ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

# A Key Exchange Protocol for PNFS in Many-To Many Communications Environment That Reduces Workload on Metadata Server

**MohanadHameed Rashid**
**Msc(Is),**
**Department Computer Science (PG),**
**Nizam College-Osmania Universtiy.**

**T. RamdasNaik**
**Assistant Professor,**
**Department Computer Science (PG),**
**Nizam College-Osmania Universtiy.**

## Abstract:

In present day scenario, authentication of protocols is very important for secure many-many communication. The problem arises by proliferation of large scale distributed file system support the parallel access file system i.e. parallel Network File System (pNFS) the communication between clients to storage system may establish the session keys of Kerberos file system. Parallel storage based on pNFS is the next evolution beyond clustered NFS storage and the best way for the industry to solve storage and I/O performance blockages. But the file system of Kerberos drawbacks are Meta data does not provide more security between client and server.

Meta data servers generate themselves for all session key which lied between client and server. Meta data exchange exhibits heavy work load that restrict the scalability of protocols. In this paper we present a Varity of security key exchange protocols that are design to solve the above issue which arises. We show that our protocol have been reduced significant amount of workload of the Meta data server and concurrently support guarantee upon certain condition and forward secrecy and we improve performance with effectively.

## Keywords:

Network File systems, Kerberos, Metadata, pNFS, Key exchange, authentication.

## Introduction:

In a parallel file system, file data is distributed across multiple storage devices or nodes to allow concurrent access by multiple tasks of a parallel application. This is typically used in large-scale cluster computing that focuses on high performance and reliable access to large datasets. That is,higher I/O bandwidth is achieved through concurrent access to multiple storage devices within large compute clusters; while data loss is protected through data mirroring using fault-tolerant striping algorithms. Some examples of high performance parallel file systems that are in production use are the IBM General Parallel File System (GPFS), Google File System (Google FS), Lustre, Parallel Virtual File System (PVFS), and Panasas File System; while there also exist research projects on distributed object storage systems such as Usra Minor, Ceph, XtreemFS, and Gfarm.

These are usually required for advanced scientific or data-intensive applications such as, seismic data processing, digital animationstudios, computational fluid dynamics, and semiconductor manufacturing. In these environments, hundreds or thousands of file system clients share data and generate very high aggregate I/O load on the file system supporting petabyte- or terabyte-scale storage capacities. Independent of the development of cluster and high performance computing, the emergence of cloud, and the MapReduce programming model has resulted in

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

file systems such as the Hadoop Distributed File System (HDFS), Amazon S3 File System, and Cloud-Store. This, in turn, has accelerated the wide-spread use of distributed and parallel computation on large datasets in many organizations. Some notable users of the HDFS include AOL, Apple, eBay, Facebook, Hewlett-Packard, IBM, LinkedIn, Twitter, and Yahoo!. In this work, we investigate the problem of secure manyto- many communications in large-scale network file systems that support parallel access to multiple storage devices. That is, we consider a communication model where there are a large number of clients (potentially hundreds or thousands) accessing multiple remote and distributed storage devices (which also may scale up to hundreds or thousands) in parallel. Particularly, we focus on how to exchange key materials and establish parallel secure sessions between the clients and the storage devices in the parallel Network File System (pNFS) the current Internet standard—in an efficient and scalable manner.

The development of pNFS is driven by Panasas, Netapp, Sun, EMC, IBM, and UMich/CITI, and thus it shares many common features and is compatible with many existing commercial/proprietary network file systems. Our primary goal in this work is to design efficient and secure authenticated key exchange protocols that meet specific requirements of pNFS. pNFS removes the performance bottleneck in traditional NAS systems by allowing the compute clients to read and write data directly and in parallel, to and from the physical storage devices. The NFS server is used only to control metadata and coordinate access, allowing incredibly fast access to very large data sets from many clients.

**Related work:**

1. Authenticated key exchange secure against dictionary attacks.

AUTHORS: M. Bellare, D. Pointcheval, and P. Rogaway

Description:

Password-based protocols for authenticated key exchange (AKE) are designed to work despite the use of passwords drawn from a space so small that an adversary might well enumerate, off line, all possible passwords. While several such protocols have been suggested, the underlying theory has been lagging. The author begin by defining a model for this problem, one rich enough to deal with password guessing, forward secrecy, server compromise, and loss of session keys. The one model can be used to define various goals. The author takes AKE (with "implicit" authentication) as the "basic" goal, and they give definitions for it and for entity-authentication goals as well. Then they prove correctness for the idea at the center of the Encrypted Key-Exchange (EKE) protocol of Bellovin and Merritt: they prove security, in an idealcipher model, of the two-flow protocol at the core of EKE.

2. Analysis of key-exchange protocols and their use for building secure channels

AUTHORS: Ran Canetti and Hugo Krawczyk

Description:

In this paper author presents a formalism for the analysis of key-exchange protocols that combines previous definitional approaches and results in a definition of security that enjoys some important analytical benefits: (i) any key-exchange protocol that satisfies the security definition can be composed with symmetric encryption and authentication functions to provide provably secure communication channels (as defined here); and (ii) the definition allows for simple modular proofs of security: one can design and prove security of key-exchange protocols in an idealized model where the communication links are perfectly authenticated, and then translate them using general tools to obtain security in the realistic setting of adversary-controlled links. This paper adopts a methodology for the analysis of key-exchange protocols. They follow the approach of the adversarial model.

3. Authenticated Key Exchange Protocols for parallel Network File Systems

AUTHORS: Hoon Wei Lim Guomin Yang

Description: In this paper the authors study the problem of key establishment for secure many-to-many communications. The problem is inspired by the

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

proliferation of large-scale distributed file systems supporting parallel access to multiple storage devices. Their work focuses on the current Internet standard for such file systems, i.e., parallel Network File System (pNFS), which makes use of Kerberos to establish parallel session keys between clients and storage devices. They overcome the number of limitations: (i) a metadata server facilitating key exchange between the clients and the storage devices has heavy workload that restricts the scalability of the protocol; (ii) the protocol does not provide forward secrecy; (iii) the metadata server generates itself all the session keys that are used between the clients and storage devices, and this inherently leads to key escrow.

4. Block level security for network-attached disks

AUTHORS: Marcos K. Aguilera, MinwenJi, Mark Lillibridge

Description:

They propose a practical and efficient method for adding security to network-attached disks (NADs). Their design requires no changes to the data layout on disk, minimal changes to existing NADs, and only small changes to the standard protocol for accessing remote block-based devices. They have implemented a prototype NAD file system, called Snapdragon that incorporates their ideas. They also evaluated Snapdragon's performance and scalability. In this paper they have presented a new block-based security scheme for network-attached disks (NADs). In contrast to previous work, their scheme requires no changes to the data layout on disk and only minor changes to the standard protocol for accessing remote block-based devices.

## EXISTING SYSTEM:

- Independent of the development of cluster and highperformance computing, the emergence of clouds and the MapReduce programming model has resulted in file systems such as the Hadoop Distributed File System (HDFS), Amazon S3 File System, and Cloud-Store. This, in turn, has accelerated the wide-spread use of distributed and parallel computation on large datasets in many organizations.
- Some of the earliest work in securing large-scale distributed file systems, for example, have already employed Kerberos for performing authentication and enforcing access control. Kerberos, being based on mostly symmetric key techniques in its early deployment, was generally believed to be more suitable for rather closed, well-connected distributed environments.
- On the other hand, data grids and file systems such as, OceanStore, LegionFS and FARSITE, make use of public key cryptographic techniques and public key infrastructure (PKI) to perform cross-domain user authentication.

## DISADVANTAGES OF EXISTING SYSTEM:

- The current design of NFS/pNFS focuses on interoperability, instead of efficiency and scalability, of various mechanisms to provide basic security. Moreover, key establishment between a client and multiple storage devices in pNFS are based on those for NFS, that is, they are not designed specifically for parallel communications. Hence, the metadata server is not only responsible for processing access requests to storage devices (by granting valid layouts to authenticated and authorized clients), but also required to generate all the corresponding session keys that the client needs to communicate securely with the storage devices to which it has been granted access.
- Consequently, the metadata server may become a performance bottleneck for the file system. Moreover, such protocol design leads to key escrow. Hence, in principle, the server can learn all information transmitted between a client and a storage device. This, in turn, makes the server an attractive target for attackers.
- Another drawback of the current approach is that past session keys can be exposed if a

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

storage device's long-term key shared with the metadata server is compromised. We believe that this is a realistic threat since a large-scale file system may have thousands of geographically distributed storage devices. It may not be feasible to provide strong physical security and network protection for all the storage devices.

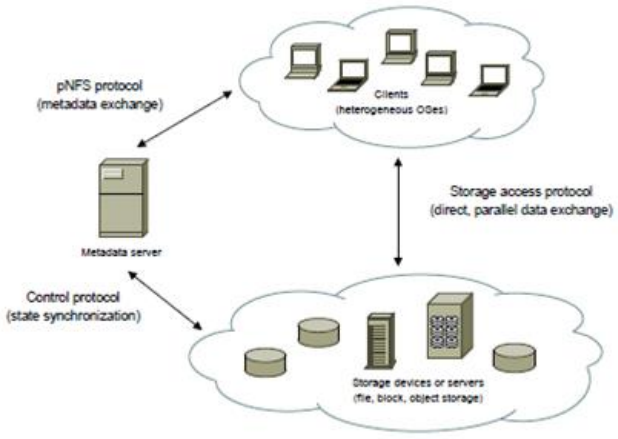
## PROPOSED SYSTEM:

- In this work, we investigate the problem of secure many to- many communications in large-scale network file systems that support parallel access to multiple storage devices. That is, we consider a communication model where there are a large number of clients (potentially hundreds or thousands) accessing multiple remote and distributed storage devices (which also may scale up to hundreds or thousands) in parallel.
- Particularly, we focus on how to exchange key materials and establish parallel secure sessions between the clients and the storage devices in the parallel Network File System (pNFS)—the current Internet standard—in an efficient and scalable manner. The development of pNFS is driven by Panasas, Netapp, Sun, EMC, IBM, and UMich/CITI, and thus it shares many common features and is compatible with many existing commercial/proprietary network file systems.
- Our primary goal in this work is to design efficient and secure authenticated key exchange protocols that meet specific requirements of pNFS.
- The main results of this paper are three new provably secure authenticated key exchange protocols. Our protocols, progressively designed to achieve each of the above properties, demonstrate the trade-offs between efficiency and security.
- We show that our protocols can reduce the workload of the metadata server by approximately half compared to the current Kerberos-based protocol, while achieving the desired security properties and keeping the computational overhead at the clients and the storage devices at a reasonably low level. We define an appropriate security model and prove that our protocols are secure in the model.

## ADVANTAGES OF PROPOSED SYSTEM:

- The proposed system achieves the following three:
- Scalability – the metadata server facilitating access requests from a client to multiple storage devices should bear as little workload as possible such that the server will not become a performance bottleneck, but is capable of supporting a very large number of clients.
- Forward secrecy – the protocol should guarantee the security of past session keys when the long-term secret key of a client or a storage device is compromised.
- Escrow-free – the metadata server should not learn any information about any session key used by the client and the storage device, provided there is no collusion among them.

## SYSTEM ARCHITECTURE:




## MODULES

**The module description is as below,**

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

### 1. Parallel Sessions

The parallel sessions are the parallel secure sessions between the clients and storage devices which are in the parallelnetwork file system (pNFS). These are the current internet standards in an efficient and scalable manner. These aresimilar to the situations where once the adversary compromises the long-term secret key, it can learn the subsequencesessions. If an honest client and an honest storage device complete the matching sessions then they compute the samesession key. Secondly two of our protocols provide forward secrecy: one is partially forward secure with respect tomultiple sessions within a time period.

### 2. Authenticated key exchange

The primary goal in this work is to design the efficient and secure authenticated key exchange protocol that meets thespecific requirements of pNFS. Three new provably secure authenticated key exchange protocols are the main results ofthis paper.We describe our design goals and give some intuition of a variety of pNFS authenticated key exchanged(pNFS-AKE) protocols that we consider in this work.

### 3. Forward secrecy

The protocol should guarantee the security of the past session keys when the long-term secret key of a client or a storagedevice is compromised as the protocol does not provide any forward secrecy. To address key escrow while achievingforward secrecy simultaneously, we incorporate a Diffie-Hellman key agreement technique into Kerberos-like pNFSAKE-I.However, it is to be noted that we gain only partial forward secrecy, by trading efficiency over security.

### 4. Client

The Client performs the following tasks,Share Data

### Upload Data

The user can upload the file to the cloud. And then the Admin can allow the data to store on the cloud.

### Download File

The user can also download the cloud file by the conditions.

### 5. Server Authentication

### Accept user

The admin can accept the new users request and can also block the users.

### Allow user file

The users can upload the file to cloud. And the admin can allow the files to cloud then only the file can store the cloud.

### 6. CLOUD

### Upload Data

The cloud can mostly upload the 3 types of files to users namely JAVA, DOT NET and PHP.

## CONCLUSION:

We implemented the three authenticated key exchange protocols for the parallel network file system (pNFS). The three appealing advantages are offered by our protocols over the existing Kerberos-based pNFS protocol. Firstly the metadata server which is executing our protocols has much lower workload as compared to that of the Kerberos-based approach. Secondly, two of our protocols provide the forward secrecy: one which is partially forward secure, while other is the fully forward secure. Thirdly we also have designed a protocol which provides forward secrecy as well as is escrow-free.

## REFERENCES:

[1] Hoon Wei Lim Guomin Yang, “Authenticated Key Exchange Protocols for Parallel Network File Systems”, IEEE Transactions on Parallel and Distributed Systems 2015.

[2] A. Adya, W.J. Bolosky, M. Castro, G. Cermak, R. Chaiken, J.R. Douceur, J. Howell, J.R. Lorch, M. Theimer, and R. Wattenhofer. FARSITE: Federated, available, and reliable storage for an incompletely trusted environment. In Proceedings of the 5th Symposium on Operating System Design and Implementation (OSDI). USENIX Association, Dec 2002.

[3] M.K. Aguilera, M. Ji, M. Lillibridge, J. MacCormick, E. Oertli, D.G. Andersen, M. Burrows, T. Mann, and C.A. Thekkath. Blocklevel security for network-attached disks.In Proceedings of the 2nd International Conference on File and Storage Technologies (FAST). USENIX Association, Mar 2003.

[4] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. A view of cloud computing. Communications of the ACM, 53(4):50–58. ACM Press, Apr 2010.

[5] D. Boneh, C. Gentry, and B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys.In Advances in Cryptology – Proceedings of CRYPTO, pages 258– 275. Springer LNCS 3621, Aug 2005.

[6] R. Canetti and H. Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In Advances in Cryptology – Proceedings of EUROCRYPT, pages 453–474. Springer LNCS 2045, May 2011.

[7] J. Dean and S. Ghemawat. MapReduce: Simplified data processing on large clusters. In Proceedings of the 6th Symposium on Operating System Design and Implementation (OSDI), pages 137–150. USENIX Association, Dec 2004.

[8] S. Emery. Kerberos version 5 Generic Security Service Application Program Interface (GSSAPI) channel binding hash agility. The Internet Engineering Task Force (IETF), RFC 6542, Mar 2012.

[9]K Bhoopal, P Manikumar, P Priyaraaga& G Deepthi, Efficient Co-Operative Key Exchange Protocol, IJMETMR, Volume No: 2 (2015), Issue No: 4 (April), http://www.ijmetmr.com/olapril2015/KBhoopal-PManikumar-PPriyaraaga-GDeepthi-70.pdf

[10] M. Abd-El-Malek, W.V. Courtright II, C. Cranor, G.R. Ganger, J. Hendricks, A.J. Klosterman, M.P. Mesnier, M. Prasad, B. Salmon, R.R. Sambasivan, S. Sinnamohideen, J.D. Strunk, E. Thereska, M. Wachs, and J.J. Wylie. Ursa Minor: Versatile cluster-based storage. In Proceedings of the 4th USENIX Conference on File and Storage Technologies (FAST), pages59–72. USENIX Association, Dec 2005.