International Journal of Research
Available at https://edupediapublications.org/journals

p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 11
July 2016

A Parallel Network file System Protocol(Pnfs) for Generating a Session key to Provide A File Security

[1] K. LALITHA, [2] M. ERANNA

[1]M.Tech Dept of CSE, PVKK Institute of Technology, Affiliated to JNTUA, AP, India .
[2]Assistant Professor, Dept of CSE, PVKK Institute of Technology, Affiliated to JNTUA, AP, India

Abstract-The problem is motivated by the proliferation of large-scale distributed file systems supporting parallel access to more than one storage devices i.e. to the multiple devices. Our work mainly focuses on the current Internet standard for such file systems, i.e. the parallel Network File System. This makes use of Kerberos for establishing the parallel session keys between the clients and the storage devices. Our analysis about the existing Kerberos-based protocol displays that it has a number of limitations. In this paper, we are trying to propose a variety of authenticated key exchange protocols which are designed to address the issues which were faced by the existing system. We show that our protocols are efficient to handle the reducing up to approximately of the workload of the metadata server and concurrently supporting forward secrecy and escrow-freeness. Only a small fraction of increased computation overhead at the client is what all required.

Keywords – **Parallel Network File system (pNFS), Kerberos, forward secrecy, Escrow-free.**

I. INTRODUCTION

In parallel file system, the file data is spread across the multiple storage devices or nodes to allow the concurrent access by many different tasks of a parallel application. [7]This is frequently 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. [1]In this work, we examine the problems of secure many-to-many communications in large-scale network file systems which support the parallel access to multiple storage devices.

That is, we examine a communication model where there are a huge 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. Especially, we focus on how to exchange key materials and build parallel secure sessions between the clients and the storage devices in the parallel Network File System (pNFS).[6] The development of pNFS is driven by Netapp, Panasas, Sun, IBM, EMC and UMich/CITI, and thus it shares many common features and is compatible with many existing commercial/proprietary network file systems.

The primary goal here is to design an efficient and secure authenticated key exchange protocol that meets the specific requirements of pNFS. The main aim is to achieve the properties like scalability, forward secrecy, Escrow-free. The main aim of this paper is to propose a variety of authenticated key exchange protocol which is very efficient to handle the reducing up to approximately of the workload of the metadata server and concurrently supporting forward secrecy and escrow-freeness. All this requires only a small fraction of increased computation overhead at the client. We define an appropriate security model and prove that our protocols are secure in the model.

II. LITERATURE REVIEW

1. FARSITE: Federated, Available, and Reliable Storage for an Incompletely Trusted Environment AUTHORS: A. Adya, W.J. Bolosky, M. Castro

Description: This paper Farsite provides the file availability and reliability with the help of randomized replicated storage. They ensure the secrecy of the file contents with the help of cryptographic techniques. They also maintain the integrity of the file and the directory data with a Byzantine-fault-tolerant protocol. They designed a system which is scalable by using a distributed hint mechanism and delegation certificates for path name translations and also achieve good performance by locally caching file data, lazily propagating file updates, and varying the duration and the granularity of the content leases. Farsite is designed to support the files and also the I/O workload of desktop computers in a large company or university. It provides availability and reliability through replication; privacy and authentication through cryptography; integrity through Byzantine-fault-tolerance techniques; consistency through leases of variable granularity and duration; scalability through namespace delegation; and reasonable performance through client caching, hint based pathname translation, and lazy update commit.

2. Block level security for network-attached disks AUTHORS: Marcos K. Aguilera, Minwen Ji, 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.

3. 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 ideal-cipher model, of the two-flow protocol at the core of EKE.

4. 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.

5. 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 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.

III. PROPOSED SYSTEM

Our work mainly focuses on the current Internet standard for such file systems, i.e., parallel Network File System. This makes use of Kerberos for establishing the parallel session keys between the clients and the storage devices. Our review of the existing Kerberos-based protocol shows that it has a number of limitations. In this paper, we propose a variety of authenticated key exchange protocols that are designed to address the above issues. We show that our protocols are efficient to handle the reducing up to approximately of the workload of the metadata server and concurrently supporting forward secrecy and escrow-freeness. All this requires only a small fraction of increased computation overhead at the client.

IV. MODULES

System Model

In the first module, we implement the system model. The metadata server is trusted to act as a reference monitor, issue valid layouts containing access permissions, and sometimes even generate session keys (for example, in the case of Kerberos-based pNFS) for secure communication between the client and the storage devices. The storage devices are trusted to store data and only perform I/O operations upon authorized requests. However, we assume that the storage devices are at a much higher risk of being compromised compared to the metadata server, which is typically easier to monitor and protect in a centralized location. Furthermore, we assume that the storage devices may occasionally encounter hardware or Kerberos-based pNFS Protocol Security model with forward secrecy software failure, causing the data stored on them no longer accessible. We note that this work focuses on communication security. Hence, we assume that data transmitted between the client and the metadata server, or between the client and the storage device can be easily eavesdropped, modified or deleted by an adversary. However, we do not address storage related security issues in this work. Security protection mechanisms for data at rest are orthogonal to our protocols.

Kerberos-based pNFS Protocol

The pNFS protocol that transfers file metadata, also known as a layout,1 between the metadata server and a client node. For the sake of completeness, we describe the key establishment protocol recommended for pNFS in RFC 5661 between a client C and n storage devices Si, through a metadata server M. Since the session keys are generated by M and transported to Si through C, no interaction is required between C and Si (in terms of key exchange) in order to agree on a session key. This keeps the communication overhead between the client and each storage device to a minimum in comparison with the case where key exchange is required. Moreover, the computational overhead for the client and each storage device is very low since the protocol is mainly based on symmetric key encryption. The message serves as key confirmation, that is to convince C that Si is in possession of the same session key that C uses.

Security model with forward secrecy

In this module, we implement security model with forward secrecy. We first introduce some notation required for our protocols. Let F(k;m) denote a secure key derivation function that takes as input a secret key k and some auxiliary information m, and outputs another key. Let sid denote a session identifier which can be used to uniquely name the ensuing session. Let also N be the total number of storage devices to which a client is allowed to access. We are now ready to describe the construction of our protocols. We now employ a Diffie-Hellman key agreement technique to both provide forward secrecy and prevent key escrow. In this protocol, each Si is required to pre-distribute some key material to M at Phase I of the protocol.

Security Analysis

We work in a security model that allows us to show that an adversary attacking our protocols will not able to learn any information about a session key. Our model also implies implicit authentication, that is, only the right protocol participant is able to learn or derive a session key. The above security model for pNFS-AKE does not consider forward secrecy (i.e., the corruption of a party will not endanger his/her previous communication sessions). Below we first define a weak form of forward secrecy we call partial forward secrecy (PFS).

CONCLUSION

We proposed 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] 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), pages 59–72. USENIX Association, Dec 2005.

[2] C. Adams. The simple public-key GSS-API mechanism (SPKM). The Internet Engineering Task Force (IETF), RFC 2025, Oct 1996.

[3] 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.

[4] 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.

[5] 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.

[6] Amazon simple storage service (Amazon S3). http://aws.amazon.com/s3/.
[7] M. Bellare, D. Pointcheval, and P. Rogaway. Authenticated key exchange secure against dictionary attacks. In Advances in Cryptology – Proceedings of EUROCRYPT, pages 139–155. Springer LNCS 1807, May 2000.

[8] M. Eisler. LIPKEY - A Low Infrastructure Public Key mechanism using SPKM. *The Internet Engineering Task Force (IETF)*, RFC 2847, Jun 2000.

[9] M. Eisler. XDR: External data representation standard. *The Internet Engineering Task Force (IETF)*, STD 67, RFC 4506, May 2006.

[10] M. Eisler. RPCSEC GSS version 2. *The Internet Engineering Task Force (IETF)*, RFC 5403, Feb 2009.

[11] M. Eisler, A. Chiu, and L. Ling. RPCSEC GSS protocol specification. *The Internet Engineering Task Force (IETF)*, RFC 2203, Sep 1997.

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

[13] M. Factor, D. Nagle, D. Naor, E. Riedel, and J. Satran. The OSD security protocol. In *Proceedings of the 3rd IEEE International Security in Storage Workshop (SISW)*, pages 29–39. IEEE Computer Society, Dec 2005.