IMPROVING KERBEROS SECURITY USING DYNAMIC PASSWORD BASED AUTHENTICATION

VIJENDRASINH THAKUR[1], K. N. HANDE[2]

[1]M Tech Student, department of Computer Science & Engineering, BCCE Nagpur, 9028464089.
[2]Assistant Professor, department of Computer Science & Engineering, BCCE Nagpur, 9822422666.

ABSTRACT

Password-based authentication is not suitable for use on distributed systems. Kerberos is a widely deployed network authentication protocol used in distributed systems. Many works have analyzed its security, identifying flaws and often suggesting fixes, thus promoting the protocol's evolution. Several recent results present successful, formal methods-based verifications of a significant portion of the current version, v.5. The aim is on developing an improved authentication protocol which should defend most of the attacks on Kerberos known as replay attack, password guessing attack. The goal is to achieve a secure authentication for Kerberos environment with removal of weakness of existing Kerberos protocol. Through proposed process, the passwords that the hackers detected for each time will not be the same, so that the hackers are not able to validate those passwords.

Key words: Authentication, Distributed System, Access Control, Kerberos, Cryptography & Network Security, Authorization.

1. INTRODUCTION

Nowadays, distributed systems are very popular and widely used throughout the globe. Most companies uses some flavor of distributed system to connect their various branches located in different geographic locations. In computer science, distributed Systems studies the coordinated use of physically distributed computers. While it is a desirable concept to use distributed systems widely, some concerns remain to be addressed. One of the main concerns that users face when using distributed system is 'security, authentication, integrity, confidentiality, and authorization'. Password-based authentication is not suitable for use on computer networks [14][15]. Eavesdroppers can intercept passwords sent across the network and use it to impersonate a legitimate user. Massachusetts Institute of Technology MIT began a project research named Athena, during which they analyze the way users authenticate themselves to a network and also what happen at other nodes in the network. The study reveals many flaws in the system and consequently, Kerberos was developed to answer a very complicated question-how do you authenticate a user to the server without worrying about impersonation and replay attack from attackers. The purpose of Kerberos was to provide secure authentication between client and server and to prevent interception by eavesdroppers; however, many studies reveals that Kerberos is not altogether one hundred percent immune against some security attacks.
The process of verifying the user's identity is called authentication. Authentication is a service related to identification. It is a fundamental building block for a secure networked

environment. If a server knows the identity of a client, it can decide whether to provide the service, whether the user should be given special privileges, and so forth [2].
With the advent of computer the need for automated tools for protecting files and other information stored on the computer became evident. One thing to keep in mind is that network security costs money: It costs money to hire, train, and retain personnel; to buy hardware and software to secure an organization's networks; and to pay for the increased overhead and degraded network and system performance that result from firewalls, filters, and intrusion detection systems (IDSs). As a result, network security is not cheap.

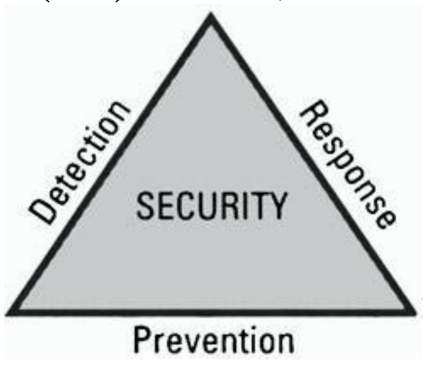

Fig. 1 The security trinity

The three legs of the "security trinity," prevention, detection and response, comprise the basis for network security. The security trinity should be the foundation for all security policies and measures that an organization develops and deploys [8].

Prevention

The foundation of the security trinity is prevention. To provide some level of security, it is necessary to implement measures to prevent the exploitation of vulnerabilities[8].

Detection

Once preventative measures are implemented, procedures need to be put in place to detect potential problems or security breaches; in the event preventative measures fail. It is very important that problems be detected immediately.

The sooner a problem is detected the easier it is to correct and cleanup[8].

Response

Organizations need to develop a plan that identifies the appropriate response to a security breach. The plan should be in writing and should identify who is responsible for what actions and the varying responses and levels of escalation[8].

Kerberos originated at the Massachusetts Institute of Technology (MIT) in the early 1980s. As computer models began to evolve from central system/dumb terminal to client/server, researchers at MIT realized there were a whole new set of issues to resolve. Network users were now able to cause mischief, since they now controlled at least a portion of the computer power. Administrators would need a way to limit and track user actions. MIT research project was created – named Athena. One of the most important issues they found was the passing of passwords across a network in plaintext. The old model had a dedicated line and a single logon – there was no threat of attackers listening in. The Kerberos protocol was the result of the work on project Athena. The earliest versions were limited to internal use at MIT, mainly for testing and interpretation of the results of development (and testing). Kerberos 4 was the first officially distributed release. It was made available to the public in 1989. Kerberos 5 is the most current version available [2][3][4].

The Kerberos model is based in part on Needham and Schroeder's trusted third-party authentication protocol and on modifications suggested by Denning and Sacco. Kerberos provides a means of verifying the identities of principals, (e.g., a workstation user or a network server) on an open (unprotected) network. This is accomplished without relying on authentication by the host operating system, without basing trust on host addresses, without requiring physical security of all the hosts on the network, and under the assumption that packets traveling along the network can be read, modified, and inserted at will. Kerberos performs authentication under these conditions as a trusted third party authentication service by using conventional cryptography, i.e., shared secret key.

The authentication process proceeds as follows: A client sends a request to the authentication server (AS) requesting "credentials" for a given server. The AS responds with these credentials, encrypted in the client's key. The credentials consist of :

1) a "ticket" for the server and
2) a temporary encryption key (often called a "session key").

The client transmits the ticket (which contains the client's identity and a copy of the session key, all encrypted in the server's key) to the server.

The session key (now shared by the client and server) is used to authenticate the client, and may optionally be used to authenticate the server. It may also be used to encrypt further communication between the two parties or to exchange a separate sub-session key to be used to encrypt further communication. The implementation consists of one or more authentication servers running on physically secure hosts. The authentication servers maintain a database of principals (i.e., users and servers) and their secret keys. In order to add authentication to its transactions, a typical network application adds one or two calls to the Kerberos library, which results in the transmission of the necessary messages to achieve authentication.

There are two methods by which a client can ask a Kerberos server for credentials. Usually client requests for a ticket-granting ticket (TGT) which can be used with the ticket-granting server (TGS). In the second method, the client sends a request to the TGS. The client sends the TGT to the TGS in the same manner as if it were contacting any other application server which requires Kerberos credentials. The reply is encrypted in the session key from the TGT. Once obtained, credentials may be used to verify the identity of the principals in a transaction, to ensure the integrity of messages exchanged between them, or to preserve privacy of the messages. The application is free to choose whatever protection may be necessary.

To verify the identities of the principals in a transaction, the client transmits the ticket to the server. Since the ticket is sent "in the clear" (parts of it are encrypted, but this encryption doesn't prevent replay) and might be intercepted and reused by an attacker, additional information is sent to prove that the message was originated by the principal to whom the ticket was issued. Encrypting the authenticator in the session key proves that it was generated by a party possessing the session key [5].

2. RELATED WORK

Massachusetts Institute of Technology (MIT) developed Kerberos to protect network services provided by Project Athena. Several versions of the protocol exist; versions 1–3 occurred only internally at MIT. Many members of Project Athena contributed to the design and implementation of Kerberos. Steve Miller and Clifford Neuman are the primary designers of Kerberos version 4 with contributions from Jerome Saltzer and Jeffrey Schiller. They published that version in the late 1980s, although they had targeted it primarily for Project Athena. Version 5, designed by John Kohl and Clifford Neuman,

appeared as RFC 1510 in 1993 [5] (made obsolete by RFC 4120 in 2005 [6]), with the intention of overcoming the limitations and security problems of version 4 [2].
Kerberos is also introduced to be used in IPv6 networks. S. Sakane, N. Okabey, K. Kamadaz, and H. Esakix describe a method to establish secure communication using Kerberos in IPv6 networks [2].
They propose a mechanism to achieve access control using Kerberos and to deal with address resolution using Kerberos with modification.
Nitin et. al present an image based authentication system using the Kerberos protocol at 2008 [2][7]. That paper is a comprehensive study on the subject of using images as a password and the implementation of Jaypee University of Information Technology (JUIT) Image Based Authentication (IBA) system called as JUIT-IBA using Kerberos protocol.
In 2007, MIT formed the Kerberos Consortium along with some of the major vendors and users of Kerberos such as Sun Microsystems, Apple, Google, Microsoft, etc., to foster continued development. The MIT Kerberos Consortium was created to establish Kerberos as the universal authentication platform for the world's computer networks [2].
Dr. S. Santhosh Baboo, K. Gokulraj (2010), Authentication is one of the essential security features in network communication. Authentication process ascertains the legitimacy of the communicating partners in communication. The authors introduced a new authentication scheme based on dynamicity which is relatively a different approach to ensure and enhance the smart card based remote authentication and security. This method discusses about the authentication for smart card based network systems. This method introduces a dynamic authentication scheme which includes number of factors, among them the password, password index, and date of modification are important factors which decides the dynamicity [8].
K. Aruna et. al (2010), The aim of this paper is to establish a collaborative trust enhanced security model for distributed system in which a node either local or remote is trustworthy. They have also proposed a solution with trust policies as authorization semantics. Kerberos, a network authentication protocol is also used to ensure the security aspect when a client requests for certain services. In the proposed solution, they have also considered the issue of performance bottlenecks [8].
Dr.Mohammad N. Abdullah & May T. Abdul-Hadi (2009), they try to establish a secure communication between the clients and mobile-bank application server in which they can use their mobile phone to securely access their bank accounts, make and receive payments, and check their balances [8][9].
Kerberos has grown to become the most widely deployed system for authentication and authorization in modern computer networks. Kerberos is currently shipped with all major computer operating systems and is uniquely positioned to become a universal solution to the distributed authentication and authorization problem of communicating parties [2][3][4].

3. KERBEROS

3.1 Kerberos Message Exchange

In an unprotected network environment, any client can apply to any server for service. The obvious security risk is that of impersonation. An opponent can pretend to be another client and obtain unauthorized privileges on server machines. To counter this threat, servers must be able to confirm the identities of clients who request service. Each server can be required to undertake this task for each client/server interaction, but in an open environment, this places a substantial burden on each server.

The Kerberos Key Distribution Center, or KDC for short, is an integral part of the Kerberos system. The KDC consists of three logical components: a database of all principals and their associated encryption keys, the Authentication Server, and the Ticket Granting Server. While each of these components are logically separate, they are usually implemented in a single program and run together in a single process space.

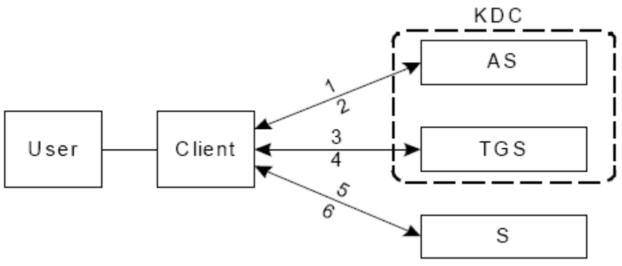

Fig. 2 Kerberos Architectural Components

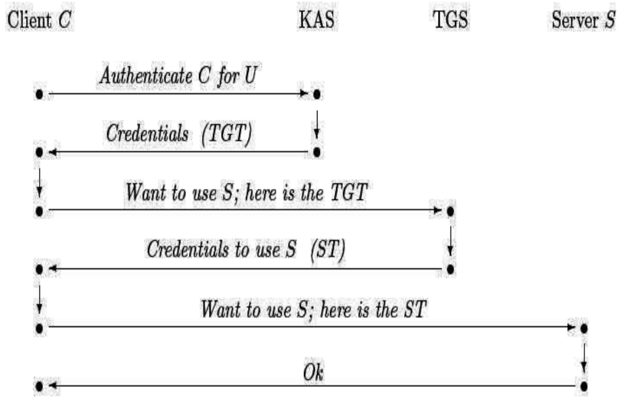
Client C
KAS
TGS
Server S
Authenticate C for U
Credentials (TGT)
Want to use S; here is the TGT
Credentials to use S (ST)
Want to use S; here is the ST
Ok

Fig. 3 Overview of the Kerberos Authentication

The client's goal is to be able to authenticate him to various application servers (e.g., email, file, print, web servers). This is done by first obtaining credentials, called the “ticket-granting ticket” (TGT), from a “Kerberos Authentication Server” (KAS) and then by presenting these credentials to a “Ticket-Granting Server” (TGS) in order to obtain a “service ticket” (ST), which is the credentials that the client finally presents to the application servers in order to authenticate himself. A TGT might be valid for a day, and may be used to obtain several STs for many different application servers from the TGS, while a single ST might be valid for a few minutes (although it, too, may be used repeatedly while it is still valid) and is used for a single application server. The KAS and the TGS are altogether known as the “Key Distribution Center” (KDC). The client's interactions with the KAS, TGS, and application servers are called the Authentication Service (AS), Ticket-Granting (TG), and Client-Server (CS) exchanges, respectively.
If we talk about the unprotected environment, any client can apply to any server for service. This has a security risk of impersonation. An opponent can pretend to be another client and obtain unauthorized privileges on server machines. In the above scheme the transaction will be highly secured in the sense that Authentication server creates a ticket which is further encrypted using the secret key shared by the server and authentication

Server. This ticket then sends back to client. Because the ticket is encrypted, it cannot be altered by client or by an opponent.

3.2 Security Considerations

One weakness of the standard Kerberos protocol is that the key Kc used to encrypt the client's credentials is derived from a password, and passwords are vulnerable to dictionary attacks [14]. Moreover, since the initial request is entirely plaintext, an active attacker can repeatedly make requests for an honest client's credentials and amass a large quantity of plaintext ciphertext pairs, the latter component being encrypted with the client's long-term key kC. While the attacker is unable to use these credentials to authenticate to the system, he is given considerable opportunity to perform an active dictionary attack against the key. Kerberos can optionally use pre-authentication, a feature that is designed to prevent an attacker from actively requesting and obtaining credentials for an honest user. In brief, pre-authentication works by requiring the client to include a timestamp encrypted with her long-term key (kC) in the initial request. The authentication server will only return credentials if the decrypted timestamp is sufficiently recent. This method successfully prevents an attacker from actively obtaining ciphertext encrypted with the long-term key which is what our proposed research work.

3.3 Kerberos Drawbacks

1) "Password guessing" attacks:

Password guessing attacks are not solved by Kerberos. If a user chooses a poor password, it is possible for an attacker to successfully mount an offline dictionary attack by repeatedly attempting to decrypt messages obtained which are encrypted under a key derived from the user's password.

The aim is on designing a user authentication protocol that is not susceptible to password guessing attacks. The main goal is to remove this password guessing attack [1][2][5][6][14].

2) Principals must keep their secret keys secret. If an intruder somehow steals a principal's key, it will be able to masquerade as that principal or impersonate any server to the legitimate principal [5][6][14].

3) KDC spoofing:

This refers to an attack which relies basically on the ability to spoof KDC responses. Having in mind the Kerberos protocol description, spoofing KDC response should not be a security concern. Indeed, Kerberos has been design to bear an untrusted network. IP spoofing is something that happens on untrusted networks. Kerberos protocol performs mutual authentication. End user's and server's identities need to be proven. This ensures protection against Man-in-the-Middle attacks. Yet circumstances still exit under which this might represent a real risk [14].

4) "Denial of service" attacks

Denial of service attacks are not solved with Kerberos. There are places in these protocols where an intruder can prevent an application from participating in the proper authentication steps. Detection and solution of such attacks (some of which can appear to be not-uncommon "normal" failure modes for the system) is usually best left to the human administrators and users.

An attacker can mount a DoS attack by flooding the KDC with authentication requests, which may result in poor response time to legitimate requests and in worst cases might even crash the KDC. To prevent a denial of service attacks, one solution could be to place the KDC behind a firewall and/or place redundant KDC slaves to service the requests and balance load [14][15].

5) Compromise of the KDC Server: KDCs maintain an encrypted database of all principals/verifiers (i.e., users and servers) and their secret keys. If the security of the KDC is compromised, the security of the entire network is compromised even though the principal keys are stored in an encrypted form using the master key; the master key itself is stored in the KDC. An attacker can gain control of the entire network, can create or modify any principal's credentials. To prevent such attack, ensure the security of the KDC and limited the access to KDC to very limited personnel. Even then, it does not protect against an insider attack, which is to an attack by an internal user who misuses his/her privileges i.e. an administrator.

6) Compromise of a verifier/server: If the security of the server is compromised, all the services on that server is compromised. The attacker will be able to impersonate any service running on the server and/or decrypt any communication between the service and a client/principal. The security of the services running on a server is dependent upon the security of the server. Security measures of servers should be proportional to value of the services and resources stored on that server.

4. PROPOSED RESEARCH WORK

Firstly Client or user has to authenticate itself to the authentication server. We will add a mechanism by which dynamicity in passwords can be added according to figure 4 & 5.

There are three fields which are sent to authentication server or KDC. Those are Principal ID, Principal Password & current system timestamp of client's machine. Principal password & timestamp are hashed first & then sent.

Now at server side, server checks to see that user is the right one or not who it pretends to be. Server has its database of valid Principal ID & Principal Password pairs. Server firstly checks for replay attack by comparing the timestamps. Then server checks to see a right password is supplied or not by comparing hash values of received & server generated values. Now next step is generation of secret key used to encrypt the ticket. It is shown in fig 6.

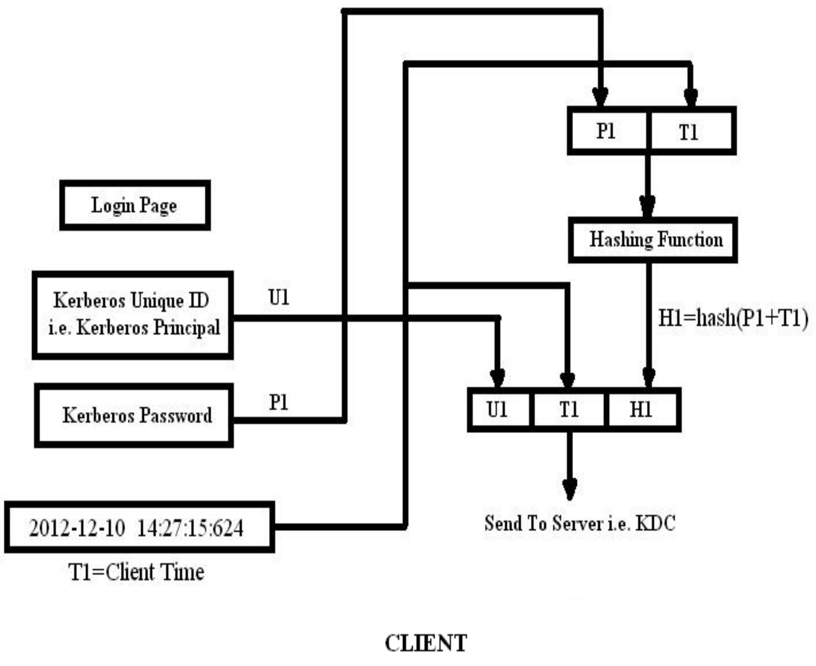

Fig 4 Design of New Authentication Client Side

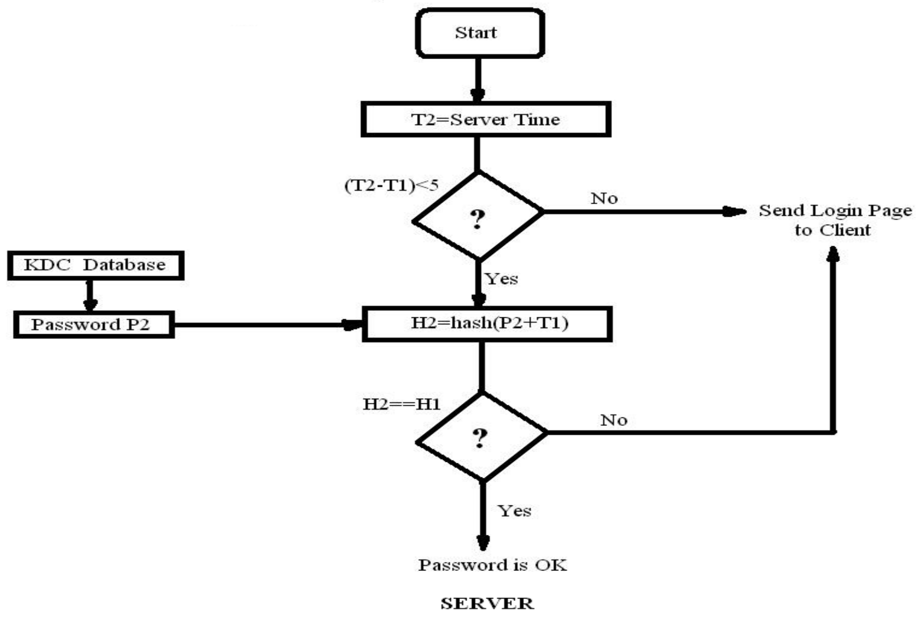

Fig 5 Design of New Authentication Server Side

The proposed method to improve Kerberos towards password guessing attack as the use of strong hashing algorithm on user's profile. It is obvious that Kerberos is vulnerable to password guessing attacks. There is one suggestion on an authentication protocol based on Kerberos with a little modification in the Kerberos database. It will be independent of the user password. Instead, the KDC will save a profile for every principal in the realm that it manages. The contents of the profile may be audio, video, image, or text data. The KDC database may have profiles of mixed data contents (some profiles may be audio, others may be images, and so on). The realm principal may be a client or a server instance that participates in the network communication. Every principle (user or server) has to register with the Kerberos database. The principal will register with the Kerberos server by the principal ID. Then, the KDC will map this ID to the principal profile. The Kerberos server will generate the principal secret key by applying a hashing algorithm to the principal profile. The input to the hashing algorithm will be the principal profile and the output will be encrypted to generate the principal secret key. The block diagram of Fig. 5 summarizes proposed scheme to generate the principle secret key. It is also suggested to control the lifetime of that secret key. There is a simple idea for that. Since the system clocks of the hosts that are involved in the protocol should be synchronized (this can be maintained manually or assured by using Network Time Protocol daemons), it will append the current system timestamp to the principal profile every certain predefined period (this period is a design parameter; i.e. a site constant). Consequently, the input to the hashing algorithm will change, and thus the secret key will change too [2][3][4].

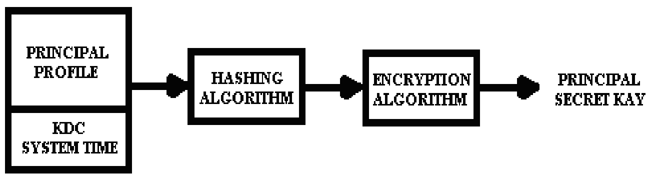

Fig. 6 Proposed Secret key generation Architecture

The machine which houses this database is called the master machine. It is extremely important that the master KDC will be installed on a carefully protected and physically secure machine. If possible, the machine should be dedicated to running the authentication server and the number of users with access should be limited. Also, there may be one more read-only copy of the Kerberos database on another machine called the slave. However, all changes to the database must be made on the master computer system. Changing or accessing the contents of a Kerberos database requires the Kerberos master password. At the principle side (a client or a server), the secret key may be obtained by one of two ways depending on the network administrator choice. The first option will be chosen if the administrator decided to keep the type of the profile contents secret. Then the principles secret keys will be distributed using another secure method. This can be achieved using hardware equipments or by using a secure delivery system. The second option will be chosen if the administrator decided to announce the type of the profile contents. In that case, every principle may keep a copy of his or her profile and prompt to enter the path of that profile during the run of the Kerberos protocol [2][3][4].

5. CONCLUSION

Adding Kerberos to a network can increase the overall security available to the users and administrators of that network. Remote sessions can be securely authenticated and encrypted. In addition, Kerberos allows the user and service principal's database to be managed securely from any machine that supports the Kerberos protocol. Traditional Kerberos is vulnerable to password guessing attack. Hence the aim is on developing a secure authentication mechanism for Kerberos Environment. Dynamicity in password is added to Kerberos protocol by making use of hashing algorithm & timestamp in proposed scheme. Resulting scheme will be independent of the user password. By this way, we will overcome weak passwords chosen by the network principal that are susceptible to password guessing attacks, the main drawback of the Kerberos protocol.

REFERENCE

[1] Emir Accilien CMPT 585001 "Security issues in Distributed Systems:Is Kerberos the Answer?"

[2] Eman El-Emam, Magdy Koutb, Hamdy Kelash, and Osama Farag Allah, "An Authentication Protocol Based on Kerberos 5", IJNS International Journal of Network Security, Vol.12, No.3, PP.159-170, May 2011.

[3] Eman El-Emam, Magdy Koutb, Hamdy Kelash , and Osama Farag Allah,"A Network Authentication Protocol Based on Kerberos", IJCSNS International Journal of Computer Science and Network Security, Vol. 9 (8), August 2009.

[4] El-Emam, "An optimized Kerberos authentication protocol", ICCES 2009.International Conference on Computer Engineering & Systems, pp. 508-523, 14-16 Dec. 2009.

[5] [RFC1510] Kohl, J., Neuman, C., "The Kerberos Network Authentication Service (V5)", RFC 1510, Sept 1993.

[6] C. Neuman, T. Yu, S. Hartman, and K. Raeburn, "The Kerberos network authentication service (V5)". Network Working Group. Request for Comments: 4120. Available at http: //www.ietf.org/rfc/rfc4120.txt, 2005.

[7] Nitin et al. "Security Analysis and Implementation of JUIT—Image Based Authentication System Using Kerberos Protocol". Proceedings of the 7th IEEE/ACIS International Conference on Computer and Information Science, (ICIS 2008).

[8] Prof R.P. Arora, Garima Verma, "Implementation of highly efficient Authentication and Transaction Security", International Journal of Computer Applications (0975 – 8887) Volume 21– No.3, May 2011.

[9] Dr. Mohammad N. Abdullah, May T. Abdul-Hadi, "A Secure Mobile Banking Using Kerberos Protocol", Eng. & Tech. Journal, Vol. 27, N0. 6, 2009.

[10] William Stallings, "Cryptography and Network Security principles and practices", Fourth edition. Pearson Prentice Hall, (2006). pp. 401-419, pp. 433-435.

[11] A. Boldyreva and V. Kumar, "Provable-Security Analysis of Authenticated Encryption in Kerberos". IEEE Symposium on Security and Privacy (SP'07). May 2007.

[12] Shital S. Thorat, Prof. H. K. Sawant, Mrs. Sarita S. Gaikwad, Prof. G. T. Chavan, "Comparative study of various PKINIT methods used in Advanced Kerberos", IJCSE International Journal of Computer Science and Engineering, Vol. 2 (7), 2010.

[13] L. Hornquist Astrand, "Deprecate DES, RC4-HMAC-EXP, and other weak cryptographic algorithms in Kerberos" February 27, 2012, Accessed July 31, 2012. Available Online URL: http://tools.ietf.org/pdf/draft-ietf-krb-wg-des-die-die-die-04.pdf

[14] Thomas Wu, "A Real-World Analysis of Kerberos Password Security", Available Online: www.isoc.org/isoc/conferences/ndss/99/proceedings/papers/wu.pdf Accessed: July 23, 2012.

[15] Alexandra Boldyreva, Virendra Kumar, "Provable-Security Analysis of Authenticated Encryption in Kerberos", IEEE Symposium on Security & Privacy Proceedings, pp. 92–100, 2007 & in IET Information Security Journal, Vol. 5(4), pp. 207–219, 2011.

[16] Rafael Marín-Lópe, Fernando Pereñíguez, Gabriel López, Alejandro Pérez-Méndez, "Providing EAP-based Kerberos pre-authentication and advanced authorization for network federations", Computer Standards & Interfaces Volume 33, Issue 5, September 2011, Pages 494–504 ScienceDirect.com, Accepted 22 February 2011.

[17] Jason Garman , "Kerberos the Definitive Guide", Publisher : O'Reilly, Pub Date : August 2003, ISBN : 0-596-00403-6, Pages : 272. Accessed: October 9 2012.