IJMTST | Volume: 2 | Issue: 09 | September 2016 | ISSN: 2455-3778

Exchange Protocols on Network File Systems Using Parallel Sessions Authenticated & Improved Keys

V. Prathyusha[1] | Aaruni Giriraj[2] | Dr. Ramakrishna[3]

[1]PG Scholar, Department of IT, Sridevi Women's Engineering College, Hyderabad.
[2]Assistant Professor, Department of IT, Sridevi Women's Engineering College, Hyderabad.
[3]Professor and HOD, Department of IT, Sridevi Women's Engineering College, Hyderabad.

ABSTRACT

In this work we studied the key establishment for secure many-to-many communications. The main problem is inspired by the rapid increase of large-scale distributed file systems supporting parallel access to multiple storage devices. The system focus on the current Internet standard for such file systems, i.e., parallel Network File System (pNFS), which makes use of Kerberos key exchange protocols to implement parallel session keys between clients and storage servers. Our study of the existing Kerberos protocol shows that it has a number of limitations: (i) a metadata server providing key exchange among the clients and the storage devices has heavy workload that limits the scalability of the protocol; (ii) the protocol cannot provide forward secrecy; (iii) the metadata server generates all the session keys for securing communication between clients and storage devices, and this inadvertently leads to key escrow. In this paper, we put forward three different authenticated key exchange protocols that are designed to address the above issues. We prove that our protocols are capable for minimizing up to almost50% of the workload of the metadata server and at the same time supporting forward secrecy and escrow-prevention. All this requires only a small fraction of increased computation overhead at the client.

KEYWORDS: *Parallel network sessions; fully encrypted authenticated key exchange; network file systems; forward secrecy; key escrow; protocols.*

I. INTRODUCTION

In parallel file system, the file data is situated throughout multiple storage nodes to allow the simultaneous access by many different functions of a parallel application [7]. This is frequently used in large-scale cluster computer networks that concentrate on increased performance and easy and error free access to large data stores. That is, higher I/O bandwidth is achieved through simultaneous access to multiple storage units inside o f the large compute clusters; while data loss is prevented by data duplication using fault-tolerant striping algorithms[1].

In this work, we analyzed the issues of secure many-to-many communications systems in large-scale network file systems which facilitate the parallel access to multiple storage servers. We examine a communication system where there are a large number of clients (may be hundreds or thousands) accessing multiple remote and distributed storage servers (which also may be counts to hundreds or thousands) in parallel.

In this paper, we looks on how to exchange key parameters and build parallel secure sessions between the clients and the storage servers 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 they posses many common features and is compatible with many of the present commercial proprietary network file entities. The prime and important goal here is to design an efficient and secure fully encrypted authenticated key exchange protocol that meets the unique requirements of pNFS. The main aim is to achieve the desirable properties such as scalability, forward secrecy, Escrow-prevention. This paper propose three different fully encrypted authenticated key exchange protocols which is highly efficient to

handle the reducing up to 90% of the workload of the metadata server and simultaneously providing forward secrecy and escrow-prevention[3]. This is achieved by some increased computation overhead by the client. We define an appropriate security model and prove that our protocols are efficient and enable for this purpose in the model.

pNFS separates the file system protocol management into two parts: metadata management and data management. Metadata is the details about a file system object, such as its name, location within the namespace, byte storage area, owner permissions and other attributes.

pNFS comprises a collection of three protocols: (i) the *pNFS protocol* that sends file metadata *layout*, between the metadata server and a client (ii) the *storage access protocol* that describes how a client accesses data from the respective storage servers according to the corresponding metadata; and (iii) the *control protocol* that synchronizes between the metadata server and the storage servers.

II. Related Work

Tele care Medical Information Systems (TMIS) give a successful approach to enhance there storative system between specialist doctors, attendants and patients. By improving the security and protection of TMIS, it is vital while testing to enhance the TMIS so that a patient and a specialist doctor can perform synchronized verification and session key foundation utilizing a 3-party curative server while the safe information of the patient can be guaranteed. In proposed procedure amysterious three-party secret word authorised key swapping (3PAKE) convention for TMIS is utilized. The convention depends on the competent elliptic bend cryptosystem. For security, weapply the pi math based formal confirmation device ProVerif to show that our 3PAKEconvention for TMIS can give namelessness to patient and specialist doctor and also accomplish synchronized verification and session key secrecy.

Authenticated key exchange secure against dictionary attacksby M. Bellare,D. Pointcheval, and P. Rogaway [7]. Password-based protocols for authenticated key exchange (AKE) are formulated to work in addition to the use of passwords drawn from a space so small that an adversary might well guess, off line, all possible passwords. While several such protocols have been suggested, the underlying theory has been lagging. The author start by designing a model for this problem, one effective enough to deal with password enumeration, forward secrecy, server compromise, and loss of session keys. The one model can be used to define various goals. The author uses AKE (with “implicit” authentication) as the “basic” goal, and they give definitions for it and for entity-authentication tasks as well. Then they prove correctness for the idea at the centre of the Encrypted Key-Exchange (EKE) protocol by Bellovin and Merritt: they prove security, in an idealcipher model, of the two-flow protocol at the core of EKE.

Analysis of key-exchange protocols and their use for building secure channels by Ran Canetti and Hugo Krawczyk [11].

In this paper authors put forward a formalism for the analysis of key-exchange protocols that substantiates previous definitional approaches and results in a definition of security that enables some vital analytical advantages: (a) any of the key inter change protocol that fulfils the security definition, can be composed using symmetric encryption methods and authentication procedures to provide proven secure communication channels (as defined here); and (b) the definition permits for simple modular proofs of security: one can design and prove security of key-exchange protocols in an ideal model where the communication links are clearly authenticated, and then translate them using general tools to achieve security in the realistic setting of adversary-controlled links. This paper adopts a procedural steps for the analysis of key-exchange protocols. They follow the approach of the adversarial modelling.

Authenticated Key Exchange Protocols for parallel Network File Systems published by Hoon Wei Lim Guomin Yang [10].

In this paper they discussed the problem of key authentication for secure many-to-many communications. The problem is inspired by the enormous growth of large sized distributed file systems supporting parallel access to multiple storage devices. Their work targeted on the current Internet standard for such file systems, i.e., parallel Network File System (pNFS), which makes use of Kerberos protocols to implement parallel session keys between clients and storage devices. They overcome the following limitations of pNFS (i) a metadata server providing key exchange between the clients and the storage units has heavy workload that limits the scalability of the protocol; (ii) the protocol does not provide forward secrecy; (iii) the metadata server produces itself all the session keys that are used between the clients and storage systems, and this intrinsically leads to key

escrow. The noted disadvantage of this system is that some of the key materials and layouts are sending without encryption that may lead to information leakage.

III. Existing System

Independent of the advancement of cluster and high performance computing, the emergence of clouds and the Map Reduce programming model has output a file systems such as the Hadoop Distributed File System (HDFS), Amazon S3 File System, and Cloud-Store. This, in turn, has increased the extensive use of distributed and parallel computation on large datasets in many organizations

Some of the past work in securing large distributed file systems, for example, have already employed Kerberos for carrying out authentication and enforcing access control. Kerberos, being based on mostly symmetric key techniques in its early implementation, was generally believed to be more suitable for rather closed,well-connected distributed environments.

On the other side, data grids and file systems such as, OceanStore, LegionFS and FARSITE, make use of public key cryptographic methods and public key infrastructure (PKI) to perform cross-domain user authentication.

3.1 Disadvantages

The current design of NFS/pNFS concentrates on *interoperability*, instead of efficiency and scalability, of various mechanisms to provide basic security. However, key establishment between a client and multiple storage devices in pNFS are based on those for NFS, that is, they are not designed alone for parallel communications. Hence, the metadata server is doing the following functions (i) processing access requests to storage devices (by granting valid layouts to authenticated and authorized clients), (ii) Generate all the corresponding session keys that the client needs to communicate securely with the storage systems to which it has been granted access.

Consequently, the metadata server may become a performance barrier for the file system. Moreover, such protocol design leads to key escrow. Hence, in fact, the server can learn all information transacted among a client and a storage device. This, in turn, makes the server an attractive aim for attackers.

Another disadvantage of the current approach is that past session keys can be revealed if a storage unit's long- period key shared with the metadata server is compromised. We firmly believe that this is a real threat since a large sized file system may have thousands of geographically distributed storage devices. It may not be practicable to protect withhigh physical security and network protection for all the storage devices. In short the disadvantages are

- Devices have heavy workload that limits the scalability of the protocol.
- The protocol not able to provide forward secrecy.
- The metadata server produces itself all the session keys that are used between the clients and storage devices, and this intrinsically leads to key escrow.

3.2 Existing Kerberos pNFS

(1) $C \rightarrow M : ID_C$

(2) $M \rightarrow C: E(K_C;K_{CT}),E(K_T; ID_C,K_{CT},t)$ (3)

$CT:ID_{S1},..,ID_{Sn},E(K_T; ID_C,t,K_{CT}), E(K_{CT};t, ID_C)$

(4) $T \rightarrow C: \sigma_1,..,\sigma_n, E(K_{MS1} ; sk_1,t, ID_C),.., E(K_{MSn}; ID_C,t, sk_n), E(K_{CT}; sk_1,....,sk_n)$

(5) $C \rightarrow Si :\sigma_i, E(K_{MSi} ;sk_i,t, ID_C,) \quad E(sk_i;t, ID_C)$

(6) $SiC \rightarrow : E(sk_i; t + 1)$

IV. Proposed System

Our primary goal in the proposed system is to design efficient and secure authenticated key exchange protocols that meet specific Security and scalability requirements of pNFS.

The main results of this paper are three new proven secure fully encrypted authenticated key exchange protocols. Our protocol rules, progressively designed to achieve each ofthe above properties, demonstrate the balance between efficiency and security.

We show that our protocols can reduce the workload of the metadata server by approximately 50% compared to the present Kerberos - based protocol, while achieving the desired security qualitiessuch as forward secrecy , key escrow prevention, key data security (by full symmetric key encryption of the key materials and layouts which contain client identity, file mapping information of the server to directly access the byte range location at the storage server) and keeping the computational overhead at the clients and the storage devices at a reasonably low level.

All the information exchange in connection with the establishment of secure and efficient communication between client, file servers and metadata server has been encrypted with

symmetric key encryption. Thus the process is almost safe from adversary attacks.

4.1 Proposed protocols

4.1.1 pNFS- FEAKE I

This is a modified version of Kerberos. In this the client generates its session keys. Symmetric key encryption is used to protect information. There is no forward secrecy and key escrow problem also exists.

Phase I – For each validity period υ:
(1) CM : $E(K_{CM}; ID_C, K_{CS1}, .., K_{CSN})$
(2) M C :$E(K_{MS1}; ID_C, ID_{S1}, \upsilon, K_{CS1}), .., E(K_{MSN}; ID_C, ID_{SN}, \upsilon, K_{CSN})$

Phase II –For every access request during time t:
(1) C M : $E(K_{CM}; ID_C, ID_{S1}, ..., ID_{Sn})$
(2) M C : $E(K_{CM}; \sigma_1, ..., \sigma_n)$
(3) C Si :$E(K_{MSi}; \sigma_i, ID_C, ID_{Si}, \upsilon, K_{CSi}), E(sk_i^0; ID_C, t)$
(4) SiC : $E(sk_i^0; t + 1)$

4.1.2 pNFS- FEAKE II

By applying Diffie Hellman key agreement rules in pNFS-FEAKE I , we can solve key escrow and implement partial forward secrecy. Session key is produced from client and server Diffie Hellman components. After the time period 'υ'all the key components are deleted. In this system the compromise of a long term key can expose the session key used in the time period 'v', but the past session keys in the elapsed time period cannot be accessed by an adversary.

Phase I – For each validity period υ:
(1) SiM : $E(K_{MSi}; ID_{Si}, g^{si})$
(2) C M : $E(K_{CM}; ID_C, g^c)$
(3) M C : $E(K_{CM}; g^{s1}, ..., g^{sN})$,
$T(K_{MS1}; ID_C, ID_{S1}; \upsilon, g^c; g^{s1}), ..., T(KMS_N; ID_C, IDS_N, \upsilon, g^c, g^{sN})$

Phase II –For every access request during time t:
(1) CM : $E(K_{CM}; ID_C, ID_{S1}, ..., ID_{Sn})$
(2) M C :$E(K_{CM}; \sigma_1, ..., \sigma_n)$
(3) C Si :$T(KMSi; \sigma_i, g^c, ID_C, ID_{Si}, \upsilon, g^c, g^{si}), E(sk_i^0; ID_C, t)$
(4) SiC : $E(sk_i^0; t + 1)$

4.1.3 pNFS- FEAKE III

This is the most advanced protocol in which we have achieved all the desirable qualities of a pNFS.

Phase I – For each validity period υ:
(1) SiM :$E(K_{MSi}; ID_{Si}, g^{si})$
(2) C M :$E(K_{CM}; ID_C, g^c)$
(3) M C :$E(K_{CM}; g^{s1}, ..., g^{sN})$
(4) M Si:$E(K_{MSi}; ID_{Si}, ID_C, \upsilon, g^c, g^{si})$

Phase II –For every access request during time t:
(1) CM :$E(K_{CM}; ID_C, ID_{S1}, ..., ID_{Sn})$
(2) M C :$E(K_{CM}; \sigma_1, ..., \sigma_n)$
(3) C Si :$E(sk_i^{j,0}; \sigma_i, ID_C; t)$
(4) Si C :$E(sk_i^{j,0}; t + 1)$

4.2Advantages

The proposed system pNFS – FEAKE III achieves the following four desirable properties.

- Scalability – the metadata server supporting 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 barrier, but is capable of supporting a very large number of clients.
- Forward secrecy – the protocol should ensures the security of past session keys when the long-period secret key of a client or a storage device is compromised.
- Key Escrow prevention – the metadata server should not learn any information about any of the session keysgenerated and used by the client and the storage servers, provided there is no collusion among them.
- All the communications between Metadata server, Clients, Parallel file servers are fully encrypted to preserve the information during transit.

V. Implementation

Client – C1,C2,C3.. Metadata Server –M, Storage servers – S1, S2, S3.... Sn.

5.1 Establishment of secure channels between Client and Metadata server

- Obtain Metadata server's certificate
- Verified that it is signed by trusted CA.
- Generate random Session Symmetric key.
- Encrypt the session key with metadata servers public key.
- Send Encrypted key to the metadata server.

5.2 Establishment of secure communication between clients and parallel servers through the help of Metadata server.

Phase1 for each validity period υ

a) Each server distribute some key materials to Metadata server .Each Si generate Diffie Hellman key component g^{si} . This is forwarded to and stored by Metadata server.

$Si \longrightarrow M : \mathcal{E}(KMS_i ; ID_{Si} , g^{si})$

b) Similarly Client generates its Diffie Hellman Key componentg^c and send to Metadata server.

$C \longrightarrow M : \mathcal{E}(KCM ; ID_C , g^c)$

c) M sends all key components to C for N storage servers that it may access within a period v

$M \longrightarrow C : \mathcal{E}(KCM ; g^{S1}.....,g^{SN})$

d) M also sends Client's Diffei Hellman Components to g^c to each Si.

$M \longrightarrow Si : \mathcal{E}(KMS_i ; ID_C, ID_{Si}, v, g^c, g^{si})$

After this stage C and Si are able to agree a Diffei Hellman value g^{csi}

e) C and Si set F1(g^{csi}, ID_C, ID_{Si}, v) as their initial shared secret state K^0CS_i

Phase2 for each access request at time t

a) C submits an access request M which contains all identities of storage devices Si

$C \longrightarrow M : \mathcal{E}(KCM, IDc, IDS_1, ..., IDSn)$

b) M issues layout σi(Layout contains Client's identity, File object mapping information and Access permissions)

$M \longrightarrow C : \mathcal{E}(KCM, \sigma_1,....., \sigma_n)$

c) Cestablish secure session with Si by computing session key $Sk_i^{j,z}$.

$Sk_i^{j,z} = F2(KCS_i^{j-1}; IDc, IDsi, j, sid, z)$

$z = 0,1$

C sends encrypted layout and identity and time to Si

$C \longrightarrow S_i : \mathcal{E}(Sk_i^{j,0}, \sigma i, t)$

d) Si decrypt encrypted message and check if the layout and IDc matches the identity of C and if t is within the current validity period v.

e) If all previous checks pass , Si replies C with a key confirmation message using key $Sk_i^{j,0}$

f) Both C and Si then set and update their internal shared secret state as K^jCS_i

VI. Conclusion

We put forward the three authenticated key exchange protocols for the parallel network file system (pNFS). The four main appealing advantages offered by the proposed system than the existing Kerberos-based protocol are(i) The metadata server which make use of our protocols has much lower workload as compared to that of the Kerberos-based approach.(ii) The 2nd and 3rd protocols provide the forward secrecy: one which is partially forward secure, while other is the fully forward secure. (iii)The third protocol which provides forward secrecy as well as is key escrow-prevention.(iv) As all the communications between the entities are fully encrypted, the chances of unauthorized information/key material access can be prevented.

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

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,.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 5thSymposium 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: 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 cipher texts and privatekeys. In Advances in Cryptology –

Proceedings ofCRYPTO, pages 258–275. Springer LNCS 3621, Aug2005.

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

[10] Hoon Wei Lim Guomin Yang: "Authenticated Key Exchange Protocols for Parallel Network File Systems", IEEE Transactions on Parallel and Distributed Systems-2015.

[11] 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.