A Key Authentication Scheme for Non-repudiation *

Min-Shiang Hwang Li-Hua Li Cheng-Chi Lee

Department of Information Management
Chaoyang University of Technology
168, Gifeng E. Rd., Wufeng,
Taichung County, TAIWAN 413, R.O.C.
Fax: 886-4-3742337
Email: mshwang@mail.cyut.edu.tw
http://www.cyut.edu.tw/~mshwang/

Abstract

In 1996, Horng and Yang had proposed a key authentication scheme that requires no authorities (Computer Communication 19 (1996) 848-850). However, it is vulnerable to the guessing attack. An intruder can try out a password and forge the public key. To amend this problem, an improved authentication scheme intended to prevent the guessing attack and the forging problem was proposed by Zhan et al. (Computer Communication 22 (1999) 739-741). However, their scheme did not achieve non-repudiation of public key. In order to achieve non-repudiation of public key, a key authentication scheme for non-repudiation is proposed in this paper. Our scheme will show that the attacker can neither repudiate a public key nor forge a public key. The security of our scheme is based on the difficulty of solving the discrete logarithm problem.

Keywords: Public key, Authentication, Certificate, Discrete logarithm, Non-repudiation, Security

1 INTRODUCTION

Recently, a key authentication scheme, HY-scheme, was proposed by Horng and Yang [3]. In their scheme, the certificate of user's public key can be verified without any authorities because the certificate is combined with user's password and private key. However, Zhan et al. [6] pointed out the vulnerability of HY-scheme by using the guessing attack [4]. An intruder can obtain a user password with the guessing attack. With the guessed password, the private key can be derived. Therefore the public key, calculated from the private key, can be forged. To prevent the forging problem, an improved key authentication scheme , ZLYH-scheme, was proposed by Zhan et al. [6]. With ZLYH-scheme, an intruder cannot forge the public key by using the guessing attack because a random number r in Z_p* is added to the password making it a discrete logarithm problem. However, we point out ZLYH-scheme do not succeed non-repudiation of public key.

In business transactions, non-repudiation is essential that provides proof to enable dispute resolution [1, 7]. The sender cannot deny what he transmits if the process is legal. In addition to this, the digital signature utilizing the public and private key must be recognized and not repudiated. Any authentication scheme should resolve non-repudiation problem in the study. To achieve this, we proposed an improved key authentication scheme for non-repudiation.

The content of this paper is organized as follows: in the next section, we review the ZLYH-scheme. In section 3, we point out the weaknesses of the ZLYH-

*This research was partially supported by the National Science Council, Taiwan, R.O.C., under contract no.: NSC89-2213-E-324-006.

scheme which do not succeed non-repudiation. Our improved key authentication scheme is introduced in section 4 and the security of our scheme is analyzed in section 5. Finally, the conclusion of this paper is presented.

2 THE REVIEW OF THE ZLYH-SCHEME

In order to prevent the guessing attack in HY-scheme, a long random number r is added into the certificate in the ZLYH-scheme [6]. The technique of ZLYH-scheme is briefly reviewed as follows. Every user is distributed a user identification, user-id, and a password, pwd, of a system. Each user of a system has a private key, Prv, and a password, pwd. Let Pub be the public key and $Pub = g^{Prv} \bmod p$, where p is a large prime, g is a generator in Z_p*, and Prv is a private key. The p, g and one-way function f is opened in the public domains.

In the user registration phase, every user chooses a random number r in Z_p* and then computes $f(pwd + r)$, where pwd is the user password. Then he sends the $f(pwd + r)$ and $R = g^r \bmod p$ to the server secretly, and the $f(pwd + r)$ is stored in the public password table of the server. To authenticate a legal transmission, the server verifies the equation $f(pwd + r) = f(pwd) \times R$. If the equation is equal, the server proves that the $f(pwd + r)$ is sent by the legal user. After the registration, each user can generate his own certificate by computing the equation

$$C = pwd + Prv + r \bmod p - 1. \tag{1}$$

The certificate C and the public key Pub is stored in the public key directory.

In the key authentication phase, when a sender wants to communicate with other, the sender must check the certificate and the public key of the receiver by computing the equation

$$f(C) = f(pwd + r) \times Pub \bmod p. \tag{2}$$

The sender can obtain the Pub, C and $f(pwd+r)$ of the receiver from the public tables. He then verifies the Equation (2). If the Equation (2) is satisfied, the sender can then use public key (Pub) to encrypt the transmission message, otherwise, the sender rejects.

3 THE WEAKNESSES OF THE ZLYH-SCHEME

Since an intruder knows only the $f(pwd + r)$ and r is a long random number in Z_p* in ZLYH-scheme [6], it is difficult to obtain the pwd and r with the guessing attack. Assume the server is trustworthy, an intruder cannot obtain the password and private key and forge the public key by using the guessing attack. Therefore, it is guessing secured.

However, some weaknesses in ZLYH-scheme are found. ZLYH-scheme cannot achieve non-repudiation of user public key. Although an attacker cannot obtain the private key Prv, pwd, and r, an attacker can forge a public key to interfere the verification and, hence, send on noise to the user. Hence, non-repudiation of public key in ZLYH-scheme is vulnerable. The weaknesses of the ZLYH-scheme are illustrated as following cases:

- Case 1: an attacker cannot revise $f(pwd + r)$ with the server.
  In this case, an attacker can get C, Pub, and $f(pwd + r)$ from the public directory. The attacker then choose a C' to derive Pub' by using

$$Pub' = \frac{f(C')}{f(pwd + r)} \bmod p. \tag{3}$$

  With the new C' and Pub', an attacker changes the original C and Pub to the fake one in the public directory. When other user wants to certify if this user with C' and Pub' is legal or not, the verification steps will be satisfied by computing the equation $f(C') = f(pwd + r) \times Pub' \bmod p$. Therefore, the authentication phase will be passed. If a message is sent with the phony public key, repudiation problem of the public key is hence occurred.

- Case 2: an attacker can revise the $f(pwd + r)$ with the server.
  In this case, an attacker can get C, Pub and $f(pwd+r)$ from the public directory as previous case. The attacker then choose C' to find Pub' and $f(pwd + r)'$ which satisfies:

$$f(C') = f(pwd + r)' \times Pub' \bmod p, \tag{4}$$

  where $f(pwd + r)' = f(pwd) \times R'$, where R' is a forged value. With the new C', Pub', and $f(pwd + r)'$, an attacker then revises the

original C, Pub, and $f(pwd + r)$ to the fake one in the public directory. When other user wants to certify if this user with C', Pub', and $f(pwd + r)'$ is legal or not, the verification steps will be succeed by computing the equation $f(C') = f(pwd + r)' \times Pub' \bmod p$. Therefore, the authentication phase will be passed. If a message is sent with the forged public key, then it will create the repudiation problem.

No matter which case may occur, an intruder can forge a Pub' to create repudiation dispute. This is easy to see, because when a sender wants to transmit messages secretly to the receiver, he uses the public key Pub' of the receiver to encrypt the messages, where Pub' has been revised by the intruder. Once the receiver receives the encrypted messages, he uses his own private key Prv to decrypt the messages. The receiver will find that he cannot decrypt the messages because of the interference of the forged public key from an intruder.

Moreover, in digital signature system, an evil intention signer may forge his own public key Pub' and C'. He then sends the signature using his private key to other user. The message and the signature will be successfully passed the system. However, when the other user receives the signature, he simply cannot verify the signature. Therefore, another dispute problem occurs. Obviously, the receiver will blame for the wrong signed message and reject the message. However, the purposed signer may claim that the message is legally encrypted and the system should be responsible for the problem. In addition, the signer can repudiate his signature because of there are two pairs of public key and private key (ie., (Pub, Prv) and (Pub', Prv')). The signer uses his private key (Prv) to sign the message. After some days, he forge his own public key (Pub') and then repudiates his signature. With problem like these, we conclude that non-repudiation problem is not complete in the ZLYH-scheme.

4 OUR SCHEME

In our scheme, we can overcome the weaknesses of ZLYH-scheme and achieve non-repudiation of public key. The main idea is utilize the difficulty of solving the discrete logarithm problem. An intruder cannot forge a public key and, therefore, no interfere of the verification is created and no noise to the users is generated.

As ZLYH-scheme, each user is distributed a user identification user-id and a password pwd of a system in our scheme. Each user of a system has Prv of the user's private key and pwd of his password. Let Pub of the user's public key is $Pub = g^{Prv} \bmod p$, where p is a large prime, g is a generator in Z_p* and Prv is the user's private key. The p, g and one-way function f is opened.

In the user registration phase, our scheme is the same as ZLYH-scheme. However, the user's certificate is strengthen as :

$$C = Prv \times f(pwd + r) + Pub \times (pwd + r) \bmod (p - 1). \quad (5)$$

The certificate C and public key Pub of the user is stored in the public key directory.

In the key authentication phase, when a sender wants to communicate with other user, the sender must check receiver's certificate of the public key by computing the equation

$$f(C) = Pub^{f(pwd+r)} \times f(pwd + r)^{Pub} \bmod p. \quad (6)$$

The sender can obtain Pub, C and $f(pwd + r)$ of a receiver from the public tables. For legal user verification, he then checks the equation (6). If the equation (6) is equal, the sender uses Pub to encrypt the confidential message and then sends the encrypted message to the receiver, otherwise, the sender rejects.

5 SECURITY ANALYSIS

Our scheme not only withstands guessing attack but also achieves non-repudiation of the user's public key. The security of our scheme is based on discrete logarithm problem. It is difficult to solve the discrete logarithm problem [2, 5]. If an attacker wants to forge the user's public key, he must obtain the pwd and r by using guessing attack. The attacker will find them extremely difficult to obtain by using guessing attack because r is a long random number. Obviously, an attacker knows only the announced information like $f(pwd + r)$. If he wants to forge a user's public key, he must compute the equation (7) or (8) as follows:

$$C' = f^{-1}(Pub'^{f(pwd) \times R} \times (f(pwd) \times R)^{Pub}) \bmod p \quad (7)$$

or

$$C' = f^{-1}(Pub') \times (f(pwd) \times R) + Pub' \times (f^{-1}(f(pwd)) + f^{-1}(R)) \bmod (p-1). \quad (8)$$

However, it is noticed that *pwd* could be guessed but not the long random number logarithmic r. He must solve the discrete logarithm problem to find R and r. Therefore, the attacker cannot forge user's public key even though he knows the *pwd* and $f^{-1}(Pub')$,

In ZLYH-scheme, although we cannot obtain the *Prv*, *pwd* and r to forge the user's public key, we can still interfere a user's public key and send noise to that user. Under this situation, the non-repudiation trap is created. In our scheme, an attacker will find it extremely hard to repudiate because it is difficult to find C', Pub' and $f(pwd+r)'$ that satisfies the equation

$$f(C') = Pub'^{f(pwd+r)'} \times f(pwd+r)'^{Pub'} \bmod p. \quad (9)$$

Therefore, it is shown that our scheme can achieve non-repudiation of the user's public key.

6 CONCLUSIONS

In order to avoid the guessing attack of HY-scheme, ZLYH proposed an improved key authentication scheme. However, their scheme does not achieve the non-repudiation of the user's public key. We propose an improved scheme in this paper. An attacker cannot forge a public key because it's based on the discrete logarithm problem. This scheme can achieve non-repudiation of the user's public key. Our scheme not only withstands guessing attack but also achieves non-repudiation of the user's public key. Furthermore, our scheme also requires no authorities.

References

[1] S. Chokhani, "Toward a national public key infrastructure," *IEEE Communications Magazine*, vol. 32, pp. 70–74, Sept. 1994.

[2] W. Diffie and M. Hellman, "New direction in cryptography," *IEEE Transactions on information theory*, vol. IT-22, no. 6, pp. 472–492, 1976.

[3] G. Horng and C.S. Yang, "Key authentication scheme for cryptosystems based on discrete logarithms," *Computer Communications 19*, pp. 848–850, 1996.

[4] G. Li, M.A. Lomas, R.M. Needham, and J.H. Saltzer, "Protecting poorly chosen secrets from guessing attacks," *IEEE Journal on Selected Areas in Communications*, vol. 11, pp. 648–656, June 1993.

[5] U.M. Maurer and Y. Yacobi, "A non-interactive public-key distribution system," *Designs, Codes and Cryptography*, vol. 9, no. 3, pp. 305–316, 1996.

[6] B. Zhan, Z. Li, Y. Yang, and Z. Hu, "On the security of HY-key authentication scheme," *Computer Communications 22*, pp. 739–741, 1999.

[7] J. Zhou and D. Gollmann, "Evidence and non-repudiation," *Journal of Network and Computer Applications*, vol. 20, no. 3, pp. 267–281, 1997.