Modified Kerberos Model for Sustaining the Integrity Privacy & Authentication of Critical Data in Cloud Computing

Neeraj Kumar[(1)], Vibhav Prakash Singh[(2)], Himanshu Mishra[(3)], Atul Barsaiyan[(4)]

Abstract— **As the new age of internet, Cloud Computing becoming the most powerful virtual supercomputing where we use the computing and other resources over the internet. So the challenging task for every user who is invoking the services provided by cloud over the internet is security. Security services just like authentication integrity and privacy becomes the main problem while users moving there critical data over internet for invoking the services provided by the cloud. So the main issues are how to maintain the integrity and privacy of those critical data over cloud. Here in this paper we have proposed a modified Kerberos model which will maintain the integrity, privacy & Authentication of the critical data which is transmit over cloud.**

Index Terms— Authentication; Authorization; Kerberos, Hash Function; Message Chaining; Nonce;

1 INTRODUCTION

Cloud computing is a technology it employ internet and multiple servers at different locations to maintain the transactions of data and other types of required works. Cloud computing allows its users to apply different applications at different sites without the installation and can access their files in the cloud with the application .Cloud computing is a type of front end which provides user to work and the actual work is done at the back end of the infrastructure. In cloud computing users can easily access their data anytime anywhere without use of any hardware & software equipment. NIST has identified three basic types of cloud service offerings. In figure-1, these models are: (i) Software as a service (SaaS) which offers renting application functionality from a service provider rather than buying, installing and running software by the user. (ii) Platform as a service (PaaS) which provides a platform in the cloud, upon which applications can be developed and executed. (iii)Infrastructure as a service (IaaS) in which the vendors offer computing power and storage space on demand. [3] Kerberos provides a ways of confirming the identities of subjects, (e.g., a workstation user or a network server) on an open unprotected network. This is accomplished without relying on assertions 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 (shared secret key) cryptography. [19].

- *Neeraj Kumar[(1)] Department of Computer Science & Engineering, Hindustan College of Science &Technology, Mathura, India E-mail: neeraj.iiita2009@gmail.com*
- *Vibhav Prakash Singh[(2)] Department of Information Technology, Anand Engineering College Agra, India, E-mail: vibhav.gla @gmail.com*
- *Himanshu Mishra[(3)] Department of Computer Science & Engineering, Hindustan College of Science &Technology, Mathura, India E-mail:himanshu.ims2009026@gmail.com*
- *Atul Barsaiyan[(4)] Department of Computer Science & Engineering, Hindustan College of Science &Technology, Mathura, India E-mail: atulbarsaiyan@gmail.com*

In the Figure-2 Provides the cloud computing infrastructure environment where it describe the Cloud computing process that how the user can move in the cloud by using his credential with the authentication system and access the resources provided by the cloud server [10], [11], [12].Where there are lots of security issues and limitations involved such as the integrity of the processed data, privacy issues with processed data validity of the processed data. There are also lots of people who are enjoying by sharing and transferring there data over cloud every day to day life. [13], [14].

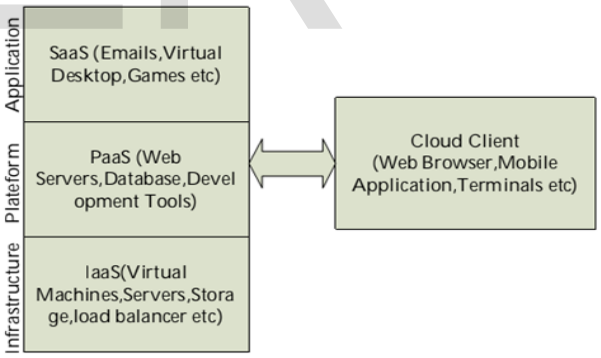

Figure1. Cloud Computing Layers

They do their online transactions using cloud computing applications with the internet connectivity form anywhere anytime in the world [1], [2].So security issues in cloud computing become so important for maintaining the integrity of the data and it is categorized into the following three broad classes:

- Traditional security concerns
  - ☐ Cloud service providers' vulnerabilities
  - ☐ Phishing cloud provider
  - ☐ Expanded network attack surface
  - ☐ Authentication and authorization
  - ☐ Forensics in the cloud

- Availability issues
  - ☐ Third Party Data Control
  - ☐ Due diligence
  - ☐ Auditability
  - ☐ Contractual obligations
  - ☐ Cloud provider espionage
- Third party data control-related issues
  - ☐ Side channel attacks
  - ☐ Denial of service attacks
  - ☐ Social networking attacks
  - ☐ Mobile device attacks
  - ☐ Increased authentication demands

We can take full benefit of cloud computing if we solve the security, privacy and authentication issues involved with cloud computing. Security and Privacy Issues in Cloud Computing can be categorized as [6].

- Identity Management.
- Authentication & Authorization of User.
- Physical security of Critical data.
- Confidentiality of the data.
- Integrity of the data.
- Availability of the data.
- Application Security.
- Privacy Issues with user's data.
- Legal issues.

The cloud computing infrastructure environment is very important to understand for the cloud users that is how we are using the data in the cloud environment here client will access the resources provided by the cloud providers by giving the user ID. Next it will be authenticated and based on this ID client will use the resources of the cloud providers here we present a diagram of cloud computing infrastructure environment which will explain the process of user access with cloud resources.

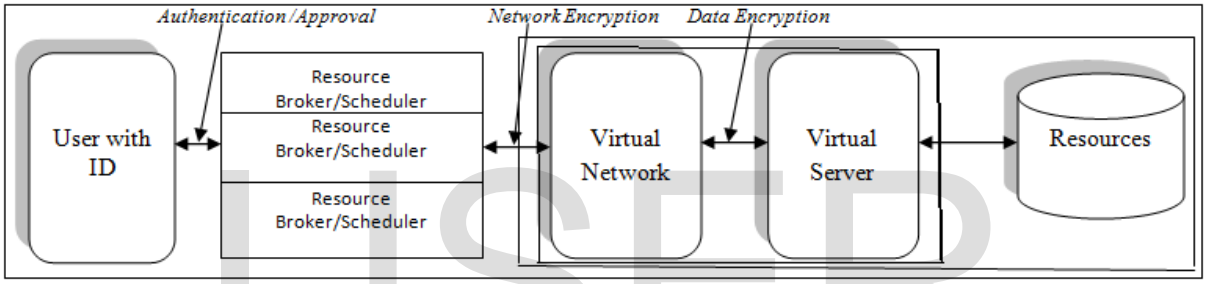

Figure2. Cloud computing infrastructure environment

2 Proposed Model

In this proposed model we are presuming the states of the authentication server, ticket granting ticket and the server is same because of the null state [Stateas=Statetgs=Statev]. Here we are using the concept of hashing for secure accessing of services provided by cloud providers. We are using Kerberos version 4 [6], [7], [8], [9], [10] as a base for this model and modifying this version for secure accessing of information. In this model we have use the state of client and state of server as Null at the initial state before processing any transaction. [State client=Null, State server=Null]. And there are three sections which is the base for our proposed model these sections are describes as follows-

2.1 Authentication Services Exchange to obtain ticket-granting ticket-

In this section we have initialize the client and server with a null value here client take the hash of his state which is null and store this hash value to his database. Then he sends the hash value with the message to the Authentication server in order to access the service provided by the cloud provider. Now at the server side the server receive the message with hash value and compare with his hash value by doing the same. If the match is same then Authentication server sends the ticket to the client by the encrypted message with the key derived by the users password (K_c). And authentication server saves his the state of the server. Now client again take the hash value of received message and compare this hash value with the received hash if the received hash value is same as computed at client side and message is accepted and client will open this message with the K_c and get a $ticket_{tgt}$ for the next transaction to the ticket granting server [18].

The Process flow of the model is describes as follows-

C → AS

$[ID_C||ID_{tgs}||TS1||Nonce_c]$

AS → C

$[E(K_c,[K_{c,tgs}||ID_{tgs}||TS_2||Lifetime_2||Ticket_{tgs}||Nonce_{as}])]$

$[Ticket_{tgs}= (K_{t\text{-}gs},[K_{c,tgs}||ID_c||AD_c||ID_{tgs}||TS_2||Lifetime_2])]$

The abbreviation used in the model is describes as-

Message (1) Client requests ticket-granting ticket

ID_C-Tells AS identity of user from this client

ID_{tgs}-Tells AS that user requests access to TGS

TS_1-Allows AS to verify that client's clock is synchronized with that of AS

$Nonce_c$-Provide randomness to user

Message (2) AS returns ticket-granting ticket
K_c-Encryption is based on user's password, enabling AS and client to verify password, and protecting contents of message (2) $K_{c,tgs}$-Copy of session key accessible to client created by AS to permit secure exchange between client and TGS without requiring them to share a permanent key.
ID_{tgs}-Confirms that this ticket is for the TGS
TS_2-Informs client of time this ticket was issued
$Lifetime_2$-Informs client of the lifetime of this ticket
$Ticket_{tgs}$ -Ticket to be used by client to access TGS
Nonce $_{as}$-Provide Randomness to the AS

2.2 Ticket granting service exchange to obtain service-granting ticket- [Null=Store previous step's value] [$State_{as}$=? $State_{tgs}$]

In this section the hash value of the previous state of the client and server is taken as a reference and which is shown here as a Null value for the simply to understand. In order to access the service provided by cloud provider the user will send the message and computed hash value to the ticket granting server. After receiving the hash value and message from user the ticket granting server will compute the hash value with the received message and then match with the received hash value if they are equal then server will grant a ticket to client for accessing to the server. Now client will compute hash based on received message and compare with the received hash [18].
The Process flow of the model is describes as follows-

C ⟶ TGS
$[ID_v||Ticket_{tgs}||Authenticator_c||Nonce_C]$
TGS ⟶ C
$[E(K_{c,tgs}, [K_{c,v}||ID_v||TS_4||Ticket_v||Nonce_c])]$
$[Ticket_{tgs}=E(K_{tgs},[K_{c,tgs}||ID_C||AD_C||ID_{tgs}||TS_2||Lifetime_2])]$
$[Ticket_v = E(K_v, [K_{c,v}||ID_C||AD_C||ID_v||TS_4||Lifetime_4])]$
$[Authenticator_c = E(K_{c,tgs}, [ID_C||AD_C||TS_3])]$

The abbreviation used in the model is describes as-
Message (3) Client requests service-granting ticket
ID_V-Tells TGS that user requests access to server V
$Ticket_{tgs}$ -Assures TGS that this user has been authenticated by AS
$Authenticator_c$-Generated by client to validate ticket
$Nonce_c$-Provide randomness to user
Message (4) TGS returns service-granting ticket
$K_{c,tgs}$-Key shared only by C and TGS protects contents of message (4)
$K_{c,v}$-Copy of session key accessible to client created by TGS to permit secure exchange between client and server without requiring them to share a permanent key
ID_v-Confirms that this ticket is for server V
TS_4-Informs client of time this ticket was issued
$Ticket_v$-Ticket to be used by client to access server V
$Ticket_{tgs}$ -Reusable so that user does not have to reenter password
K_{tgs}-Ticket is encrypted with key known only to AS and TGS, to prevent tampering
$K_{c,tgs}$-Copy of session key accessible to TGS used to decrypt authenticator, thereby authenticating ticket
ID_C-Indicates the rightful owner of this ticket
AD_C-Prevents use of ticket from workstation other than one that initially requested the ticket
ID_{tgs}-Assures server that it has decrypted ticket properly
TS_2-Informs TGS of time this ticket was issued
$Lifetime_2$-Prevents replay after ticket has expired
$Authenticator_c$-Assures TGS that the ticket presenter is the same as the client for whom the ticket was issued has very short lifetime to prevent replay
$K_{c,tgs}$-Authenticator is encrypted with key known only to client and TGS, to prevent tamperig
ID_c-Must match ID in ticket to authenticate ticket
AD_c-Must match address in ticket to authenticate ticket
TS_3-Informs TGS of time this authenticator was generated
$Nonce_{tgs}$-Provides Randomness

2.3 Client/Server Authentication Exchange to obtain Service-[$State_{tgs}$=?$State_v$]

In this step we are taking the same as the previous last step state of the ticket granting ticket as the initial step state of the server. In this step client will authenticate to the server with the ticket provided by the ticket granting server. Client will send the authenticator in message and computed hash value to the server in order to access the service. The server will compute the hash value, ticket and authenticator if verified the server will allow to the client to access the services of the server.The Process flow of the model is describes as follows-

C ⟶ V
$[Ticket_v||Authenticator_c||Nonce_c]$
V ⟶ C
$[E(K_{c,v}, [TS_5 + 1]|Nonce_v|)$ (for mutual authentication)$]$
$[Ticket_v = E(K_v, [K_{c,v}||ID_c||AD_c||ID_v||TS_4||Lifetime_4])]$
$[Authenticator_c = E(K_{c,v},[ID_c||AD_C||TS_5])]$

The abbreviation used in the model is describes as-
Message (5) Client requests service
$Ticket_v$-Assures server that this user has been authenticated by AS
$Authenticator_c$-Generated by client to validate ticket
$Nonce_c$-Provides Randomness
Message (6) Optional authentication of server to client
$K_{c,v}$-Assures C that this message is from V
TS_5+1-Assures C that this is not a replay of an old reply
$Ticket_v$-Reusable so that client does not need to request a new ticket from TGS for each access to the same server
K_v-Ticket is encrypted with key known only to TGS and server, to prevent tampering
$K_{c,v}$-Copy of session key accessible to client; used to decrypt authenticator, thereby authenticating ticket

IJSER

ID_C-Indicates the rightful owner of this ticket
AD_c-Prevents use of ticket from workstation other than one that initially requested the ticket
ID_v-Assures server that it has decrypted ticket properly
TS_4-Informs server of time this ticket was issued
$Lifetime_4$-Prevents replay after ticket has expired
$Authenticator_c$-Assures server that the ticket presenter is the same as the client for whom the Ticket was issued; has very short lifetime to prevent replay
$K_{c,v}$-Authenticator is encrypted with key known only to client and server, to prevent tampering
ID_C-Must match ID in ticket to authenticate ticket
AD_c-Must match address in ticket to authenticate ticket
TS_5-Informs server of time this authenticator was generated
$Nonce_v$-Provide randomness

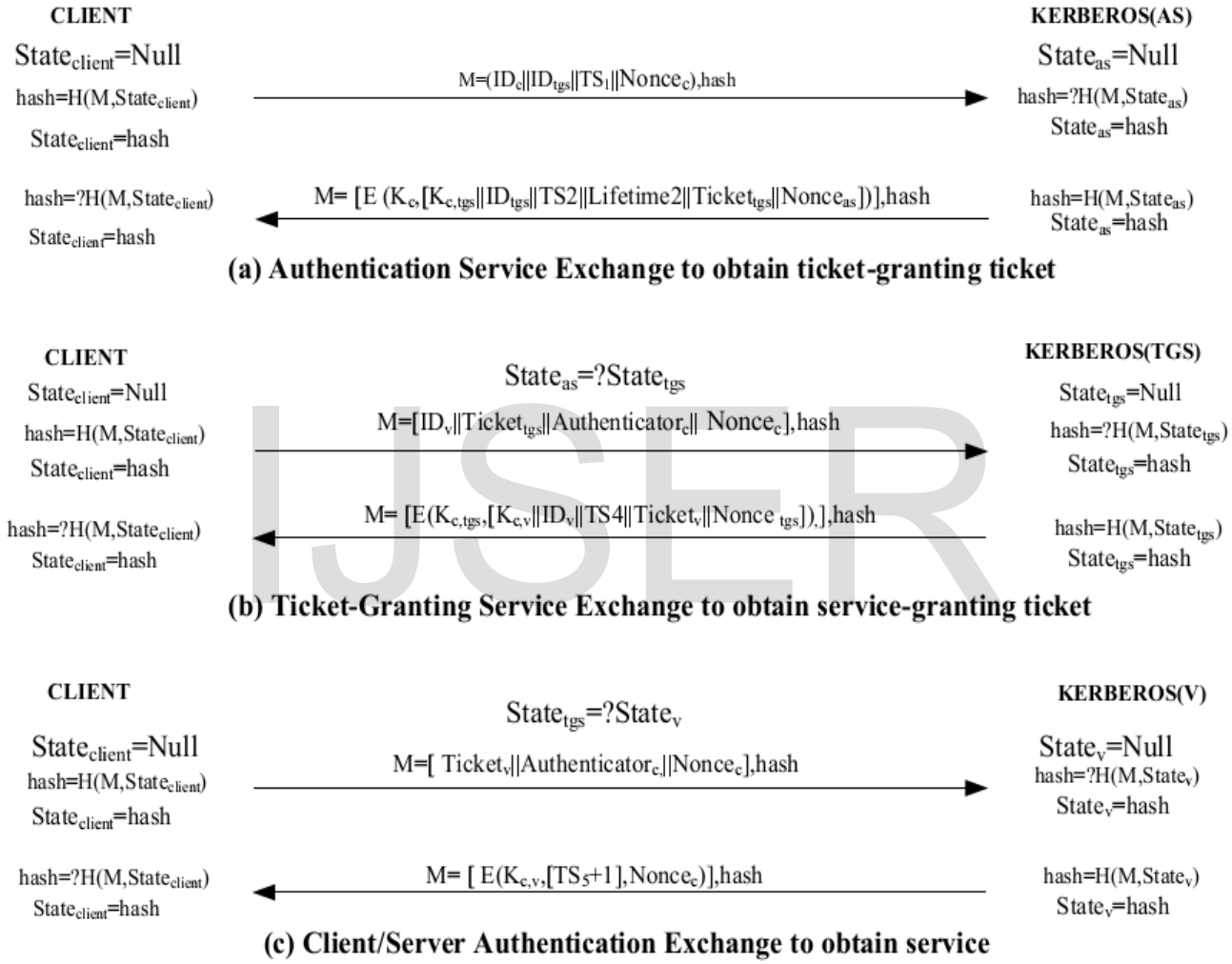

Figure3. Constituents of the Proposed Model

3 State Diagram-

State diagram is working on the basis of the proposed model in this we are trying to explain the step by step procedure that how client moving by Kerberos system to access the desire services provided by the cloud provider in a secure manner. Hear in this diagram client first try to authenticate himself with the authentication system here it is Kerberos by giving his credential and for the randomness to user behavior we are using nonce value so that an opponent will not be able to easily guess the user behavior. We are also sending the hash value computed by the user to the Kerberos. Now again Kerberos will compute and compare the hash value send by the user. In the second step Kerberos send the ticket to the client encrypted by the secret key and also send the computed hash value to the client. Here again client will Compare the hash value if they are equal then it will be accepted by the client.

IJSER

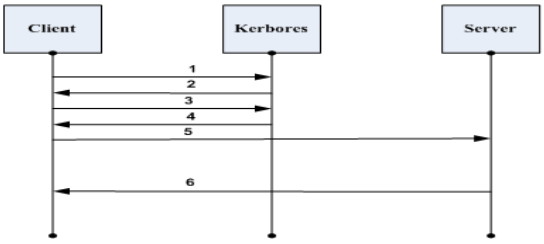

Figure4.Exchanged of secure transaction in Kerberos

After authentication completed in the step third the client will again send this request to the Kerberos for access to the services provided by the server. Again hash chaining will work as it was working in the previous steps of the model. In the fourth step Kerberos will send the ticket for accessing the server along with timestamp value in the encrypted form. Client will receive the hash value and will compare with computed hash value if they are equal it is accepted and valid.in step client will try to access the server by sending the ticket received in step 4 and server will verify it and allow to user for access the services.

4 Features Provided by Proposed model-

This table provides the services provided by the proposed model we can achieve more security by applying this model.

Features	Purposed Model
Confidentiality	Yes
Message Authentication	Yes
Client Authentication	Yes
Server Authentication	Yes
MITM Attack	Yes
Replay Attack	Yes

Table1. Features of Proposed Model

5 Analysis & Discussion-

In this proposed model user will be authenticated through out at the each step of the transaction. By using this proposed model user will access the services securely which is provided by the server. At each step client and server will verify the hash value based on their current state. If message in between the transaction is modified by any attacker this can be easily traced by the user or by the server based on their hash value. We also claim that the user will access the data from the server in a secure manner without any tempering. The confidentiality of the data which is in transaction is maintained through the use of strong symmetric encryption key. Message authentication is also performed because only the authenticated user will be able to open the message which is encrypted by the key. This proposed model also provides the protection against the type MITM (Man-In-The-Middle) attack because we are using the Nonce value for each transaction. Also provides the protection against the Replay attack because of the timestamp.

6 Conclusions and Future Work-

In this paper we first discuss the working of Kerberos protocol and then address the security related issues involved with the transactions in cloud computing. And then we have proposed the model using modified version of the Kerberos protocol where we tried to cover all these security issues which is addressed in this paper. The creation of hash value involved the message chaining and the state variables are stop to performing replay attack. And the nonce value is providing the protection against the MITM attack. In future, we can also authenticate the user in place of user entity and after that we will try to solve the problem of key exchange with the Asymmetric cryptography.

References-

[1] "Baburajan, Rajani, "The Rising Cloud Storage Market Opportunity Strengthens Vendors," infoTECH, August 24, 2011".It.tmcnet.com. 2011-08-24.Retrieved 2011-12-02.

[2] Oestreich, Ken, (2010-11-15). "Converged Infrastructure".CTO Forum. Thectoforum.com. Retrieved 2011-12-02.

[3] "The NIST Definition of Cloud Computing". National Institute of Standards and Technology. Retrieved 24 July 2011.

[4] Security and Privacy Issues in Cloud Computing JaydipSen Innovation Labs, Tata Consultancy Services Ltd., Kolkata, INDIA.

[5] "Security Guidance for Critical Areas of Focus in Cloud Computing". Cloud Security Alliance.2011.Retrieved2011-05-04.

[6] John T. Kohl and B. Clifford Neuman, ''the Kerberos Network Authentication Service Revision 5, Project Athena, Massachusetts Institute of Technology (April 1992).

[7] S. M. Bellovin and M. Merritt, ''Limitations of the Kerberos Authentication System,''*Computer Communications Review* 20(5), pp. 119-132 (October 1990).

[8] John T. Kohl, B Neuman, YTs'o," The Evolution of the *Kerberos* Authentication Service" , Spring EurOpen Conference in Tromso, Norway, 1991.

[9] John T. Kohl ''The Use of Encryption in Kerberos for Network Authentication,'' in Crypto '89 Conference Proceedings, International Association for Cryptologic Research, Santa Barbara, CA (August 1989)

[10] S. P. Miller, B. C. Neuman, J. I. Schiller, and J. H. Saltzer, Section E.2.1: Kerberos Authentication and Authorization System, M.I.T. Project Athena, Cambridge, Massachusetts (December 21 1987).

[11] A.R. Butt, A. Sumalatha, N.H. Kapadia, Grid computing portals and security issues, Journal of Parallel and Distributed Computing 63 (10) (2003) 1006–1014..

[12] H. Cassanova, Distributed computing research issues in grid computing, ACM SIGACT News 33 (3) (2002) 50–70.

[13] B. Jacob, Grid computing—what are the key components?http://www-106.ibm.com/developerworks/grid/library/gr-overview/?ca=dgr-lnxw09Gri, Document view: March 28 2006.

[14] B. Jacob, How grid infrastructure affects application design, http://www106.ibm.com/developerworks/library/grinfra.html,(Document view: March 28 2006).

[15] K. Klaus, R. Buyya, M.Maheshwaren, A taxonomy and survey of grid resource management systems for distributed computing, Software Practice and Experience, 00:1–7, 2001, Effectively, Future Generation Computer Systems 9 (4) (May 2003) 563–573.

[16] F. Kon, M. Roman, P. Liu, Monitoring, security and dynamic configuration with the dynamic TAO reflective ORB, IFIP/ACM International Conference on Distributed Systems Platforms,2000, pp.

121–143, New York, United States.
[17] Peer to Peer and Grid: Synergies and Opportunities, Workshop Notes,http://www-csag.ucsd.edu/P2P-Grid/P2P-Grid-Workshop10-7-2003.pdf (Document view: March 28, 2006).
[18] William Stalling, Cryptography and Network Security, Pearson Publication, Singapore, 2004.
[19] http://www.ietf.org/rfc/rfc4120.txt

IJSER