A Key Authentication Scheme With Non-repudiation

Min-Shiang Hwang[†] Li-Hua Li[‡] Cheng-Chi Lee[§]
Department of Management Information System, National Chung Hsing University[†]
250 Kuo Kuang Road, 402 Taichung, Taiwan, R.O.C.
Email: mshwang@nchu.edu.tw
Department of Information Management, Chaoyang University of Technology[‡]
168 Gifeng E. Rd., Wufeng, Taichung County, Taiwan 413, R.O.C.
Department of Computer Science, National Chung Hsing University[§]
250 Kuo Kuang Road, 402 Taichung, Taiwan, R.O.C.

Abstract

In 1996, Horng and Yang proposed a key authentication scheme that requires no authorities. 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. in 1999. However, their scheme did not achieve non-repudiation of public key. In order to achieve non-repudiation of public key, two improvements of key authentication scheme for non-repudiation are discussed in this paper.

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

I Introduction

Recently, a key authentication scheme, HY-scheme, was proposed by Horng and Yang [1]. 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. [2] pointed out the vulnerability of HY-scheme by using the guessing attack [3]. 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. [2]. 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 this paper.

In business transactions[4, 5], non-repudiation is essential that provides proof to enable dispute resolution [6, 7]. The sender cannot deny what he/she 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 propose an improved key authentication scheme with 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-scheme which do not succeed in non-repudiation. Our improved key authentication schemes are discussed in last section 4.

II Overview of 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 [2]. 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. The one-way function f is $f(x) = g^x \bmod p$.

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/she 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/her own certificate by computing the equation

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

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 the others, the sender must check the certificate and the public key of the receiver by computing the equation

$$f(C) = f(pwd + r) \times Pub \bmod p. \tag{1}$$

The sender can obtain the Pub, C and $f(pwd + r)$ of the receiver from the public tables. He/She then verifies the Equation (1). If the Equation (1) is satisfied, the sender can then use public key (Pub) to encrypt the transmission message, otherwise, the sender rejects.

III The Weakness Of ZLYH-scheme

Since an intruder knows only the $f(pwd + r)$. And r is a long random number in Z_p* in ZLYH-scheme [2], 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, a weakness in ZLYH-scheme is 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 weakness of the ZLYH-scheme is illustrated as follows:

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:

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

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.

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/she 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/she uses his/her own private key Prv to decrypt the messages. The receiver will find that he/she 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/her own public key Pub' and C'. He/She then sends the signature using his/her 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/she 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/her signature because of there are two pairs of public key and private key (ie., (Pub, Prv) and (Pub', Prv')). The signer uses his/her private key (Prv) to sign the message. After some days, he/she forge his/her own public key (Pub') and then repudiates his/her signature. With problem like these, we conclude that non-repudiation problem is not complete in the ZLYH-scheme.

IV Discussion and Conclusion

We can see that ZLYH-scheme cannot achieve the non-repudiation of the user's public key because their scheme cannot against modification attack. To avoid this attack, two simple improved methods are discussed as follows. First method is that their scheme additionally adds access control scheme to it. Access control scheme is a very important subject in the field of information protection systems and can prevent files from being destroyed, altered, disclosed or copied by unauthorized users. Until to now, many access control schemes had been proposed in computer protection systems [8, 9, 10]. Therefore, we suggest that their scheme can add the access control scheme against modification attack.

Second method is that their scheme additionally adds the digital signature of the server to public key directory. This method must add a public key cryptosystem such as ElGamal system [11, 12]. Two signature functions are $D_{sprv}()$ and $E_{spub}()$, where *sprv* is the server's private key and *spub* is the server's public key stored in public table. The server uses the function $D_{sprv}()$ to signs $(C, Pub, f(pwd + r))$ and stores $D_{sprv}(C, Pub, f(pwd + r))$ to its public key directory. Before verifying the public key of the user, each verifier must use the function $E_{spub}()$ to verify the signature $D_{sprv}(C, Pub, f(pwd + r))$. In this method, no attacker can forge $(C, Pub, f(pwd + r))$ because he/she does not know the server's private key.

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 two simple improved schemes above. An attacker cannot forge a public key because it is protected by the access control scheme and the digital signature scheme of server. These simple improved schemes can achieve non-repudiation of the user's public key.

ACKNOWLEDGEMENTS

Part of this research was supported by the National Science Council, Taiwan, R.O.C., under contract no. NSC92-2213-E-324-023.

References

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

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

[3] 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.

[4] Min-Shiang Hwang, Iuon-Chung Lin, and Li-Hua Li, "A simple micro-payment scheme," *Journal of Systems and Software*, vol. 55, no. 3, pp. 221–229, 2001.

[5] Min-Shiang Hwang, Eric Jui-Lin Lu, and Iuon-Chung Lin, "Adding timestamps to the secure electronic auction protocol," *Data & Knowledge Engineering*, vol. 40, no. 2, pp. 155–162, 2002.

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

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

[8] Min-shiang Hwang, Wen-Guey Tzeng, and Wei-Pang Yang, "An Access Control Scheme Based on Chinese Remainder Theorem and Time Stamp Concept," *Computers & Security*, vol. 15, no. 1, pp. 73-81, 1996.

[9] Min-Shiang Hwang and Wei-Pang Yang, "A New Dynamic Access Control Scheme Based on Subject-Object-List," *Data & Knowledge Engineering*, vol. 14, no. 1, pp. 45-56, 1994.

[10] Min-Shiang Hwang, Wen-Guey Tzeng, and Wei-Pang Yang, "A Two-Key-Lock-Pair Access Control Method Using Prime Factorization and Time Stamp," *IEICE Transactions on Information and Systems*, vol. E77-D, no. 9, pp. 1042-1046, 1994.

[11] T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. IT-31, pp. 469–472, July 1985.

[12] Min-Shiang Hwang, Chin-Chen Chang, and Kuo-Feng Hwang, "An ElGamal-like cryptosystem for enciphering large messages," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 2, pp. 445–446, 2002.