An Improved Remote User Authentication Scheme using Bilinear Pairings

Sunder Lal[a] & K.K.Goyal[b]

[a] Department of Mathematics, Institute of Basic Science, Dr.B.R.Ambedkar University, Khandari,
Agra-282002 (U.P)
E-mail:sunder_lal2@rediffmail.com

[b] Faculty of Management & Computer Application, R.B.S.College, Khandari, Agra-282002 (U.P).
E-Mail:kkgoyal@gmail.com

Abstract

In 2005 Das et al. [5] proposed a remote user authentication scheme using bilinear pairings. Fang and Huang [7] analyzed the scheme and pointed out some weaknesses. They also proposed an improvement. Recently, Giri and Srivastava [9] observed that the improved scheme is still insecure to off-line attack and an improvement. However, the improved scheme is still insecure. In this paper, we show some weaknesses in the existing scheme and propose an improvement. The proposed scheme also enables users to choose and change the password without the help of the remote server.

Keywords: Authentication; Smart Card; Attacks; Password; Timestamp.

1. Introduction

In computer network systems, user authentication is an important mechanism for preventing unauthorized network access. The password-based authentication schemes with smart cards are usual parts of security for simpler and convenient authentication mechanisms to deal with secret data over insecure networks. In 1981, Lamport [11] proposed a well-known hash-based password authentication scheme for secure communication. His scheme resists replay attacks, but requires a verification table to verify the legitimacy of a login user. This approach also introduces risk and cost of managing and protecting the table. To avoid such problems, several authentication schemes without the verification table have been proposed [10, 13, 16]. Also, it is difficult for a user to memorize a long key or a server generated password. To overcome this problem, several schemes have been proposed [14, 16] so that the legitimate users can choose their passwords freely. Recently, some related schemes have been proposed [3, 5] for the authentication using smart cards. In 2005, Das et al. [5] proposed a scheme for smart card authentication using bilinear pairings that provides the users to choose and change their passwords freely. However, this scheme has some security flaws, which are described in [4, 8]. In 2006, Fang and Huang [7] proposed an improvement of Das et al's scheme [5] to remedy these weaknesses. In 2006, Giri and Srivastava [9] proposed an improvement on Fang and Huang [7] scheme to prevent some weaknesses. In this paper, we show some weaknesses in the existing scheme and propose a modified scheme that provides better security. The, proposed scheme also enables users to choose and change their password without the help of the remote server.

The remainder of this paper is organized as follows. Section 2, briefly introduces some mathematical concepts for our proposed scheme. Section 3, briefly reviews the scheme of Giri and Srivastava[9]. In this section we also describe a weakness of this scheme. In Section 4, we introduce our scheme and compare computational efficiency of our proposed scheme with some previously published schemes. Section 5, concludes the paper.

2. Preliminaries

In this section, we briefly review the basic concepts on bilinear pairings and a related mathematical problem.

Bilinear pairings derived from the Weil pairings or Tate pairings on elliptic curves have been used in cryptography to construct identity (ID)-based cryptographic schemes. Let $< G_1 , + >$ be an additive cyclic group of order q, where q is prime and let $< G_2 , X >$ be a multiplicative cyclic group of the same order. A mapping $e : G^2_1 \rightarrow G_2$ is called a bilinear mapping if it satisfies the following properties:

1. Bilinear property: For all Q, R, S $\in G_1$, e(Q+R; S) = e(Q, S) X e(R,S) and e(Q, R+S) = e(Q,R) X e(Q,S). As a result e(a *Q, b* R) = $(Q,R)^{a.b}$ for all Q,R $\in G_1$ and for all a,b $\in Z_q^*$, where a * Q means a times additions of Q, over the group $< G_1 , + >$.

2. Non-degeneracy property: There exist Q,R $\in G_1$ such that e(Q,R) $\neq 1_{G_2}$, where 1_{G_2} is the identity element of G_2.

3. Computability property: There is an efficient algorithm to compute e(Q,R) for all Q, R $\in G_1$.

For implementation point of view, G_1 will be the group of points on an elliptic curve and G_2 will denote a multiplicative subgroup of a finite field. The mapping e will be derived from either the Weil or the Tate pairing on an elliptic curve over a finite field. We refer to [1, 6, 12] for more comprehensive description on how these groups, pairings and other parameters are defined.

Discrete Logarithm Problem (DLP): Given two elements Q,R $\in G_1$, find an element x $\in Z_q^*$, such that Q = x * R whenever such an element exists.

It is known that on suitable chosen elliptic curves DLP is a computationally infeasible problem.

3. Brief review of the Giri - Srivastava scheme

In this section, we review the scheme proposed by Giri and Srivastava.

3.1 Set-up

The setup phase by the remote server(RS) proceeds as follows. The RS selects two groups: (i) G_1, an additive cyclic group of order prime, say, q, and (ii) G_2, a multiplicative cyclic group of the same order, a bilinear pairing $e : G_1^2 \rightarrow G_2$ mapping and a cryptographic hash function $H : \{0; 1\}^* \rightarrow G_1$. The RS chooses randomly a secret key (private key) s and computes the public-key as Pub_{RS} = s*P, where P is a generator of the group G_1. The RS also selects a public key cryptosystem, with $E_{Pub_{RS}}$(.) and E_s(.) as the encryption and decryption algorithms respectively. Finally, the RS publishes the system parameters: G_1, G_2, q, Pub_{RS}, e(.;.), H(.) and $E_{Pub_{RS}}$(.).The RS keeps the parameter s as secret.

3.2 Registration

In this phase, a user U_i submits his/her identifier ID_i and password PW_i (an integer in Z^*_q) to the RS. These private data must be sent over a secure channel. Then RS computes:
(i) a secret parameter $SP_i = PW_i * Pub_{RS,}$ and
(ii) registration identifier $Reg_{ID_i} = s * H(ID_i) + SP_i$.

RS loads Pub_{RS} ; ID_i ; Reg_{ID_i} ; SP_i and H(.) in the memory of the smart card and issues it to U_i.

3.3 Authentication

The authentication process consists of two phases: (a) the login phase and (b) the verification phase.

(a) Login Phase

If the user U_i wants to log into the RS, he/she must insert his/her smart card into a card reader and keys in his identifier ID_i and password PW_i. Then the smart card performs the following:

(i) computes $A = PW_i * Pub_{RS,}$ and $B = Reg_{ID_i} - A$.

(ii) randomly selects a number r and computes $C_i = E_{Pub_{RS}}(r)$.

(iii) computes $D_i = T * B + r * Pub_{RS}$, where T is the user system's current timestamp. It sends the login request message M = <ID_i ; C_i ; D_i ; Ti> to the RS over a public channel.

(b) Verification

On receiving the login request message M = <ID_i ; C_i ; D_i ; Ti> at time T', the RS and the smart card will perform the following steps for mutual authentication:

(i) RS verifies the validity of the time interval between T' and T. If (T'- T) >ΔT, RS rejects the login request, where ΔT denotes the expected valid time interval for transmission delay. Otherwise, it goes for the next step.

(ii) computes $X = E_s(C_i)$ and $Y = X * Pub_{RS}$.

(iii) checks if $e(D_i - Y, P) = e(H(ID_i); Pub_{RS})^T$. RS accepts the login request iff the equality holds.

3.4 Password change

Our scheme also enables user to change their password freely and securely. If the user U_i wants to change his password from PW_i to PW'_i, he/she inserts his smart card into a card reader and keys in his identifier ID_i and password PW_i. Then the smart card performs the following steps:
(i) The smart card computes $SP'_i = PW_i * Pub_{RS}$ and accepts the change in password request iff $SP^*_i = SP_{i.}$
(ii) The smart card if accepts the change in request, computes $Reg'_{ID_i} = Reg_{ID_i} – SP'_i + PW'_i * Pub_{RS} = s * H(ID_i) + PW'_i * Pub_{RS}$.

(iii) The password has been changed now with the new password PW'_i and the smart card stores new SP'_i and Reg'_{ID_i} in place of SP_i and Reg_{ID_i} respectively.

3.5 Weaknesses in the scheme

We know that the *<Pub_{RS} , ID_i , Reg_{ID_i} , SP_i* and *H(.)* > are stored in the memory of the smart card. If these information are revealed to an adversary X then by using *Reg_{ID_i}* and *SP_i* , X can easily make the new message M as follows :-

1. Already have A = *SP_i*
2. X can compute B= Reg_{ID_i} - A.
3. Select r' then C_i= $E_{Pub_{RS}}$(r') where r' is any random number.
4. D_i = T' * B + r' * Pub_{RS} where T' is the adversary timestamp.

Now M=< ID_i ; C_i ; D_i ; T' > can be sent to the RS. So the scheme of Giri and Srivastava is insecure against this offline attack.

4. Our scheme

As we observed above, if all the information stored in the smart card of user U_i are revealed to an adversary X, he can make access to RS. We modify the information to be stored in the smart in such a way that the revelation of these information to do not help in the verifying equation at the RS side. In this section, we present our authentication scheme with smart cards. The proposed scheme has four phases, namely, setup, registration, authentication, and password change phases.

4.1 Set-up phase

The system set-up proceeds as follows. The RS selects two groups: (i) G_1, an additive cyclic group of order prime, say, q, and (ii) G_2, a multiplicative cyclic group of the same order. A bilinear mapping e : $G_1^2 \rightarrow G_2$ and a cryptographic hash function H : {0; 1}*→G_1. The RS chooses randomly a secret key (private key) s and computes the public-key as Pub_{RS} = s*P, where P is a generator of the group G_1. Again, the RS selects a public key cryptosystem, where $E_{Pub_{RS}}$ (.) and E_s(.) are the encryption and decryption algorithms respectively. Finally, the RS publishes the following system parameters: G_1, G_2, q, Pub_{RS}, e(.;.), H(.) and $E_{Pub_{RS}}$(.).The parameter s is kept secret by RS.

4.2 Registration

In this phase, an user U_i submits his/her identifier ID_i and password PW_i to the RS. This data must be sent over a secure channel. Then RS computs:

(i) a secret parameter SP_i=PW_i * Pub_{RS}, and

(ii) registration identifier Reg_{ID_i} = (s + PW_i)H(ID_i).

RS loads Pub_{RS} ; ID_i ; Reg_{ID_i} ; SP_i and H(.) in the memory of the smart card and issues it to U_i .

4.3 Authentication

The authentication phase has two phases: (a) the login phase and (b) the verification phase.

(a) Login Phase

If the user U_i wants to log into the RS, he/she must insert his/her smart card into a card reader and keys in his identifier ID_i and password PW_i. Then the smart card performs the following:

(i) computes $A = Reg_{IDi} – Pwi.H(ID)$.
(ii) randomly selects a number r and computes $B_i = E_{PubRS}(r)$.
(iii) computes $C_i = T * A + r * Pub_{RS}$, where T is the user system's current timestamp.
It sends the login request message M = <ID_i ; B_i ; C_i ; Ti> to the RS over a public channel.

(b) Verification

On receiving the login request message M = <ID_i ; C_i ; D_i ; Ti> at time T', the RS and the smart card will perform the following steps for mutual authentication:

(i) RS verifies the validity of the time interval between T' and T. If (T'- T) >ΔT RS rejects the login request, where ΔT denotes the expected valid time interval for transmission delay. Otherwise, it goes for the next step.

(ii) computes $X = E_S(B_i)$ and then $Y = X * Pub_{RS}$.

(iii) checks if $e(C_i - Y, P) = e(H(ID_i); Pub_{RS})^T$. RS accepts the login request iff the equality holds.

4.4 Password change

Our scheme also enables user to change their password freely and securely. If the user U_i wants to change his password from PW_i to PW'_i, he/she insert his smart card into a card reader and keys in his identifier ID_i and old password PW_i. Then the smart card performs the following:
(i) computes $SP^*_i = PW_i * Pub_{RS}$ and accepts the change in password request iff $SP^*_i = SP_i$.
(ii) If accepts the change in request then computes $A = PW_i * H(ID_i)$, $B = PW'_i * H(ID_i)$, and $Reg'_{IDi} = Reg_{IDi} –A+ B$, where PW' is the new password.
The smart card now stores Reg'_{IDi} and SP'_i in place Reg_{IDi} and SP_i respectively.

To compare our scheme computationally with the previous schemes with respect to time complexity we use the following notations:

- t_+ is the time for addition of two elements in the additive group < G_1 ; + >.
- t_{AG} is the time for x Є Z^*_q times additions in the additive group < G_1 ; + >.
- t_{MG} is the time for x Є Z^*_q times multiplication in the multiplicative group < G_2 ; X >.
- t_e is the time for bilinear pairing operation.
- t_H is the time for executing the one-way hash function.

- t_E is the time for encrypting/decrypting a message.

Items Schemes	registration	Login	Verification	Password Change
Das et al	$2t_H + t_{AG}$	$2t_{AG} + t_H$	$t_H + 2t_e + t_{MG} + t_+$	$2t_H + 2t_+$
Fang et al	$2t_H + t_{AG}$	$t_{AG} + t_E$	$t_H + t_E + 2t_e + t_{MG}$	$2t_H$
Giri et al	$t_H + 2t_{AG} + t_+$	$3t_{AG} + 2t_+ + t_E$	$t_H + t_{AG} + t_+$ $+$ $t_E + 2t_e + t_{MG}$	$2t_{AG} + 2t_+$
Our	$t_H + 2t_{AG} + t_+$	$3t_{AG} + 2t_+ + t_E + t_H$	$t_H + t_{AG} + t_+$ $+$ $t_E + 2t_e + t_{MG}$	$3t_{AG} + 2t_+$

Table 1: Time complexity for different phases

We observe that in our scheme in the registration phase and the verification phase the computational time is same as to Giri and Srivastava[9] scheme but in login phase our scheme takes just t_H more time than the Giri and Srivastava[9] scheme and in password change phase computation time is $3t_{AG} + 2t_+$. This is the cost that we pay in our scheme to achieve more security as compared to the scheme of Giri and Srivastava[9] and the scheme of Fang and Huang[7].

5 Conclusion

In this paper we analyzed the Giri and Srivastava[9] scheme and observed some weakness in it. We propose an improvement to remove this weakness. We modified the password change protocol, which remains offline. We also compare our proposed scheme with previously published schemes. Our scheme is more secure as compared to previously published schemes and retains the flexibility in password change.

References

[1]Boneh D., M. Franklin,”Identity-based encryption from the Weil pairing,” In J. Kilian, editor, Advances in Cryptology-CRYPTO 2001, Springer-Verlag, LNCS, #2139, pp.213- 229, 2001.

[2]Boneh D., B. Lynn and H. Shacham, ”Short signatures from the Weil pairing,” Advances in Cryptology - Asiacrypt2001, LNCS #2248, Springer-Verlag, pp. 514-532, 2002.

[3]Chien H., J. Jan and Y.Tseng, ”An efficient and practical solution to remote authentication: smart card,” Computers and Security, **21**(4), 372-375, 2002.

[4]Chou J. S., Y. Chen, and J. Y. Lin, ”Improvement of Manik et al.s remote user authentication scheme,” http://eprint.iacr.org/2005/450.pdf, 2005.

[5]Das M. L., A. Saxena, V. P. Gulati and D. B. Phatak, ”A novel remote user authentication scheme using bilinear pairings,” Computers and Security, **25**(3), pp.184-189, 2005.

[6]Frey G. and H. G. Ruck, ”A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves,” Math. Comp., **62(**206), pp. 865-874, 1994.

[7]Fang G. and G. Huang, ”Improvement of recently proposed remote user authentication schemes,” http://eprint.iacr.org/2006/200.pdf.

[8]Goriparthi T., M. L. Das, A. Negi, and A. Saxena, ”Cryptanalysis of recently proposed remote user authentication schemes,”http://eprint.iacr.org/2006/028.pdf, 2005.

[9]Giri Debasis and P.D. Srivastava, “An improved remote user authentication scheme with smart card using billinear pairings.” http://eprint.iacr.org/2006/274.pdf.

[10]Hwang M. and L. Li, ”A new remote user authentication scheme using smart cards,” IEEE Trans Consumer Electron, **46**(1), pp. 28-30, February 2000.

[11]Lamport L., ”Password authentication with insecure communication,” Commun ACM, **24**, pp.770-772, 1981.

[12]Menezes A. J., T. Okamoto and S. A. Vanstone, ”Reducing elliptic curve logarithms to logarithms in a finite field,” IEEE Trans. Inf. Theory, **39**(5), pp.1639-1646, 1993.

[13]Sun H., ”An efficient remote user authentication scheme using smart cards,” IEEE Trans Consumer Electron, **46**(4), pp. 958-961, November 2000.

[14]Tan K. and H. Zhu, ”Remote password authentication scheme with smart cards,” Comput Commun,**18**, pp. 390-393, 1999.

[15]Thulasi G., Manik Lal Das and Ashutosh Saxena,” Cryptanalysis of recently proposed remote user authentication schemes,” http://eprint.iacr.org/2006/028.pdf

[16]Yang W. and S. Shieh, ”Password authentication schemes with smart cards,” Computers and Security, **18**(8), pp. 727-733, 1999.