A Novel Authentication Protocol Based on Elliptic Curves

Soram Ranbir Singh

Department of Computer Science & Engineering, Manipur Institute of Technology,
Takyelpat, Imphal-795001, Manipur, India.

Khomdram Memeta Chanu

Department of Electronics Accreditation of Computer Courses Centre,
Akampat, Imphal-795001, Manipur, India.

Abstract- **Authentication is the process of determining whether someone or something is, in fact, who or that it is declared to be. It is a client-server based protocol by which a server identifies the identity of a remote client when it logs on to the server through unsecured network. In private and public computer networks, authentication is commonly done through the use of logon passwords. Each user registers initially or is registered by someone else, using an assigned or self-declared password. On each subsequent use, the user must know and use the previously declared password. Internet business and many other transactions require a more stringent authentication process. This paper proposes a new authentication protocol to authenticate remote users using elliptic curves. The proposed protocol has three phases- registration phase, login phase, and authentication phase. When a genuine user wants to login the computer system, he has to key in his identity, password and private keys.**

Keywords- Authentication, RSA, Elliptic Curve, Elliptic Curve Discrete logarithm Problem.

I. INTRODUCTION

In today's information age, information is treated as a very important asset. Information security is a matter of great concern and importance for us and so we must do everything possible to protect our information. With the introduction of computer networks, a new requirement has come up in the picture when the information is accessed from a remote computer i.e., authentication. So, the need for the public-key cryptography comes into play. RSA is a public-key cryptography algorithm and it is presently used in both encryption and authentication but it is very slow in actual practice. Elliptic Curve Cryptography is one of a few public-key algorithms that can be used in the same place where RSA is used.

II. MAIN CONCERN OF RSA

One of the main concerns of RSA is its huge key length used in today's Internet security algorithms. An RSA key length of 1024 bits is used in web site logins but for high-security applications such as online financial fund transfers or for data that need to remain confidential for more than a few years; a 2048-bit key is used. Life is changing fast, computers become more and more powerful and, therefore, security requirements constantly change resulting in the demand for higher keys. What is perfectly acceptable and more than enough today may not be sufficient enough tomorrow. But a larger key has a serious problem in practice for the decryption is very slow and the size of ciphertext also becomes huge considerably. What worries us a lot is the speed of the decryption as the decryption takes place at the server. Smaller parameters can be used in Elliptic Curve Cryptography (ECC) than with RSA system at a given security level. The advantages that can be gained from smaller parameters include faster computations, smaller keys and certificates.

III. ELLIPTIC CURVE

Elliptic curves are a specific class of algebraic curves. The "Weierstrass form"of an elliptic curve equation is [2, 4]:-

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

The constant $a_1,\ a_2,\ a_3,\ a_4,\ a_6$ and the variables x, y can be complex, real, integers, polynomials, or even any other field elements. But in practice we must specify which field, F, these constants and the variables, x, y belong to and $\Delta \neq 0$, where Δ is the discriminant of E and is defined as follows [2,4]:-

$$\Delta = -d_2^2 d_8 - 8d_4^3 - 27d_6^2 + 9d_2 d_4 d_6$$
$$d_2 = a_1^2 + 4a_2$$
$$d_4 = 2a_4 + a_1 a_3$$
$$d_6 = a_3^2 + 4a_6$$
$$d_8 = a_1^2 a_6 + 4a_2 a_6 - a_1 a_3 a_4 + a_2 a_3^2 - a_4^2$$

We say that E is defined over K when the coefficients a_1, a_2, a_3, a_4, a_6 (and of course, the variables x and y) of the equations come from the elements of the field K. So, we sometimes write $E(K)$ to emphasize that E is defined over K, and K is called the underlying field.

A. *Elliptic Curve over prime Galois Fields*

An elliptic group over a prime Galois Field uses a special elliptic curve of the form

$$y^2 (\text{mod } p) = x^3 + ax + b (\text{mod } p)$$

where $a, b \in GF(p), 0 \le x \le p$ and $-16(4a^3 + 27b^2) \bmod p \neq 0$. The constants ***a*** and ***b*** are non-negative integers smaller than the prime p. The condition that $-16(4a^3 + 27b^2) \bmod p \neq 0$ implies that the curve has no "singular points" [2,4].

B. *Group Law*

The mathematical property that makes elliptic curves useful for cryptography is simply that if we take two distinct points on the curve, then the chord joining them intercepts the curve in a third point for because we have a cubic curve. If we then reflect that point in the x-axis we get another point on the curve as the curve is symmetric about the x-axis. This is the "sum" of the first two points. Together with this addition operation, the set of points $E(K)$ forms an abelian group with **0** serving as its identity [2,4]. It is this group that is used in the construction of elliptic curve cryptographic systems.

Group law for $y^2 = x^3 + ax + b$ over $GF(p)$.

(1) Identity: $P + 0 = 0 + P = P$ for all $P \in E(K)$.

(2) Negative: If $P = (x, y) \in E(K)$, then $(x, y) + (x, -y) = 0$. The point $(x, -y)$ is denoted by -P and is called the negative of P; note that -P is indeed a point in $E(K)$. Also, $-0 = 0$.

(3) Point addition: Let $P = (x_1, y_1) \in E(K)$ and $Q = (x_2, y_2) \in E(K)$ where $P \neq \pm Q$ .Then $P + Q = R(x_3, y_3)$, where $x_3 = \lambda^2 - x_1 - x_2$, $y_3 = \lambda(x_1 - x_3) - y_1$ and $\lambda = \dfrac{y_2 - y_1}{x_2 - x_1}$.

(4) Point doubling: Let $P = (x_1, y_1) \in E(K)$, where $P \neq \pm P$. Then $2P = R(x_3, y_3)$, where $x_3 = \lambda^2 - 2x_1$, $y_3 = \lambda(x_1 - x_3) - y_1$ and $\lambda = \dfrac{3x_1^2 + a}{2y_1}$.

The geometrical interpretation of the above group law is given here. Let's take a point $P = (x, y)$. The formula for finding $-P$ is $-P = (x, -y)$ as shown in the fig. 1.

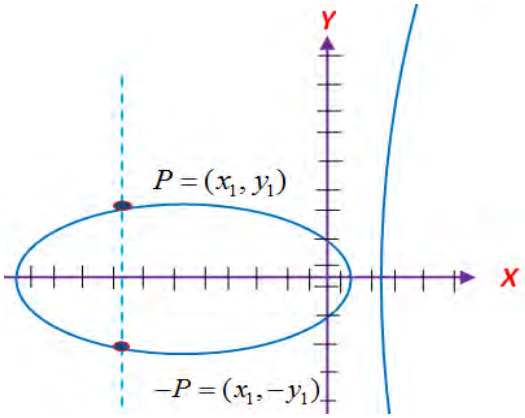

Fig.1.Negative of a Point

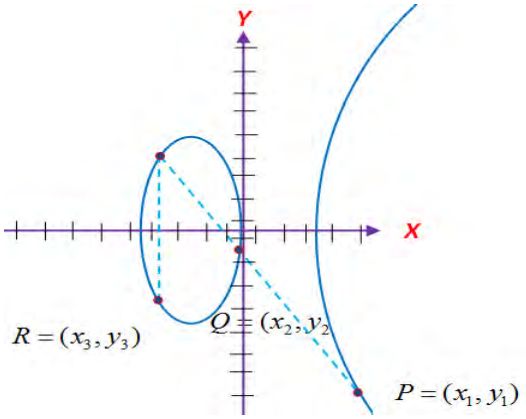

Fig. 2. Addition of two Points

We can define the addition of any two points on an elliptic curve by drawing a line between the two points and finding the point at which the line intersects the curve. The negative of the intersection point is defined as the "elliptic sum" of the two point and is shown in fig. 2. Mathematically we write:

R = *P* + *Q*.

This "addition" satisfies all the usual algebraic properties that we associate with integers, provided we define a single additional point "the point at infinity", which plays the role of 0 in the integers. In mathematical terms, we can define a finite additive abelian group on the points of the curve, with the zero being the point at infinity.

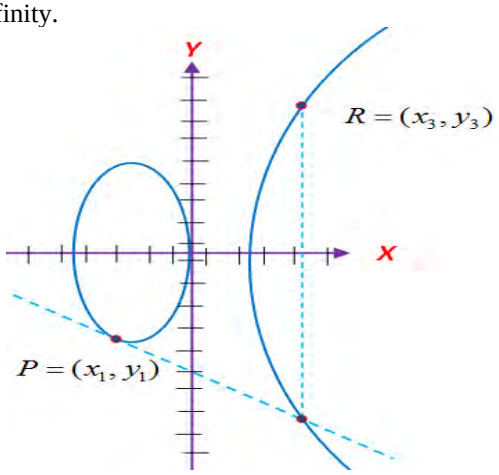

Fig. 3. Doubling a Point

If $P = (x_1, y_1)$, then the double of *P*, denoted by, $R = (x_3, y_3)$, is defined as follows. First draw the tangent line to the elliptic curve at *P*. This line intersects the elliptic curve in a second point. Then *R* is the reflection of this point in the x –axis. This is depicted in fig. 3. We can extend this idea to define $P + P + P = 3P$, and extending this idea further, we can define $P + P + P + ... + k$ times $= kP$, for any integer *k*, and hence define the order of P, being the smallest integer k such that $kP = 0$, where 0 denotes the point at infinity.

IV. HASSE THEOREM

Let *E* be an elliptic curve defined over F_q. The number of points in $E(F_q)$, denoted by $\#E(F_q)$, is called the *order* of *E* over F_q. Then Hasse's theorem says that the order of $E(F_q)$ satisfies the inequality-

$$q + 1 - 2\sqrt{q} \le \#E(F_q) \le q + 1 + 2\sqrt{q}.$$

For example, let E be the elliptic curve $y^2 = x^3 - 10x + 21$ over $GF(557)$. Hasse's theorem says that the order of $E(F_{557})$ satisfies the inequality given below:-

$$557+1-2\sqrt{557} \le \left|E(F_{557})\right| \le 557+1+2\sqrt{557}$$
$$\text{i.e, } 511 \le \left|E(F_{557})\right| \le 605.$$

V. SECURITY OF ECC

Let E be an elliptic curve defined over a finite field and let, P be a point (called base point) on E of order *n* and *k* is a scalar. Calculating the point $Q = kP$ from P is very easy and $Q = kP$ can be computed by repeated point additions of P. However, it is very hard to determine the value of k knowing the two points: kP and P. This lead leads to the definition of Elliptic Curve Logarithm Problem (ECDLP) [11], which is defined as: "Given a base point *P* and the point $Q = kP$, lying on the curve, find the value of scalar k". The integer k is called the elliptic curve discrete logarithm of Q to the base P, denoted as $k = \log_P Q$.

VI. PROPOSED AUTHENTICATION PROTOCOL

Before we explain our protocol, we give a few important notations used in this section.

Alice: The user.
Bob: The authentication server.
$ID_{AB}(x, y)$: Identity of Alice and is a point on Bob's curve.
$PW_{AB}(x, y)$: Password of Alice and is a point on Bob's curve.
$f : R_A(X, Y) \to R_B(X, Y)$: A mapping function that is used to map a point from Alice's curve to a point on Bob's curve.
$\oplus$: The concatenation operator.

The proposed protocol has three phases, registration phase, login phase, and authentication phase and each of them is explained below.

A. Registration Phase

We use two different curves in this protocol. One curve is used by Alice and the other curve is used by Bob. Each one of them exchanges with each other the curve parameters $D = (q, FR, a, b, G, n, h)$ comprising of the following:-

q	is the order of the field used
FR	Field representation used for elements of F_q.
a, b	a, b in $y^2 = x^3 + ax + b$
G	base point of the curve
n	order of the base point
h	cofactor and $h = \frac{\#E(F_q)}{n}$

The exchange can take place in an unsecure medium as the curves are public.

Bob chooses using his curve a private key d_B such that $d_B \in [1, n_B - 1]$ and a public key $Q_B(x, y) = d_B.G_B(x, y).$

Alice chooses m integers $d_{A1}, d_{A2}, ..., d_{Am} \in [1, n_A - 1]$ as his private keys. He then chooses m points $P_{A1}(x, y), P_{A2}(x, y), ...,$ and $P_{Am}(x, y)$ on his curve. Then, he calculates a point

$$P_D(x, y) = \sum_{i=1}^{m} d_{Ai}.P_{Ai}(x, y)$$
$$= d_{A1}.P_{A1}(x, y) + d_{A2}.P_{A2}(x, y) + ... + d_{Am}.P_{Am}(x, y).$$

His public key is the tuple $\left(P_D(x, y), P_{A1}(x, y), P_{A2}(x, y), ..., P_{Am}(x, y)\right)$.

Alice submits his public key $\left(P_D(x,y), P_{A1}(x,y), P_{A2}(x,y),..., P_{Am}(x,y)\right)$ to Bob for registration. Bob calculates Alice's identity and password as

$$ID_{AB}(x,y) = f\left(P_D(x,y)\right)$$

$$PW_{AB}(x,y) = d_B.ID_{AB}(x,y).$$

Then Bob issues to Alice the public parameter $ID_{AB}(x,y)$. This value is unique for every user, and maintained by Bob. Bob also despatches $PW_{AB}(x,y)$ to Alice through a secure channel.

B. Login Phase

When Alice wants to login to Bob, he keys in $ID_{AB}(x,y)$, $PW_{AB}(x,y)$ and his private keys. Then Alice will perform the following steps:

(1) Generates m random numbers r_{A1}, $r_{A2}, ..., r_{Am}$ and calculates a point $P_R(x,y) = \sum_{i=1}^{m} r_{Ai}.P_{Ai}(x,y)$

(2) Send the login request message $\left(ID_{AB}(x,y), \quad P_R(x,y)\right)$ to Bob.

Then Bob calculates an integer

(3) $e_B = g\left(T_{CB} \oplus ID_{AB}(x,y) \oplus PW_{AB}(x,y)\right)$. Here, T_{CB} is the current timestamp of Bob. Bob sends e_B to Alice.

(4) Alice keys in his private keys, password and calculates the followings:-

(i) $x_{A1} = r_{A1} + e_B.d_{A1}$

(ii) $x_{A2} = r_{A2} + e_B.d_{A2}$

(iii) ...

(iv) $x_{Am} = r_{Am} + e_B.d_{Am}$

(v) $C_A(x,y) = e_B.ID_{AB}(x,y)$

(vi) $D_A(x,y) = C_A(x,y) + e_B.PW_{AB}(x,y)$

(vii) $t_A = g\left(T_{CA} \oplus e_B \oplus PW_{AB}(x,y)\right)$

$(viii)$ $E_A(x,y) = t_A.G_A(x,y)$

(5) Alice sends the tuple $\left(x_{A1}, x_{A2}, ..., x_{Am}, C_A(x,y), ID_{AB}(x,y), D_A(x,y), E_A(x,y), T_{CA}\right)$ to Bob. Here T_{CA} is the current timestamp of the Alice.

C. Authentication phase

Bob receives the login request and performs the following steps:

(1) Check whether $ID_{AB}(x,y)$ is a valid user identity, if not, then Bob rejects the login request.

(2) Check, whether $\left(\widetilde{T_{CB}} - T_{CA}\right) \leq \triangle T$, where $\widetilde{T_{CB}}$ is current timestamp and $\triangle T$ is the permissible transmission delay. If $\triangle T$ is not reasonable, then Bob rejects the login.

(3) Bob calculates $t_B = g\left(T_{CA} \oplus e_B \oplus PW_{AB}(x,y)\right)$, where T_{CA} is the timestamp sent by Alice.

(4) Evaluate the following equations

$$(i)\quad P_R(x,y)=\left(\sum_{i=1}^{m}x_{Ai}.P_{Ai}(x,y)\right)-e_B.P_D(x,y) \qquad (1)$$

$$(ii)\quad D_A(x,y)-d_BC_A(x,y)=C_A(x,y) \qquad (2)$$

$$(iii)\quad E_A(x,y)-t_B.G_A(x,y)=0 \qquad (3)$$

If any of the above equations is not satisfied, then login is rejected otherwise login is allowed.

(5) If the login request is rejected three times then automatically the user account is locked for the day.

D. Algorithms

To show that my protocol is practical, I describe an efficient algorithm for each required operation.

Algorithm 1. Bob's Private and Public Keys generation

INPUT: Domain parameters $D_B=(q_B,FR_B,a_B,b_B,G_B(x,y),n_B,h_B)$

OUTPUT: Bob's Private and Public key.

1. Select $d_B\in[1,n_B-1]$.
2. Compute $Q_B(x,y)=d_B.G_B(x,y).$
3. Private Key $\leftarrow d_B.$
4. Public Key $\leftarrow Q_B(x,y).$
5. Return (Private Key, Public Key).

End Algorithm.

The run time complexity of the above algorithm 1 is discussed below. I will assume that $\#E(F_q)=n_Bh$ where n_B is prime and *h* is small (so $n_B\approx q$), and multipliers such as d_B are randomly selected integers from the interval $[1,n_B-1]$. The number of digits in the binary representation of d_B is denoted by t where $t\approx\lfloor\log_2 q\rfloor$. Hence the expected running time of Algorithm 1 is approximately $\frac{t}{2}$ point additions(A) and *t* point doublings(D), denoted by

$$\frac{t}{2}A+tD.$$

Algorithm 2. Alice's Private and Public Keys generation

INPUT: Domain parameters $D_B=(q_A,FR_A,a_A,b_A,G_A(x,y),n_A,h_A)$

OUTPUT: Alice's Private and Public key.

1. Select m integers $d_{A1},d_{A2},...,d_{Am}\in[1,n_A-1].$
2. Chooses m points $P_{A1}(x,y),\ P_{A2}(x,y),..,$ and $P_{Am}(x,y)$ on his curve.
3. Compute the following point on his curve.

$$P_D(x,y)=\sum_{i=1}^{m}d_{Ai}.P_{Ai}(x,y)$$
$$=d_{A1}.P_{A1}(x,y)+d_{A2}.P_{A2}(x,y)+...+d_{Am}.P_{Am}(x,y).$$

4. Private Key $\leftarrow(d_{A1},d_{A2},...,d_{Am}).$
5. Public Key $\leftarrow(P_D(x,y),P_{A1}(x,y),P_{A2}(x,y),...,P_{Am}(x,y)).$
6. Return (Private Key, Public Key).

End Algorithm.

To estimate the run time complexity of the algorithm 2, I will assume that $\#E(F_q) = n_A h$ where n_A is prime and *h* is small (so $n_A \approx q$), and the multipliers $d_{A1}, d_{A2}, ..., d_{Am}$ are randomly selected integers from the interval $[1, n_A - 1]$. The number of digits in the binary representation of each d_{Ai} is denoted by t where $t \approx [\log_2 q]$. Hence the expected running time of Algorithm 2 is approximately $m\frac{t}{2}$ point additions(A) and mxt point doublings(D), denoted by

$$m\frac{t}{2}A + mtD + (m-1)A$$
$$= \left(m\frac{t}{2} + m - 1\right)A + mtD.$$

Algorithm 3. Alice's Identity and Password generation

INPUT: Alice's Public key $\left(P_D(x,y), P_{A1}(x,y), P_{A2}(x,y), ..., P_{Am}(x,y)\right)$.

OUTPUT: Alice's Identity and Password.

1. Compute $ID_{AB}(x,y) = f\left(P_D(x,y)\right)$.
2. Compute $PW_{AB}(x,y) = d_B.ID_{AB}(x,y)$.
3. Identity $\leftarrow ID_{AB}(x,y)$.
3. Password $\leftarrow PW_{AB}(x,y)$.
4. Return (Identity, Password).

End Algorithm.

The number of digits in the binary representation of d_B is approximately t where $t \approx [\log_2 q]$. Hence the expected running time of Algorithm 3 is approximately $\frac{t}{2}$ point additions(A) and *t* point doublings(D), denoted by

$$\frac{t}{2}A + tD.$$

Algorithm 4. Alice's Login

INPUT: Domain parameters $D_A = (q_A, FR_A, a_A, b_A, G_A(x,y), n_A, h_A)$, Alice's identity $ID_{AB}(x,y)$, Password and Private keys.

1. Alice generates m random numbers $r_{A1}, r_{A2}, ..., r_{Am} \in [1, n_A - 1]$.
2. Alice calculate a point as follows-
   $$P_R(x,y) = \sum_{i=1}^{m} r_{Ai}.P_{Ai}(x,y) = r_{A1}.P_{A1}(x,y) + r_{A2}.P_{A2}(x,y) + ... + r_{Am}.P_{Am}(x,y)$$
3. Alice send the login request message $\left(ID_{AB}(x,y), \quad P_R(x,y)\right)$ to Bob.
4. Bob calculates an integer
   $e_B = g\left(T_{CB} \oplus ID_{AB}(x,y) \oplus PW_{AB}(x,y)\right)$, where T_{CB} is the current timestamp of Bob.
5. Bob sends e_B to Alice.
6. Alice calculates the followings:-

$I)\quad x_{A1} = r_{A1} + e_B.d_{A1}$

...

$III)\quad x_{Am} = r_{Am} + e_B.d_{Am}$

$IV)\quad C_A(x,y) = e_B.ID_{AB}(x,y)$

$V)\quad D_A(x,y) = C_A(x,y) + e_B.PW_{AB}(x,y)$

$VI)\quad t_A = g\left(T_{CA} \oplus e_B \oplus PW_{AB}(x,y)\right)$

$VII)\quad E_A(x,y) = t_A.G_A(x,y)$

7. Alice sends the tuple $\left(x_{A1}, x_{A2}, ..., x_{Am}, C_A(x,y), ID_{AB}(x,y), D_A(x,y), E_A(x,y), T_{CA}\right)$ to Bob. Here T_{CA} is the current timestamp of the Alice.
8. **End Algorithm.**

To calculate the run time complexity of the algorithm 4, I will assume that $n_A \approx n_B \approx q$ and the multipliers are randomly selected integers from the interval $[1, n_A - 1]$. The number of digits in the binary representation of each multiplier is denoted by t where $t \approx \lfloor \log_2 q \rfloor$. Hence the expected running time of Algorithm 4 is approximately

$\left(m\frac{t}{2}A + mtD\right) + (m-1)A + \left(m\frac{t}{2}A + mtD\right) + \left(m\frac{t}{2}A + mtD\right) + \left(m\frac{t}{2}A + mtD\right)$. Simplifying this, we get the run time complexity as

$(2mt + m - 1)A + 4mtD.$

Algorithm 5. Alice's Authentication

INPUT: Domain parameters $D_A = (q_A, FR_A, a_A, b_A, G_A(x,y), n_A, h_A)$, Alice's Login request message.

OUTPUT: Acceptance or rejection of Authentication.

Bob receives the login request message and performs the following steps:

1. Bob checks whether $ID_{BA}(x,y)$ is a valid user identity, if not, then Bob rejects the login request.
2. Bob checks whether $\left(\widetilde{T_{CB}} - T_{CA}\right) \le \Delta T$, where $\widetilde{T_{CB}}$ is current timestamp and ΔT is the permissible transmission delay. If ΔT is not reasonable, then Bob rejects the login.
3. Bob calculates
   $t_B = g\left(T_{CA} \oplus e_B \oplus PW_{AB}(x,y)\right)$, where T_{CA} is the timestamp sent by Alice.
4. Bob evaluates the following equations-

   $I.\quad P_R(x,y) = \left(\sum_{i=1}^{m} x_{Ai}.P_{Ai}(x,y)\right) - e_B.P_D(x,y)$

   $= x_{A1}.P_{A1}(x,y) + x_{A2}.P_{A2}(x,y) + ... + x_{Am}.P_{Am}(x,y) - e_B.P_D(x,y)$

   $II.\quad D_A(x,y) - d_B C_A(x,y) = C_A(x,y)$

   $III.\quad E_A(x,y) - t_B.G_A(x,y) = 0$
5. Bob checks if any of the above equations is not satisfied. If so, the login is rejected otherwise login is allowed.
6. If the login request is rejected three times then automatically the user account is locked for the day by Bob.
7. **End Algorithm.**

The subtraction operation in Elliptic Curves is as fast as the addition operation as $P(x_1, y_1) - Q(x_2, y_2) = P(x_1, y_1) + Q(x_2, -y_2)$. So, the expected running time of Algorithm 5 is approximately

$$\left(m\frac{t}{2}A + mtD\right) + \left(\frac{t}{2}A + tD\right) + (m-1)A + 1A + \left(\frac{t}{2}A + tD\right) + 1A + \left(\frac{t}{2}A + tD\right) + 1A$$

Simplifying this, we get the run time complexity as

$$\left(m\frac{t}{2} + \frac{3}{2}t + m + 2\right)A + (mt + 3t)D$$

E. Proof that the Algorithm works

Rewriting equation (1), we have,

$$P_R(x, y) = \left(\sum_{i=1}^{m} x_{Ai}.P_{Ai}(x, y)\right) - e_B.P_D(x, y)$$
$$= x_{A1}.P_{A1}(x, y) + x_{A2}.P_{A2}(x, y) + ... + x_{Am}.P_{Am}(x, y) - e_B.P_D(x, y)$$

$$RSH = x_{A1}.P_{A1}(x, y) + x_{A2}.P_{A2}(x, y) + ... + x_{Am}.P_{Am}(x, y) - e_B.P_D(x, y)$$
$$= (r_{A1} + e_B.d_{A1})P_{A1}(x, y) + (r_{A2} + e_B.d_{A2})P_{A2}(x, y) + ... + (r_{Am} + e_B.d_{Am})P_{Am}(x, y)$$
$$- e_B\left(\sum_{i=1}^{m} d_{Ai}.P_{Ai}(x, y)\right)$$

$$[\because x_{A1} = r_{A1} + e_B.d_{A1},\ x_{A2} = r_{A2} + e_B.d_{A2}]$$
$$...$$
$$[\because x_{Am} = r_{Am} + e_B.d_{Am}]$$
$$\left[\because P_D(x, y) = \sum_{i=1}^{m} d_{Ai}.P_{Ai}(x, y)\right]$$

$$= r_{A1}P_{A1}(x, y) + e_B.d_{A1}P_{A1}(x, y) + r_{A2}P_{A2}(x, y) + e_B.d_{A2}P_{A2}(x, y) + ... + r_{Am}P_{Am}(x, y)$$
$$+ e_B.d_{Am}P_{Am}(x, y) - (e_B d_{A1}.P_{A1}(x, y) + e_B d_{A2}.P_{A2}(x, y) + ... + e_B d_{Am}.P_{Am}(x, y))$$

$$= r_{A1}P_{A1}(x, y) + r_{A2}P_{A2}(x, y) + ... + r_{Am}P_{Am}(x, y) + e_B.d_{A1}P_{A1}(x, y) + e_B.d_{A2}P_{A2}(x, y)$$
$$+ ... + e_B.d_{Am}P_{Am}(x, y) - e_B d_{A1}.P_{A1}(x, y) - e_B d_{A2}.P_{A2}(x, y) - ... - e_B d_{Am}.P_{Am}(x, y)$$

$$= r_{A1}P_{A1}(x, y) + r_{A2}P_{A2}(x, y) + ... + r_{Am}P_{Am}(x, y)$$
$$= P_R(x, y)$$
$$= LSH$$

Rewriting equation (2), we have,

$$D_A(x,y) - d_B C_A(x,y) = C_A(x,y)$$

$$\begin{aligned} LSH &= D_A(x,y) - d_B C_A(x,y) \\ &= C_A(x,y) + e_B.PW_{AB}(x,y) - d_B e_B.ID_{AB}(x,y) \\ &\qquad \left[\because D_A(x,y) = C_A(x,y) + e_B.PW_{AB}(x,y)\right] \\ &= C_A(x,y) + e_B d_B.ID_{AB}(x,y) - d_B e_B.ID_{AB}(x,y) \\ &\qquad \left[\because PW_{AB}(x,y) = d_B.ID_{AB}(x,y)\right] \\ &= C_A(x,y) \\ &= RSH \end{aligned}$$

Again rewriting equation (3), we get,

$$E_A(x,y) - t_B.G_A(x,y) = 0$$

$$\begin{aligned} LSH &= \\ &= E_A(x,y) - t_B.G_A(x,y) \\ &= t_A.G_A(x,y) - t_B.G_A(x,y) \\ &\qquad \left[\because E_A(x,y) = t_A.G_A(x,y)\right] \\ &= g\left(T_{CA} \oplus e_B \oplus PW_{AB}(x,y)\right).G_A(x,y) - g\left(T_{CA} \oplus e_B \oplus PW_{AB}(x,y)\right).G_A(x,y) \\ &\qquad \left[\because t_A = g\left(T_{CA} \oplus e_B \oplus PW_{AB}(x,y)\right)\right] \\ &\qquad \left[\because t_B = g\left(T_{CA} \oplus e_B \oplus PW_{AB}(x,y)\right)\right] \\ &= 0 \\ &= RHS \end{aligned}$$

F. Conditions under which the Algorithm will work

From section VI, we know that Alice's public key is $\left(P_D(x,y), P_{A1}(x,y), P_{A2}(x,y), ..., P_{Am}(x,y)\right)$ where

$$\begin{aligned} P_D(x,y) &= \sum_{i=1}^{m} d_{Ai}.P_{Ai}(x,y) \\ &= d_{A1}.P_{A1}(x,y) + d_{A2}.P_{A2}(x,y) + ... + d_{Am}.P_{Am}(x,y) \end{aligned}$$

Let

$P_{A1}(x,y) = \lambda_1 G_A(x,y)$

$P_{A2}(x,y) = \lambda_2 G_A(x,y)$

...

$P_{Am}(x,y) = \lambda_m G_A(x,y)$, where λ_i are some integers and $G_A(x,y)$ is the generator of the Alice's curve.

$$\therefore P_D(x,y) = \sum_{i=1}^{m} d_{Ai}.P_{Ai}(x,y)$$
$$= d_{A1}.P_{A1}(x,y) + d_{A2}.P_{A2}(x,y) + ... + d_{Am}.P_{Am}(x,y)$$
$$= d_{A1}\lambda_1 G_A(x,y) + d_{A2}\lambda_2 G_A(x,y) + ... + d_{Am}\lambda_m G_A(x,y)$$
$$= \left(d_{A1}\lambda_1 + d_{A2}\lambda_2 + ... + d_{Am}\lambda_m\right) G_A(x,y)$$

But the order of the Elliptic Curve of Alice is n_A, so Alice will have to make sure that the following inequality is satisfied.

$\left(d_{A1}\lambda_1 + d_{A2}\lambda_2 + ... + d_{Am}\lambda_m\right) \le (n_A - 1).$

But, in most cases, it will not create any problem as the order of any curve used in cryptography is very very huge.

VII. SECURITY ANALYSIS

It is assumed here that Eve steals the identity of a person. He keys in $ID_{AB}(x,y)$, $\widehat{PW_{AB}(x,y)}$ and private keys. Then Bob and he will perform the following steps:

(1) Generates m random numbers r_{A1}, r_{A2}, ..., r_{Am} and calculates

$$P_R(x,y) = \sum_{i=1}^{m} r_{Ai}.P_{Ai}(x,y)$$
$$= r_{A1}.P_{A1}(x,y) + r_{A2}.P_{A2}(x,y) + ... + r_{Am}.P_{Am}(x,y)$$

(2) Send the login request message $\left(ID_{AB}(x,y),\quad P_R(x,y)\right)$ to Bob.

(3) Then Bob calculates an integer $\widehat{e_B} = g\left(T_{CB} \oplus ID_{AB}(x,y) \oplus \widehat{PW_{AB}(x,y)}\right)$ and sends it to him.

(4) Eve keys in private keys, password and calculates the followings:-

(i) $\widehat{x_{A1}} = r_{A1} + \widehat{e_B}.\widehat{d_{A1}}$

(ii) $\widehat{x_{A2}} = r_{A2} + \widehat{e_B}.\widehat{d_{A2}}$

(iii) ...

(iv) $\widehat{x_{Am}} = r_{Am} + \widehat{e_B}.\widehat{d_{Am}}$

(v) $\widehat{C_A(x,y)} = \widehat{e_B}.ID_{AB}(x,y)$

(vi) $\widehat{D_A(x,y)} = \widehat{C_A(x,y)} + \widehat{e_B}.\widehat{PW_{AB}(x,y)}$

(vii) $\widehat{t_A} = g\left(T_{CE} \oplus \widehat{e_B} \oplus \widehat{PW_{AB}(x,y)}\right)$

$(viii)$ $\widehat{E_A(x,y)} = \widehat{t_A}.G_A(x,y)$

(5) The tuple $(\widehat{x_{A1}}, \widehat{x_{A2}}, ..., \widehat{x_{Am}}, \widehat{C_A(x,y)}, ID_{AB}(x,y), \widehat{D_A(x,y)}, \widehat{E_A(x,y)}, T_{CE})$ is sent to Bob by Eve. Here T_{CE} is the current timestamp of the Eve.

A. Authentication Phase

Bob receives the login request and performs the following steps:

(1) Check whether $ID_{AB}(x, y)$ is a valid user identity, if not, then Bob rejects the login request.

(2) Check, whether $\left(\widehat{T_{CB}} - T_{CE}\right) \leq \triangle T$, where $\widehat{T_{CB}}$ is current timestamp and $\triangle T$ is the permissible transmission delay. If $\triangle T$ is not reasonable, then Bob rejects the login.

(3) Bob calculates $\widehat{t_B} = g\left(T_{CE} \oplus \widehat{e_B} \oplus \widehat{PW_{AB}(x, y)}\right)$, where T_{CE} is the timestamp sent by Eve.

(4) Now Eve gets the following equations

$$(i) \quad P_R(x, y) = \left(\sum_{i=1}^{m} \widehat{x_{Ai}}.P_{Ai}(x, y)\right) - \widehat{e_B}.P_D(x, y)$$

$$= \widehat{x_{A1}}.P_{A1}(x, y) + \widehat{x_{A2}}.P_{A2}(x, y) + ... + \widehat{x_{Am}}.P_{Am}(x, y) - \widehat{e_B}.P_D(x, y)$$

$$(ii) \quad \widehat{D_A(x, y)} - d_B\widehat{C_A(x, y)} = \widehat{C_A(x, y)}$$

$$(iii) \quad \widehat{E_A(x, y)} - \widehat{t_B}.G_A(x, y) = 0$$

From the above equations, let see whether Eve can recover the private keys and password in polynomial time.

Consider the equation

$$P_R(x, y) = \left(\sum_{i=1}^{m} \widehat{x_{Ai}}.P_{Ai}(x, y)\right) - \widehat{e_B}.P_D(x, y)$$

$$= \widehat{x_{A1}}.P_{A1}(x, y) + \widehat{x_{A2}}.P_{A2}(x, y) + ... + \widehat{x_{Am}}.P_{Am}(x, y) - \widehat{e_B}.P_D(x, y)$$

We have

$$\widehat{x_{A1}} = r_{A1} + \widehat{e_B}.\widehat{d_{A1}}$$

$$\widehat{x_{A2}} = r_{A2} + \widehat{e_B}.\widehat{d_{A2}}$$

...

$$\widehat{x_{Am}} = r_{Am} + \widehat{e_B}.\widehat{d_{Am}}$$

$$P_R(x, y) = \sum_{i=1}^{m} r_{Ai}.P_{Ai}(x, y)$$

$$= r_{A1}.P_{A1}(x, y) + r_{A2}.P_{A2}(x, y) + ... + r_{Am}.P_{Am}(x, y)$$

So,

$$\widehat{P_R(x,y)} = \left(\sum_{i=1}^{m} \widehat{x_{Ai}}.P_{Ai}(x,y)\right) - \widehat{e_B}.P_D(x,y)$$

$$= \widehat{x_{A1}}.P_{A1}(x,y) + \widehat{x_{A2}}.P_{A2}(x,y) + ... + \widehat{x_{Am}}.P_{Am}(x,y) - \widehat{e_B}.P_D(x,y)$$

$$= (r_{A1} + \widehat{e_B}.\widehat{d_{A1}}).P_{A1}(x,y) + (r_{A2} + \widehat{e_B}.\widehat{d_{A2}}).P_{A2}(x,y) + ... + (r_{Am} + \widehat{e_B}.\widehat{d_{Am}}).P_{Am}(x,y)$$
$$- \widehat{e_B}.\left(d_{A1}.P_{A1}(x,y) + d_{A2}.P_{A2}(x,y) + ... + d_{Am}.P_{Am}(x,y)\right)$$

$$= r_{A1}.P_{A1}(x,y) + \widehat{e_B}.\widehat{d_{A1}}.P_{A1}(x,y) + r_{A2}.P_{A2}(x,y) + \widehat{e_B}.\widehat{d_{A2}}.P_{A2}(x,y) + ... + r_{Am}.P_{Am}(x,y)$$
$$+ \widehat{e_B}.\widehat{d_{Am}}.P_{Am}(x,y) - \widehat{e_B}.d_{A1}.P_{A1}(x,y) - \widehat{e_B}.d_{A2}.P_{A2}(x,y) - ... - \widehat{e_B}.d_{Am}.P_{Am}(x,y)$$

$$= T_A(x,y) + d_{A1}.U_A(x,y) + d_{A2}.V_A(x,y) + ... + d_{Am}.W_A(x,y)$$

$$\left[T_A(x,y) = r_{A1}.P_{A1}(x,y) + \widehat{e_B}.\widehat{d_{A1}}.P_{A1}(x,y) + r_{A2}.P_{A2}(x,y) + ... + r_{Am}.P_{Am}(x,y)\right]$$

$$\left[U_A(x,y) = -\widehat{e_B}.P_{A1}(x,y)\right]$$

$$\left[V_A(x,y) = -\widehat{e_B}.P_{A2}(x,y)\right]$$

...

$$\left[W_A(x,y) = -\widehat{e_B}.P_{Am}(x,y)\right]$$

$$\Rightarrow \widehat{P_R(x,y)} - T_A(x,y) = d_{A1}.U_A(x,y) + d_{A2}.V_A(x,y) + ... + d_{Am}.W_A(x,y)$$

$$\therefore Z_A(x,y) = d_{A1}.U_A(x,y) + d_{A2}.V_A(x,y) + ... + d_{Am}.W_A(x,y)$$

$$[Z_A(x,y) = \widehat{S_A(x,y)} - T_A(x,y)]$$

Now, d_{A1}, d_{A2},..., and d_{Am} can't be found out in polynomial time because of the ECDLP of elliptic curves.

Next consider the equation $\widehat{D_A(x,y)} - d_B\widehat{C_A(x,y)} = \widehat{C_A(x,y)}$.
We have,

$$\widehat{D_A(x,y)} - d_B\widehat{C_A(x,y)} = \widehat{C_A(x,y)}$$

$$\widehat{D_A(x,y)} = d_B\widehat{C_A(x,y)} + \widehat{C_A(x,y)} = (d_B + 1)\widehat{C_A(x,y)}$$

$$\widehat{D_A(x,y)} = \widehat{d_B}.\widehat{C_A(x,y)} \quad [\widehat{d_B} = d_B + 1]$$

Again, we cannot solve $\widehat{d_B}$ in polynomial time because of the ECDLP of elliptic curves. In the same way, we cannot solve for $\widehat{t_B}$ in the third equation $\widehat{E_A(x,y)} - \widehat{t_B}.G_A(x,y) = 0$ in polynomial time. So, Eve cannot masquerade as Alice.

ACKNOWLEDGMENT

The second author thanks her husband for his understanding and cooperation in all the time she spent in the preparation of the manuscript.

REFERENCES

[1] Rotman, Galois Theory, Springer International Edition, 2010
[2] Ian Blake, Gadiel Seroussi, Higel Smart, Elliptic Curves in Cryptography, Cambridge University Press, 1999.
[3] Joseph H. Silverman, John Tate, Rational Points on Elliptic Curves, Springer, 1992.
[4] Lawrence C. Washington, Elliptic Curves, Number Theory and Cryptography, CRC Press, 2008.
[5] Henri Cohen, Gerhard Frey, Handbook of Elliptic and Hyperelliptic Curve Cryptography, CRC Press, 2006.

[6] Atul Kahate, Cryptography and Network Security, 2E, Tata McGraw, 2011.
[7] Bhattacharya, Jain, Nagpaul, Basic Abstract Algebra, Cambridge University Press, 2002.
[8] Bruice Schneier, Applied Cryptography, Wiley India, 2007.
[9] William Stallings, Cryptography & Network Security, PHI, 2006
[10] [Joseph H. Silverman, The Arithmetic of Elliptic Curves, Springer, 1986.
[11] [Ian Blake, Gadiel Seroussi, Higel Smart, Advances in Elliptic Curve Cryptography, Cambridge University Press, 2005
[12] Thomas Koshy, Elementary Number Theory with Applications, Academic Press, 2009.
[13] Erdinc Ozturk, "Low Power Elliptic Curve Cryptography" M.Sc thesis, Worcester Polytechnic Institute, April 2004
[14] Menezes, Okamoto, Vanstone, "Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field, IEEE Transaction on Information Theory, vol. 39, 1993.
[15] Authentication from The Wikipedia website. [Online]. Available: http://en.wikipedia.org/
[16] J. J. Shen, C. W. Lin and M. S. Hwang, "A modified remote user authentication scheme using smart cards," IEEE Trans. Consumer Electronic, vol. 49, no. 2, pp. 414-416, May 2003.
[17] R. Anderson, Security Engineering: A Guide to Building Dependable Distributed Systems. Wiley, 2001.
[18] Neal Koblitz, Alfred J. Menezes, "A survey of public-key cryptosystems,", Aug 7. 2004.
[19] Rotman, Galois Theory, Springer International Edition, 2010.
[20] R.L.Rivest, A.Shamir & L.M.Adleman, " A method for obtaining Digital Signature and Public Key Cryptosystems", ACM, 1978.
[21] Kristin Lauter, "The Advantages of Elliptic Curve Cryptography for Wireless Security", Microsoft Corporation.
[22] W. Ford and M. Baum. Secure Electronic Commerce: Building the Infrastructure for Digital Signatures and Encryption. Prentice Hall, 2nd edition, 2000.
[23] ANSI X9.62, "Public key cryptography for the financial services industry – the elliptic curve digital signature algorithm (ECDSA)", 1999.
[24] ANSI X9.62. Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA). American National Standards Institute, 1999.
[25] A. Antipa, D. Brown, A. Menezea, R. Struik, and S. Vanstone,. Validation of elliptic curve public keys. Public Key Cryptography—PKC 2003, 211–223, 2003.
[26] M. Bellare , A. Boldyreva,and S. Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. Advances in Cryptology—EUROCRYPT 2000 , 259–274, 2000.
[27] W. Diffie, P Vanoorschot, and M. Wiener. Authentication and authenticated key exchanges. Designs, Codes and Cryptography, 2:107–125, 1992.
[28] W. Ford and M. Baum. Secure Electronic Commerce: Building the Infrastructure for Digital Signatures and Encryption. Prentice Hall, 2nd edition, 2000.
[29] R. Canetti and H. Krawczyk , Analysis of key-exchange protocols and their use for building secure channels. Advances in Cryptology—EUROCRYPT 2001, 453–474, 2001.