Analysis of a security protocol in a cluster storage system

A. K. M. Nazmus Sakib & M. Nijim
Department of Electrical Engineering & Computer Science, Texas A & M University- Kingsville, USA

Abstract

This paper represents a review of security vulnerabilities in a cluster storage system. In particular, we focus on identifying the different network level vulnerabilities including attack on the authentication, key exchange and data transfer protocol. We provide a brief description of the different functional entities and we investigate several technical issues including infrastructure and aspects related to the security of the existing system. This paper proposes a secure authentication technique for the network layer of DAS, SAN and NAS. We also bring unencrypted data communication under an encryption process. We implement the concept of shared key by using a one way chain algorithm for block transfer of data. We integrate a new protocol to maintain the level of security according to the data priority concept.
Keywords: cluster, authentication, hash chain, time stamp.

1 Introduction

A cluster file system is shared by being simultaneously mounted on multiple servers. There are several approaches to clustering, most of which do not employ a clustered file system. While many computer clusters don’t use clustered file systems, unless servers are underpinned by a clustered file system the complexity of the underlying storage environment increases as servers are added [1]. Cluster storage systems have emerged as a high performance and cost-effective storage infrastructure for large-scale real-time data-intensive applications. With the technological advances on processors, fast disk I/O interconnections as well as low-cost storage systems, clusters of commodity PCs have emerged as a primary and cost-effective infrastructure for large-scale and data intensive applications [1]. A large fraction of scientific and web-based applications are parallel and data-intensive since the same applications deal with a large amount of data transferred either between memory and storage systems or among nodes of a cluster via interconnection networks. More importantly, cluster storage systems are most suitable for data-intensive applications such as simulations [1] with long execution durations remote-sensing applications and biological sequence analysis running on high-performance computing platforms and assuring the effectiveness of national critical infrastructures [1], since the applications need extensive computation coupled with access to diverse data repositories. The concept of cluster storage systems has been introduced in conjunction with different domains of applications like Internet data caches and video servers. Next-generation data grids are expected to utilise some clusters as data storage systems to support distributed data manipulation functionalities [1]. Cluster storage systems can provide scalable disk bandwidth by stripping and storing large files across a number of cluster nodes, reaching over 1 GB/sec aggregate bandwidth, 33 times faster than a single disk system capacity, by distributing large files across multiple nodes [1].

Cluster systems may experience security threats both from inside and outside of the systems. Since cluster storage systems are built to execute a broad spectrum of unverified user-implemented data-intensive applications from an enormous number of different users, both applications and users may pose as security threats [1]. Malevolent users, i.e., hackers, in an attempt to compromise clusters can take advantage of the vulnerabilities of applications or may access systems to launch denial of service attacks. In addition, legitimate users may tamper with shared data or consume computation cycles excessively to disrupt services available to other cluster users [1, 2].

Many existing clusters do not utilize any security mechanism to counter the security threats, worsening the security status of storage systems [1]. Therefore, it has become mandatory to deploy security services to protect security-critical applications executed on clusters. The most common security threats in cluster computing environments are snooping, alteration, and spoofing. Snooping, i.e., an unauthorized interception of information can be countered by confidentiality services. Alteration, i.e., an unauthorized change of information, can be countered by integrity services. Spoofing, i.e., an impersonation of one entity by another can be countered by authentication services [1, 3].

In this paper, our main focus is to identify different vulnerabilities and remove threat on authentication; key generation and data integrity by introduce a secure authentication and encryption process. Both time stamping and random number based two ways authentication techniques are presented. We also proposed chain algorithm for block transfer of data and measure performance curve for number of iteration.

1.1 Computer cluster

Consist of loosely connected computer that work together so that in many respects they can be viewed as a single system. It is like multi processor systems but multiple systems working together [4], usually sharing storage via a storage area network. Components of a cluster are usually connected to each other through fast local area networks. Each *node* (computer used as a server) running its own instance of an operating system. Computer clusters emerged as a result of convergence of a number of computing trends including the availability of low cost microprocessors, high speed networks and software for high performance distributed computing [4].

1.1.1 Asymmetric clustering

A two-computer type of clustering in which one computer (said to be in hot-standby mode) serves solely as a backup to the active computer [5].

1.1.2 Symmetric clustering

A type of clustering in which all computers in the cluster run applications simultaneously while also monitoring each other [5].

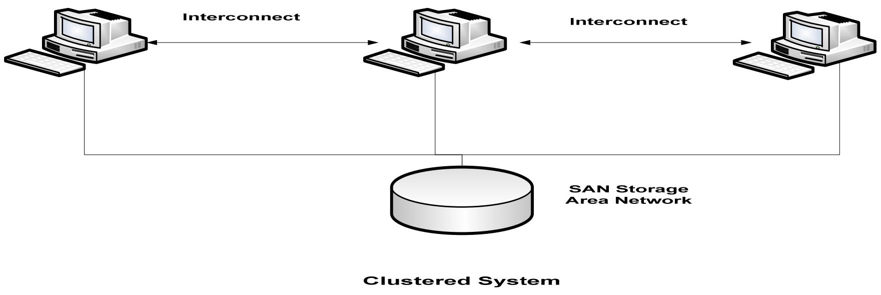

Figure 1: Clustered system.

1.2 Distributed file system

A distributed file system is any file system that allows access to files from multiple hosts sharing via a computer network. This makes it possible for multiple users on multiple machines to share files and storage resources [5].

1.2.1 Network Attached Storage (NAS)

Allow computer data storage connected to a computer network providing data access to a heterogeneous group of clients. It enables centralized storage and files sharing and typically connected via gigabit Ethernet [5].

1.2.2 Storage Area Network (SAN)

Is a dedicated network that provides access to consolidated, block level data storage. SANs are primarily used to make storage devices, such as disk arrays, tape libraries, and optical jukeboxes [6]. A SAN typically has its own network of storage devices that are generally not accessible through the local area network by other devices. A SAN does not provide file abstraction, only block-level operations. However, file systems built on top of SANs do provide file-level access, and are known as SAN file systems or shared disk file systems [6].

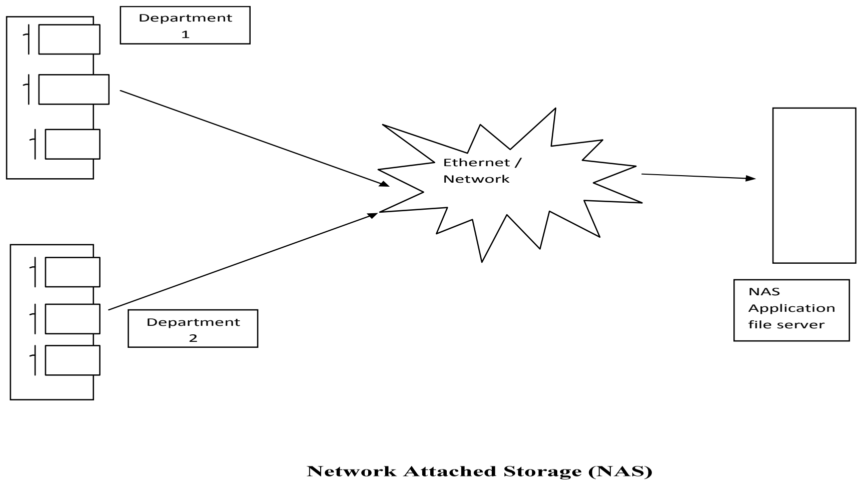

Figure 2: Network Attached Storage (NAS).

Table 1: Difference between SAN and NAS.

	SAN	NAS
Protocol	• Fiber Channel • Fiber Channel to SCSI	TCP/IP
Applications	• Mission-critical transaction-based database application processing • Centralized data backup • Disaster recovery operations Storage consolidation	• File sharing in NFS and CIFS • Small-block data transfer over long distances • Limited read-only database access
Advantages	• High availability • Data transfer reliability • Reduced traffic on the primary network • Configuration flexibility • High performance • High scalability • Centralized management	• Few distance limitations • Simplified addition of file sharing capacity • Easy deployment and maintenance

1.3 Priority data service according to the level of security

Depends on data type the security level must be maintained. According to the security level need to define which data will be sent as a uni-cast or multicast way, which needs multiple key generation and authorization request. Security process is divided into several steps and each steps required different level of security services. For example the negotiation parameter and key generation technique require the highest priority security services. In that case, uni-cast is the better way which may not be possible for some other data type because of bandwidth or other complex computation.

Table 2: Data type with security level.

Data Type	Security Level
Initial Negotiation	Level 1
Authentication	Level 2
Authorization Key	Level 3
Key Generation	Level 4
Data Key Exchange	Level 5
Data type 1	Level 6
Data type 2	Level 7
……………	……………
Data type n	Level n

Table 3: Comparison between PKI Technologies to other types of security solution.

	Authentication	Confidentiality	Integrity	Non repudiation
Firewall	x	x		
Access Control	x	x		
Encryption		x	x	
Public Key Infrastructure	x	x	x	x

1.3.1 Security threats

Attacks on authentication can be described by the way which a network can be intruded and the privacy of the users be compromised. The secure access of network services is becoming an important issue in the present communication infrastructures [7]. Any attempts of an intruder to get registered with the network illegitimately or to create chaos in it, is possible; if the user authorization and authentication is compromised. The ways to breach the authentication frameworks are termed as attacks on privacy and key management protocols and their variants. Several security threat might be possible like; Man-In-The middle attack, Water Torture attack, Interleaving Attack, Suppress Reply Attack, Interceptions, Fabrication, Modification, Reply and reaction, Interruptions, Denial of Service attack, Spoofing etc [7].

2 Possible solution

In this section we proposed two different types of authentication techniques by using hash function and time stamping model. Our first priority is to establish a secure channel between server and client. Then we can use hash chaining algorithm for sending keys and other data [8, 9].

2.1 Authentication process by using hash algorithm

The steps are as follows:

1 Client has to requests for communication and sends out a string as a challenge to Sever.

2 Server also sends out a string as a challenge to Client.

3 Client calculates the message digest of the string by applying hash algorithm and sends the challenging string value and other security parameters to Server.

4 Server need to calculate the message digests for the corresponding string and sends to the Client. Only the legitimate Server and Client know the hash algorithm. But the evil Client is not able to produce correct value for the given string. Now both the Client and Server compare the corresponding message digest value. If it matches then further communication will continued [8]. Otherwise, the Communication will immediately be ceased. This is illustrated in figure 3.

2.2 Authentication process by using time stamp model

Time stamping (T.S) is the process of securely keeping track of the creation and modification time of a document. Here security means that once the document has been recorded, no one can be able to change it, provided that the time stamper’s integrity is never compromised [8, 9]. The technique is based on hash functions and digital signature. First a hash is calculated from the data which is a sort of digital fingerprint of the original data: a string of bits that is different for each set of data. If the original data is changed, hash will also change. Anyone trusting the time stamper can then verify that the document had not been posed after the date that the times tamper vouches and also it can no longer be repudiated that the requester of the time stamp was in possession of the original data at the time given by the time stamp [8, 9].

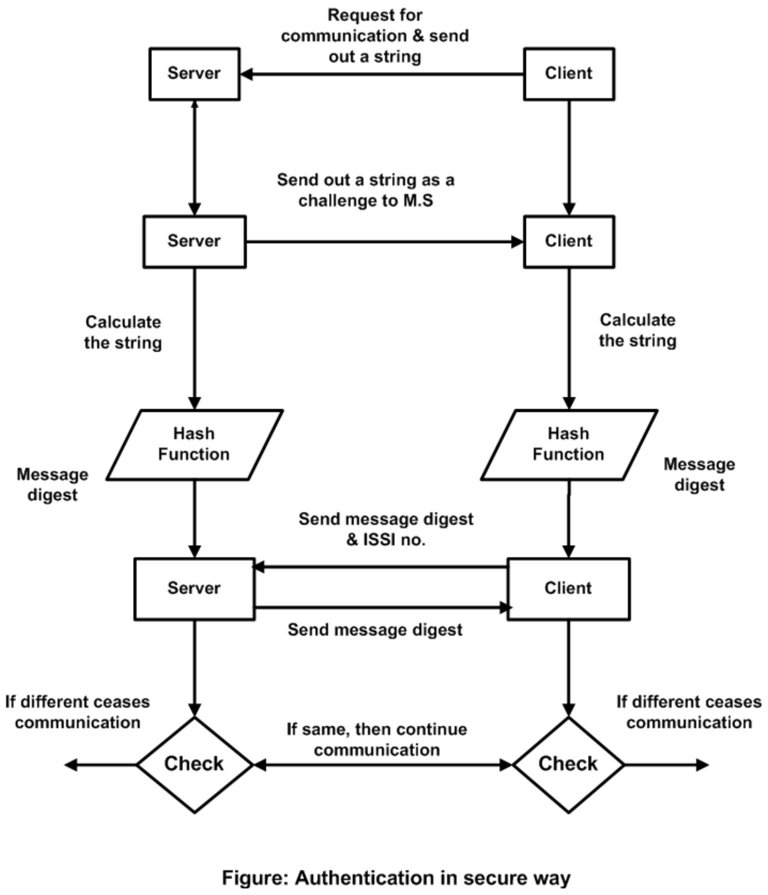
Request for communication & send out a string
Server
Client
Send out a string as a challenge to M.S
Server
Client
Calculate the string
Calculate the string
Hash Function
Hash Function
Message digest
Message digest
Send message digest & ISSI no.
Server
Client
Send message digest
If different ceases communication
If different ceases communication
If same, then continue communication
Check
Check
Figure: Authentication in secure way

Figure 3: Authentication process using hash algorithm.

Steps are given below:

1 Client sends communication request to Server.

2 Server generates the hash (H1) of the data and sends it to the Client.

3 Client now adds the T.S to H1 and generates hash H2. Then H2 is encrypted with the private key of Client. Now encrypted H2 and T.S of Client are to be sent to Server.

4 Server has to add its data with the T.S of Client and to generate hash H3. Now H2 (Which was encrypted by private key of Client) should be decrypted by the public key of Client. If H3=H2 then further communication is continued, otherwise the communication should immediately be ceased.

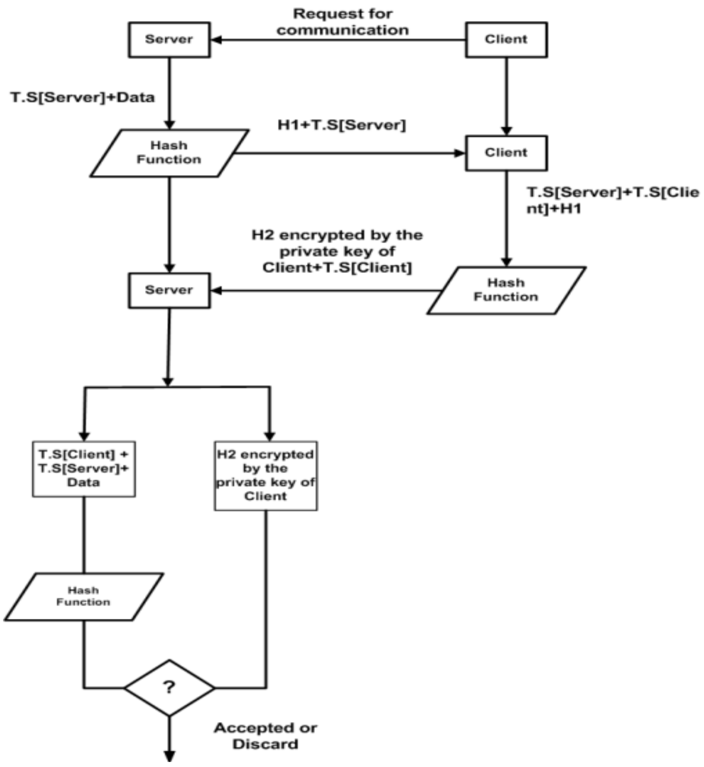
Request for communication
Server
Client
T.S[Server]+Data
H1+T.S[Server]
Hash Function
Client
T.S[Server]+T.S[Client]+H1
H2 encrypted by the private key of Client+T.S[Client]
Server
Hash Function
T.S[Client] + T.S[Server]+ Data
H2 encrypted by the private key of Client
Hash Function
?
Accepted or Discard

Figure 4: Authentication process by using time stamping model.

2.3 Hash chaining algorithm

After the Authentication Technique a secure channel has been created. Now we can generate Keys for further communication [8]. We can use Hash Chaining Algorithm for secure Multicast service. This can be used for Block transfer of data securely. Data can be break into may block and then we can use chain algorithm to transfer them securely. Each block can be verified at the client end. This process is time consuming and so it’s not feasible to apply for each type of data. In that case we need to follow the level of security.

In that process, the server has to generate a random number which represents the initial key K0. Then the other Ks are generated by applying a one way hash function to the previous Ks respectively.

This is iterated n times.

K0 = random ( )
K1 = f (K0)
K2 = f (K1)
……………………….
Kn = f (Kn-1)

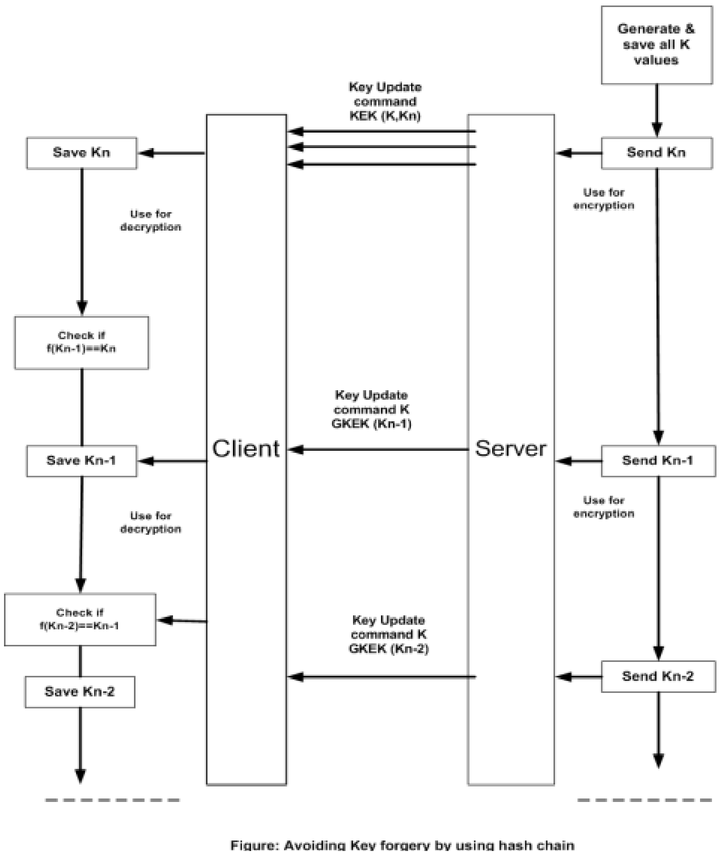

Figure 5: Block transfer of data by using hash chain.

This hash chain allows verifying each Key by applying the same one way function to the previous one. To achieve this chained authentication the last Key has to be distributed to each Client in a secure way as it is the only key in the chain which cannot be authenticated by another one. One possibility is to distribute Kn in the Key update command message which is a uni-cast message and encrypted by a Client related key. If a Client receives a new Key via a broadcasted Key update command message it can verify its integrity by applying the one way hash function to it. If the authentication is positive, the current K can be overwritten and the received one is established. If the authentication fails, the MS discards the message and requests a new K.

3 Experimental results

Table 4: Comparisons between keys [8].

	Unicasting	Asymmetric signature	Hash chain authentication
Computing requirements(Client End)	Low	High	Low
Computing requirements(Server end)	Low	High	Low

Table 5: Symmetric and asymmetric key crypto systems [8].

Approaches	Asymmetric	Symmetric
Encryption	Slower	Faster
Decryption	Slower	Faster
Key Distribution	Easy	Difficult
Security	Highest	Moderate

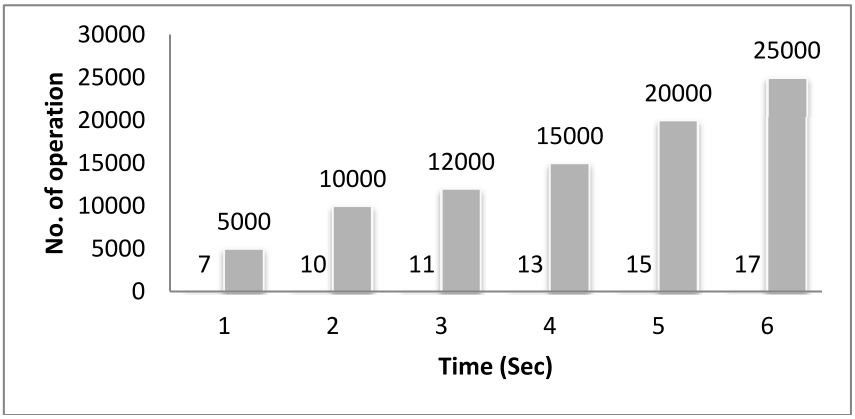

Figure 6: Performance analysis of hash chain algorithm.

4 Conclusion

In this paper, an overview of security protocol in cluster storage system is presented. We investigate various vulnerabilities in the existing system (Specially distributed file system) and give possible solution to eliminate them. We propose a secure Authentication technique for the Network layer of DAS, SAN, NAS etc. We modify the initial network entry process and bring unencrypted data communication under encryption process. We also implement the concept of shared key for block transfer of data and maintain the security level according to the data priority concept.

References

[1] “CaPaS: an optimal security-aware cache replacement algorithm for cluster storage systems” - Dr. Mais Nijim, Int. J. High Performance Systems Architecture, Vol. 3, No. 4, 2011.
[2] “Adaptive Quality of Security Control in Storage Systems”- Dr. Mais Nijim
[3] “AWARDS: An Adaptive Write Strategy for Secure Local Disk Systems”- Dr. Mais Nijim, Xiao Qin, Tao Xie and Mohamed Alghamidi.
[4] “Security for Network Attached Storage Devices”- Howard Gobioff , Garth Gibson, Doug Tygar, October 23, 1997, CMUCS97185.
[5] “An Overview of Security Issues in Cluster Interconnects”- Manhee Lee, Eun Jung Kim, Ki Hwan Yum, and Mazin Yousif.
[6] “Quality of security adaptation in parallel disk systems” - Mais Nijim , Ziliang Zong, Shu Yin, Kiranmai Bellam, Xiao Qin.
[7] “Security Enhancement & Solution for Authentication Frame work in IEEE 802.16”- A.K.M. NAZMUS SAKIB1, Academic and Industrial Collaboration Centre [International Journal of Computer Science and Information Technology] Vol 2, No 6, 2010.
[8] “Shared key Vulnerability in IEEE 802.16e: Analysis & Solution”- A.K.M. NAZMUS SAKIB1, Mir Md Saki Kawsor, International Conference on Computer and Information Technology 2010.
[9] “Key Agreement & Authentication Protocol for IEEE 802.11”, A.K.M. Nazmus Sakib, Fauzia Yasmeen, Samiur Rahman, Md.Monjurul Islam, Prof. Dr. Md. Matiur Rahaman Mian, Md. Rashedur Rahman, Global Journal of Computer Science and Technology, Vol 11, Issue 20, 2011.