Authenticated Key Exchange Protocols for Parallel Network File Systems

Mr. R.Akash
Mr. K.Raju

ABSTRACT:

We 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. Our 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. Our review of the existing Kerberos-based protocol shows that it has a 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. 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 capable of reducing up to approximately 54% 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.

Keywords-Parallel sessions, authenticated key exchange, network file systems, forward secrecy, key escrow.

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) [48], Google File System (GoogleFS) [21], Lustre [35], Parallel Virtual File System (PVFS) [43], and Panasas File System [53]; while there also exist research projects on distributed object storage systems such as Usra Minor [1], Ceph [52], XtreemFS [25], and Gfarm [50]. These are usually required for advanced scientific or data-intensive applications such s, seismic data processing, digital animation studios, computational fluid dynamics, and semiconductor manufacturing. we attempt to meet the following desirable properties, which either have not been satisfactorily achieved or are not achievable by the current Kerberos-based solution

• *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;

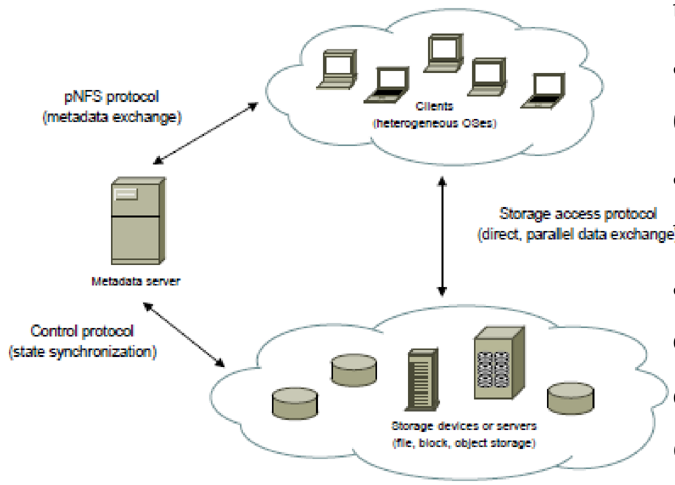
• *Forward secrecy* – the protocol should guarantee the security of past session keys when the long-term secret key of a client or a storage device is compromised [39]; and

• *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.

Security Consideration

Earlier versions of NFS focused on simplicity and efficiency, and were designed to work well on intranets and local networks. Subsequently, the later versions aim to improve access and performance within the Internet environment. However, security has then become a greater concern. Among many other security issues, user and server authentication within an open, distributed, and cross-domain environment are a complicated matter. Key management can be tedious and expensive, but an important aspect in ensuring security of the system. Moreover, data privacy may be critical in high performance and parallel applications, for example, those associated with biomedical information sharing [28], [44], financial data processing & analysis [20], [34], and drug simulation & discovery [42].

p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 13
September 2016



Single Sign-on. In NFS/pNFS that employs kerberos, each storage device shares a (long-term) symmetric key with the metadata server (which acts as the KDC). Kerberos then allows the client to perform single sign-on, such that the client is authenticated once to the KDC for a fixed period of time but may be allowed access to multiple storage devices governed by the KDC within that period. This can be summarized in three rounds of communication between the client, the metadata server, and the storage devices as follows: 1) the client and the metadata server perform mutual authentication through LIPKEY (as described before), and the server issues a ticket-granting ticket (TGT) to the client upon successful authentication; 2) the client forwards the TGT to a ticket-granting server (TGS), typically the same entity as the KDC, in order to obtain one or more service tickets (each containing a session key for access to a storage device), and valid layouts (each presenting valid access permissions to a storage device according to the ACLs); 3) the client finally presents the service tickets and layouts to the corresponding storage devices to get access to the stored data objects or files.

CONCLUSIONS

We proposed three authenticated key exchange protocols for parallel network file system (pNFS). Our protocols offer three appealing advantages over the existing Kerberos-based pNFS protocol. First, the metadata server executing our protocols has much lower workload than that of the Kerberos-based approach. Second, two our protocols provide forward secrecy: one is partially forward secure (with respect to multiple sessions within a time period), while the other is fully forward secure (with respect to a session). Third, we have designed a protocol which not only provides forward secrecy, but is also 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] 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.

p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 13
September 2016

[9] B. Callaghan, B. Pawlowski, and P. Staubach. NFS version 3 protocol specification. *The Internet Engineering Task Force (IETF)*, RFC 1813, Jun 1995.

[10] 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 2001.

[11] CloudStore. http://gcloud.civilservice.gov.uk/cloudstore/.

[12] Crypto++ 5.6.0 Benchmarks. http://www.cryptopp.com/benchmarks. html.

[13] 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.

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

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

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

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

[18] 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.

[19] 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.

[20] Financial Services Grid Initiative. http://www.fsgrid.com/.

[21] S. Ghemawat, H. Gobioff, and S. Leung. The Google file system. In *Proceedings of the 19th ACM Symposium on Operating Systems*

Principles (SOSP), pages 29–43. ACM Press, Oct 2003.
[22] G.A. Gibson, D.F. Nagle, K. Amiri, J. Butler, F.W. Chang, H. Gobioff,
C. Hardin, E. Riedel, D. Rochberg, and J. Zelenka. A costeffective,
high-bandwidth storage architecture. *ACM SIGPLAN Notices*,
33(11):92–103. ACM Press, Nov 1998.
[23] Hadoop Wiki. http://wiki.apache.org/hadoop/PoweredBy.
[24] J.H. Howard, M.L. Kazar, S.G. Menees, D.A. Nichols, M. Satyanarayanan, R.N. Sidebotham, and M.J. West. Scale and performance
in a distributed file system. *ACM Transactions on Computer Systems*
(TOCS), 6(1):51–81. ACM Press, Feb 1988.
[25] F. Hupfeld, T. Cortes, B. Kolbeck, J. Stender, E. Focht, M. Hess, J. Malo,
J. Marti, and E. Cesario. The XtreemFS architecture – a case for objectbased
file systems in grids. *Concurrency and Computation: Practice and*
Experience (CCPE), 20(17):2049–2060. Wiley, Dec 2008

Author's Profile

Mr. K.Raju received M.Tech degree from Jayamukhi Institute of Technological Science,Narsampet, Warangal affiliated to JNTUH, Hydearabad. He is currently working as Assistant professor, Department of CSE, in Vinuthna Institute of Technology & Science ,Hasanparthy, Warangal, Telangana, India. Hisinterest includes Data Base Management Systems.

Mr. R.Akash received B.Tech Degree fromVinuthna Institute of Technology & Science Hasanparthy, Warangal affiliated to KU, Warangal. He is currently pursuing M.Tech Degree in Software Engineering specialization in Vinuthna Institute of Technology & Science Hasanparthy, Warangal, Telangana, India.