International Journal of Engineering and Technology Volume 1 No. 1, October, 2011

EWOL AUTHENTICATION SCHEME FOR GRID COMPUTING

[1]Debajyoti Karmaker, [2] Mohammad Saiedur Rahaman, [3] Hafizur Rahman, [4] Mohammad Saidur Rahman

American International University Bangladesh
Dhaka, Bangladesh
Email: [1] d.karmaker@aiub.edu, [2] saied@aiub.edu, [3] hafizur@aiub.edu, [4] saidur@aiub.edu

ABSTRACT

Issues regarding security have become vital in the world of Grid computing. Users deserve a threat free environment where they are not at risk of providing their password to potential hackers when they are using it to access any protected resource. To prevent the grid resources from being illegally visited, a strong password authentication system, that requires input from both the user and server, is necessary. In this paper, based on the elliptic curve cryptosystem, we have proposed an efficient and most importantly a secure user authentication scheme for grid computing to guard against attacks like server spoofing, On-line and off-line password guessing and replay attack. The proposed scheme only requires a one-way hash function and Weil pairing, which makes it simpler and no hassle of safe guarding and private key. More over it makes uses of Lattice which is computationally must faster and efficient.

Key words: Weil pairing, Elliptic curve cryptosystem, grid computing, security, user authentication

I. INTRODUCTION

Grid computing, as a distributed computing model, stands for the new kind of systems that combine heterogeneous computational resources, such as computers, storage space, sensors, application software, and experiment data, connected by the Internet and make them easy access to a wide user community. When a user wants to request some computing and data resources, the grid can seamlessly, transparently and dynamically supply them to him over the Internet, which is similar to the power grid supplies electricity to end users. However, as the goal of grid computing is to only provide secure grid service resources to legal users, the security issue becomes an important concern of grid computing. To prevent the illegal users from visiting the grid resources, it should be guaranteed that strong mutual authentication needed for users and server.

Authentication based on passwords is a popular way for user authentication in the client-server model because of its easy-to-memorize property. However, several security concerns such that a password selected from a small space allows an adversary to mount, off-line, a dictionary attack. To prevent this ever present attack, various protocols have been proposed to achieve secure password-authenticated key exchange based on different cryptographic assumptions. Password-authenticated key exchange schemes assume that two entities have a priori shared password. Two parties use their shared password to generate a secure common session key and perform key confirmation with regard to the session key. Most password-authenticated key exchange schemes in the literature consider authentication between a client and a sever. Most of the earlier authentication systems [3, 4, 5, 6] does not work for grid computing as time stamp is used.

Inspired by Rongxing Lu's scheme [12] we propose our new scheme, where we take the benefits of Rongxing Lu scheme and earlier papers along with introducing weil pairing and taking the computation one more step further i.e.

Introduction of ECDDLP which is more secure than ECCDHP.

II. RELATED WORKS

A. Preliminaries

Calculations with real numbers generally produce round of error and thus inaccurate for cryptographic applications. Fast and precise arithmetic operations are required for cryptographic applications. In practice elliptic curve groups are used over the finite fields of Fp. Fp takes the numbers starting from 0 to p - 1, and computations finishes with taking the remainder on division by p. For example, in F23 the field is populated by 23 integers starting from 0 to 22, any operation within this arena will result in an integer which will also be between 0 and 22.

An elliptic curve over a finite field can be formed by picking variables a and b ; where both a and b are within the field of Fp. The curve takes all the points (x,y) that suit the elliptic curve equation mod p. (x and y are both member of Fp). 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 resultant elliptic curve, simultaneously with a special point O called the point at infinity. On such an elliptic curve there may exist finitely many points. Choose a generator point P = (xP , yP) whose order is a large prime number q over E(Fp), where G ≠ 0 In sucha way, a subgroup G of the elliptic curve group E(Fp) with order q is constructed.

Given a point element Q in G, find an integer x Z*q such that Q = xP, where xP indicates that the point P is added to itself for x times by the elliptic curves operation.

Every cryptosystem is based on a hard mathematical problem that is computationally infeasible to answer. The discrete logarithm problem is at the base for the Elliptic Curve Cryptosystem. The ECC resides upon the complexity of the Elliptic Curve Discrete Logarithm Problem (ECDLP). One can simple take a point on the elliptic curve and can obtain 2P by doubling. The point 3P can be calculated by adding P and 2P. A point nP can be generated in this manner which can be referred to as Scalar Multiplication of a point. The intractability of scalar multiplication products is the base of ECDLP. In the domain of Zp*, the discrete logarithm problem is described as: find a number k such that r = qk mod p; given elements r and q of the group, and a prime p. when the elliptic curve groups is described by multiplicative notation, the elliptic curve discrete logarithm problem is: find a number that Pk = Q; k is called the discrete logarithm of Q to the base P; given points P and Q in the group.

The Weil pairing, which is denoted by em, takes as input a pair of points P,Q E[m] and gives as output an m^{th} root of unity em(P,Q).

The Weil pairing em has many useful properties.

(a) The values of the Weil pairing satisfy
(b) The Weil pairing is bilinear
(c) The Weil pairing is alternating
(d) The Weil pairing is nondegenerate

The of the Weil pairing is that it can be computed quite efficiently without one's having to express P and Q in terms of a basis for E[m]. This is good, since expressing a point in terms of the basis P1 and P2 is even more complicated than solving the ECDLP.

B. Review of Rongxing Lu's Schemes

Rongxing Lu's scheme will consist of three phases: the registration phase, the authentication phase and the password change phase.

- U, S: U stands for user and S for server in grid computing.
- ID: public identifier for user U.
- G, P: subgroup of the curve and its generator point of order q
- D: distributed dictionary having capacity $|D| = 2^k$
- pw: low-entropy password extracted from D.
- K: server's secret key, which is kept secret and needs to be safeguarded.
- h: stands for secure one-way hash function, where $h : \{0, 1\}^* \rightarrow \{0, 1\}^l$ and l = 160.
- [m]k: the most significant k bits of any string m.
- i: low-entropy password's shelf life.

i. Registration Phase

First, user U have to submit his ID to get registered to the server S. if the given ID is valid, Server S chooses a shelf life i and by using its own private key K , it generates a hash value v = h (K || ID || i). Then the server S also generates user U's password $pw = [v]^K$ and sends back the user (pw, i) .The secret key K must be safeguarded to ensure the cryptography secure.

ii. Authentication Phase

Step 1: U chooses a random $r_1 \in Z^*q$, computes $R_1 = (pw \cdot r_1)P$, and sends (ID, R_1, i) to S.

Step 2: S first checks the shelf life i. If it is valid, continue; otherwise, stop. Then, S computes v =h(K || ID || i), $pw = [v]^K$ and $R_1' = pw^{-1}R_1 = (pw^{-1} \cdot pw \cdot r_1)P = r_1P$.

S chooses another random $r_2 \in Z^*q$, computes $R_2 = r_2P$, $sk = r_2R_1' = r_1r_2P$ and $h_1 = h(sk \| R_2)$. Finally, S sends (R_2, h_1) to U.

Step 3: U computes $sk = r_1R_2 = r_1r_2P$ and checks whether $h(sk \| R_2) = h_1$ holds. If it does hold, S is authenticated. Then, U computes $h_2 = h(sk \| ID)$ and sends it to S.

Step 4: S computes h_2' = h(sk || ID) and compares whether $h_2' = h_2$ or not. If they are equal, U is authenticated and granted to access the resources by S. In addition, after the mutual authentication between U and S, $sk = r_1r_2P$ will be used as a session key for further operations.

iii. Password Change Phase

After a common session key $sk = r_1r_2P$ is shared between U and S as above, they can establish a secure channel between them. Then, when U wants to change his password in its shelf life, he can securely request a new password as follows.

Step 1: U sends his ID along with old password pw and the shelf life i to S through the secure channel.

Step 2: S checks whether pw = [h(K || ID || i)]k holds or not. If it does hold, S chooses a new shelf life i and pw =

[h(K || ID || i)]k, then sends (pw , i) back to U using the secure channel. Thus, U can hold a new password pw and its shelf life i .

III. PROPOSED SCHEME AND ITS FEATURES

- U, S: U stands for user and S for server in grid computing.
- ID: public identifier for user U.
- G, p: subgroup of the curve and its generator point of order q
- D: distributed database Stores user ID , pw , p ,P [x0 , y0] , Q [x1 , y1] .
- pw: low-entropy password extracted from D.
- K: value of weil pairing solution.
- h: stands for secure one-way hash function, where h : {0, 1}* → {0, 1}l and l = 160.
- [m]k: the most significant k bits of any string m.
- i: shelf life of a low-entropy password.
- r1, r2: session-independent random exponents [1, q − 1] chosen by client and server, respectively.
- E(A,m),E2(A,m): symmetric encryption scheme of message m with A.

i. Registration Phase

(1) U → S: id,E(pw, gr_1 mod p)

U chooses a random number r1 and computes E(pw, gr1). Then, U sends the computation result with id to S as a registration request.

(2) S → U: gr2,H(sk, gr_1 mod p)

After receiving a registration request, S uses pw to retrieve gr1. S generates a random number r2 and computes the session key sk = (gr1)r2 . Thereupon, S computes gr2 and h(sk, gr1), and sends the results to U.

(3) U → S: id,H(sk´, gr_2 mod p)

U computes the session key sk´= (gr2)r1 and authenticates S by checking whether h(sk, gr1) = h(sk´, gr1) holds. If it holds, U computes h(sk´, gr1) and sends it with id to S.

(4) S → U: Access granted or denied

S computes the hash value h(sk, gr2) using its own copies of sk and gr2 and determines whether h(sk, gr2) = h(sk´, gr2) holds or not. If it holds, S will register the user.

ii. Authentication Phase

Step 1: U chooses a random r1 € Z*q , computes R1 =(pw · r1)P, and sends (ID,R1, i) to S.

Step 2: S checks the validity of shelf life i. If it is valid, proceeds; else, stop.

S chooses another random r2 € Z*q, computes R2 = r2P, sk = r2 = r1r2P and

h1 = h(sk || R2). After that it computes the value of K with points P[x0 , y0] and Q [r1 , r2] by weil pairing and lactics , Where Q = KP. Then S computes v =h(K || ID || i), pw = [v]K and h1 = pw−1R1 = (pw−1 ·pw· r1)P = r1P. Finally, S sends back (R2, h1) to U.

Step 3: U computes sk = r1R2 = r1r2P and checks whether h(sk || R2) = h1 holds. If it does hold, S is authenticated. Then, U computes h2 = h(sk || ID) and sends it back as response to S.

Step 4: S computes h2´ = h(sk || ID) and compares whether h2´ = h2 or not. If they are equal, U is authenticated and granted to access the resources by S. In addition, after the mutual authentication between U and S, sk = r1r2P will be used as a session key for further operations.

IV. SECURITY ANALYSIS

Replay attack: Reply attack will not work as r_1 and r_2 is chosen independently by client and server respectively.

On-line password guessing attack: Online password guessing attack is detectable. If an adversary tires to guess user U's password, he should use the guessed password to compute $\mathbf{h_2}$ in Step 3 for S's verification.

Off-line password guessing attack: if adversary obtains all exchanged messages (R_1, R_2, h_1, h_2, P, Q) he has to solve ECDDLP which is more secure than ECCDHP on which Rongxing Lu's Schemes relies on.

Server spoofing attack: since both way authentication is done before handshaking , eliminates the chances of server spoofing. $\mathbf{R_2}$ ensues that.

V. COMPARISON

In Rongxing Lu's scheme Hash Function, Server Private Key, Dictionary are needed where as Verification Table, Safeguarded data, Timestamp, Server Public Key, Smart Card are not needed in Rongxing Lu's schemes. In our proposed scheme only Hash function and verification table is needed. Table 1 shows the comparison of previous three schemes with our new scheme.

** R = Required; NR = No Required

Table 1: Comparison with [1], [2] & [3]

comparison Items	Scheme [10]	Scheme [11]	Scheme [12]	Our scheme
Hash Function	R	R	R	R
Server Private Key	R	R	R	NR
Dictionary	NR	NR	R	NR
Verification Table	R	R	NR	R
Safeguarded data	R	R	NR	NR
Timestamp	R	NR	NR	NR
Server Public Key	NR	NR	NR	NR
Smart Card	NR	NR	NR	NR

VI. CONCLUSION

In this paper, based on the elliptic curve cryptosystem, we have proposed a secure and effective password-based user authentication scheme for grid computing. As in Rongxing Lu's scheme [12] there is big chance of being attacked by any attacker just by knowing a valid unregistered id. Our scheme does block this lacking and goes one step forward in security which can be simple implemented.

REFERENCES

[1] I. Damgard, "A design principle for hash functions,"Advances in Cryptology, CRYPTO '89, LNCS 1989,no. 435, pp. 416-427, 1989.

[2] R. Lu and Z. Cao, "Efficient remote user authentication scheme using smart card," Computer Networks,vol. 49, pp. 535-540, 2005.

[3] H. Sun, "An efficient remote user authentication scheme using smart cards," IEEE Transactions on Consumer Electronics, vol. 46, no. 4, pp. 958 961,2000.

[4] S. Wang and J. Chang, Smart Card Based Secure Password Authentication Scheme, Computers and security, vol. 15, no. 3, pp. 231-237, 1996.

[5] S. Wu and B. Chieu, A User Friendly Remote Authentication Scheme with Smart cards, Computers and Security, vol. 22, no. 6, pp. 547-550, 2003.

[6] C. Yang, J. Li, M. Hwang, "A new mutual authentication and key exchange protocol with balanced computational power for wireless settings," European Transactions on Telecommunications, vol. 15, pp. 91-99, 2004.

[7] A. Menezes, P. Oorschot, S. Vanstone, Handbook of Applied Cryptograph, CRC Press. New York, 1997

[8] A solution to ecdlp ;http://www.certicom.com/index.php/531-a-solution-to-the-ecdlp

[9] Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman "An Introduction to Mathematical Cryptography" ISBN: 978-0-387-77993-5

[10] Y. Chang, C. Chang, Y. Liu, "Password authentication without the server public key," IEICE Trans-actions on Communications, vol. E87-B, no. 10, pp.3088-3091, 2004.

[11] E. Yoon and K. Yoo, "An efficient password authentication schemes without using the server public key for grid computing," GCC 2005, LNCS 3795, pp.149-154, 2005.

[12] Rongxing Lu, Zhenfu Cao, Zhenchuan Chai, and Xiaohui Liang "A Simple User Authentication Scheme for Grid Computing"