

## Parallel Network File System with Authenticated Key Exchange Protocols

HUSNA NABEELA<sup>1</sup>, AMENA SAYEED<sup>2</sup>

<sup>1</sup>PG Scholar, Dept of CS, Shadan Women’s College of Engineering and Technology, Hyderabad, TS, India.

<sup>2</sup>Assistant Professor, Dept of CSE, Shadan Women’s College of Engineering and Technology, Hyderabad, TS, India.

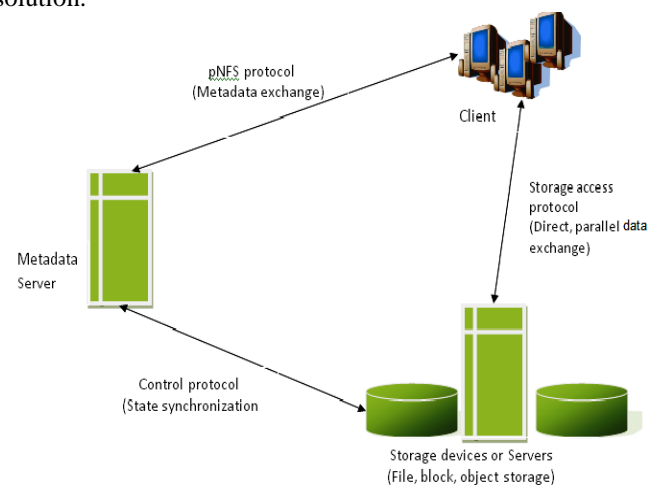
**Abstract:** We study the problem of key generation for secure many to many communications. The problem is raised by the proliferation of large scale distributed file system supporting parallel access to multiple storage devices. Our work focuses on current internet standards for such file systems, 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.

**Keywords:** Parallel Sessions, Authenticated Key Exchange, Network File Systems, Forward Secrecy, Key Escrow.

### I. 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 map reduce 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 as shown in Fig.1. 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.

## II. RELATED WORK

Tele care 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. Qi Xie<sup>1</sup>\*, 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 Bellare 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 Diffie-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 Diffie-Hellman problem, which are of independent interest. We also provide concrete relations between the new variants and the standard Diffie-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. - Michel Abdalla, 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 proposed a 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. Sai Kumar, et al [3].

Proposed scheme Uses compositional method for proving cryptographically sound security properties of key 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. Anupam Datta<sup>1</sup>, 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 message passing interface (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. Here AES 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 esmpich2 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. Feng Hao<sup>1</sup>, et al [6]

We present a mechanized proof of the password-based protocol one-encryption key exchange (OEKE) using the computationally-sound protocol prover crypto verif. OEKE is

## Parallel Network File System with Authenticated Key Exchange Protocols

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 cryptoverif to support the computational diffiehellman assumption. We have also added support for proofs that rely on shoup's lemma 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. Bruno Blanchet [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 (JPAKE), 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. Peter Ryan, Eta [8]

### III. SYSTEM FLOW

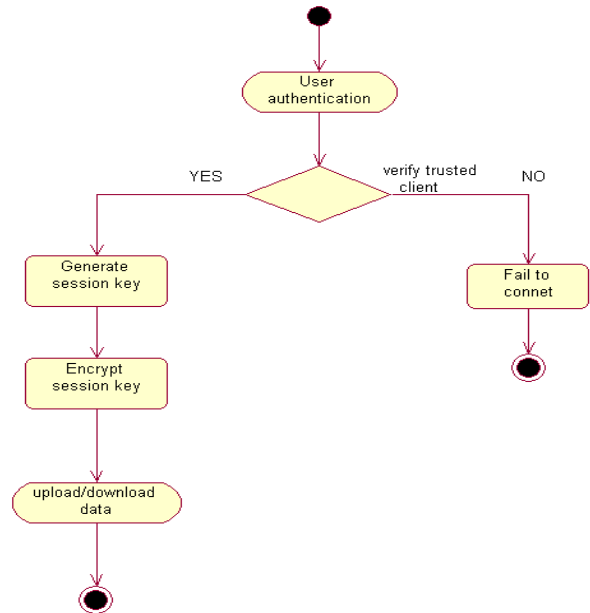
We have introduced metadata in our work; metadata plays a vital role in managing the client operation. Metadata performs the major task of authentication of user as shown in Fig.2. It generates one-time-password (OTP) to authenticate the user access. Once the user/client gets verified the metadata create session key which enables user to access resources for specific period of time.

#### A. Execution of System

**Send ( )**

Send( $A_i$ , start); ifActiveSessionIndex  $\neq 0$  then;  
Abort AActiveSessionIndex; return M;

Else  
passive attack index = M then;  
sent  $\pi ip$ ;  
return A.



**Fig.2. System Flow.**

**Corrupt ( )**

Corrupt (p);  
If session Expire then;  
 $P \in SS \cup CS$ ;  
Else return corrupt\_message.

**Reveal ( )**

reveal( $\mu_i$ );  
proceed as follows : -  
for instance  $\mu_i$  then;  
return sk ;

**Execute ( )**

execute( $A_i$ ,  $B_j$ );  
 $SkA \leftarrow H(A, B, k)$ ;  
 $SkA \leftarrow SkA$ ;  
Return( $A, B$ );

**Test ( )**

Test( $P^*$ ,  $i^*$ );  
Instance is defined  $\pi ip$ , where  $P^* \in SS \cup CS$   
If instance is defined  $\pi ip$  be session key  $Sk_i * p^*$  ;  
SIM;  
If else can  $b=1$  SIM then;  
Return  $Sk_i * p^*$ ;  
Else A;

#### B. Algorithm

Upon receiving an I/O request for a file object from c, each SI performs the following:

- Check If The Layout  $\Sigma_i$  Is Valid;
- Decrypt The Authentication Token And Recover Key  $K_{csi}$ ;

- Compute Keys  $Skz\ I = F(Kcsi; IDC, Idsi, V, Sid, Z)$  For  $Z = 0, 1$ ;
- 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  $Sk0\ I$ .

### C. Explanation of 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 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.

### IV. OVERVIEW OF OUR 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- Hellman key 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.

### V. MODULES

This project is divided into five modules. They are:

**User Interface Design:** This is the first module of our project. The important role for the Network user is to move login window to cloud user window. This module has created for the security purpose. In this login page we have to enter login user id and password. It will check username and password is match or not (valid user id and valid password). If we enter any invalid username or password we can't enter into login window to user window it will shows error

message. So we are preventing from unauthorized user entering into the login window to user window. It will provide a good security for our project. So server contain user id and password server also check the authentication of the user. It well improves the security and preventing from unauthorized user enters into the network. In our project we are using JSP for creating design. Here we validate the login user and sever authentication.

**Parallel Sessions:** Parallel secure sessions between the clients and the storage devices in the parallel Network File System (pNFS) .The current Internet standard—in an efficient and scalable manner. This is similar to the situation that once the adversary compromises the long-term secret key, it can learn all the subsequence sessions. If an honest client and an honest storage device complete matching sessions, they compute the same session key. Second, two our protocols provide forward secrecy: one is partially forward secure with respect to multiple sessions within a time period.

**Authenticated Key Exchange:** Our primary goal in this work is to design efficient and secure authenticated key exchange protocols that meet specific requirements of pNFS. The main results of this paper are three new provably secure authenticated key exchange protocols. We describe our design goals and give some intuition of a variety of pNFS authenticated key exchange6 (pNFS-AKE) protocols that we consider in this work.

**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. However, the protocol does not provide any forward secrecy.To address key escrow while achieving forward secrecy simultaneously, we incorporate a Diffie- Hellman key agreement technique into Kerberos-like pNFS-AKE-I. However, note that we achieve only partial forward secrecy (with respect to v), by trading efficiency over security.

**Server Authentication:** The admin can accept the new user request and also block the users. The users can upload the file to Network. And the admin can allow the files to Network then only the file can store the cloud. If the file uploaded by the user is not permitted from the Server means the file cannot be uploaded by the Client.

### VI. CONCLUSION AND FUTURE WORK

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. This is the application for the network File System. The system employs NLP techniques, heuristic reasoning, and ontologies for the knowledge acquisition processes. DOM-Sortze has been tested using an

## Parallel Network File System with Authenticated Key Exchange Protocols

network File System and comparing the automatically generated elements with the Domain Module manually developed by instructional designers. LDO is the cornerstone of The Semantic Web, yet there still very few commercial LDO apps. In the latest issue of Nodalities, a magazine about the Semantic Web by UK Company Talis, there is an article by Talis CTO Ian Davis about the state of Semantic Web applications. An LDO store in the DB2 database server is a set of user tables within a database schema that stores an LDO data set. A unique store name is associated with each set of these tables. Each LDO store has a table that contains metadata for the store. This table has the same name as the store.

**Future Scope:** Authentication using Password authenticated key exchange using distributed server (PAKEUDE) is done where a cryptographic key - exchange of messages. Security analysis has shown that our protocol is secure against passive and active attacks in case that one of the two servers is compromised

### VII. REFERENCES

- [1] Qi Xie<sup>1\*</sup>, Bin Hu<sup>1\*</sup>, Na Dong<sup>1</sup>, Duncan S. Wong<sup>2</sup> ., “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 Datta<sup>1</sup>, Ante Derek<sup>1</sup>, John C. Mitchell<sup>1</sup>, And Bogdan Warinschi<sup>2</sup>., “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 Hao<sup>1</sup> And Peter Ryan<sup>2</sup>., “J-PAKE: Authenticated Key Exchange Without PKI”
- [7] Bruno Blanchet., “Automatically Verified Mechanized Proof Of One-Encryption Key Exchange”
- [8] Feng Hao<sup>\*1</sup> And Peter Ryan<sup>2</sup>., “Password Authenticated Key Exchange By Juggling

#### Author's Profile:

**Ms. Husna Nabeela** has completed her B.Tech in (IT) from Shadan Women's College of Engineering And Technology, Jawaharlal Nehru Technological University, Hyderabad. Presently, she is pursuing her masters in computer science from Shadan Women's College of Engineering and Technology, Khairtabad, Hyderabad, T.S, India.

**Ms. Amena Sayeed** has completed her M.Tech (CSE) from Shadan Women's College of Engineering And Technology, Hyderabad, Telangana. She is having 3 years of experience in academic, currently working as Asst. Professor in department of CSE, Shadan Women's College of Engineering and Technology, Khairtabad, Hyderabad, T.S, India.