Establishing a Session Key Between user and Storage Device in Parallel Network File System

Namitha K Y
M.Tech Scholar
Department of Computer Science & Engineering
Don Bosco Institute of Technology
Bangalore,India

Chandrakala
Assistant Professor
Department of Computer Science & Engineering
Don Bosco Institute of Technology
Bangalore,India

***Abstract*— Ensuring Security with respect to device to device communication has been extensively studied and many efficient techniques has been proposed in the recent days where providing security in many-to many communications is of major focus in the recent days. There is increase in use of large-scale distributed file systems supporting parallel access to more than one storage space devices i.e. to the multiple devices. Our job mainly focuses on the current Internet standard for such file systems, i.e. the parallel Network File scheme. This make use of Kerberos for establishing the parallel session keys amid the clients and the storage devices. An analysis is carried out about the existing Kerberos-based protocol displays that it has a figure of limits In this paper, Number of authenticated key exchange protocols are proposed and intended to speak to the issues which were face by the existing system. We show that our protocols are well-organized to handle the falling up to approximately of the workload of the metadata server plus concurrently supporting forward secrecy and escrow-freeness. simply a minute part of increased computation overhead at the client is what all required.**

Keywords*— Parallel Network File System, Kerberos, Escrow-free, Secure Key Exchange, Security

I. INTRODUCTION

In parallel file systems, the file data is distributed across multiple storage devices or nodes to allow concurrent access by numerous undertakings of a parallel application. That is normally utilized as a part of scale cluster computing that spotlights onhigh performance and reliable fetch huge datasets. That higher I/O transmission capacity is accomplished through simultaneous bringing information to various storage devices within large computing clusters, while data loss is protected through data reflecting utilizing imperfection tolerant striping algorithms. Couple of case of high performance parallel file system that are in the creation use are the IBM General Parallel Files System. which are normally required for highly developed technical or data intensive applications, for example, digital animation studios, computational fluid dynamics, and semiconductor manufacturing.

In these situations, hundreds or a large number of file system clients share information and produce very much high aggregate I/O load on the file system supporting petabytes or terabytes scale storage capacities. Autonomous of the development of the cluster and high performance computing, the emergence of clouds and the MapReduce programming model has brought about file system, for example, the Hadoop Distributed File System (HDFS). In this work, we research the issue of the safe numerous to numerous interchanges in the huge scale file system which bolster parallel bring to different putting away storage device. That we considering the communication model where there are an expansive number of the clients getting to numerous remote and conveyed storage devices in parallel. Especially, we tries to focus on the most proficient method to trade the key materials and foundation of the parallel secure sessions amongst client and capacity storage devices in the parallel Network File System (pNFS), the present Internet gauges in productive and adaptable way. The development of pNFS is driven by Sun, EMC, IBM, plus UMich/CITI, and thus it shares many comparable features and is perfect with numerous existing commercial network file systems. Our primary objective in this work is to plan proficient and secure confirmed key exchange protocols that address particular issues of pNFS.

All the more particularly, pNFS contains an accumulation of three protocols: (i) the pNFS protocol that exchanges file metadata, otherwise called a design, between the metadata server and a client hub; (ii) the capacity access convention that determines how the client gets to information from the related storage devices as per the comparing metadata; and (iii) the control protocol that synchronizes the state between the metadata server and the storage devices.

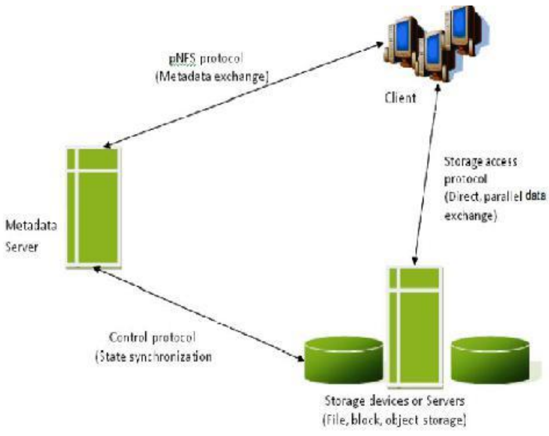

Figure 1: The Parallel File System Conceptual model

11.RELATED STUDY

Telecare Medical Information Systems (TMIS) give a compelling approach to enhance the therapeutic procedure between specialists, entourage and patients. By pretty the safety and shield of TMIS, it is critical while testing to enhance the TMIS so that a patient and a professional can perform coordinated verification and session key foundation utilizing a 3-party medicinal server while the safe information of the patient can be guaranteed. In proposed framework a mysterious three-party secret key verified key trade (3PAKE) convention for TMIS is utilized. The convention depends on the effective elliptic bow cryptosystem. For defense, we affect the pi math based formal verification device ProVerif to demonstrate that our 3PAKE convention for TMIS can give secrecy to patient and specialist and also accomplishes synchronized verification and session key security. The benefit of proposed plan is security and productivity that can be utilized as a part of TMIS. For this J-PAKE based convention are utilize The inconvenience of proposed plan is of it lessened session keys. - Qi Xie1*, estimated time of arrival [1]

Watchword based encoded key trade are conventions that are intended to give pair of clients conveying over an untrustworthy channel with a protected session key notwithstanding when the mystery key or secret key shared between two clients is haggard from a modest understanding of key. In proposed plan, two straightforward passwords based scrambled key trade conventions taking into account that of Bellovin and Merritt. While one convention is more appropriate to situations in which the watchword is shared over various servers, alternate gives better security. Both conventions are as productive, if worse, as any of the current scrambled key trade conventions in the writing, but then they just require a solitary irregular prophet occurrence. The confirmation of security for both conventions is in the arbitrary prophet display and in view of hardness of the computational Diffe-Hellman issue. In any case, a portion of the systems that we utilize are very not the same as the typical ones and make utilization of new variations of the Diffe-Hellman issue, which are of free intrigue. We additionally give solid relations between the new variations and the standard Diffe-Hellman issue. constructive position of this map it is conceivable to determine a few kinds of key. In this distinctive sorts of conventions are utilized like SIGMA, IKE and so forth - Michel Abdalla, estimated time of arrival [2]

Proposed plan Uses compositional strategy for demonstrating cryptographically solid security properties of key trade conventions, in light of a typical rationale that is translated over ordinary keeps running of a convention against a probabilistic polynomial time assailant. Since considerations around an unbounded figure of continues running of a meeting incorporates affectation like disputes about hisproperties saved by every run, we figure a detail of secure key trade that, dissimilar to customary key in recognize capacity, is shut under general organization with steps that utilization the key. We exhibit formal evidence rules in view of this diversion based condition, and demonstrate that the confirmation guidelines are sound over a computational semantics. - Anupam Datta1, estimated time of arrival [3]

In an open system, when various bunches associated with each other is expanded turns into a potential risk to security applications running on the groups. To lecture to this question, a memo Passing Interface (MPI) is created to save security administrations in an unsecured system. The future work concentrate on MPI in its place of different conventions in light of the fact that MPI is a stick out amongst the most well-known messages convention on disseminated bunches. Here AES calculation is utilized for encryption/decoding and insertion polynomial calculation is utilized for key administration which is then incorporated into Message Passing Interface Chameleon variant 2 (MPICH2) with standard MPI interface that gets to be ES-MPICH2. This ES-MPICH2 is another MPI that gives security and confirmation to conveyed bunches which is brought together into cryptographic and numerical idea. The real craving of ES-MPICH2 is supporting an expansive assortment of calculation and correspondence stages. The planned scaffold depends on both cryptographic and technical idea which prompts loaded with blunder free message passing interface with upgraded security. - R.S.RamPriya, estimated time of arrival [4]

Passwords are a show up amongst the most famous reason for framework crashes, in light of the fact that the low entropy of passwords makes frameworks defenseless against animal power speculating assaults. Because of new innovation passwords can be hacked effortlessly. Computerized Turing Tests keep on being a powerful, simple to-convey way to deal with distinguish robotized vindictive login endeavors with sensible expense of disservice to clients. Henceforth in this proposed plan the deficiency of existing and proposed login conventions intended to address extensive scale online lexicon assaults e.g. from a botnet of a huge figure of center In this map potential a essential plan that fortifies secret word based validation conventions and avoids online lexicon assaults and in addition numerous to-numerous assaults normal to 3-pass SPAKA conventions. - *A. Sai Kumar ,estimated time of arrival [5]

A motorized evidence of the secret word based convention One-Encryption Key Exchange (OEKE) is proposed utilizing the computationally-solid convention prover CryptoVerif. OEKE is a non-insignificant convention, and subsequently motorizing its verification gives extra certainty that it is right. This contextual analysis was additionally a chance to execute a few vital augmentations of CryptoVerif, helpful for demonstrating numerous different conventions. We have for sure stretched out CryptoVerif to bolster the computational Diffie-Hellman presumption. We have likewise included backing for confirmations that depend on Shoup's lemma and extra diversion changes. Specifically, it is currently conceivable to embed case qualifications physically and to union cases that no more should be recognized. In the long run, a few changes have been included the calculation of the likelihood limits for assaults, giving better decreases. Specifically, we improve over the standard calculation of probabilities when Shoup's lemma is utilized, which permits us to enhance the bond given in a past manual verification of OEKE, and to demonstrate that the foe can test at most one secret word for every session of the protocol. In this paper, we display these expansions, with their application to the evidence of OEKE. All progressions of the confirmation, both programmed and physically guided, are checked by CryptoVerif. - Bruno Blanchet [6]

Password Authenticated Key Exchange (PAKE) is one of the critical points in cryptography. It intends to address a viable security issue: how to build up secure correspondence between two gatherings exclusively in view of a common password without requiring a Public Key Infrastructure (PKI). After over 10 years of broad exploration in this field, there have been a few PAKE protocols accessible. The EKE and SPEKE plans are maybe the two most striking cases. Both procedures are however protected. In this paper, we survey these systems in point of interest and abridge different hypothetical and functional shortcomings. Likewise, we exhibit another PAKE arrangement called J-PAKE. Our methodology is to rely on upon settled primitives, for example, the Zero-Knowledge Proof (ZKP). As such, the greater part of the past arrangements have abstained from utilizing ZKP for the worry on effectiveness. We exhibit how to adequately incorporate the ZKP into the convention configuration and in the mean time accomplish great productivity. Our convention has similar computational productivity to the EKE and SPEKE plans with clear favorable circumstances on security. - Feng Hao1, estimated time of arrival [7]

Secret word Authenticated Key Exchange (PAKE) concentrates how to set up secure correspondence between two remote gatherings exclusively in view of their common watchword, without requiring a Public Key Infrastructure (PKI). In spite of broad examination in the previous decade, this issue stays unsolved. Patent has been one of the greatest brakes in conveying PAKE arrangements practically speaking. Furthermore, notwithstanding for the protected plans like EKE and SPEKE, their security is just heuristic; specialists have reported some unobtrusive however stressing security issues. In this paper, we propose to handle this issue utilizing a methodology unique in relation to every past arrangement. Our convention, Password Authenticated Key Exchange by Juggling (J-PAKE), accomplishes common verification in two stages: initial, two gatherings send vaporous open keys to each other; second, they scramble the mutual secret word by juggling people in general keys obviously. The principal utilization of such a juggling strategy was found in taking care of the Dining Cryptographers issue in 2006. Here, apply it to tackle the PAKE issue, and demonstrate that the convention is zero-learning as it uncovers nothing aside from one-piece data: whether the supplied passwords at two sides are the same. With clear favorable circumstances in security, our plan has practically identical proficiency to the EKE and SPEKE conventions.. - Peter Ryan, estimated time of arrival [8]

III.PROPOSED TECHNIQUES

Our work mainly interested on the current Internet standard for such file systems, i.e., parallel Network File System. This make use of Kerberos for establishing the parallel session keys between the clients and the storage devices. Our evaluation of the existing Kerberos-based protocol shows that it has a number of limitations. In this paper, we propose a various 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 around of the workload of the metadata server and concurrently behind forward secrecy and escrow-freeness. All this require only a small fraction of increased computation overhead at the client.

1. Upon receiving an I/O request for a file object from *C*, each *Si* performs the following:
2. Check if the layout σi is valid;
3. Decrypt the authentication token and recover key KCSi;
4. Compute keys skz i = F(KCSi;IDC,IDSi,v,sid,z) for z = 0,1;
5. Decrypt the encrypted message, check if IDC matches the identity of C and if t is within the current validity period v;
6. if all previous checks pass, Si replies C with a key confirmation message using key sk0 i.

In first step we are checking if available layout is valid or not for further operations and communication. In second step we do the decryption operation on the token which is generated by metadata server for authentication process. By performing decryption we will recover the key for client set. In this third step we will compute the key for storage set for accessing the data\information within the storage set. We will compute key by checking the key of client set as well as id for users. As per the result we will return access to user or denied to communicate. Fourth step will perform the task of decryption of encrypted message. And it will also check for validation for user access. In this final step if all the above process is successfully validated then it will return key confirmation message to User\client.

pNFS-AKE-I: Our first protocol can be regarded as a modified version of Kerberos that allows the client to generate its own session keys.

pNFS-AKE-II: To address key escrow while achieving forward secrecy simultaneously, we incorporate a Diffie-Hellman key agreement technique into Kerberos-like pNFS-AKE-I. Particularly, the client *C* and the storage device *Si* each now select a secret value (that is known only to itself) and pre-computes a Diffie-Hellman key component. A session key is then generated from both the Diffie-Hellman components.

pNFS-AKE-III: Our third protocol aims to achieve *full* forward secrecy, that is, disclosure of a long-term key affects only a current session key (with respect to *t*), but not

all the other past session keys

The Flow and the Implementation is as shown in the below steps.

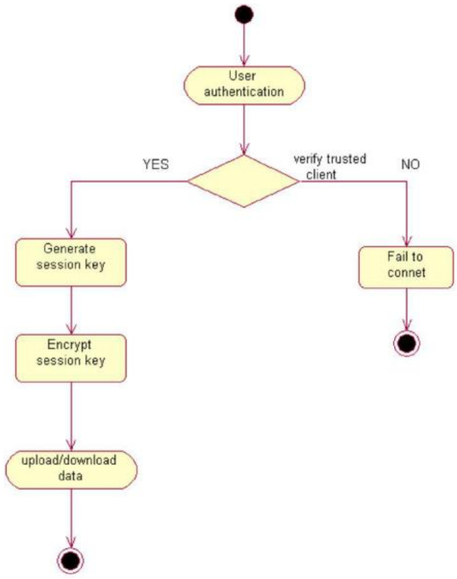

Figure 2: The Flow of execution

Send ( )

```
Send(Ai, start);
ifActiveSessionIndex ≠ 0
then; Abort AActivesessionIndex;
return M;
Else passive attack index = M then;
sent πip;
return A.
```

Corrupt ( )

```
Corrupt (p);
 If session Expire then; P ∈ SS ∪ CS;
 Else return corrupt_message.
```

Reveal ( )

```
reveal(μi);
proceed as follows : − for instance μi then;
return sk ;
```

Execute ( )

```
execute(Ai, Bj); SkA←H(A,B,k); SkA←SkA;
Return(A,B);
```

Test ( )

```
Test(P*,i*);
Instance is defined πip, where P* ∈ SS ∪ CS
If instance is defined πip be session key Ski*p* ; SIM;
If else can b=1 SIM then;
Return Ski*p*;
Else A;
```

IV. PERFORMANCE EVALUATION

We judge the computational above your head for w access requests over time period v for a metadata serverM, a clientC, and storage devices Si for I belongs to [1,N]. We assume that a layout s is of the form of a MAC, and the computational price intended for authenticated symmetric encryption E is similar to that for the non-authenticated version E.10

Assuming fresh session keys are used to secure communications between the client and multiple storage devices, clearly all our protocol have condensed bandwidth rations. This is because during each contact request, the client does not need to get the required authentication token set from M. for this reason, the reduction in bandwidth consumption is approximately the size of n authentication tokens.

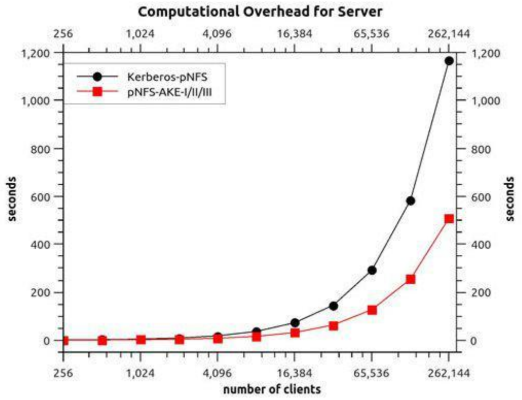

Figure 3: Computational Overhead for server

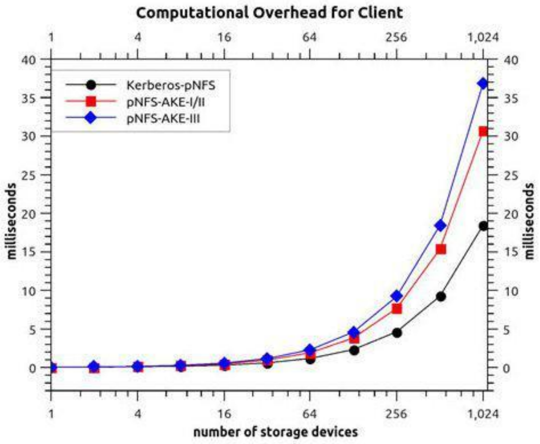

Figure 3: Computational Overhead for Client

VI. CONCLUSION

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

REFERENCES

[1] Qi Xie1*, Bin Hu1*, Na Dong1, Duncan S. Wong2 ., "Anonymous Three-Party Password-Authenticated Key Exchange Scheme for Telecare Medical Information Systems."

[2] Michel Abdalla, David Pointcheval., "Simple Password-Based Encrypted Key Exchange Protocols."

[3] *A. Sai Kumar **P. Subhadra., "User Authentication to Provide Security against Online Guessing Attacks."

[4] Anupam Datta1, Ante Derek1, John C. Mitchell1, and Bogdan Warinschi2., "Key Exchange Protocols: Security Definition, Proof Method and Applications ."

[5] R.S.RamPriya, M.A.Maffina., "A Secured and Authenticated Message Passing Interface for Distributed Clusters."

[6] Feng Hao1 and Peter Ryan2., "J-PAKE: Authenticated Key Exchange Without PKI"

[7] Bruno Blanchet., "Automatically Verified Mechanized Proof of One-Encryption Key Exchange"

[8] Feng Hao⋆1 and Peter Ryan2.,"Password Authenticated Key Exchange by Juggling"

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.