iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 4 **Issue:** VIII **Month of publication:** August 2016

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Authenticated Key Exchange Protocols For pNFS

Kolse G.G.[1], Pathan N.S.[2], Patil A.B.[3], Sinare P.D.[4]
Computer, SCSCOE, Maharashtra, India

Abstract— We study the problem of key generation for secure many to many communications. The problem is raised by the rise 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 workload of a metadata server and concurrently supporting forward secrecy and escrow-freeness.
Keywords— Sessions, Key exchange, NFS, 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 [4]. 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 [6]. 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 [1]. 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) [2]. We provide some background on pNFS and describe its existing security mechanisms associated with secure communications between clients and distributed storage devices [3]. Moreover, we identify the limitations of the current Kerberos-based protocol in pNFS for establishing secure channels in parallel. The main results of this paper are three new provably secure authenticated key exchange protocols [7]. Our protocols, progressively designed to achieve each of the above properties, demonstrate the trade-offs between efficiency and security. We show that our protocols can reduce the workload of the metadata server by approximately half compared to the current Kerberos-based protocol, while achieving the desired security properties and keeping the computational overhead at the clients and the storage devices at a reasonably low level [5]. We define an appropriate security model and prove that our protocols are secure in the model.

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 [6].

II. CURRENT LIMITATIONS

The current design of NFS/pNFS focuses on *interoperability*, instead of efficiency and scalability, of various mechanisms to provide basic security. Moreover, key establishment between a client and multiple storage devices in pNFS are based on those for NFS, that is, they are not designed specifically for parallel communications. Hence, the metadata server is not only responsible for processing

©IJRASET: All Rights are Reserved

Volume 4 Issue VIII, August 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

access requests to storage devices (by granting valid layouts to authenticated and authorized clients), but also required to generate all the corresponding session keys that the client needs to communicate securely with the storage devices to which it has been granted access. Consequently, the metadata server may become a performance bottleneck for the file system. Moreover, such protocol design leads to key escrow. Hence, in principle, the server can learn all information transmitted between a client and a storage device. This, in turn, makes the server an attractive target for attackers. Another drawback of the current approach is that past session keys can be exposed if a storage device's long-term key shared with the metadata server is compromised. We believe that this is a realistic threat since a large-scale file system may 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.

III. PROPOSED SYSTEM

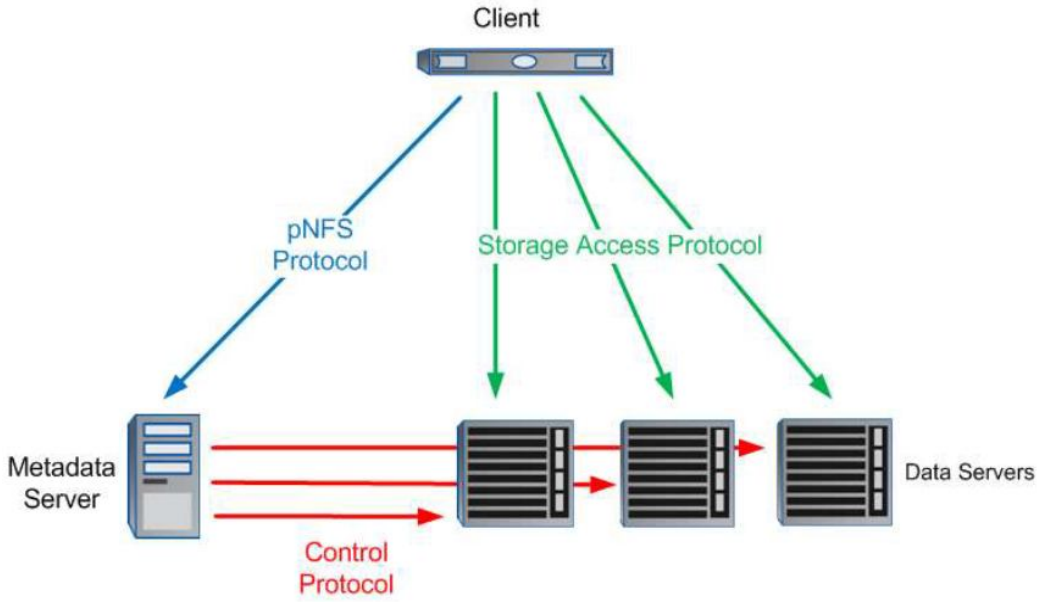


Fig 1: System Architecture

A. *User and Server Authentication*

Kerberos, a widely deployed network authentication protocol supported by all major operating systems, allows nodes communicating over a non-secure network to perform mutual authentication as mention in fig 1.

B. *Secure storage access*

The session key generated by the ticket-granting server (metadata server) for a client and a storage device during single sign-on can then be used in the storage access protocol. It protects the integrity and privacy of data transmitted between the client and the storage device. Clearly, the session key and the associated layout are valid only within the granted validity period.

Particularly, we attempt to meet the following desirable properties, which have not been satisfactorily achieved or are not achievable by current Kerberos-based solution. 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.

IV. SYSTEM FLOW

We have introduced metadata in our work as mentioned below in fig [2], 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.

©IJRASET: All Rights are Reserved

Volume 4 Issue VIII, August 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

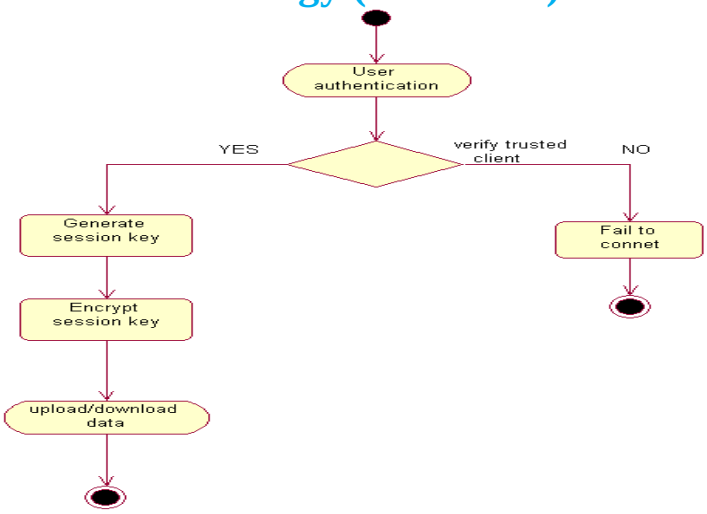


Figure 2: System Flow

V. OUR PROTOCOL

We first introduce some notation required for our protocols. Let $F(k;m)$ denote a secure key derivation function that takes as input a secret key k and some auxiliary information m, and outputs another key. Let *sid* denote a session identifier which can be used to uniquely name the ensuing session. Let also N be the total number of storage devices to which a client is allowed to access. We are now ready to describe the construction of our protocols.

A. *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. That is, the key material used to derive a session key is pre-computed by the client for each v and forwarded to the corresponding storage device in the form of an authentication token at time t (within v). As with Kerberos, symmetric key encryption is used to protect the confidentiality of secret information used in the protocol. However, the protocol does not provide any forward secrecy. Further, the key escrow issue persists here since the authentication tokens containing key materials for computing session keys are generated by the server.

B. *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. Upon expiry of a time period v, the secret values and Diffie-Hellman key components are permanently erased, such that in the event when either C or Si is compromised, the attacker will no longer have access to the key values required to compute past session keys. However, note that we achieve only *partial* forward secrecy (with respect to v), by trading efficiency over security. This implies that compromise of a long-term key can expose session keys generated within the current v. However, past session keys in previous (expired) time periods v' (for $v' < v$) will not be affected.

C. *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. We would also like to prevent key escrow. In a nutshell, we enhance pNFS AKE- II with a key update technique based on any efficient one-way function, such as a keyed hash function. In Phase I, we require C and each Si to share some initial key material in the form of a Diffie-Hellman key. In Phase II, the initial shared key is then used to derive session keys in the form of a keyed hash chain. Since a hash value in the chain does not reveal information about its pre-image, the associated session key is forward secure.

©IJRASET: All Rights are Reserved

VI. ALGORITHM

A. Phase I – For each validity period v

1) Si send diffie hellman component to M : gsi
2) C send diffie hellman component to M : IDC, E(KCM; gc)
3) M send all key component & gsi to C
4) M issue authentication token (KMSN ; IDC; IDSN ; v; gc; gsN) to C

B. Phase II – For each access request at time t

1) C submit access request to M contain all IDs of storage device IDS1…… IDSn
2) M returns layout to C
3) C compute key and send to Si : i, E(skj,0i ; IDC; t)
4) Si verify layout and identity of C : E(skj,0i ; t + 1)

a) In first step we are checking if available layout is valid or not for further operations and communication.
b) 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.
c) 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.
d) Fourth step will perform the task of decryption of encrypted message. And it will also check for validation for user access.
e) In this final step if all the above process is successfully validated then it will return key confirmation message to User\client.

VII.RELETED WORK

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.

- Michel Abdalla, eta [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 ,eta [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 Datta1, eta [4]

©IJRASET: All Rights are Reserved

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 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, eta [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 Hao1, eta [6]

We 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 CryptoVerif to support the computational Diffie-Hellman 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 (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 variable 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]

VIII. CONCLUSION & FUTURE SCOPE

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).

©IJRASET: All Rights are Reserved

Third, we have designed a protocol which not only provides forward secrecy, but is also escrow-free.

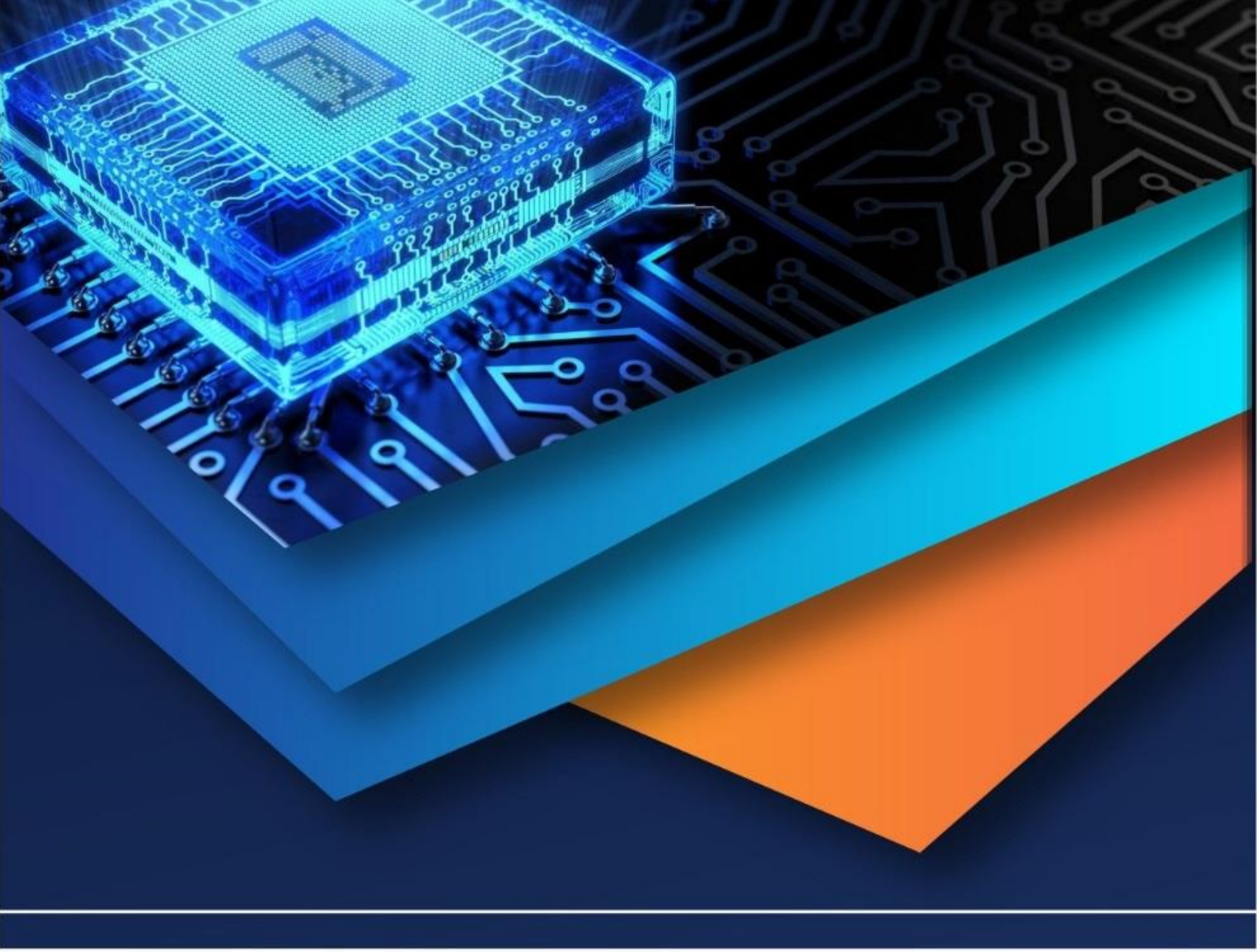
IX. ACKNOWLEDGEMENTS

We would like to thank our guide Prof. P.D.Sinare, for his valuable guidance. We wish to express our sincere gratitude to Prof. Dighe M.S. project coordinator for his encouragement in carrying out this project work. We also wish to express our gratitude to the all lecturers of SCSCOE who render their support during the period of project work.

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 Hao1 and Peter Ryan2.,"Password Authenticated Key Exchange by Juggling"

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)