Asian Journal of Computer Science and Technology
ISSN: 2249-0701 (P) Vol.1 No.1, 2012, pp.6-10

DOI: https://doi.org/10.51983/ajcst-2012.1.1.1697

A Formal Analysis and Some Properties of Kerberos in Network Security

R.Aghila, K.Nagalakshmi, B.S.Roopan, S.Rajakrishnan and T.Anandaraj

Department of Computer Science and Engineering,

KLN College of Information and Technology, Sivagangai District, India.

E-mail: laxmirengaraj@gmail.com, raja1810tamil@gmail.com

***Abstract* - Network is a chain, a weak link in that can break a system. Network plays a vital role in modern computer industry. Making secure network is more important because a lot of confidential transactions are performed over the network. In this world of universal electronic connectivity "Hackers" threaten the prosperity and productivity of individuals and corporations, to prevent that we introduce a new type of security with KERBEROS. This paper deals with networking in a most secured manner. In this advanced Hacking world network security seems to be more important and it plays a vital role in communication over networks. This is made possible by using the KERBEROS, the best way of implementing the network security.**

1. Introduction

The generic name for the collection of tools designed to protect data and to thwart hackers is known as Computer Security. The measures that are needed to protect data during their transmission and to guarantee that transmissions are authentic is Network Security.

The key aspects to securing communications over a distributed environment are authentication, integrity, confidentiality and authorization [1]. Kerberos is a network protocol that addresses the authentication part. This application aims at using Kerberos V5 to secure the communication between a J2Me MIDlet communicating over the GPRS, and a Banking Transaction Server. This project mainly covers at gaining the initial ticket provided by the authentication server. Authentication server checks its database to authenticate the user. If the username and the password matches it generates and initial ticket by which we can obtain many service tickets to access the services provided by the banking transaction server.In [9], This paper is a survey on how the Kerberos V5 can be transported over TLS protocol in order to provide additional security features. This document describes an extensibility mechanism for Kerberos V5 protocol which improves the security in TLS protocol.

The rest of the paper is organized as follows. Chapter 2 discusses about the need for security, Chapter 3 highlights the various Technologies involved in cryptography. Chapter 4 highlights the Working of Kerberos key distribution. The Paper ends in Chapter 5 highlights the Working of sub protocols and Chapter 6 with the Conclusion and the proposed work.

II. Need for Security

Security and privacy are important because many people in many places have access to a computer network. The information stored in some of the network machine may be of great valves to a corporation. It must not to be lost, stolen or damaged. It is important to protect the data and programs from hardware and software failure, from catastrophes and from criminals, vandals, incompetents and people who would misuse it. Users for whom security is of little or no importance and users for whom it is vital often share a network. Network is prone to attacks such as

1) Interception : An unauthorized party gains access to asset. This is an attack on confidentially. The unauthorized party could be a person, program or a computer.

Examples: wire taping to capture data in network and the illicit copying of files or programs.

2) Modification : An unauthorized party does not gain access but tampers with an asset. This is an attack on integrity.

Examples: Changing values in a data file, altering programs so that it performs differently and modifying the content off messages being transmitted in a network.

3) Fabrification : An unauthorized party inserts counterfeit objects into the system. This is an attack on authenticity.

Examples: the insertion of spurious messages in a network or the addition of records to a file.

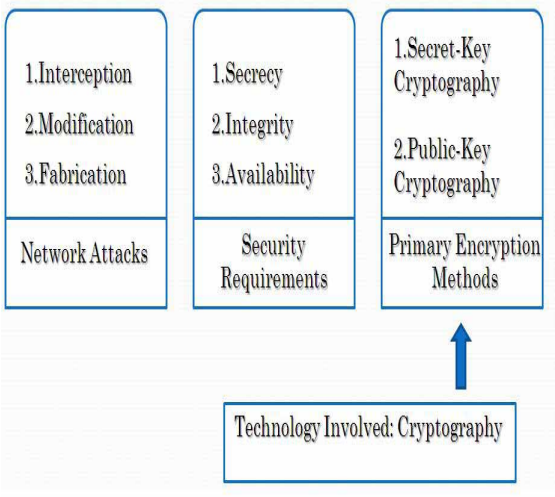

Fig.1 Need for security

A. Security Requirements

1) Secrecy: Requires that the information in a computer system only be accessible for reading by authorized parties. This type of access includes printing, displaying and other forms of disclosure including simply revealing the existence of an object.

2) Integrity: Requires that only authorized parties can modify the computer system assets. Modification includes writing, changing status, deleting and creating.

3) Availabilty: Requires that the computer system assets are available to authorized parties.

III. Technology Involved in Cryptography

Cryptography is the science of using mathematics to encrypt and decrypt data. Cryptography enables you to store sensitive information or transmit it across insecure networks (like the internet) so that it cannot be read by anyone expect the intended recipient.

A. Encryption and Decryption

The method of disguising plain text in such a way as to hide its substance is called encryption. Encryption ensures that information is hidden from anyone for whom it is not intended, even those who can see the encrypted data. The process of reverting cipher text to its original plaintext is called decryption.

A cryptography algorithm works in combination with a key word, number or phrase to encrypt the plain text, which is then decrypted using the same key to produce original message. There are two primary encryption methods. They are as follows.

B. Secret - Key Cryptography

Secret-key cryptography, also known as symmetric cryptography uses the same to encrypt and decrypt the message. Therefore, the sender and recipient of a message must share a secret, namely the key. A well secret-key cryptography algorithm is the data encryption standard (DES), which is used by financial institutions to encrypt PIN (personal identification numbers)

Disadvantages: Secret-key cryptography is impractical for exchanging messages with a large group of previously unknown corresponding over a public networks. For a merchant to conduct transactions with millions of Internet subscribers, each transmitted over a separate secure channel.

Because conventional cryptography was once the only available means for relaying secret information, the expense of secure channels and key distribution relegated its use only to those who could afford it such as government and large banks.

C. Public-Key Cryptography

Public-key cryptography also known as asymmetric cryptography uses two keys: one key to encrypt the message and the other key to decrypt the message. The two keys are mathematically related so that data encrypted with either key can only be decrypted using the other. Each user has two keys: a public key and a private key. The user distributes the public key. Because of the relationship between the two keys, the user and anyone receiving the public key can be assured that the data encrypted with the public key and send to the user can only decrypted when the user uses the private key. This assurance is maintained if the user ensures that the private key is not disclosed to anyone else. Therefore the user should generate the key pair. The user should generate the best-known public-key pair. The best-known public-key cryptography algorithm is RSA (Rivest-Shamir-Adalman). The primary benefit of public-key cryptography is that it allows people who have no pre-existing security arrangement to exchange messages securely. The need for sender and receiver to share secret key via some secure channels is eliminated; all communication needed only public keys and a no private key is transmitted or shared. Cryptography strength is measured in the time and resources it would require to recover the plain text.

D. Key Authentication

Many forms of authentication are based on the idea that an entity can prove its identity if it can prove it knows a key, such as a password, that only it can know. Authentication techniques that rely on a secret, such as a password, need to have a way to keep the secret from becoming public knowledge. A password owner cannot walk up to a door and give the password. Some one besides the doorkeeper might be listening; or it might be the wrong door. In order to keep the secret, there must way to prove a user knows the password without revealing the password. That is the idea behind secret key authentication, a method of verification used throughout the KERBEROSPROTOCOL.

In the Kerberos protocol model, client/server connection begins with the authentication. Client and server, in turn, step through a sequence of actions designed to verify to the party on each end of the connection that the party on the other end is genuine. If authentications is successful, session setup completes and a secure client/server session is established. The Kerberos protocol makes use of key authentication; authenticator messages, the Kerberos key distribution center, session tickets, and ticket granting tickets to provide a secure session.

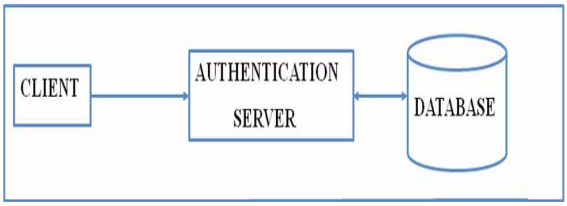

Fig. 2 Key Authentication

IV. Working of Kerberos Key Distribution

A client will want to communicate with many servers and will need different keys for each of them. A server communicates with many clients and needs different keys for each of them, as well. If each client needs a different key for every server, and each server needs a different key for every client, key distribution becomes a problem. In addition, the need to store and protect many keys on many computers creates an enormous security risk.

The name of the Kerberos protocol suggests its solution to the problem of key distribution. Kerberos (or Cerberus) was a figure in classical Greek mythology- a fierce, three-headed dog who kept living intruders from entering the under world. Like the mythical guard, the Kerberos protocol has three heads: a client, a server, and a trusted intermediary in this protocol are the Key Distribution Center (KDC).

The KDC is the service running on a physically secure server. It maintains a database with account information for all security principals in its realm. A realm is the Kerberos equivalent of domain in Microsoft windows. Along the other information each security principal, the KDC stores a cryptographic key known only to the principal of the KDC. This is the master key used in exchanges between each security and the KDC.

In most implementations of the Kerberos protocol, this master key is derived using a hash function from a security principal's password. When a client wants to create a secure connection with a server, the client begins by sending a request to the KDC, not to the server that it wants to reach. The KDC creates and sends to the client a unique session key for the client and a server to use to authenticate each other. The KDC has access to both the client's master key and the server's master key. The KDC encrypts the server's copy of the session key using the server's master key, and the client's copy using the client's master key.

The KDC could fulfill its role as trusted intermediary by sending the session key directly to each of the security principals involved, but in practice this procedure will not work, for a number of reasons. Instead, the KDC sends both encrypted session keys to the client. The session key for the server is included in a session ticket.

A. Session Tickets

Instead of sending the encrypted session keys to both of the principals, the KDC sends both the client's and the server's copies of the session key to the client. The client's copy of the session key is encrypted with the client's master key and therefore cannot be decrypted by any other entity. The server's copy of the session key is embedded, along with authorization data about the client, in a data structure called a ticket. The ticket is entirely encrypted or other entity that does not have access to the server's master key. It is the responsibility of the client to store the ticket safely until contact with the server. The KDC only provides ticket-granting service.

The client and server are responsible for keeping their respective master keys secure. When the client receives the kdc's response, it extracts the ticket with its own copy of the session key, putting both aside in a secure cache. To establish a secure session with the secure, it sends the server a message consisting of the ticket, still encrypted with the server's master key, and an authenticator message encrypted with the session key. Together, the ticket and authenticator message are the client's credentials to the server. When the server receives credentials from a client, it decrypts the ticket with its master keys, extracts the session key, and uses the session key to decrypt the client's authenticator message. If everything check's out, the server knows that the client's credentials were issued by the KDC, a trusted authority. For mutual authentication, the server responds by encrypting the timestamp from the client's authenticator message using the session key. This client then decrypts the message. If the returned message is the same as the timestamp in the original authenticator message, the server is authenticated. As an added benefit, the server does not need to store the session keys it users with its client's. It is client's responsibility to manage the ticket for the server in its ticket cache and to present that ticket each time it accesses the server. Whenever the server receives a ticket from a client, it uses it accesses the server. Whenever the server receives a ticket from a client, it uses its master key to decrypt the ticket and extract the session key.

When the server no longer needs the session key, the key is purged. The client does not need to access the KDC each time it wants access to this particular server. Tickets can be reused. As a precaution against the possibility of ticket theft, tickets have an expiration time, specified by the KDC in the ticket structure. How long a ticket is valid depends on kerberos policy for the realm.

Typically, tickets are good for no longer than eight hours about length of a normal logon session. When the user at a client workstation logs off, the client ticket cache is flushed and all tickets and clients session keys are destroyed.

B. Key Distribution Center

Windows 2000 implements the key Distribution Centre (KDC) as a domain service. It uses the Active Directory as its account database and the Global Catalog for directing referrals to KDC's in other domains. When a user logs on, the client requests a ticket for the KDC just as it would request a ticket for any other service. The KDC responds by creating a logon session key and a ticket for special server, the KDC's full ticket granting service.

One copy of the logon session key is embedded in the ticket, and the ticket is encrypted with the KDC's master key. Another copy of the logon session key is encrypted with the master

key derived from the user's logon password. Both the ticket and encrypted session key are sending to the client. When the client gets the KDC's reply, it decrypts the logon session key with the user's logon password. The client no longer needs the key derived from the user's password because the client will now use the logon session key to decrypt its ticket cache along with its ticket for the KDC's full ticket granting service. The ticket for the full Ticket-granting service is called a Ticket-granting Ticket (TGT).

Table 1 Kerberos of Sub protocol

SUBPROTOCOL	MEANING
Authentic service (AS) exchange	In this sub protocol, the KDC gives the client a logon session key and a TGT.
Ticket-granting service (TGS) exchange	In this sub protocol, the KDC distributes a service session key and a ticket for the service
Client/service (CS) exchange	In this sub protocol, the client presents the ticket for admission to a service

V. Working of Sub Protocols

A. Authentication Service (As) Exchange

The user begins logging on to the network by typing a logon name and password. The Kerberos client on the user's workstation converts the password to an encryption key and saves the result in a program variable. The client then requests credentials for the KDC's ticket-granting service by sending the KDC's authentication service a message of type KRB_AS_REQ (Kerberos Authentication Service Request). The first part of these message identifiers the user and the TGS service being requested. The second part of the message contains pre-authentication data intended to prove that user knows the password. This is simply an authenticator message encrypted with the master key derived from the user's login password. When the KDC receives KRB_AS_REQ, it looks up the user in its database, gets the associated user's master key, decrypts the pre-authentication data, and evaluates the timestamp inside. If the timestamp is valid the KDC can be assured that the pre-authentication data was encrypted with the user's master key and thus that the client is genuine. Once it has verified the user's identity, the KDC creates credentials that the client can present to the ticket-granting service. It invents a logon session key and encrypts a copy with the user's master key. It embeds another copy of logon session key and the user authorization data in a TGT, and encrypts the TGT with the KDC's own master key.

The KDC sends these credentials back to the client by replying with a message o type KRB_AS_REP (Kerberos Authentication Service Reply). When the client receives the reply, it uses the key derived from the user's password to decrypt the new logon session key. The client stores the new in its ticket cache. The client extracts the TGT from the message and stored that in its cache as well.

B. Ticket Granting Service (Tgs) Exchange

To request a ticket for another service from the KDC's the following process is used. The Kerberos client on the user's workstation requests credentials for the service by sending a message or type KRB_TGS_REQ (Kerberos Ticket Granting Service Request). This message consists of the identity of the service for which the client is requesting credentials, an authenticator message encrypted with the user's new logon session, and the TGT obtained from the Authentication Service (AS) exchange. When it receives a KRB_TGS_REQ, the KDC Decrypt the TGT with its secret key and extras the user' logon session key. The KDC users the logon session key to decrypt the user's authenticator message and evaluate that, if the authenticator passes the test, the KDS extracts the user's authorization data from the TGT and invents a service session key for the user to share with the desired server.

The KDC encrypt one copy of the service session key with the user's logon session key. The KDC embeds another copy of the service session key in a ticket, along with the user's authorization data and encrypt the tickets with the server's master key. The KDC sends these credentials back to the client by replying with a message of type KRB_TGS_REP (Kerberos Ticket-Granting Service Reply). When the client receives the reply, it decrypts the service session key with the user's logon session key, and stores the service session key in its ticket cache. The client extracts the ticket to the server and stores that in its tickets cache.

C. Client/Server (Cs) Exchange

One a user has a ticket to a server. The workstation client can establish a secure communications session with that server. This is done using the following steps. The client sends the server a message of type KRB_AP_REQ (Kerberos application request). This message contains an authenticator message encrypted with the key sent by the KDC for the session the server. The ticket for sessions with the server, and a flag indicating whether the client requests mutual authentications. Setting of the flag requesting mutual authentication is options of the options in configuring Kerberos. The user is never asked whether mutual authentication should be used or not. The server receives KRB_AP_REQ, decrypts the ticket, and extracts user's authentication data and the session key. The server uses the session key from ticket to decrypts the user's authenticator message and evaluates the timestamp inside.

If the authenticator message is valid, the server checks the mutual authentication flag in the client's request. If the mutual authentication flag is set, the server uses the session key to encrypt the time from the user's authenticator message and return the results in a message of type KRB_AP_REP (Kerberos Application Reply). When the client receives KRB_AP_REP, it decrypts the server's authenticator message with the session key it shares with the server, and compares the time sent back by the service with the time in its original

authenticator message. If they match, the client is assured that the service is genuine and the connection proceeds.

VI. Conclusion

Kerberos proves to be much efficient when compared to other network security methods. It provide a better level of security. Kerberos, the strongest encryption obtainable today will hold up tomorrow's the more computing power.

References

[1] Jaiganesh, M.; Ramdoss, B.; P.S.N.A. Coll. of Eng. & Technol., Dindigul Computing, Communication and Networking, 2008. ICCCn 2008. International Conference on 18-20 Dec. 2008 ,1 - 7 ,St. Thomas, VI ,ISBN: 978-1-4244-3594-4 ,24 February 2009.

[2] Raluca, C.; Monica, B.; Electronics and Telecommunications (ISETC), 2010 9th International Symposium on 11-12 Nov. 2010 ,237 - 240 ,Timisoara ,ISBN: 978-1-4244-8457-7 ,06 January 2011

[3] S. M. Bellovin and M. Merritt. Limitations of the Kerberos Authentication System. In *Proceedings of the Winter 1991 Usenix Conference*. January 1991.

[4] B. Clifford Neuman and Stuart G. Stubblebine. A Note on the Use of Timestamps as Nonces. *Operating Systems Review*, 27(2):10-14, April 1993. (unrefereed)

[5] T. Wu, "A Real-World Analysis of Kerberos Password Security", in Proceedings of the 1999 Internet Society Network and Distributed System Security Symposium, San Diego, CA, Feb 1999. Pdf. presentation.

[6] K. Hildrum (UC Berkeley, UNITED STATES), Security of Encrypted rlogin Connections Created With Kerberos IV [Paper], [Overview], NDSS 2000.

[7] S. Josefsson. On Active Attacks to Kerberos Telnet, August 2001, unpublished.

[8] F. Butler, I. Cervesato, A. Jaggard, A. Scedrov. In: S. Schneider, ed.,A formal analysis of some properties of Kerberos 5 using MSR, , "15-th IEEE Computer Security Foundations Workshop, Cape Breton, Nova Scotia, Canada, June, 2002", *IEEE Computer Society Press*, 2002, pp. 175-190.

[9] Raluca, C.; Monica, B.; Electronics and Telecommunications (ISETC), 2010 9th International Symposium on 11-12 Nov. 2010 ,237 - 240 ,Timisoara ,ISBN: 978-1-4244-8457-7 ,06 January 2011