International Journal of Research

Available at https://edupediapublications.org/journals

p-ISSN: 2348-6848
e-ISSN: 2348-795X

Volume 04 Issue 01
January 2017

Novel Efficient and secure protocol design to exchange information in network

M.SRIDHAR[1], Dr. S.MURUGAN[2]

Research Scholar, CSE, Sathyabama University, Chennai, Tamilnadu

Prof, CSE, Sathyabama University, Chennai, Tamilnadu

ABSTRACT

We study the issues of key creation for secure many-to-many communications. issue is impressed by the proliferation of large-scale distributed file systems supporting parallel access to multiple storage devices. Our work focuses on the current internet normal for such file systems, i.e., parallel Network file system (pNFS), that makes use of Kerberos to establish parallel session keys between senders and storage devices. Our review of the prevailing Kerberos-based protocol shows that it has variety of limitations: (i) a information server facilitating key exchange between the senders and also the storage devices has heavy work that restricts the quantifiability of the protocol; (ii) the protocol doesn't offer forward secrecy; (iii) the information server generates itself all the session keys that are used between the data owners and storage devices, and this inherently results in key escrow. During this paper, we propose a range of authenticated key exchange protocols that are designed to handle the higher than issues. we show that our protocols are capable of reducing up to approximately fifty four of the work of the information server and concurrently supporting forward secrecy and escrow-freeness. All this needs only atiny low fraction of enhanced computation overhead at the data owner.

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

1. **INTRODUCTION:**

In a parallel file system, file information is distributed across multiple storage devices or nodes to permit concurrent access by multiple tasks of a parallel application. this can be usually utilized in giant-scale cluster computing that focuses on high performance and reliable access to large datasets. That is, higher I/O information measure is achieved through concurrent access to multiple storage devices among giant compute clusters; whereas information loss is protected through information mirroring mistreatment fault-tolerant marking algorithms. Some samples of high-performance parallel file systems that are in production use are the IBM General Parallel file system (GPFS) , Google file system (GoogleFS) , Lustre , Parallel Virtual file system (PVFS) , and Panasas file system ; whereas there also exist analysis comes on distributed object storage systems like Usra Minor , Ceph , and Gfarm . These are sometimes needed for advanced scientific or data-intensive applications like, seismic processing, digital animation studios, procedure fluid dynamics, and semiconductor producing. In these environments, hundreds or thousands of file system data owners share information and generate very high aggregate I/O load on the file system supporting petabyte- or terabyte-scale storage capacities vices. in this work, we investigate issue of secure severalto-many communications in large-scale network file systems that support parallel access to multiple storage devices. That is, we consider a communication model wherever there area unit a large variety of data owners (potentially tons of or thousands) accessing multiple remote and distributed storage devices (which also could scale up to tons of or thousands) in parallel. significantly, we target the way to exchange key materials and establish parallel secure sessions between the data owners and the storage devices within the parallel Network file system (pNFS) —the current web standard—in an economical and scalable manner. the development of pNFS is driven by Panasas and so it shares several common options and is compatible with several existing commercial/proprietary network file systems.

2. METHODOLOGY

The current style of NFS/pNFS focuses on ability,instead of efficiency and scalability, of assorted mechanisms to produce basic security. Moreover, key creation between a data owner and multiple storage devices in pNFS are supported those for NFS, that is, they're not designed specifically for parallel communications. Hence, the information server isn't only responsible for process access requests to storage devices (by granting valid layouts to authenticated and licensed clients), but also needed to come up with all the corresponding session keys that the data owner must communicate securely with the storage devices to that it's been granted access. Consequently, the information server may become a performance bottleneck for the classification system. Moreover, such protocol style ends up in key escrow. Hence, in theory, the server will learn all info transmitted between a consumer and a device. This, in turn, makes the server an attractive target for attackers. Another problem of the present approach is that past session keys is exposed if a storage device's long-run key shared with the information server is compromised. we believe that this can be a realistic threat since a large-scale file system could 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.

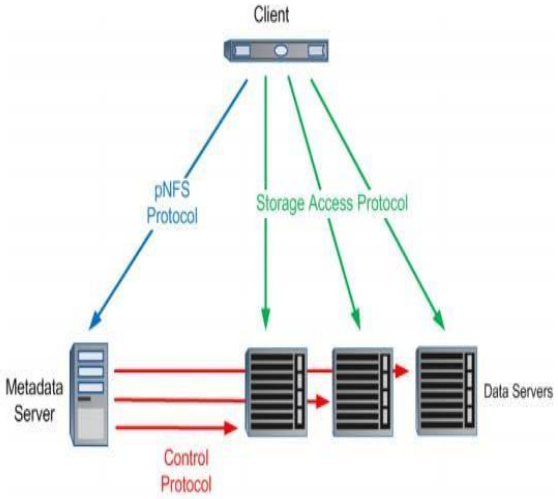

Fig 1:System architecture

We implement 3 authenticated key exchange protocols for parallel network classification system (pNFS). Our protocols supply 3 appealing advantages over present Kerberos-

based pNFS protocol. First, the information server execution our protocols has a lot of lower workload than that of the Kerberos-based approach. Second, 2 our protocols give forward secrecy: one is part forward secure (with relation to multiple sessions among a time period), whereas the other is totally forward secure (with respect to a session). Third, we have designed a protocol that not only provides forward secrecy, however is additionally escrow-free.

3.Related Works:

The multi-core revolution has brought the difficulty of parallel programming to the forefront of computer code development generally. several parallel programming models are developed within the past, however, there's no solidly established technology to choose up because the obvious alternative for exploiting correspondence on today's thought multi-cores and on networks of multicores, serving as affordable superior platforms. Even a lot of seriously, several of the established technologies area unit tied to 1 specific category of parallel machines, and are ill-suited to cope with radical fine arts changes. To additional boost procedure capabilities, usually graphics process units (GPUs) or co-processors are added to the current configuration, leading to a extremely heterogeneous computing platform. Programming such a platform is difficult for parallel computing specialists, in addition to domain specialists new to parallel programming. The programming models for such a platform thus need to strike a balance between simplifying the task of parallel programming whereas still delivering high performance on heterogeneous hardware, that is that the motivation for using these platforms within the initial place. in this survey we aim to convey guidance to domain experts in search of a high-level programming model that may exploit such heterogeneous hardware with token computer user effort. we explore the realm of parallel programming models, classify the established technologies on language as well as on system level, and assess their suitability for property parallel programming in the long-term, specially envisioning further heterogeneousness within the underlying platform, as mentioned in

Section two. we establish many major challenges of parallel programming in Section 3: computer user productivity, performance portability, scalability, and adaptively, which require to be self-addressed, since exploiting parallelism is that the main supply of increased performance. we then explore many implementation technologies, managing these challenges. Specially, we classify the prevailing parallel programming models in Table one of Section four supported their various level of abstraction over parallel coordination. In Section five, we present a variety of low- and high-level models for programming heterogeneous architectures and discuss these trends in GPU programming. Furthermore, we summary many representative policy control mechanisms in Table two of Section vi, and discuss the foremost system-level technologies that have well-tried successful across a variety of parallel architectures. we summaries our findings from this in-depth study of models and systems in Section eight. 2. Trends in Parallel Architectures Hardware design substantially affects application performance [1]. Therefore, it's necessary to require the characteristics of the target design into account, particularly as novel parallel architectures are progressively heterogeneous and stratified.

4. Conclusion:

We implement 3 authenticated key exchange protocols for parallel network classification system (pNFS). Our protocols supply 3 appealing advantages over present Kerberos-based pNFS protocol. First, the information server execution our protocols has a lot of lower workload than that of the Kerberos-based approach. Second, 2 our protocols give forward secrecy: one is part forward secure (with relation to multiple sessions among a time period), whereas the other is totally forward secure (with respect to a session). Third, we have designed a protocol that not only provides forward secrecy, however is additionally escrow-free.

5. Bibliography

[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, Dec2002.

[4] M.K. Aguilera, M. Ji, M. Lillibridge, J. MacCormick, E. Oertli,D.G. Andersen, M. Burrows, T. Mann, and C.A. Thekkath. Blocklevelsecurity 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 viewof cloud computing. *Communications of the ACM*, 53(4):50–58. ACMPress, 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.

[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.

4506,

[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 processin on large clusters. In *Proceedings of the 6th Symposium on OperatingSystem Design and Implementation (OSDI)*, pages 137–150. USENIX Association, Dec 2004.

May 2006.

[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