International Journal of Students' Research in Technology & Management
eISSN: 2321-2543, Vol 6, No 2, 2018, pp 20-24
https://doi.org/10.18510/ijsrtm.2018.624

SECURE COMMUNICATION USING PFS IN A DISTRIBUTED ENVIRONMENT

[1]Rani Kumari, [2]Dr. Parma Nand, [3]Dr. Suneet Chaudhary
[1]Computer Science & Engineeering, Chitkara University, Baddi, India.
[2] Computer Science & Engineeering, SCSE, Galgotias University, Gr. Noida, India.
[3] Computer Science & Engineeering, Maharishi Markandeswar University, Ambala, Haryana, India.
Email: ranichoudhary04@gmail.com, parmaastya@gmail.com,suneetcit81@gmail.com

Article History: Submitted on 5nd January, Revised on 15th January, Published on 2nd March 2018

Abstract. Today millions of ordinary citizens are using networks for banking, shopping and filing their tax return. Network security has become a massive problem. All this requires network to identify its legal users for providing services. An authentication protocol used is Kerberos which uses strong secret key for user authentication but it is vulnerable in case of weak passwords. Authentication & key distribution protocols requires sharing secret key(s) with a view that only the concerned users know to derive the information from it. These protocols are vulnerable to key guessing attacks. Another important consideration is perfect forward secrecy in which our proposed scheme cover cases with application servers, authentication servers or clients key are revealed & their combination. In this paper our proposed scheme deal with key guessing attacks, perfect forward secrecy and protocols for few combinations of keys. All these protocols are based on the fact that the keys are weak & can be exploited easily.

Keywords. PFS, Attacks, Security Analysis, Security requirements.

INTRODUCTION

Network security problems can be divided roughly into four areas: Secrecy, authentication, non repudiation, & integrity control. Authentication deals with determining whom you are talking to before revealing sensitive information or entering into a business deal.
In recent years, a variety of protocols for authentication and key distribution have been proposed and applied to many communication system. Diffie and Hellman described how to establish a common session key by public messages. In [7], Needham et al. proposed a key distribution center that used encryption too. An authentication protocol used in many real systems which is based on a variant of Kerberos [6]. However, it is vulnerable in case of weak passwords. In this paper our proposed scheme focused on environment with weak keys and focus on key covering attacks and also perfect forward secrecy (PFS) which ensures that a **session** key derived from a set of long- term public & private keys will not be compromised if one of the private keys is compromised in the future[3].The remainder of this paper is organized as follows section 2 summaries the description of notation & security requirements. Section 4 presents a protocol for class -3PFS. Remaining section deals with class-7 PFS.

NOTATION & SECURITY REQUIREMENTS

Table 1. Notations

U	User
AS	Application server
AU	Authentication server
P_{AU}	Key shared between U and AU
S_{AS}	Secret key shared between AS and AU
K_S	Public key of AU
x, y, a, b, c, d	Random numbers
h( )	hash function
U→AS : M	U sends a message M to AS
g	Base generator
P	Large prime (P is the modulus of all Modular exponentiations)
$[data]_K$	Symmetric encryption of "data" with commom key K
[data]K	Asymmetric encryption of "data" with common key K

International Journal of Students' Research in Technology & Management
eISSN: 2321-2543, Vol 6, No 2, 2018, pp 20-24
https://doi.org/10.18510/ijsrtm.2018.624

Key Guessing attacks:
This type of attacks can take place both in offline as well as online mode. In online mode the authentication servers can detect such an attack by noticing continuous authentication failures[5]. In offline mode, an attacker eavesdrop communication messages during a protocol & stores them, then tries to find out the our weak password by verifying a password with the information stored.

Replay Attacks:
In this the data transmission is maliciously or fraudulently repeated or delayed. This is carried out either by the originator or by an adversary as part of the data & propagates it possibly as part of a masquerade attack by IP Pack substitution.

Perfect forward secrecy (PFS)
It is a property that ensures that a session key derived from a set of long term public & private keys will not be compromised if one of the private keys is compromised in the future. Based on the capability of protecting the client's password, the application server's secret key, and the authentication server's private key[2].Seven Classes of Perfect forward secrecy based on clients, application server and authentication server are as given below.

Class1 PFS
It provides low security because in this clients key is revealed to an attacker but still it does not help in obtaining the session keys of previous session, but the application server's secret key and the authentication server's private key is still secure.

Class 2 PFS
In this only application server key is revealed to the attacker where as authentication servers and clients password are revealed to the attacker[3] which makes it impossible to obtain the keys of the previous session.

Class 3 PFS and Class 4 PFS
In Class-3 protocol only the authentication server key is secure but the other keys are revealed to the attacker still the attacker can not obtain the sessions keys of previous session. In Class-4 Protocol the authentication servers key is revealed to the attacker[4], rest of the keys are secured which makes it a little vulnerable to attack but is still secured enough.

Class 5 ,Class 6 and Class 7 PFS's
In Class 5 and Class 6 application servers and clients key are secured respectively ,due to which both are vulnerable to attack . A class providing class 7 PFS means that if the clients key, authentication servers and application servers key are simultaneously revealed to an attacker, this protocol is highly secure it will not help the attacker in obtaining the keys of the previous sessions.

A PROTOCOL FOR CLASS 1 PFS

An efficient authentication and key distribution protocol providing class-1 PFS is proposed in this section. There are three principals involved in our protocol: an application server AS who provides services to clients, a client U who requests services from the application server, and an authentication server AU who is responsible for authentication and who distributes the common session key shared between the client *U* and the application server AU.

In this protocol, our proposed scheme assume that all principals know the server's public key *KS* in the system. We also assume that a poorly chosen password P_{AU} chosen by *A*(client) is known to *S(*server) via a secure channel [3]. Similarly, the application server's secret key S_{AS} is known to AU via a secure channel.
Our proposed protocol shown in Fig. 1 and the detailed steps are described as follows:

(1) U→AU : U,{U,AS ,P_{AU}, z}K_S :
A chooses a random number z and keeps it secret. Then, A encrypts user's password ,AS, P_{AU}, z, with server S's public key K_S and transmits the encrypted message as a request to S, where PAU is the password of A.

(2) *AU→AS*: *U*,[*U,AS,* [*U,K*]z, *K*]S_{AS} :

International Journal of Students' Research in Technology & Management
eISSN: 2321-2543, Vol 6, No 2, 2018, pp 20-24
https://doi.org/10.18510/ijsrtm.2018.624

After receiving client *A*'s message, the authentication server, *AU*, decrypts {*U,AS,,PAU, z*}K_S with his private key

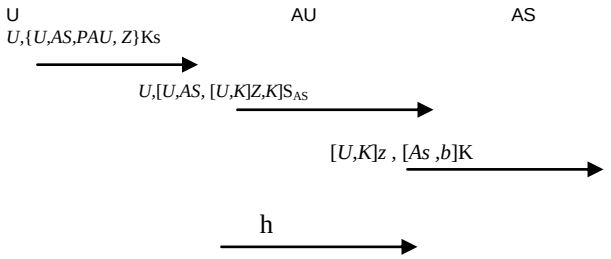

Fig. 1. A protocol providing PFS with low security

corresponding to the public key K_S and checks the authenticity of *U* by verifying *AU*'s password *PAU*. Then, he chooses a common key *K*. Hence, he can compute [*U,AS,* [*U,K*]*z, K*]S_{AS} , and transmit it to *AS*. Note that the value *z* also acts as a one-time key.

(3) *AS →U*: [*U,K*]*Z,* [*AS ,b*]*K*:
The application server *AS* first decrypts the message [*U,AS,* [*U,K*]*z ,K*]*SB* with his secret key *SAS* and gets the common key *K*. Then *AS* chooses a challenge value *b*, encrypts AS and *b* with the common key *K*, and sends [*U,K*]*z* and [*AS ,b*]*K* to the user *U*.

(4) *U→AS*: *b*:
In step 4, the user *A* decrypts message [*U,K*]*z* with *z* and gets the common key *K*. Then, he decrypts [*As ,b*]*K*, checks the validity of *K*, and sends the response value *b* to *AS*.
Finally, the user *U* and the application server *AS* can authenticate each other and compute the common session key *h(K)*.

SECURITY ANALYSIS

Guessing attacks
These can take place either in on-line or offline mode.
In online mode, a guessed password failure by an attacker can be detected and logged, which makes this highly resistant. In offline mode , clients password is used only to authenticate clients status in message 1, which is not include in any data that can be verified[3]. Thus an impersonator cannot verify his/her guess on the password unless being aware of the random number z , which makes our protocol immune to such kind of attacks.

Replay attack
Even though an attacker can replay an old message 1 as server can not decide the freshness of a message as he will be unable to know the random numbers z and servers secret key to decrypt the messages received excluding this information which does not help in compromising a future session key or to guess the password[1]. Thus, our protocol is secure against message replay attacks.

Efficiency of proposed scheme
In our scheme, if a user key is revealed, an attacker can know the key shared between him and server. But still the server private key is secure and hence random number z is secure in message 1, message 2 cannot be decrypted since server's secret key is also secure. Therefore attacker does not have any opportunity to get session keys which is what is required in class 1 PFS.

The details are described as follows:

(1) U→AS: U,{U,AS,PAU, z,g^x }K_S :

U chooses a z, a random number x and computes g^x . Then U encrypts U,AS,PAU, z,g^x by the server AU's public key K_S and sends the cipher text to AS.

International Journal of Students' Research in Technology & Management
eISSN: 2321-2543, Vol 6, No 2, 2018, pp 20-24
https://doi.org/10.18510/ijsrtm.2018.624

PROTOCOL PROVIDING PFS WITH HIGH SECURITY

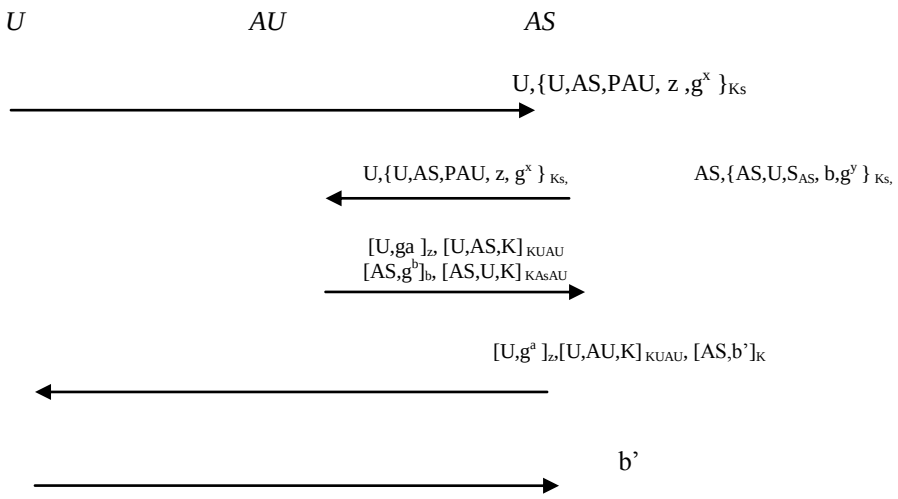

Figure2. A protocol providing PFS with high security.

(2) AS →AU: U,{U,AS,PAU, z,g^x }K_S,AS, {AS,U,SAS, b,g^y }K_S :

On receiving the client *U*'s message, the server AS chooses a b, and computes g^y mod P by choosing a random number *y*. Then, he encrypts AS, U, SAS, b, g^y with the server AU's public key KS. Both cipher texts {U,AS,PAU, z,g^x }K_S and {AS,U,SAS, b,g^y }K_S together with U and AS are sent to the server AU.

(3) AU→AS: [U,g^a]z, [U,AS,K]KUAU , [AS,g^b]b, [AS,U,K]KASAU :

After receiving Message 2, the authentication server AU decrypts it with his private key. *AU* checks the authenticity of *U* by verifying *U*'s password *PAU* and the authenticity of *AS* by verifying *AS*'s secret key *SAS*. He then chooses a common key *K*, computes {[*U*,g^a]*z*, [*U,AS,K*]*KUAU* , [*AS*,g^b]*b*, [*AS,U,K*]*KAS AU*}, and transmits it to *AS*, where *a* and *b* are chosen by AU, $KUAU = (g^x)^a = (g^a)^x = g^{xa}$ and $KASAU = (g^y)b = (g^b)^y = g^{yb}$ are used to securely pass the session key *K*.

(4) *AS →U*: [*U*,g^a]*z*, [*U,AS,K*]*KUAU* , [*AS,b'*]*K*:

The application server *AS* first decrypts the message [*AS*,g^b]*b* with *b* and computes $KASAU = (g^b)^y = g^{yb}$. He then decrypts [*AS,U,K*]*KASAU* with *KASAU* and gets the common key *K*. Then, *AS* uses *b' encr*ypts *AS*, *b'* with the common key *K* and sends [*U*,g^a]*z*, [*U,AS,K*]*KUAU* , [*AS,b'*]*K* to the client *U*.

(5) *U→AS*: *b'*:

The client *U* decrypts the message [*U*,g^a]*z* with *z* and computes $K_UAU = (g^a)^x = g^{xa}$. Then, he decrypts [*U,AS,K*]*KUAU* with K_UAU and gets the common key *K*. After that, he decrypts [*AS,b'*]*K*, checks the validity of *K*, and sends the response value *b'* to *AS*.On account of this, the client *U* and the application server *AS* can authenticate each other and compute the common session key.

SECURITY ANALYSIS AT THE SERVER SITE

Guessing attacks

The client's password is not used in this protocol[2]. This leads to data that can not be verified to make sure whether the guess is right or wrong. This makes our protocol immune to password guessing attacks.

Replay attacks

The server can not decide the freshness of the message still future session keys or password from servers reply can not be compromised. Thus, our protocol is secure against message replay attacks.

Class 7 PFS

Our protocol is based on the following well known hard problems which cannot be solved in polynomial time. A problem which cannot be solved in polynomial time is believed to be unsolvable. Even if keys related to client, application server and authentication server are all known by an attacker, session key cannot be obtained by an attacker due to the complexity of the calculation process. Therefore our scheme provides class 7 PFS.

CONCLUSION

In a distributed environment the client access services on application server through an authentication server. All the keys related to them are vulnerable to attack by an attacker. Depending upon preserving the clients, the application server and authentication servers keys have introduce several classes of perfect forward secrecy which are resistible to password guessing and replay attacks which makes them best while working in a distributed environment.

REFERENCES

[1] S. Lucks, Open key exchange: How to defeat dictionary attacks without encrypting public keys, in: Proc. of the Security Protocol Workshop,Springer-Verlag, April 1997.

[2] T. Wu, The secure remote password protocol, in: Internet Society Symposium on Network and Distributed System Security, 1998.

[3] T. Kwon, J. Song, Efficient key exchange and authentication protocol protecting weak secrets, IEICE Trans. Fundamentals E81-A (1) (January1998) 156–163.

[4] M. Steiner, G. Tsudik, M. Waidner, Refinement and extension of encrypted key exchange, Oper. Syst. Rev. 29 (3) (July 1995) 22–30.

[5] Y. Ding, P. Horster, Undetectable on-line password guessing attacks, Technical Report, TR-95-13-F, July 1995.

[6] J.T. Kohl, B.C. Neuman, T. Ts'o, The evolution of the kerberos authentication system, in: Distributed Open System, IEEE Comput. Soc. Press,1994, pp. 78–94.

[7] Needham,R.M. and M.D. Schroeder(1978) . " Using encryption for authentication in large networks of computers" Communications of the ACm, 21, 993-999.

.

