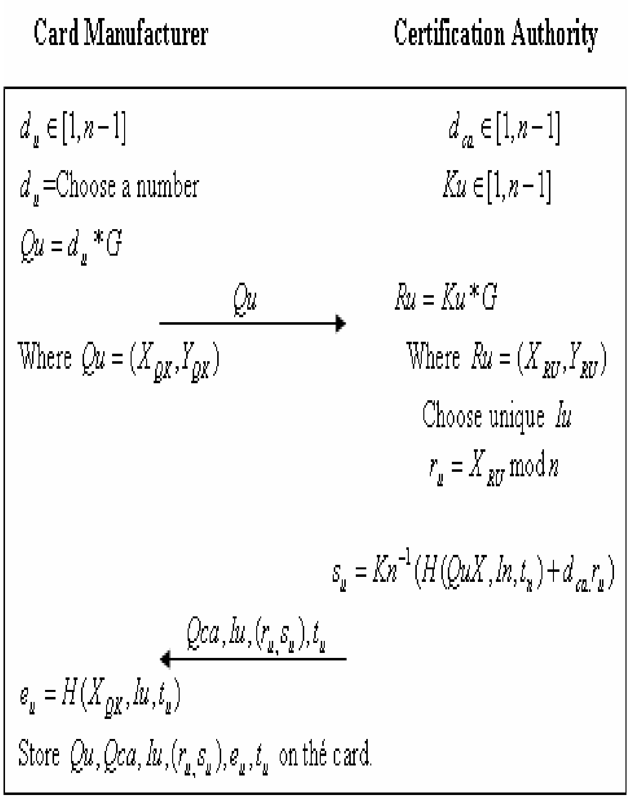

Card Manufacturer
Certification Authority
$d_u \in [1, n-1]$
$d_{ca} \in [1, n-1]$
d_u=Choose a number
$Ku \in [1, n-1]$
$Qu = d_u * G$
Qu
$Ru = Ku * G$
Where $Qu = (X_{QX}, Y_{QX})$
Where $Ru = (X_{RU}, Y_{RU})$
Choose unique Iu
$r_u = X_{RU} \bmod n$
$s_u = Kn^{-1}(H(QuX, In, t_u) + d_{ca.} r_u)$
$Qca, Iu, (r_u, s_u), t_u$
$e_u = H(X_{QX}, Iu, t_u)$
Store $Qu, Qca, Iu, (r_u, s_u), e_u, t_u$ on thé card.

Fig.2 Card Initialization

Mutual authentication and Key agreement between the Terminal and the smart card is described in Fig 3. When ever there is a service request either by the card or by the terminal there is an immediate key exchange. Once both the party's have the public key of the other then by using their private key they can generate the secret key to encrypt the data required to have the mutual authentication. To protect the certificate from eavesdropper it is send through the encrypted format using the mutually secret key x_{QK}. The server concatenates the certificate through the $e_{s,}(r_{s,}s_{s),}t_{s,}g)$, to obtain the final mutual key of authentication.

The encrypted message Co is then sent to the user then decrypts Co and then obtains the certificate and the generator g. The user then encrypts the data with the concatenation of $e_{u, (}r_{u,}$ $s_{u)}$ with the certificate expiration date t_u and the random generator g, the encrypted data is known as C1, it is the send to the server which is then decrypted with the mutually agreed key and then checks that whether g and t_u are valid or not, if it is valid then the server finds $w = S_s^{-1}$ and then finds $u1 = w * e_u$, $u2 = w * r_u$, $R = u1G + u2Q_{ca}, V = x_r \bmod n$ and by using the previously known generator g to find Kf which will be final session key and with the help of Kf similarly at the user end message is encrypted and send to the server where verification is done and message is decrypted as shown in Fig. 3.

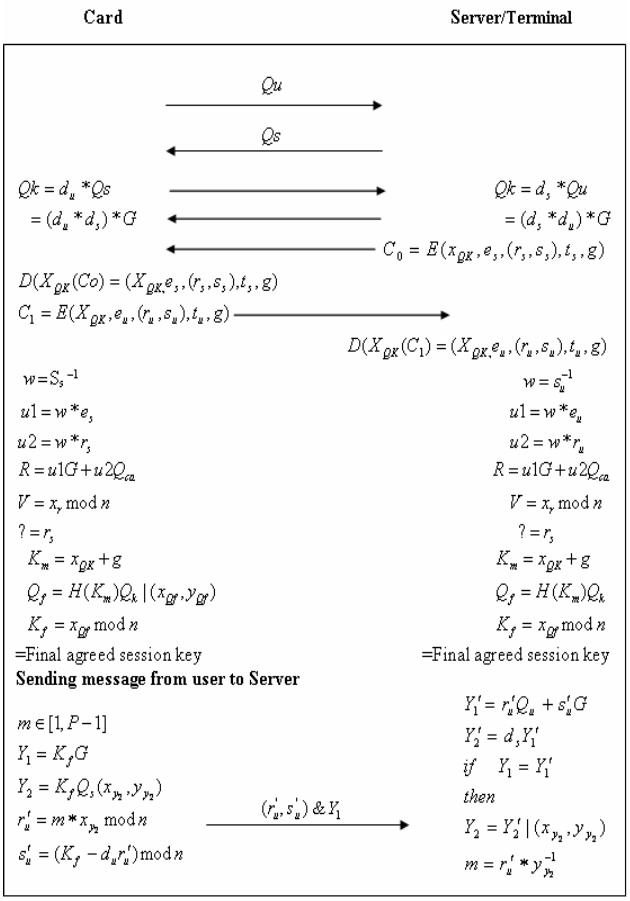

Card
Server/Terminal
Qu
Qs
$Qk = d_u * Qs$
$= (d_u * d_s) * G$
$Qk = d_s * Qu$
$= (d_s * d_u) * G$
$C_0 = E(x_{QK}, e_s, (r_s, s_s), t_s, g)$
$D(X_{QK}(Co) = (X_{QK,}e_s, (r_s, s_s), t_s, g)$
$C_1 = E(X_{QK}, e_u, (r_u, s_u), t_u, g)$
$D(X_{QK}(C_1) = (X_{QK,}e_u, (r_u, s_u), t_u, g)$
$w = S_s^{-1}$
$u1 = w * e_s$
$u2 = w * r_s$
$R = u1G + u2Q_{ca}$
$V = x_r \mod n$
$? = r_s$
$K_m = x_{QK} + g$
$Q_f = H(K_m)Q_k \mid (x_{Qf}, y_{Qf})$
$K_f = x_{Qf} \mod n$
=Final agreed session key
$w = s_u^{-1}$
$u1 = w * e_u$
$u2 = w * r_u$
$R = u1G + u2Q_{ca}$
$V = x_r \mod n$
$? = r_s$
$K_m = x_{QK} + g$
$Q_f = H(K_m)Q_k$
$K_f = x_{Qf} \mod n$
=Final agreed session key
Sending message from user to Server
$m \in [1, P-1]$
$Y_1 = K_f G$
$Y_2 = K_f Q_s (x_{y_2}, y_{y_2})$
$r_u' = m * x_{y_2} \mod n$
$s_u' = (K_f - d_u r_u') \mod n$
$(r_u', s_u') \,\&\, Y_1$
$Y_1' = r_u' Q_u + s_u' G$
$Y_2' = d_s Y_1'$
$if \quad Y_1 = Y_1'$
$then$
$Y_2 = Y_2' \mid (x_{y_2}, y_{y_2})$
$m = r_u' * y_{y_2}^{-1}$

Fig .3 Mutual Authentication & Message Recovery

III CORRECTNESS OF PROPOSED PROTOCOL

The verifier only verifies the pair (r, s) and message m by using the above equation. The correctness of the equation as follows: The verifier has only digital signature *(r,s,R)* of message m for verification. The customer extracts the signature by using above, therefore

$$
\begin{aligned}
Y_1' &= r_u' Q_u + s_u' G \\
&= r_u' d_u G + (K_f - d_u r_u') G \\
&= K_f G \\
&= Y_1
\end{aligned}
$$

If $Y_1' = Y_1$ then

$$
\begin{aligned}
Y_2' &= d_s * Y' \\
&= d_s * Y_1 \\
&= d_s * k_f G \\
&= Y_2
\end{aligned}
$$

IV. TRANSACTION USING THE ABOVE PROPOSED PROTOCOL

The following procedure explained an untraceable off-line electronic payment protocol assuming that the consumer wants to purchase some goods from the merchant and that both have bank accounts with Bank:

A. Request for certificate by user or smart card.

1. Customer asks for certificate from the certifying authority.
2. Certifying Authority issues a certificate to the card by putting the unique identity number and valid period.

B. Request for certificate by Terminal or server.

1. Terminal/Server asks for certificate from the certifying authority.
2. Certifying Authority issues a certificate to the card by putting the unique identity number and valid period.

C. Online Transaction between Card and Terminal/Server.

1. After agreeing on the initial key, final session key is generated with the help of a generator.
2. With the help of new session key message is encrypted along with the certificate then send to the receiver.
3. At the receiving end message and certificate is verified, if it is true then the encrypted message is decrypted using the above proposed protocol.

V. CONCLUSION

This paper suggests a secure and efficient protocol based on the Elliptic Curve Discrete Logarithm Problem for smart card. The scheme utilizes fewer number bits due to inherent property of elliptic curve as compared to its public key counterparts such as RSA. The proposed protocol is suitably illustrated using an online transaction for banking system. The validity of the proposed scheme has been made.

VI. REFERENCES

[1]. Alfred J. Menezes, "*Elliptic curve public key cryptosystem*" Auburn University,Kluwer Academic Publishers ,DordrechVLondon, 1993.

[2]. V. Miller, "*Uses of Elliptic Curve in Cryptography,*" Advances in Cryptography, Proceedings of Crypto'85, Lectures notes on Computer Sciences, 218, Springer-Verlag, 1986, pp. 417-426.

[3]. N. Koblitz, "*Elliptic Curve Cryptosystems,*" Mathematics of Computation, 48, 1987, pp. 203-209.

[4]. Jonson, D., Alfred Menezes. "*Elliptic curve DSA(ECDSA)):An Enhanced DSA.*" 24 ebruary 2000.URL:http://citeseer.nj.nec.com/cache/papers/cs/8755/http:zSzzSzcacr.math.uwaterloo.cazSz~ajmenezezSzpublicationszSzecdsa.pdf/johnson99elliptic.pdf (8 Feb. 2004)

[5]. T. ELGamel,"*A Public Key cryptosystem and a0 signature scheme based on Discreet logarithms*", IEEE Transaction on Information Theory, Vol.It-3 1, No. 4 July (1985), pg 469-481.

[6]. Rivest. R.L. Shamir, and L. Adleman, *"A method for obtaining digital Signatures and Public key cryptosystem''*, Communications of the ACM, v.21, n.2, (1978) 120-126.

[7]. Certicom, "Elliptic Curve Cryptography" **http://www.certicom.coiti/reaserch**

[8]. Koblitz, *"CM-Curves with Good Cryptographic Properties,"* Proceeding of Crypto'91, 1992.

[9]. Doug Stinson, "*Cryptography Theory and Practice*", Second Edition, CRC Press, Inc, 2002

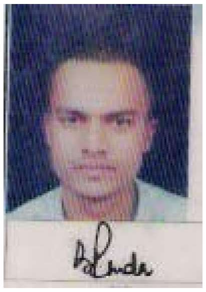
[10]. Menezes, P. van Oorschot, and S. Vanstone, "*Handbook of Applied Cryptography*," CRC Press, 1996.

[11]. Ahmed Khaleed M.Al-Kayali, "Elliptic Curve cryptography and smart cards" GISC Security Essentials Certification (GSEC),2004

[12]. C.P .Schnorr. efficient signature generation by smart cards. Journal of Cryptology, 4(3):161-174, 1991.

[13]. William Stallings, Cryptography and network security: Principles and practice, Second Edition, prentice hall, 1999.

VII. BIIOGRAPHIES



Bibhudendu Panda was born in 1st June,1976.He received his B Tech degree in Computer Science and Engineering and now pursuing M Tech(Res) from NIT,Rourkela.His research areas of interest are Information Security,Data base engineering and Networking.

Dr. S.K. Jena was born in 28 April, 1954. He received his Ph.D. from Indian Institute of Technology, Bombay and M.Tech from Indian Institute of Technology, Kharagpur. He has joined National Institute of Technology as Professor in the Department of Computer Science and Engineering in 2002. Currently he is working as Professor of Computer Science and Engineering department. He has more than 35 publications in International Journals and conferences. His research areas of interest are Database Engineering, Distributed Computing, Parallel algorithm, Information Security and Data Compression.

B.Majhi is presently working as a professor and Head of the department of computer science & engineering in NIT Rourkela. He has completed 16 years of teaching in NIT Rourkela and 3 years of Industry experience in a reputed firm. He has completed his M.Tech and Ph.D. in Computer Science & Engineering from NIT Rourkela and Sambalpur University respectively. He has 13 research publications in international and national journals and more than 30 publications in national/international conferences to his credit. His research areas include soft computing applications, image processing, and cryptography.