Middle-East Journal of Scientific Research 20 (12): 2289-2293, 2014
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2014.20.12.324

Efficient Password Multicast Authentication

P. Gayathri and A. Jeyamathi

Department of Information Technology,
Bharath University, Chennai, 600073, India

Abstract: This paper contain secure- password authentication involve third party. The previous methods focus on two party authentications who share the password. In the previous method is week against dictionary attack. Our method provide single sign like previous method and also against for on/off-line dictionary attack. This method provides forward securacy and also reduces the damage of single point failure.

Key words: Contain secure-password authentication · Against dictionary attack

INTRODUCTION

Password based authentication is less expansive to use compared to biometric or hardware based scheme. Telnet authentication or http authentication, have problem ranging from being totally insecure to certain types of attack such as off-line dictionary attack.

More secure password-based Authentication protocol, encrypted key exchange (EKE) which is secure against passive dictionary attacks [1].

The plaintext-equivalent is a piece of data, as the hashed password, which can be used to obtain the same level of access that the adversary gets when the adversary gets the password. These authentication-key exchange protocols have been developed for two parties. These cannot provide Single Sign on (SSO) feature in the distributed computing without modification.

Kerberos, which provides SSO, is focused on a password-based authentication protocol evolving on-line trusted third parties. Kerberos is based on a symmetric cryptosystem and maintains shared secret keys of entities in the Key Distribution Center (KDC) [2]. Therefore, the Kerberos server must be extremely secure, as it represents a single point of failure for the entire system and all entities. Moreover, Kerberos is fragile to off-line dictionary attacks and cannot provide forward secrecy.

This paper describes, the secure password authentication involving third party the previous methods only concentrate two parties this is also one of the advantage.

In this paper, we concentrate on a secure password authentication scheme involving a trust third party to provide SSO without degrading security comparing to the previous secure password authentication schemes [3]. Our design goals are following:

- Preventing the on/off-line dictionary attack: This is a common attack for password.
- Providing the prefect forward secrecy: A compromised password will not allow an adversary to decrypt past sessions. Also, a compromised session key will not allow an adversary to find out a password.
- Tolerating the compromise of a password file: For this, a server must keep not plaintext-equivalent [4].
- Avoiding or reducing the damage of the single point of failure: When KDC is compromised in Kerberos, all users and all target servers have to change the password and the key. Also, all past session can be decrypted.
- Providing different trust level of servers: The trusted third party should not provide any information about password to target servers.

Secure Password Authentication for Distributed Computing (SPADC) Registration: We assume that a connection must be protected against eavesdropping and any kinds of modification during registration.

For example, the TLS/SSL, which provides integrity, encryption and authentication, can be used for registration.

Corresponding Author: P. Gayathri, Department of Information Technology, Bharath University, Chennai, 600073, India.

Initialization: Before registration protocol execution, a TIS has to initialize as followings [5].

- The TIS chooses large primes p, q such that $q \mid p$, a generator g of a multiplicative subgroup of $Z\,p$ with order q and a collision resistance hash function $h(\bullet)$ where $h(\bullet) : \{0, 1\}^* \rightarrow Z^*_q$.
- The (*TIS*) chooses $X_{TIS}\, R\, Z^*_q$ and computes $Y_{TIS} = G^{X}{}_{TIS}$ mod p, where (X_{TIS}, Y_{TIS}), is a key pairs of a private key and a public key of the TIS and publishes a tuple (p, q, g, Y_{TIS}) as public value, which is known publicly in the network. The TIS must keep securely X_{TIS}.
- The TIS can submit (p, q, g, Y_{TIS}) to a certificate authority (CA) to get a X.509 Certificate. (p, q, g, Y_{TIS}) can be published as a Certificate $Cert_{TIS}$.

Registration:

- The communication channel must be secure for registration, so we assume that a client and TIS open a SSL connection. The client sends his/her ID and g^{π} mod p to register himself to the TIS, where $\pi = h(password||v)$ and a salt $v \in_R Z^*_q$.
- The TIS chooses random numbers k and n in Z^*_q. The TIS computes and sends s, r and a nonce n, where a tuple (s, r) is a TIS's signature on the user identity and g^{π} with Nyberg-Rueppel signature scheme.
- The client sends two encrypted values and $u = (s+t)$ mod q to the token issuer.
- The TIS computes a decryption key $SK\_$ and checks whether the encrypted value is equal to $n + 1$. If not, the TIS sends the error message and stops the registration process. Otherwise, the TIS keeps a tuple ($ID, r, E_{f(\pi)(s)}, v$). Note $g^{u} \bullet y^{(h(ID||r)}{}_{TIS}$. $r = g^{-x} TIS^{h(ID||r)-k+t)}$. $g^{-x} TIS^{h(ID||r)}$. $g^{\pi+t}$ *mod p. Therefore, SK = SK'.*

Token Issuing and Authentication Between the Client and the TSE: To get the services from TSEs, a client has to show the TSE the client's attributes such as attribute certificate in secure way. For that, the client and the TIS have to authenticate each other [6]. Once authentication is success, the client requests the tokens, which include the attributes for access control and the secure information for authentication between the client and TSE.

- The client requests a token to the TIS in order to access a TSE with the TSE identity, timestamp, the client's attributes such as role name and supplement information. The hash value of a request message and the request message are encrypted with his session key before sending.
- The TIS decrypts the message and checks the integrity of the message and whether the requested attributes are belong to the client. If not, TIS sends an error message. The TIS checks the integrity of a TSE's public key Y_{TSE} in his database. Finally, a token, Y_{TSE} and the hash value of Y_{TSE} are encrypted and sent to the client.
- The client decrypts $e1$. The client gets the Token and the TSE public key.

Authentication Between the Client and the TSE:

- The client chooses a random value t_4 and a nonce n_c in Z^*_q. The client sends the Token,$g^{t}{}_4$ and n_c.
- The TSE checks whether the receiver in Token is her/him. If not, the TSE rejects the request. Otherwise, the TSE chooses a random number w and a nonce n_t in Z^*_q. The TSE computes the session key $SK2$ and encrypts n_t, n_c and ID. The TSE sends encrypted value e_2 and g^w.
- The client computes the session key SK_2' and decrypts e_2 with SK_2'. He checks whether the nonce n_c is same as he sent and the receiver identity is her/him. If true, he encrypts n_c, n_t and TSE and sends it.

Alternative Version for Mobile Devices: In some cases, it is not necessary that a client downloads $E_{f(\pi)}(s)$ and v. For example, a client uses a mobile device such as Laptop, PDA for registration and authentication. An other possible scenario is as following. The client has a personal computer at home or the client's office [7]. A client keeps a s in systems belonging to him use to login. The advantage of this version is more secure against the Single Point of Failure than the previous version.

The different things for registration are as following.

Security Analysis

Threats: We consider following attacks:

Replay Attack: An impersonation or other deception involving use of information from single previous protocol execution on the same or a different communication party [8].

Interleaving Attack: An impersonation or other deception involving selective combination of information from one or more previous or simultaneously ongoing protocol execution (parallel sessions), including possible origination of one or more protocol execution by an adversary itself [9].

Reflection Attack: An special interleaving attack involving sending information from an ongoing protocol execution back to the originator of such information. A general solution to prevent reflection attacks is that the protocol must be asymmetric [10].

Man-in-the-Middle-attack: When the public key is not authenticated, an adversary sends his public key as the intended communication party's public key [11].

Forward Secrecy We defines this term as following:

- Compromise of current long term key should not compromise future long term key.
- Compromise of old long term key should not compromise current long term key.
- Compromise of current long term key should not compromise current or past session keys.
- Compromise of current session key should not compromise current long term key.

Exhaustive Password Search: A password that can be memorized by a human has limited length. Moreover, the entropy of such a password is low [12]. Therefore, the key space of a password is so small as to be conducted a exhaustive search by an adversary.

Moreover, the computer will become more powerful continuously according to Moore's Law. Therefore, we need longer keys in the future, while the password size cannot be lengthen, because the limitation of the human memory. The other main obstacle of lengthen the password size is the design of password system.

For example, Unix systems limit the length of the password to eight characters.

Password-guessing and Dictionary Attack: People tend to choose memorable password. It means that a password is easy to guess and most users select passwords from a small subset of the full password space (e.g., dictionary words, names, lower-case and so on) while ideally arbitrary strings of n characters would be choose as user-selected passwords.

Security Analysis against Threats

Replay Attack: We uses always time variant parameters to protect replay attack.

Interleaving Attack: A TIS and a TSE never initiate the protocol. That is, the protocol itself is asymmetric [13].

Moreover, in the authentication between the client and the TSE, n_c which is sent by initiator (the client) is checked always to prevents the interleaving attack.

Reflection Attack: The messages for authentication between the client and the TIS includes the receiver's identity. The authentication between the client and the TSE also includes the receiver's identity.

Man-In-Middle-Attack: We need actually the certificate *CertTIS* because that is a way to protect this type of attack on a TIS public key.

A TSE's public key Y_{TSE} is provided by a TIS through authenticated channel. A TIS issues the token for providing authenticity of the client's public value.

Forward Secrecy: The long term keys (the private key of TIS and TSE) and passwords are always chosen independently.

Therefore, compromise of current long term keys and current passwords cannot compromise old long term keys and old passwords. By the same reason, compromise of an old long term key and old password cannot compromise current long term key and current password.

When an adversary gets the password, the adversary must solve the discrete log problem.

When an adversary gets the private key of a TIS, he cannot compromise past session because g_w is independent on Y_{TIS}. By same reason, the compromise of TSE's private key cannot compromise the past session.

Exhaustive Password Search and Dictionary Attack: In order to conduct dictionary attack or exhaustive password attack, an adversary has to have verifiable plaintext to know whether the guessed password is right by comparison the verifiable plaintext with the computation result using the guessed password. We summarize verifiable plaintext as following.

- g^{π}: to getting this an adversary must be able to decrypt a underlying protocol which is required for registration to provides authenticity and encryption.

- $E_{f(\pi)}(s)$, s and v: An adversary can get v $E_{f(\pi)}(s)$ from the observation of the message. However, the adversary needs a verifiable plaintext s to conduct the password guessing attack against $E_{f(p)}(s)$. Therefore, an adversary has to decrypt the underlying protocol (TLS/SSL) in the registration.

The first and the second are dependent on the security of the underlying protocol. The third and the forth are archived when an adversary compromises a TIS or a TIS and a client both. We assume that the underlying protocol is secure against eavesdropping and modification. Under this assumption, an adversary cannot get the first or the second information. Therefore, a passive adversary, who observes messages in the whole network, cannot conduct off-line dictionary attack.

On-line dictionary attack or exhaustive password search can be easily detected and thwarted, by counting consecutive authentication failure.

We explained that compromise of a session key cannot compromise the password. Also, long term keys of the TIS and TSE are independent on session keys, so compromise of other long term key of TIS or TSE and old password cannot compromise of current password.

Conclusively, only one way to get a password is compromise of a password file in a TIS or a TIS and a client and conducting off-line exhaustive password search or dictionary attack.

To make exhaustive password search and dictionary attack difficult when the adversary compromises password file, 1)Passphrase of which length is not limited is used, so the adversary conduct exhaustive password search in Z^*_q with exponentiation computation.2) Password rules to discourage or prevent users from using weak password must be imposed at the client side.

Single Point of Failure: We assume that a password file and a TIS's private key is kept in separate location. A TIS's private key could be kept in tamper-resistance hardware device.

When a password file in scheme is compromised, an adversary can get some user's password after successful dictionary attack with expensive exponentiation computations. Even if the adversary gets the password, he cannot decrypt the current session and past session.

When a password file at the TIS is compromised, an adversary cannot get any information about clients' passwords.

When s and v at clients in scheme is compromised, an adversary cannot get any information about client's password.

Compromise of the TIS's private is very dangerous, because all entity could be impersonated by adversaries, because he can issue tokens of any clients to him. However,

the current sessions and past sessions are protected by adversaries because of forward secrecy of our scheme. Moreover, only TIS needs to change his private key. The clients and the TSE does not need to change their keys.

Compromising of a TSE's private key, an adversary can impersonate the TSE. The current and past sessions could be protected by forward secrecy of our scheme.

Different Trust Level: The TIS can get $E_{f(\pi)}(s)$, a salt v and s during registration protocol execution. The TIS can conduct a dictionary attack to get passwords.

In scheme TIS cannot get a salt without compromise of a client system. In this case, the TIS cannot conduct a dictionary attack to get password.

The TSE and adversaries cannot conduct dictionary attack because t_1 is a random value. Getting $\pi+t_1$ is reduced to the discrete log problem.

Conclusively, a client must more trust the TIS than TSEs in case of scheme.

CONCLUSION

Password authentication protocols are easy to use, not expensive and pervasively used in the real world. Previous secure password authentication protocols for two parties are not comfortable in the distributed computing. This is due to the fact that a user has to type his password to log in the each server. It is not comfortable. Therefore, we more focus on authentication involving third trust parties, such like Kerberos. We developed two version of secure passwords authentication protocol for Single Sign On. The first provides the mobility of user but is weak against single point of failure. In second version, the user has to keep s and a salt v. Therefore, it limits the mobility of user, but is immune to single point of failure. Our secure password authentication protocols: (1)are secure against on/off-line dictionary attacks, (2) provide perfect forward secrecy, (3) does not keep plaintext-equivalent. Moreover, our protocols: (4) reduce harm of a single point of failure,

(5)combine secure information for authentication and attributes for authorization (6) provide no information about password to the target servers, who authenticates a user. Finally, our protocol inherently includes functions of generating attribute certificate, key distribution and secure authentication. Therefore, we do not need many efforts to integrate several technologies and mechanisms with our protocols, while previous password authentication protocols are integrated with other technologies to provide same functionalities which are provided by our protocols.

REFERENCES

1. Bellovin, S.M. and M. Merritt, 1991. Limitations of the Kerberos authentication system. In Proc. Winter Usenix Conf., Dallas TX (USA), pp: 253-267.
2. Bellovin, S.M. and M. Merritt, 1992. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In Proc. IEEE Symposium on Research in Security and Privacy, pp: 72-84.
3. Bellovin, S.M. and M. Merritt, 1993. Augmented encrypted key exchange: A password-based protocol secure against dictionary atttacks and password file compromise. In V. Ashby, editor, Proceedings of the 1st ACM Conference on Computer and Communications Security, Fairfax, Virginia, ACM Press, pp: 244-250.
4. Dierks, T. and C. Allen, 1999. The TLS Protocol Version 1.0. IETF.
5. Jablon, D.P., 1997. Extended password key exchange protocols immune to dictionary attack. In Proceedings of the WETICE' 97 Workshop on Enterprise Security, Cambridge, MA, USA.
6. Nyberg, K. and R.A. Rueppel, 1994. Message recovery for signature schemes based on the discrete logarithm problem. In A. D. Santis, 1995. editor, Advances in Cryptology-EUROCRYPT 94, volume 950 of Lecture Notes in Computer Science, Springer-Verlag, pp: 182-193.
7. Housley, S.R., 2002. An Internet Attribute profile for Authorization.
8. Steiner, J., B. Neuman and J. Schiller, 1988. Kerberos: an authentication service for open network systems. In Usenix Conference Proceedings, Dallas, Texas, pp: 191-202.
9. Wu, T., 1998. The secure remote password protocol. In Proceedings of the Symposium on Network and Distributed Systems Security (NDSS '98), San Diego, California, Internet Society, pp: 97-111.
10. Udayakumar, R., V. Khanna, T. Saravanan and G. Saritha, 2013. Retinal Image Analysis Using Curvelet Transform and Multistructure Elements Morphology by Reconstruction, Middle-East Journal of Scientific Research, ISSN: 1990-9233, 16(12): 1798-1800.
11. Udayakumar, R., V. Khanna, T. Saravanan and G. Saritha, 2013. Cross Layer Optimization For Wireless Network (Wimax), Middle-East Journal of Scientific Research, ISSN: 1990-9233, 16(12): 1786-1789.
12. Thooyamani, K.P., V. Khanaa and R. Udayakumar, 2013. A frame work for modelling task coordination in Multi-agent system, Middle-East Journal of Scientific Research, ISSN: 1990-9233, 15(12): 1851-1856.
13. Thooyamani, K.P., V. Khanaa and R. Udayakumar, 2013. An Integrated Agent System for E-mail Coordination using Jade, Indian Journal of Science and Technology, ISSN: 0974-6846, 6(6): 4758-4761.