Integrating Authenticated Key Exchange in Parallel Network File Systems enables Secure Data Transfer

[1]Sharmila

[1]Selection Grade Lecturer

Department of Computer Science & Engineering

Government Polytechnic, Arakere, Srirangapatna, Karnataka, India 571415

Abstract

We study the problem of key generation for secure many to many communications. The problem is raised by the proliferation of large scale distributed file system supporting parallel access to multiple storage devices. Our work focuses on current Internet standards for such file systems, i.e. the parallel Network File System (pNFS), which makes use of Kerberos to establish parallel session keys between client and storage devices. Our review of the existing Kerberos-based protocol has a number of limitations: (i) a metadata server facilitating key exchange between clients and storage devices has heavy workload which restricts the scalability of the protocol; (ii) the protocol does not provide forward secrecy; (iii) metadata server establish itself all the session keys that are used between the clients and storage devices, and this inherently leads to the key escrow. In this paper, we propose a variety of authenticated key exchange protocols that are designed to address above issues. We show that our protocols are capable of reducing up to approximately 54% of workload of a metadata server and concurrently supporting forward secrecy and escrow-freeness.

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

INTRODUCTION

In parallel file systems, the file data is distributed across multiple storage devices or nodes to allow concurrent access by multiple tasks of a parallel application. That is typically used in large scale cluster computing that focuses on high performance and reliable fetch to large datasets. That higher I/O bandwidth is achieved through concurrent fetching data to multiple storage devices within large computing clusters, while data loss is protected through data mirroring using defect tolerant striping algorithms. Few examples of high performance parallel file systems that are in the production use are the IBM General Parallel Files System. which are usually required for advanced scientific or data intensive applications such as digital animation studios, computational fluid dynamics, and semiconductor manufacturing. In these environments, hundreds or thousands of file system clients share data and generate very much high aggregate I/O load on the file system supporting petabytes or terabytes scale storage capacities. Independent of the development of the cluster and high performance computing, the emergence of clouds and the MapReduce programming model has resulted in file system such as the

Hadoop Distributed File System (HDFS).

In this work, we investigate the issue of the secure many to many communications in the large scale network file systems which support parallel fetch to multiple storing devices. That we considering the communication model where there are a large number of the clients accessing multiple remote and distributed storage devices in parallel. Particularly, we tries to focus on how to exchange the key materials and establishment of the parallel secure sessions between clients and storage devices in the parallel Network File System (pNFS), the current Internet standards in efficient and scalable manner. The development of pNFS is driven by Sun, EMC, IBM, and UMich/CITI, and thus it shares many similar features and is compatible with many existing commercial network file systems. Our main goal in this work is to design efficient and secure authenticated key exchange protocols that meet specific needs of pNFS. Particularly, we attempt to meet the following desirable properties, which have not been satisfactorily achieved or are not achievable by current Kerberos-based solution.

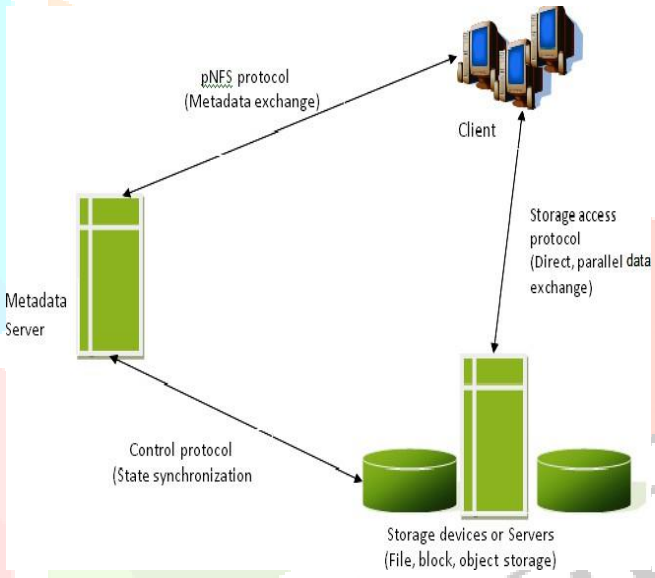

Fig-1: The conceptual model of pNFS

More specifically, pNFS comprises a collection of three protocols: (i) the pNFS protocol that transfers file metadata, also known as a layout, between the metadata server and a client node; (ii) the storage access protocol that specifies how the client accesses data from the associated storage devices according to the corresponding metadata; and (iii) the control protocol that synchronizes the state between the metadata server and the storage devices.

Related works

Telecare Medical Information Systems (TMIS)provide an effective way to improve the medical process between doctors, nurses and patients. By improving the security and privacy of TMIS, it is important while challenging to improve the TMIS so that a patient and a doctor can perform synchronized authentication and session key establishment using a 3-party medical server while the secure data of the patient can be ensured.

In proposed system an anonymous three-party password authenticated key exchange (3PAKE) protocol for TMIS is used. The protocol is based on the efficient elliptic curve cryptosystem. For security, we apply the pi calculus based formal verification tool ProVerif to show that our 3PAKE protocol for TMIS can provide anonymity for patient and doctor as well as achieves synchronized authentication and session key security. The advantage of proposed scheme is security and efficiency that can be used in TMIS. For this J-PAKE based protocols are used. The disadvantage of proposed scheme is of it reduced session keys.

QiXie1*,et al.[1], Password-based encrypted key exchange are protocols that are designed to provide pair of users communicating over an unreliable channel with a secure session key even when the secret key or password shared between two users is drawn from a small set of keys. In proposed scheme, two simple passwords based encrypted key exchange protocols based on that of Bellovin and Merritt. While one protocol is more suitable to scenarios in which the password is shared across multiple servers, the other provides better security. Both protocols are as efficient, if not better, as any of the existing encrypted key exchange protocols in the literature, and yet they only require a single random oracle instance. The proof of security for both protocols is in the random oracle model and based on hardness of the computational Diffe-Hellman problem. However, some of the techniques that we use are quite different from the usual ones and make use of new variants of the Diffe-Hellman problem, which are of independent interest. We also provide concrete relations between the new variants and the standard Diffe-Hellman problem. Advantage of this scheme it is possible to find several flavors of key. In this different types of protocols are used like SIGMA, IKE etc.

MichelAbdalla, et al.[2], Passwords are one of the most common causes of system crashes, because the low entropy of passwords makes systems vulnerable to brute force guessing attacks. Due to new technology passwords can be hacked easily. Automated Turing Tests continue to be an effective, easy- to-deploy approach to identify automated malicious login attempts with reasonable cost of inconvenience to users. Hence in this proposed scheme the inadequacy of existing and proposed login protocols designed to address large- scale online dictionary attacks e.g. from a botnet of hundreds of thousands of nodes. In this scheme proposeda simple scheme that strengthens password based authentication protocols and helps prevent online dictionary attacks as well as many-to-many attacks common to 3-pass SPAKA protocols.

*A.SaiKumar, et al.[3], Proposed scheme Uses compositional method for proving cryptographically sound security properties ofkey exchange protocols, based on a symbolic logic that is interpreted over conventional runs of a protocol against a probabilistic polynomial time attacker. Since reasoning about an unbounded number of runs of a protocol involves induction-like arguments about properties preserved by each run, we formulate a specification of secure key exchange that, unlike conventional key in distinguish ability, is closed under general composition with steps that use the key. We present formal proof rules based on this game-based condition, and prove that the proof rules are sound over a computational semantics.

AnupamDatta1,et al.[4], In a public network, when a number of clusters connected to each other is increased becomes a potential threat to security applications running on the clusters. To address this problem, a MessagePassingInterface(MPI)is developed to preserve security services in an unsecured network. The proposed work focuses on MPI rather than other protocols because MPI is one of the most popular communication protocols on distributed clusters. HereAES algorithm is used for encryption/decryption and interpolation polynomial algorithm is used for key management which is then integrated into Message Passing Interface Chameleon version 2 (MPICH2) with standard MPI interface that becomes ES-MPICH2. This ES- MPICH2 is a new MPI that provides security and authentication for distributed clusters which is unified into cryptographic and mathematical concept. The major desire of ES-MPICH2 is supporting a large variety of computation and communication platforms. The proposed system is based on both cryptographic and mathematical concept which leads to full of error free message passing interface with enhanced security.

R.S.RamPriya,et al.[5], Password Authenticated Key Exchange (PAKE) is one of the important topics in cryptography. It aims to address a practical security problem: how to establish secure communication between two parties solely based on a shared password without requiring a Public Key Infrastructure(PKI).After more than a decade of extensive research in this field, there have been several PAKE protocols available. The EKE and SPEKE schemes are perhaps the two most notable examples. Both techniques are however patented. In this paper, we review these techniques in detail and summarize various theoretical and practical weaknesses. In addition, we present a new PAKE solution called J-PAKE. Our strategy is to depend on well-established primitives such as the Zero-Knowledge Proof(ZKP). So far, almost all of the past solutions have avoided using ZKP for the concern on efficiency. We demonstrate how to effectively integrate the ZKP into the protocol design and meanwhile achieve good efficiency. Our protocol has comparable computational efficiency to the EKE and SPEKE schemes with clear advantages on security.

FengHao1,et al.[6], present a mechanized proof of the password- based protocol One-Encryption Key Exchange (OEKE) using the computationally-sound protocol prover CryptoVerif. OEKE is a non-trivial protocol, and thus mechanizing its proof provides additional confidence that it is correct. This case study was also an opportunity to implement several important extensions of CryptoVerif, useful for proving many other protocols. We have indeed extended CryptoVer if to support the computational Diffie-Hellman assumption. We have also added support for proofs that rely on Shoup'slemma and additional game transformations. In particular, it is now possible to insert case distinctions manually and to merge cases that no longer need to be distinguished. Eventually, some improvements have been added on the computation of the probability bounds for attacks, providing better reductions. In particular, we improve over the standard computation of probabilities when Shoup's lemma is used, which allows us to improve the bound given in a previous manual proof of OEKE, and to show that the adversary can test at most one password per session of the protocol. In this paper, we present these extensions, with their application to the proof of OEKE. All steps of the proof, both automatic and manually guided, are verified by CryptoVerif.

BrunoBlanch et al.[7], Password-Authenticated Key Exchange (PAKE) studies how to establish secure communication between two remote parties solely based on their shared password, without requiring a Public Key Infrastructure (PKI). Despite extensive research in the past decade, this problem remains unsolved. Patent has been one of the biggest brakes in deploying PAKE solutions in practice. Besides, even for the patented schemes like EKE and SPEKE, their security is only heuristic; researchers have reported some subtle but worrying security issues. In this paper, we propose to tackle this problem using an approach different from all past solutions. Our protocol, Password Authenticated Key Exchange by Juggling (J- PAKE), achieves mutual authentication in two steps: first, two parties send ephemeral public keys to each other; second, they encrypt the shared password by juggling the public keys in a verifiable way. The first use of such a juggling technique was seen in solving the Dining Cryptographers problem in 2006. Here, we apply it to solve the PAKE problem, and show that the protocol is zero-knowledge as it reveals nothing except one-bit information: whether the supplied passwords at two sides are the same. With clear advantages in security, our scheme has comparable efficiency to the EKE and SPEKE protocols..

System Flow

We have introduced metadata in our work; metadata plays a vital role in managing the client operation. Metadata performs the major task of authentication of user. It generates One-Time-Password (OTP) to authenticate the user access. Once the user/client gets verified the metadata create session key which enables user to access resources for specific period of time.

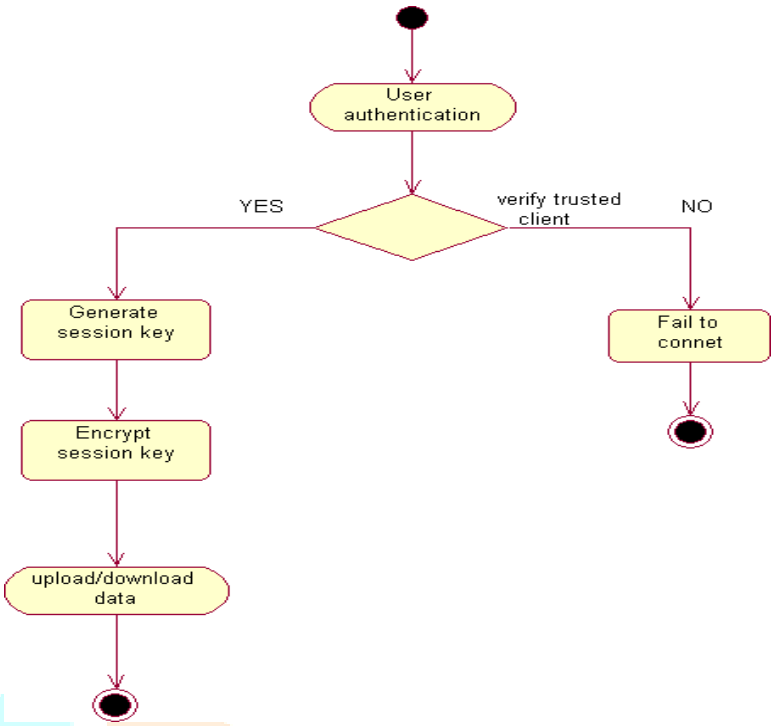

Fig-2:System Flow

Execution of System

Send()

Send(A^i,start);ifActiveSessionIndex≠0 then;

Abort$A^{ActivesessionIndex}$;

returnM;

Else

passiveattackindex=Mthen;

sent $\pi^i p$;

returnA.

Corrupt()

Corrupt(p);

IfsessionExpirethen; P ∈ SS ∪ CS;

Else

returncorrupt_message.

Reveal()

reveal(μ^i);

proceedasfollows:− for instance μ^ithen; return sk ;

Execute() execute(A^i,B^j); SkA←H(A,B,k);

SkA←SkA; Return(A,B);

Test()

Test(P^*,i^*);

Instance is defined π^i_p, where $P^* \in SS \cup CS$

If instance is defined π^i_pbe sessionkey $Sk^{i*}p^*$; SIM;

If else can b=1SIMthen; Return $Sk^{i*}p^*$;

Else A;

Algorithm

Upon receiving an I/O request for a file object from *C*, each

Si performs the following:

Check if the layout σi is valid;

Decrypt the authentication to ken and recoverkey KCSi;

Computekeys skzi=F(KCSi; IDC, IDSi, v, sid,z)forz=0,1;

decrypt the encrypted message, check if IDC matches the identity of C and if t is within the current validity period v;

if all previous checks pass, Si replies C with a key confirmation message using key sk0 i .

ALGORITHM

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 forusers. 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.

OVERVIEW OF PROTOCOL

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- Hellmankey agreement technique into Kerberos-like pNFS-AKE-I. Particularly, the client *C* and the storage device *Si* each now chooses 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, exposure of a long-term key affects only a current session key (with respect to*t*), but not all the other past session keys.

CONCLUSIONS

We 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 lower workload than that of the Kerberos-based approach. 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 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] Qi Xie1*, Bin Hu1*, Na Dong1, Duncan S. Wong2 ., *"Anonymous Three-Party Password-Authenticated Key Exchange Scheme for Telecare Medical Information Systems."*

[2] MichelAbdalla,DavidPointcheval., *"SimplePassword- 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 DistributedClusters."*

[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"*