A Simple User Authentication Scheme for Grid Computing

Rongxing Lu, Zhenfu Cao, Zhenchuan Chai, and Xiaohui Liang
(Corresponding author: Rongxing Lu)

Department of Computer Science and Engineering, Shanghai Jiao Tong University
No.800, Dongchuan Rd., Minghang District, Shanghai, 200240, China
(Email: rxlu.cn@gmail.com)

(Received July 6, 2005; revised and accepted Apr. 30, 2007)

Abstract

The security issue has become an important concern of grid computing. To prevent the grid resources from being illegally visited, the strong mutual authentication is needed for user and server. In this paper, based on the elliptic curve cryptosystem, we would like to propose an efficient user authentication scheme for grid computing. The proposed scheme only requires a one-way hash function and server private key, which makes it more simple.

Keywords: Elliptic curve cryptosystem, grid computing, security, user authentication

1 Introduction

Over the last few years, the concept of grid computing has gradually gained prominence in both the academic and the research communities [3, 4, 5]. 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. See Figure. 1.

In recent periods, many password-based user authentication schemes are proposed for solving the authentication issue. However, most of them [8, 12, 13, 14] are not ideal for grid computing, since they are based on smart card and do not provide the strong mutual authentication. Aiming at the grid computing, Chang et al. [1] proposed an efficient and practical password-based authentication in 2004. However, since it uses the timestamp, Chang et al.'s scheme requires serious time synchronization tasks. To avoid using the timestamp, in 2005, Yoon et al. [16] proposed a more efficient password-based authentication scheme for grid computing. However, like Chang et al.'s scheme [1], Yoon et al.'s scheme [16] still requires a symmetric encryption algorithm and a verification table maintained at server side.

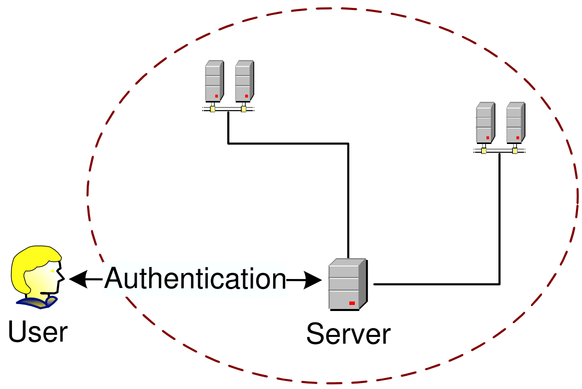

Figure 1: User authentication in grid computing

Motivated by mentioned above, in this paper, we would like to propose a new password-based user authentication scheme based on the elliptic curve cryptosystem [6]. Our proposed scheme not only inherits the advantages of Yoon et al. scheme [16] but will be more simple, since it doesn't require either the symmetric encryption algorithm or the verification table.

The rest of this paper is organized as follows. In Section 2, we first review the elliptic curve group and secure one-way hash function. Then, we propose our new password-based user authentication scheme for grid computing in Section 3. In Section 4, we analyze the security of our proposed scheme and compare it with another two efficient user authentication schemes [1, 16]. Finally, we draw our conclusions in Section 5.

2 Preliminaries

2.1 Elliptic Curve Group

Let $p > 3$ be a large prime and choose two field elements $a, b \in \mathbf{F}_p$ satisfying $4a^3 + 27b^2 \neq 0 \bmod p$ to define the equation of a non-supersingular elliptic curve $\mathbf{E}$: $y^2 = x^3 + ax + b \bmod p$ over $\mathbf{F}_p$, i.e., the set of solutions $(x, y) \in \mathbf{F}_p \times \mathbf{F}_p$ to the congruence $y^2 = x^3 + ax + b \bmod p$ together with a special point O called the point at infinity. Choose a generator point $P = (x_P, y_P)$ whose order is a large prime number q over $\mathbf{E}(\mathbf{F}_p)$, where $G \neq O$. In such a way, a subgroup $\mathbf{G}$ of the elliptic curve group $\mathbf{E}(\mathbf{F}_p)$ with order q is constructed. Next, let us consider three related mathematical problems in $\mathbf{G}$. Namely, the elliptic curve discrete logarithm problem (ECDLP), the elliptic curve computational Diffie-Hellman problem (ECCDHP) and the elliptic curve decisional Diffie-Hellman problem (ECDDHP).

Definition 1 (ECDLP). *Given a point element Q in $\mathbf{G}$, find an integer $x \in \mathbf{Z}_q^*$ such that $Q = xP$, where xP indicates that the point P is added to itself for x times by the elliptic curves operation.*

Definition 2 (ECCDHP). *For $a, b \in \mathbf{Z}_q^*$, given two point elements aP, bP in $\mathbf{G}$, compute abP in $\mathbf{G}$.*

Definition 3 (ECDDHP). *For $a, b, c \in \mathbf{Z}_q^*$, given three point elements aP, bP and cP in $\mathbf{G}$, decide whether $cP = abP$.*

Clearly, we have the relationship that the ECCDHP is no harder than ECDLP, and ECDDHP is also no harder than ECCDHP in $\mathbf{G}$. Therefore, we assume throughout this paper that ECDDHP is intractable (since $\mathbf{E}$: $y^2 = x^3 + ax + b \bmod p$ is a non-supersingular elliptic curve), which may guarantee that there is no polynomial time algorithm to solve ECDDHP, ECCDHP and ECDDLP with nonnegligible probability.

2.2 One-way Hash Function

Definition 4 (One-way Hash Function). *A one-way hash function H is said to be secure, if the following properties are satisfied [2, 9, 11, 15]:*

- *H can take a message of arbitrary-length input and produce a message digest of a fixed-length output.*
- *Given x, it is easy to compute $H(x) = y$. However, it is hard to compute $H^{-1}(y) = x$, when given y.*
- *Given x, it is computationally infeasible to find $x' \neq x$ such that $H(x') = H(x)$.*
- *It is computationally infeasible to find any two pair x and x' such that $x' \neq x$ and $H(x') = H(x)$.*

3 Proposed Scheme

In this section, we propose our password-based simple user authentication scheme for grid computing. The proposed scheme will consist of three phases: the registration phase, the authentication phase and the password change phase. The descriptions of each phase will be given as follows. First of all, we will introduce some used notations in the proposed scheme.

- U, S: user and server in grid computing.
- ID: public identity of user U.
- $\mathbf{G}$, P: subgroup of the elliptic curve group $\mathbf{E}(\mathbf{F}_p)$ and its generator point of order q, as defined in Section 2.1.
- $\mathcal{D}$: uniformly distributed dictionary of size $|\mathcal{D}| = 2^k$, as usual, $40 \leq k \leq 104$.
- pw: low-entropy human-memorable password extracted from $\mathcal{D}$.
- K: secret key of server S, which is only known by the server and must be safeguarded.
- h: secure one-way hash function, where $h : \{0,1\}^* \rightarrow \{0,1\}^l$ and $l = 160$.
- $[m]^k$: the most significant k bits of string m.
- i: shelf life of a low-entropy human-memorable password.

3.1 Registration Phase

In the registration phase, user U submits his identity ID to register himself to the server S. After checking the valid of identity ID, the server S chooses a shelf life i and uses her secret key K to compute the hash value $v = h(K \| ID \| i)$. Then, she generates U's password $pw = [v]^k$ and returns (pw, i) to U. And thus user U holds the human-memorable password pw and its shelf life i. Note that here the sever S doesn't need to maintain a verification table in database to store (ID, pw), which therefore overcomes the stolen-verifier attack. Nevertheless, the secret key K of the server S must be safeguarded.

3.2 Authentication Phase

When user U wants to login into the server S, as shown in Figure. 2, they will run the following steps:

Step 1. U chooses a random $r_1 \in \mathbf{Z}_q^*$, 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 $R_1' = pw^{-1} R_1 = (pw^{-1} \cdot pw \cdot r_1)P = r_1 P$. S chooses another random $r_2 \in \mathbf{Z}_q^*$, computes $R_2 = r_2 P$, $sk = r_2 R_1' = r_1 r_2 P$ and $h_1 = h(sk \| R_2)$. Finally, S sends (R_2, h_1) to U.

$$
\begin{array}{lcr}
\hline
U & & S \\
[ID, pw, i] & & [K] \\
\hline
r_1 \in \mathbf{Z}_q^*;\ R_1 = (pw \cdot r_1)P & & \\
& \xrightarrow{ID, R_1, i} & \\
& & v = h(K\|ID\|i);\ pw = [v]^k \\
& & R_1' = pw^{-1}R_1 = r_1P \\
& & r_2 \in \mathbf{Z}_q^*;\ R_2 = r_2P \\
& & sk = r_2R_1' = r_1r_2P \\
& & h_1 = h(sk\|R_2) \\
& \xleftarrow{R_2, h_1} & \\
sk = r_1R_2 = r_1r_2P & & \\
h(sk\|R_2) \stackrel{?}{=} h_1; & & \\
h_2 = h(sk\|ID) & & \\
& \xrightarrow{h_2} & \\
& & h_2' = h(sk\|ID) \stackrel{?}{=} h_2 \\
\hline
\end{array}
$$

Figure 2: Mutual authentication

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.

3.3 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 identity ID, old password pw and the shelf life i to S using 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'.

4 Security Analysis and Comparisons

In this section, we will analyze the security of our proposed scheme and also compare our proposed scheme with another two efficient user authentication schemes for grid computing [1, 16]. The detailed descriptions are given as follows. (We have to emphasis that, though we do not need to maintain a verification table on the server side, we must ensure the system secret key K is secure enough.)

4.1 Security Analysis

We examine the security of our proposed scheme in terms of the following security properties [10]: Replay attack, On-line password guessing attack, Off-line password guessing attack, Server spoofing attack and Perfect forward secrecy.

- **Replay attack:** Replay attack failed since nonce variables r_1 and r_2 are generated independently and both will be different in each login message. For example, if an adversary intercepts $(ID, R_1 = (pw \cdot r_1)P, i)$ in Step 1 and uses it to impersonate U to login into the server S. However, since the adversary has no knowledge of r_1, when he receives new (R_2, h_1) in Step 2, he cannot compute the right h_2 in Step 3 for S's verification. As a result, replay attack cannot follow.

- **On-line password guessing attack:** On-line password guessing attack is detectable in our proposed scheme. If an adversary tires to guess user U's password, he should use the guessed password to compute h_2 in Step 3 for S's verification. However, the probability of guessing the correct password is only $\frac{1}{|\mathcal{D}|} = 2^{-k}$, if the guessing is wrong, S can easily detect that there is an adversary trying to guess the password. Therefore, on-line password guessing attack cannot succeed.

- **Off-line password guessing attack:** To avoid the off-line password guessing attack, there must be no verification information for passwords in all exchanges. Observe our proposed scheme, if an adversary obtains all exchanged messages (R_1, R_2, h_1, h_2) by passive attack, and wants to guess U's password. He first guesses a password pw^* and uses it to compute $R_1' = {pw^*}^{-1}R_1$, then checks the correction of R_1'. If R_1' is right, then the password pw^* is correct. However, to determine the correction of R_1', he will face the ECCDHP, or ECDDHP when he also knows the session key $sk = r_1r_2P$. Therefore, our proposed scheme can resist off-line password guessing attack.

- **Server spoofing attack:** Since our proposed scheme provides mutual authentication, the server spoofing attack can be resisted. In the authentication phase, as user U sends (ID, R_1, i) to the adversary masquerading as the server, the adversary cannot generate proper R_2 and h_1 without the secret key K in Step 2. Therefore, the server spoofing attack doesn't work in our proposed scheme.

- **Perfect forward secrecy:** Perfect forward secrecy is provided in the situation that even though user's password pw or server's secret key K is compromised, an adversary still cannot derive any previous session keys. In our proposed scheme, suppose that an adversary knows user's password pw, he tries to find previous session keys from the information collected by passive attack in past communication sessions. However, due to the hardness of ECDLP and ECCDHP, he cannot do that. Therefore, our proposed scheme can provides the property of perfect forward secrecy.

4.2 Comparisons

Table 1: Comparisons in structure

Items of comparison	Scheme [1]	Scheme [16]	Ours
Server Private Key	R	R	R
Hash Function	R	R	R
Symmetric Encryption	R	R	NR
Verification Table	R	R	NR
Timestamp	R	NR	NR
Server Public Key	NR	NR	NR
Smart Card	NR	NR	NR

* R = Required; NR = No Required

Table 2: Definitions of notations

Notation	Definitions
T_{Mul}	the time for the modular multiplication
T_{Exp}	the time for the modular exponentiation
T_{Pmul}	the time for the multiplication of a number and an elliptic curve point
T_{Inv}	the time for the modular inversion
T_{Ha}	the time for the hashing operation
T_{En}	the time for the symmetric encryption operation
T_{De}	the time for the symmetric decryption operation
	*Under the conditions assumed in [7], the time complexity associated with the different operations can be roughly combined into multiplication operations. i.e., $T_{\text{Exp}} \approx 240T_{\text{Mul}}; T_{\text{Pmul}} \approx 29T_{\text{Mul}}$.

Since our proposed scheme is based on the elliptic curve cryptosystem [6], the total overhead for communication and performance can be reduced. For example, to reach a reasonable security level, it just requires a 160-bit prime p to construct the elliptic curve group $\mathbf{E}(\mathbf{F}_p)$. In addition, compared with Chang et al.'s scheme [1] and Yoon et al.'s scheme [16], our proposed scheme seems more simple. Table. 1 presents the comparisons of these three efficient user authentication schemes, and the results indicate that our proposed scheme is indeed simple and can be easily implemented.

Table 3: Estimation of performance aimed at time complexity

Items	Time complexity	Rough estimation
	Scheme [16]	
User	$2T_{\text{Exp}} + T_{\text{Ha}} + T_{\text{En}}$	$480T_{\text{Mul}} + T_{\text{Ha}} + T_{\text{En}}$
Server	$2T_{\text{Exp}} + T_{\text{Ha}} + T_{\text{De}}$	$480T_{\text{Mul}} + T_{\text{Ha}} + T_{\text{De}}$
	Our scheme	
User	$2T_{\text{Pmul}} + 2T_{\text{Ha}}$	$58T_{\text{Mul}} + 2T_{\text{Ha}}$
Server	$3T_{\text{Pmul}} + 3T_{\text{Ha}} + T_{\text{Inv}}$	$87T_{\text{Mul}} + 3T_{\text{Ha}} + T_{\text{Inv}}$

On the other hand, compared with other schemes, our scheme has more efficiency in time consuming. We could see that the user and the server cost $58T_{\text{Mul}} + 2T_{\text{Ha}}$ and $87T_{\text{Mul}} + 3T_{\text{Ha}} + T_{\text{Inv}}$, respectively. Since the cost of operation T_{Inv} in Z_p^* is negligible, the total consuming time on both user and server sides are largely less than the scheme [16], which, as far as we know, is the most efficient in that kind before ours. Table 2 defines the used notations, and Table 3 provides the details between the scheme in [16] and ours. From the results shown in Table 3, we can assure that our proposed scheme is efficient.

5 Conclusions

In this paper, based on the elliptic curve cryptosystem, we have proposed an efficient password-based user authentication scheme for grid computing. Since our proposed scheme only requires the server private key and a secure one-way hash function, compare with Chang et al. and Yoon et al.'s schemes [1, 16], it is more simple and can be easily implemented.

Acknowledgment

The authors would like to thank the anonymous referees for their suggestions which significantly improved this paper. This work was supported by the National Natural Science Foundation of China under Grant Nos. 60572155 and 60673079, the National High Technology Development Program of China under Grant No. 2006AA01Z424, and the National Research Fund for the Doctoral Program of Higher Education of China under Grant No. 20060248008.

References

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

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

[3] I. Foster, and C. Kesselman, "The globus project: a status report," *Proceedings of the IPPS/SPDP' 98 Heterogeneous Computing Workshop*, pp. 4-18, 1998.

[4] I. Foster, C. Kesselman, J. Nick, and S. Tuecke, "The physiology of the grid: an open grid services architecture for distributed systems integration," *Open Grid Service Infrastructure WG, Global Grid Forum*, 2002.

[5] I. Foster, C. Kesselman, and S. Teucke, "The anatomy of the Grid: enabling scalable virtual organizations," *International Journal of High Performance Computing Applications*, vol. 15, no. 3, pp. 200-222, 2001.

[6] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, pp. 417-426, 1987.

[7] N. Koblitz, A. Menezes, S. Vanstone, "The state of elliptic curve cryptography," *Designs, Codes and Cryptography*, vol. 19, pp. 173-193, 2000.

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

[9] R. Merkle, "One-way hash functions and DES," *Advances in Cryptography, CRYPTO'89*, LNCS 435, pp. 428-446, 1989.

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

[11] R. Rivest, *The MD5 Message Digest Algorithm*, Technical Report RFC 1321, 1992.

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

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

[14] 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.

[15] 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.

[16] 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.

Rongxing Lu received the B.Sc. and M.Sc. degrees in computer science from the Tongji University, Shanghai, China, in 2000 and 2003, respectively. In 2006, he received the Ph.D degree in computer science from Shanghai Jiao Tong University, Shanghai, China. Currently, he is a Post-doctoral fellow at the University of Waterloo, Waterloo, Canada. His current research interests include wireless network security and cryptography. Now, he is also a guest member of Trusted Digital Technology Laboratory of Shanghai Jiao Tong University.

Zhenfu Cao received his B.S. degree in computer science and technology from Harbin Institute of Technology, China, in 1983, and his Ph.D. degree in mathematics from the same university. Currently, he is a professor and a doctoral supervisor at the Department of Computer Science and Engineering of Shanghai Jiao Tong University. His main research areas are number theory, modern cryptography, theory and technology of information security etc.. Now, he is the director of Trusted Digital Technology Laboratory of Shanghai Jiao Tong University.

Zhenchuan Chai received his BS degree in electronic engineering and MS in computer science from East China University of Science and Technology in 2000 and 2003 respectively. Currently, he is a doctoral candidate in the Department of Computer and Engineering, Shanghai Jiao Tong University. His research interests in cryptography and network security.

Xiaohui Liang received his BS degree in computer science from Shanghai Jiao Tong University in 2006. Now he is a graduate student at the Department of Computer Science and Engineering of Shanghai Jiao Tong University. His research interest lies in network security.