International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

Password Prestidigitation for Key Exchange Authentication Protocol

Anila P [1], Prajeesh C B[2]

[1]MEA College of Engineering, Kerala Technological University, Vengoor, Perinthalmanna, Malappuram, Kerala, India

[2] MEA College of Engineering, Calicut University, Vengoor, Perinthalmanna, Malappuram, Kerala, India

Abstract: ***The major problem addressed in accessing storage devices in parallel Network File System(pNFS) is the generation of key for many to many communication safely and securely .Normally, pNFS uses Kerberos based key exchange authentication protocol for the establishment of key which have some limitations such as the problem of scalability ,key escrow, forward secrecy. This paper proposes authenticated key exchange protocol names as password prestidigitation protocol that overcome the limitations of existing Kerberos based protocol.***

Keywords: NFS, pNFS, Kerberos, LIPKEY, key escrow, forward secrecy**,** password prestidigitation

1. Introduction

The file system is the methods and data structures that an operating system uses to keep track of files on a disk or partitions, i.e. the way the files are organized on the disks. In the parallel file systems the files are distributed among multiple storage devices or nodes that can allow the concurrent access of multiple tasks of a parallel program.

The Network File System (NFS)[1] protocol is a distributed file system protocol that was developed by Sun Microsystems. It allows a user on a client computer, which may be diskless, can access files over networks in a manner similar to how local storage is accessed. One major advantage of NFS is that it provides central management. The main disadvantage of NFS is the problem of security. Because NFS is based on RPC, remote procedure calls, which is not yet matured, it is inherently insecure and it should only be used on a trusted network behind a firewall.

The most recent version of NFS is NFSv4.1 (NFS version 4.1) protocol.NFSv4.1 provides a feature called parallel NFS (pNFS) [1] that allows direct and concurrent client access to multiple storage devices in the network to improve the performance and scalability of the network.

The figure 1.1 shows the architecture of pNFS. One main advantage of parallel NFS is that, the application server or client can access the storage devices in parallel over multiple data path to storage servers or nodes. A Metadata server which is not stand between the data path of client and storage devices supplies the client with the location of the data that is in the storage devices. The client can read and write the data directly to the file in the storage devices. The parallel NFS work with three protocols namely:

- Parallel NFS protocol
- Control protocol
- Storage access protocol

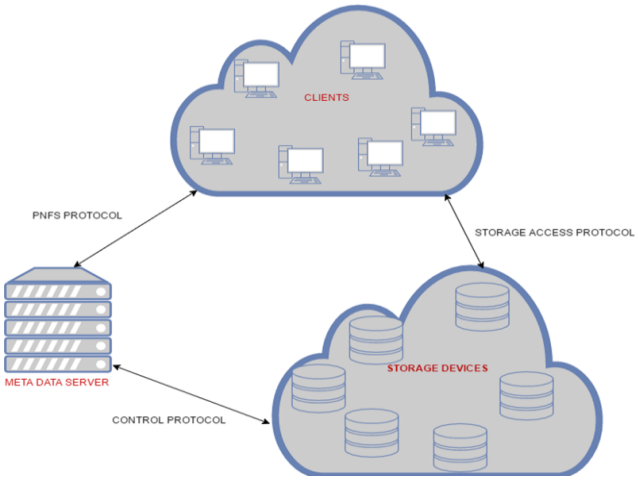


Figure 1.1: Architecture of pNFS

Parallel NFS is important because it brings together the benefits of parallel I/O and a powerful standard for network file systems (NFS). This will allows the clients to experience high performance and scalability in their storage devices with the guarantee of security.

Key-exchange protocols are mechanisms by which two parties that communicate over a controlled network. This protocol can generate a common secret key. Key-exchange protocols are very important for enabling the use of shared-key cryptography to protect transmitted data over the insecure networks. Therefore, they are the central piece for building secure communications and are the most commonly used protocol in cryptographic protocols.

Kerberos is a computer network authentication protocol that helps people to transfer the informations from one place to another place in computer networks. In Kerberos based solutions, the communication between client and storage device is taken place through the metadata server which is capable of generating the authentication key. This kerberos works on the basis of tickets to allow nodes communicating over a non-secure network to prove their identity to one another in a secure manner. Kerberos is a centralized network authentication system developed by MIT. It is the default authentication method in Windows 2000 and later. Kerberos

uses symmetric key cryptography, requires a trusted third party which leads to the problem of key escrow and can use public-key cryptography during certain phases of authentication if needed.

Scalability, Forward secrecy and Escrow free are some of the desirable properties which are not achievable by this Kerberos-based[2] solution and also they are weak in provide security to the long-term keys.

- **Scalability**: In the case of Kerberos, server is responsible for giving access to the storage devices as well as generating the session keys for the coming clients so that heavy traffic network may be a burden for the server. Therefore the scalability is limited in the case of Kerberos.
- **Forward secrecy**: It is a major property of secure communication in which compromise of a long term key does not compromise the past session keys. It protects past session keys against future compromise of the long term keys. Simply it means provide security to the past session keys when long-term keys are used.
- **Key escrow**: In key escrow the keys which are needed to decrypt or encrypt the data are held in escrow or an agreement so that under certain circumstances an authorized third party may gain access to those keys. It should be protected in key exchange protocol to secure the communication.

This work mainly focused on achieving these desirable properties. Here a single protocol is developed which is the integration of two protocols. In the first protocol client computes the key materials earlier and generate session keys based on that. So in this can achieve key escrow as well as scalability. The session keys are generated by the client and the server has no role in generating the session key so that the third party involvement is not there and also thereby decreasing the workload of server. Hence there is no third party who can view the session key for the secure communication between the client and storage devices. The second protocol use password authentication through prestidigitation method. That is, a key is generated after passing through three key generation function.

The objective of this work is to develop an authenticated key exchange protocol that can provide the following properties:

- Provide security for the long-term secret keys
- Reduce the workload of metadata server
- Provide Forward secrecy
- Provide Escrows freeness
- Provide resistance to password guessing attacks

2. Literature Survey

Encrypted key exchange [3] uses the combination of symmetric and asymmetric cryptography. A secret key is used to encrypt a randomly generated public key which is shared among two parties such as passwords, use it to exchange secret information, called EKE.

Simple password-based encrypted key exchange protocols[4] are the protocols by which the secret keys are chosen from small set of possible values rather than from large space which is uniformly distributed. This paper introduce two protocols SPAKE1(Simple non-concurrent Password-based encrypted Key Exchange) and SPAKE2 (Simple concurrent Password-based encrypted Key Exchange).SPAKE1 is a non-concurrent password based encrypted key exchange protocol similar to encrypted key exchange protocol. SPAKE2 is a concurrent password based encrypted key exchange protocol which is also similar to encrypted key exchange protocol and SPAKE1 and the only difference is in the session key derivation function which will also include the password to the key generation function.The session key is set to be the hash of the session identification, the user identities, and the Di±e-Hellman key and the password also.

Password-based authenticated key exchange in the three-party setting[5] says about 3-Party password based key exchange in which instead of using many passwords to communicate with different clients, they use a single password that will be shared with the trusted server. The advantage of this protocol is that it can communicate with many number of users by sharing only a single password. The main limitations of this protocol is that a server is needed whenever a party wants to communicate.

The SPEKE Protocol[6] is a little modification of Diffie-Hellman key exchange where the Diffie-Hellman generator g is created from a hash of the password. SPEKE prevents man in the middle attack. An adversary who is able to read and modify all messages between two parties cannot learn the shared key K and cannot make more than one guess for the password in each interaction with a party that knows it. SPEKE protocol leads to an impersonation attack when a person is engaged in two parallel sessions with an active attacker. The original SPEKE protocol is subject to a key-malleability attack. An active attacker can test multiple passwords for hacking in one protocol execution.

Generic Security Service Application Program Interface[7] is an application program interface that creates a security context by which data can be exchanged between applications and it provides security services that give protection to the data to be transmit. GSS-API is independent of the underlying protocols and the addressing modes and also independent of the underlying mechanisms. RPCSEC_GSS[8] protocol allows the Remote Procedure Call to access the Generic Security Services API.

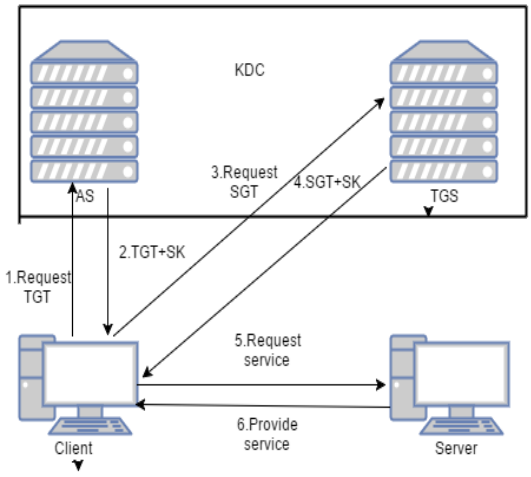
The Simple Public-Key GSS-API Mechanism[9] include the procedures, conventions, protocols to be employed by the peers that provide GSS_API security services while using simple public key mechanisms. It is important to provide GSS mechanism to public key rather than symmetric key. SPKM allows both single and mutual authentication without using a specific time stamps.

Low Infrastructure Public Key Mechanism[10] uses GSS_API mechanism to provide a secure channel between client and server by using password for client authentication and public key certificate for server authentication. Therefore it analogous to the low infrastructure usage of Transport

Layer Security (TLS).It also provide integrity and privacy. LIPKEY improves the current SPKM method.

An Authentication Protocol Based on Kerberos 5[11] is a computer network authentication protocol that helps people from purloin information that sent across one place to another place in computer networks. In Kerberos based solutions, the communication between client and storage device is taken place through the metadata server which is capable of generating the authentication key. This Kerberos works on the basis of tickets that prove the identity of people who communicate over a non-secure network in a secure manner. Kerberos is a centralized network authentication system developed by MIT. It is the default authentication method in Windows 2000 and later. Kerberos uses symmetric key cryptography and it requires a trusted third party. Kerberos can also use public-key cryptography during certain phases of authentication if needed.

In Kerberos protocol, the client will repeatedly authenticate to multiple servers by assuming that there is a long-term secret key shared between the client and Kerberos infrastructure. The long-term secret key of client was generated by using the client's password. A simplified overview of the Kerberos actions is shown in Figure 2.1.



AS →Authentication Server
TGT → Ticket Granting Ticket
KDC →Key Distribution Center
TGS →Ticket Granting Server
SGT →Server Granting Ticket
SK →Session Key

Figure 2.1 Overview of Kerberos authentications.

Kerberos have several limitations[12] such as sensitive to password guessing attacks, The system clocks of the hosts should be synchronized, KDC should be always available, Difficult to achieve scalability, Cannot achieve Forward secrecy, Cannot achieve Escrow freeness.

3. Proposed System

It contains mainly three phases:

- Developing a protocol for avoiding scalability problem and key escrow problem in Kerberos.
- Developing a protocol for avoiding forward secrecy problem of Kerberos.
- Integrating the above two protocols into one.

3.1 Developing authenticated key exchange protocol 1

It has mainly two phases .In phase 1 the client pre-compute the key materials and send that to the server and the server generate tickets and send to client. In phase 2 server generate layout and send to client, then the client then generate session key and send to storage device. After that storage device check the session key and if it genuine they can communicate. This is shown in the following figures

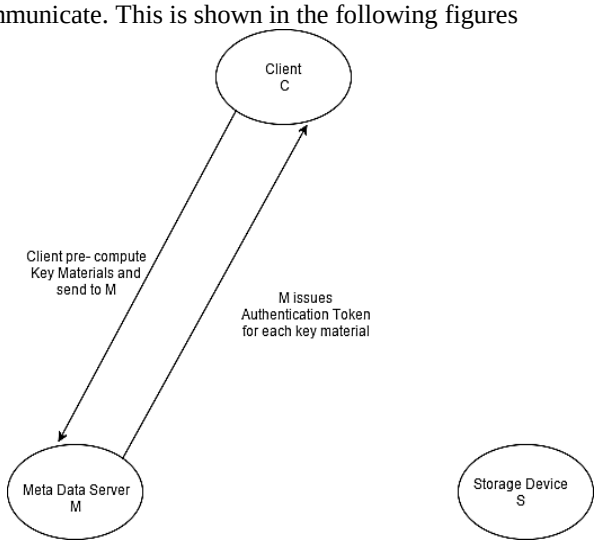


Figure 3.1.1: Data Flow Diagram of Phase 1 of AKE 1

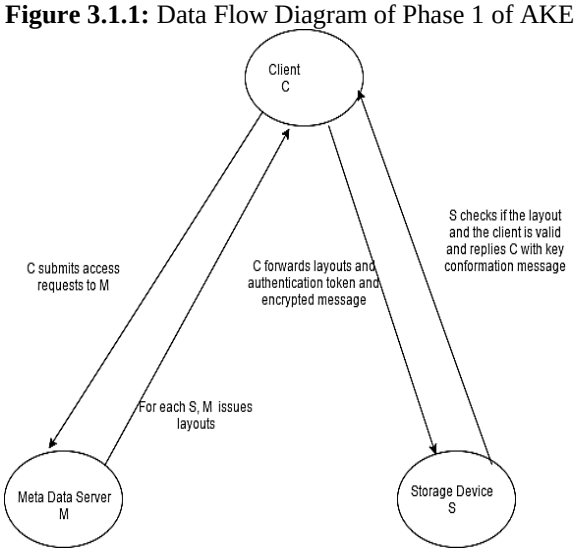


Figure 3.1.2: Data Flow Diagram of Phase 2 of AKE 1

This protocol can be regarded as a modified version of Kerberos that allows the client to generate its own session keys. That is, the key material used to derive a session key is pre-computed by the client and forwarded to the corresponding storage device in the form of an authentication token. This protocol contains two phases. In the first phase, the client set the valid time period v. It is the time the client can access the storage devices. It may take a week or month etc. In that period the client will accept the all key details of the storage devices it needed to access in that period. The server will control all these operation.

In the algorithm C represents the client, S represents the Server and Sd represent the Storage device. IDC represents the identity of the client. This IDC is generated from the password of the client. KCM is the secret key of the client to server path. The KCS represent the secret key of client to the storage device path. E() represent the encryption mechanism. V is the validity period and t is the access request time. This t will be inside the V. The SK represents the session key of the client and storage device. This session key is used for the secure communication of both. There are two phases for this protocol. AT is the authentication tokens. The layout contains the location of the file systems, its access permissions, its statistic details etc. After this process the client has the entire access to the file it requested from the storage device.

3.2 Password prestidigitation protocol

This protocol is activating at each time the encryption process will takes place. Also the key generation function is used whenever the client or server generates the secret key for the communication. Below the data flow diagram of the key generation function is shown. Each circle indicates the key generation function. The output of the one function is given as the input to the other function. The arrow indicates this**.** KGF indicates the key generation function.

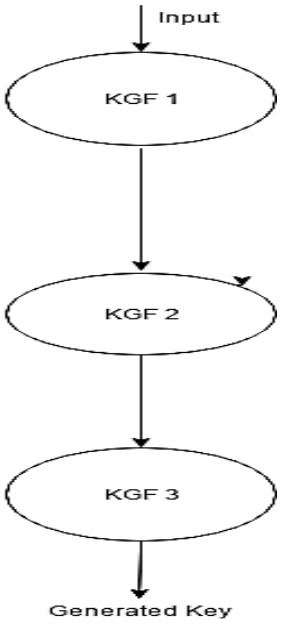


Figure 3.2.1: Password prestidigitation

4. Results and Discussion

The performance of the two protocol is compared by using the concepts number of cryptographic operations for a number of access requests from client to storage device via metadata server over a certain time period. The comparison is done in terms of number of cryptographic operations for Meta-data Server, Storage device, client and the overall operations is shown below. The comparison is done by taking the two authenticated protocols developed by Hoon Wei Lim and Guomin. Here ,the following are the comparison graphs to indicate the number of cryptographic operations included in Kerberos and password prestidigitation protocol. Here w is the Number of Access Requests, n is the Number of Storage Devices that a Client Access Concurrently and N is the Total Number of Storage Devices

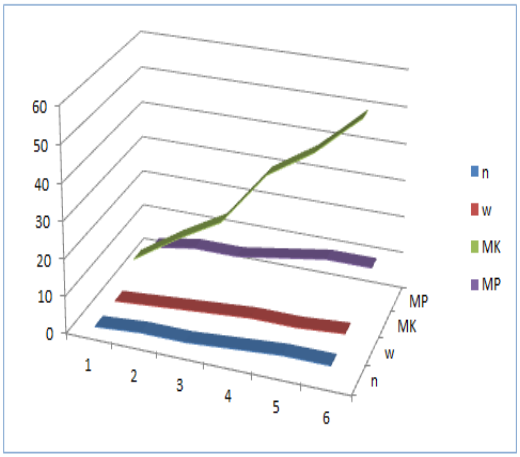


Figure 4.1: Comparison chart for Kerberos and password prestidigitation protocol in terms of number of cryptographic operations in metadata server.

Figure 4.1 shows that the password prestidigitation protocol reduce the workload of the meta data server in kerberos protocol by approximately up to half. Here, Number of Cryptographic Operations in Metadata Server in case of kerberos protocol is calculated by w(n+5) and Number of Cryptographic Operations in Metadata Server in case of Password Prestidigitation protocol is calculated by N+1 where we assume that N=2n.In the graph, MK represents the number of cryptographic operations in metadata server of Kerberos and MP represents the number of cryptographic operations in metadata server of password prestidigitation protocol.

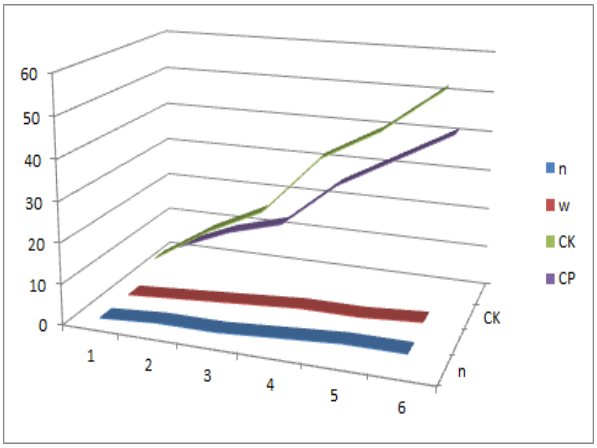


Figure 4.2: Comparison chart for Kerberos and password prestidigitation protocol in terms of number of cryptographic operations in client.

Figure 4.2 shows that the password prestidigitation protocol reduce the workload of the client in kerberos protocol by approximately up to half. Number of Cryptographic Operations in Client in case of kerberos protocol is calculated by w(2n+3)and Number of Cryptographic Operations in Client in case of Password Prestidigitation protocol is calculated by 2(wn+1). In the graph, CK represents the number of cryptographic operations in client of

Kerberos and CP represents the number of cryptographic operations in client of password prestidigitation protocol.

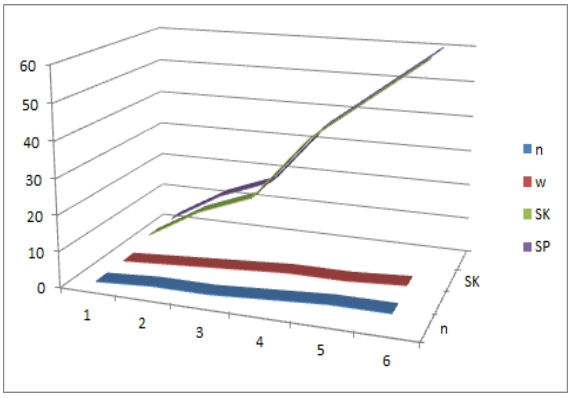


Figure 4.3: Comparison chart for Kerberos and password prestidigitation protocol in terms of number of cryptographic operations in each storage device.

Figure 4.3 shows that the password prestidigitation protocol and Kerberos having the same workload in case of storage devices. Number of Cryptographic Operations of Storage Devices in case of kerberos as well as password prestidigitation protocol is calculated by 3wn.

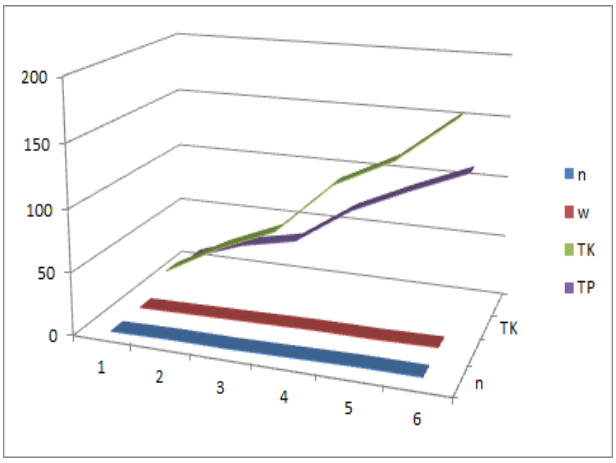


Figure 4.4: Comparison chart for Kerberos and password prestidigitation protocol in terms of number of cryptographic operations in total.

The total calculated number of cryptographic operations is less in case of password prestidigitation protocol than in kerberos protocol. Number of Cryptographic Operations in Total in case of kerberos protocol is calculated by w (6n+8) and Number of Cryptographic Operations in Total in case of Password Prestidigitation protocol is calculated by (5wn+N+2).

5. Conclusion

In this work I conclude that the password prestidigitation protocol address the limitations of Kerberos based protocol. This proposed protocol can increase the scalability of the network and can achieve the forward secrecy. The protocol can also provide the escrow freeness. The protocol is free from password guessing attack also. So that, it can be concluded that password prestidigitation protocol is a better choice for authenticated key exchange protocol than Kerberos.

References

[1] Pawlowski, Brian, et al. "The NFS version 4 protocol." *Proceedings of the 2nd International System Administration and Networking Conference (SANE 2000)*. Vol. 2. No. 5. 2000.
[2] Lim, Hoon Wei, and Guomin Yang. "Authenticated Key Exchange Protocols for Parallel Network File Systems." *IEEE Transactions on Parallel and Distributed Systems* 27.1 (2016): 92-105.
[3] Bellovin, Steven M., and Michael Merritt. "Encrypted key exchange: Password-based protocols secure against dictionary attacks." *Research in Security and Privacy, 1992. Proceedings., 1992*.
[4] Abdalla, Michel, and David Pointcheval. "Simple password-based encrypted key exchange protocols." *Cryptographers' Track at the RSA Conference*. Springer Berlin Heidelberg, 2005.
[5] Abdalla, Michel, Pierre-Alain Fouque, and David Pointcheval. "Password-based authenticated key exchange in the three-party setting." *International Workshop on Public Key Cryptography*. Springer Berlin Heidelberg, 2005.
[6] Hao, Feng, and Siamak Fayyaz Shahandashti. "The SPEKE Protocol Revisited." *SSR* 14 (2014): 26-38.
[7] Linn, John. "Generic security service application program interface version 2, update 1." (2000).
[8] Eisler, Michael, Alex Chiu, and Lin Ling. *RPCSEC_GSS protocol specification*. No. RFC 2203. 1997.
[9] Adams, Carlisle. "The simple public-key GSS-API mechanism (SPKM)." (1996).
[10] Eisler, Mike. "LIPKEY-a low infrastructure public key mechanism using SPKM." (2000).
[11] El-Emam, Eman, et al. "An Authentication Protocol Based on Kerberos 5."*IJ Network Security* 12.3 (2011): 159-170.
[12] Bellovin, Steven M., and Michael Merritt. "Limitations of the Kerberos authentication system." *ACM SIGCOMM Computer Communication Review*20.5 (1990): 119-132.

Author Profile

Anila P received her B.Tech. degree in computer science and engineering from the Mar Baselius College of Engineering And Technology and Technology, Kerala, in 2014. Right now she is pursuing her M. Tech degree in computer science at MEA Engineering College, Kerala from 2015 to 2017, Anila worked as a Program Analyst Trainee at Cognizant Technology Solutions(CTS) Tamilnadu. Her research interests lie in Network Security.

Prajeesh C B received his B.Tech. degree in computer science and engineering from the College of Engineering Adoor, Kerala, in 2007 and received his M. Tech degree in computer science at MEA Engineering College, Kerala in 2014 . Right now he is working as Assistant Professor in the Department of Computer Science at MEA Engineering College, Kerala. His research interests lie in Network Security and data mining.