Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 4, April 2013, pg.414 – 421

RESEARCH ARTICLE

Remote System Controller: An Implementation of a Secure Channel

Shanu Sharma[1]

[1]Assistant Professor, Computer Science & Engineering Department, Amity School of Engineering & Technology, Amity University, Noida, Uttar Pradesh, India

[1] *shanu.sharma1611@gmail.com*

Abstract— Now, a day's Security over a Network is a must. Everyone wants to share files with others like Facebook, Pinterest. But are they aware whether the network on which they are Sharing files is Secure? So to help these kind of People who are not aware of what's going on in Background there is a Kerberos cum Remote System Controller (RSC) System. Kerberos as a name looks like a Network Protocol, just for authentication purpose. This system allows a person to send, transfer, share any file or document over a particular Network in a Secured way. So that no one reveal the type of transaction going on between the two parties. The proposed System works on the Client Server architecture model, which says the Client will have to prove its Identity to the server before accessing anything. The objective of this System cum Controller was just to make users aware of what's happening on back-end of which they are Un-aware of. The Controller is a "window based application" to be implemented by a client or the Sender. The use of mediator is mainly just to reduce the Resources Utilization Power. The System will work based on Secured client's password, Internet Protocol address and name of the service to be used. The Concept will rotate whole about Key Distribution Centre who will use DES i.e. Data Encryption Standard Algorithm for Transmission. RSC is using LSB i.e. Least Significant Bit Algorithm for Cryptography and Steganography. So, any client can send and receive files which can be Audio, Text and Video along with Data file containing Hidden text messages. All the Functionality is under a Single Page Cum Software, so that clients have the choice what Receiver and Sender wants to do.

Key Terms: - Window based Application; GUI; Kerberos protocol; Java; Cryptography and Steganography

I. INTRODUCTION

Usually people make an overview of the other person by observing their physical appearance, voice patterns; personality etc., these few factors gives a chance to distinguish one person from another person. But during conversation on network people cannot identify the other person on the basis of said factors. So to maintain security over network there should be some key or password to judge the authenticity of other person [1].

Kerberos is a computer network authentication protocol which allows nodes communicating over a non-secure network to prove their identity to one another in a secure manner. It aims primarily at a client–server model, and provides mutual authentication to both the user and the server for verifying each other's identity. It requires a trusted third party, and use public-key cryptography by utilizing asymmetric key cryptography during certain phases of authentication. It makes use of a trusted third party, termed a key distribution center (KDC), which consists of an Authentication Server (AS) and a Ticket Granting Server (TGS)[1]. Kerberos works on the tickets sharing process which will enhance the security over network. KDC maintains a database which consists of shared secret keys of both client and server. For transmission between two entities, KDC generates a session key which is then provided to the client by the server [2].

Kerberos runs in the background means the user on, whose PC this software is running, is totally unaware of what is happening in his/ her computer system. But the information to the user about the activity of providing the network authentication and understanding of functions of Kerberos is too important for the totally secured channel. This paper presents the development of software which shows the various processes that occur in different phases of network authentication.

During sharing of a file on network no one is aware of what's going on in a particular file sharing network. They just see the file transmitting without even knowing that some-one is accessing their Secret file, which is a very serious problem for any user. The proposed system has a option of Hiding text/Audio and Video in any of the Video/Audio and Text, i.e. a client will have complete freedom to do anything, with that file. It uses both Asymmetric key cryptography and Symmetric key cryptography to achieve three goals: Security, Integrity and Availability [3].

The motivation of this paper is to present the working of a secured channel which works like a software having 3Modules: Steganography in Image, Audio, Video along with Cryptography with Text and the third module shows the process of network Authentication. The proposed system is a windows based application which helps a Lay-man to use the Software for Security. The system is used to enhance the resources in a manner that very less resources of server should be used.

II. REVIEW OF TOOLS AND RELATED WORK

The literature covers everything about Kerberos, Secured Network Algorithms like DES and LSB.

A. Kerberos

Kerberos is a network authentication protocol which basically works on sharing of tickets among senders and receivers, the working of this protocol is shown in Fig. 1. The client authenticates itself to the Authentication Server and receives a ticket (All tickets are time-stamped). It then contacts the Ticket Granting Server, and using the ticket it demonstrates its identity and asks for a service. If the client is eligible for the service, then the Ticket Granting Server sends another ticket to the client. The client then contacts the Service Server, and using this ticket it proves that it has been approved to receive the service[1][5].

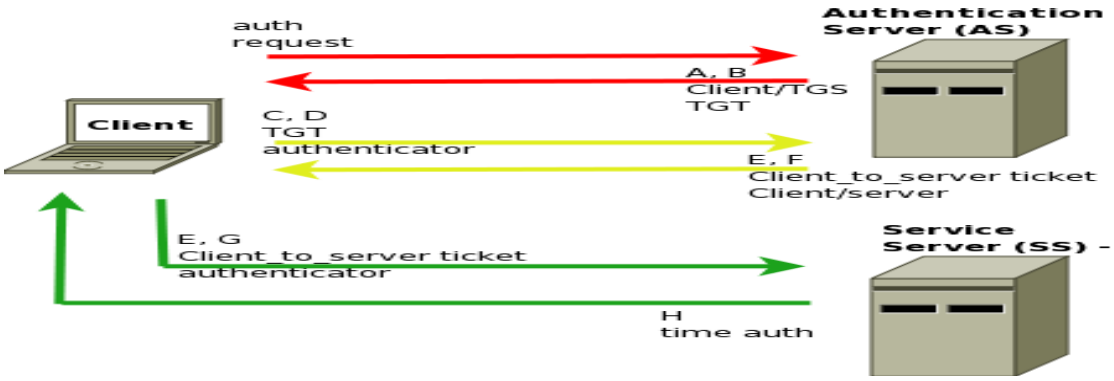


Fig 1. Kerberos Negotiation

The client authenticates to the Authentication Server (AS) once using a long-term shared secret (e.g. a password) and receives a Ticket-Granting Ticket (TGT) from the AS. Later, when the client wants to contact some Service Server (SS) , it can (re)use this ticket to get additional tickets from Ticket-Granting Server (TGS), for SS, without resorting to using the shared secret. These tickets can be used to prove authentication to SS.

B. Data Encryption Standard Algorithm

Data Encryption Standard (DES) is the block cipher algorithm that takes a fixed-length string of plaintext bits and converts it into same length Cipher text using 64-Bits Key. In the case of DES, the block size is 64 bits. DES has 16 rounds of procedure, meaning the main algorithm i.e. DES is repeated 16 times to produce the cipher text or Encrypted text. Brute force attack is not so easy because Exponential power increases each time [4][8].

1) Key Scheduling: Ever time, the 56 bits keys is rotated or moved by 1 or 2 bits. 1, 2, 6, 9, 15.. Bits are moved by 1 bit and the rest bits by 2 bits. This will result in 16 new keys, which will be used for Encryption.

DES uses a 64 bit key. The multiple of 8 bit i.e. 8.16.24.32… are removed from text to make the text 56 bits. Now after this, 16 rounds come into place which consists of:

The core functions of DES algorithm are [4]:

Expansion: The 32-bit half-block is expanded to 48 bits using the expansion permutation, denoted E in the diagram, by duplicating half of the bits[8].

Key mixing: The result is combined with a sub-key using an XOR operation. Sixteen 48-bit sub-keys, one for each round, are derived from the main key using the key scheduling process.

Substitution: After mixing in the sub-key, the block is divided into eight 6-bit pieces before processing by the S-boxes, or substitution boxes. These S.boxes are secured and made by IBM after long working for many years.

Permutation: Finally, the 32 outputs from the S-boxes are rearranged according to a fixed permutation, the P-box. The whole process is shown in Fig 2.

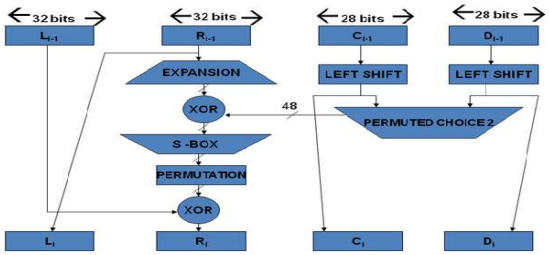


Fig 2. Structure of DES

2. Ciphertext Preparation: The final step is to apply the permutation IP^(-1) to the pre-output. The result is the completely encrypted cipher-text [12].

C. Standard LSB Algorithm

Data hiding in the least significant bits (LSBs) of audio samples in the time domain is one of the simplest algorithms with very high data rate of additional information. In Layman language, it is an algorithm which works on Bits. It retrieves the image Bit form, on which the user or the developer works. Suppose for example, if the bit form for an Image/Audio or Video is in Matrix form, then the Message which we want to hide should also be in matrix bit form and by using some pattern Either Row wise or Column wise. We can embed that message. This is how the Encryption is done. It has very less effect on the Message. However, the hidden message can-not be decrypted without knowing the secret Key/Password. This is how LSB is used in Remote System Controller [12].

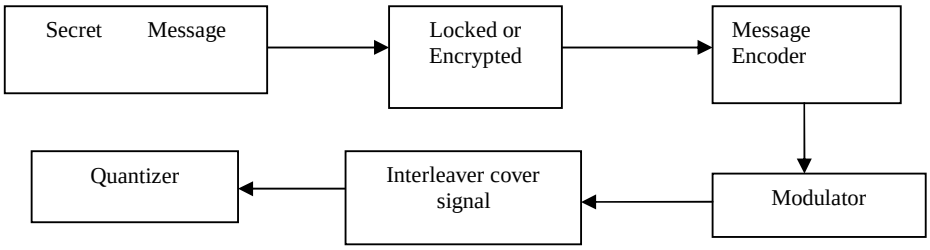


Fig 3. LSB Algorithm

III. DESIGN AND IMPLEMENTATION

A. Objective of the Proposed System

The current system of Kerberos deals with the problem of running in the backend that means the user on, whose PC this software is running is totally unaware of what is happening in his/ her computer system. Therefore, it is really important that the layman should be informed about this activity of providing network

authentication along with understanding of its function. This paper presents the application which shows the improvement of the current software. This application ensures that the time of the server is not wasted by unauthorised access henceforth, exploit the security.

B. System Requirements and Implementation

1). Software Requirements

- Jdk 1.6
- Ms-Access as backend
- Windows Vista or its family

2). Hardware Requirements

- Core 2 Duo Processor
- 1 or more GB for data storage
- 128 -512 KB RAM
- Printer for reporting and printing

C. Working of System

1) User Client Based Logon

- A user enters a username on the client machine.
- The client performs a one-way function on the entered password, and this becomes the secret key of the client/user.

2) Client Authentication

- The client sends a cleartext message of the user ID to the AS requesting services on behalf of the user. (Note: Neither the secret key nor the password is sent to the AS.) The AS generates the secret key by hashing the password of the user found at the database .

- The AS checks to see if the client is in its database. If it is, the AS sends back the following two messages to the client[6]:

Message A: Client/TGS Session Key encrypted using the secret key of the client/user.

Message B: Ticket-to Get-Ticket (which includes the client ID, client network address, ticket validity period, and the client/TGS session key) encrypted using the secret key of the TGS.

- Once the client receives messages A and B, it attempts to decrypt message A with the secret key generated from the password entered by the user. If the user entered password does not match the password in the AS database, the client's secret key will be different and thus unable to decrypt message A. With a valid password and secret key the client decrypts message A to obtain the Client/TGS Session Key. This session key is used for further communications with the TGS. (Note: The client cannot decrypt Message B, as it is encrypted using TGS's secret key.) At this point, the client has enough information to authenticate itself to the TGS[7].

3). Client Service Authorization

- When requesting services, the client sends the following two messages to the TGS[10][11]:

Message C: Composed of the TGT from message B and the ID of the requested service.

Message D: Authenticator (which is composed of the client ID and the timestamp), encrypted using the Client/TGS Session Key.

- Upon receiving messages C and D, the TGS retrieves message B out of message C. It decrypts message B using the TGS secret key. This gives it the "client/TGS session key". Using this key, the TGS decrypts message D (Authenticator) and sends the following two messages to the client:

Message E: Client-to-server ticket (which includes the client ID, client network address, validity period and Client/Server Session Key) encrypted using the service's secret key.

Message F: Client/server session key encrypted with the Client/TGS Session Key.

4). Client Service Request

- Upon receiving messages E and F from TGS, the client has enough information to authenticate itself to the SS. The client connects to the SS and sends the following two messages[10][11]:

Message E from the previous step (the client-to-server ticket, encrypted using service's secret key).
Message G: a new Authenticator, which includes the client ID, timestamp and is encrypted using client/server session key.

- The SS decrypts the ticket using its own secret key to retrieve the Client/Server Session Key. Using the sessions key, SS decrypts the Authenticator and sends the following message to the client to confirm its true identity and willingness to serve the client:

Message H: the timestamp found in client's Authenticator plus 1, encrypted using the Client/Server Session Key.

- The client decrypts the confirmation using the Client/Server Session Key and checks whether the timestamp is correctly updated. If so, then the client can trust the server and can start issuing service requests to the server.

- The server provides the requested services to the client.

5). Class Diagram: In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or) methods and the relationships between the classes.
In the class diagram these classes are represented with boxes which contain three parts:

- The upper part holds the name of the class
- The middle part contains the attributes of the class
- The bottom part gives the methods or operations the class can take or undertake

In the system design of a system, a number of classes are identified and grouped together in a class diagram which helps to determine the static relations between those objects. With detailed modeling, the classes of the conceptual design are often split into a number of subclasses.
In order to further describe the behavior of systems, these class diagrams can be complemented by state diagram or UML state machine. Also instead of class diagrams Object role modeling can be used if you just want to model the classes and their relationships.

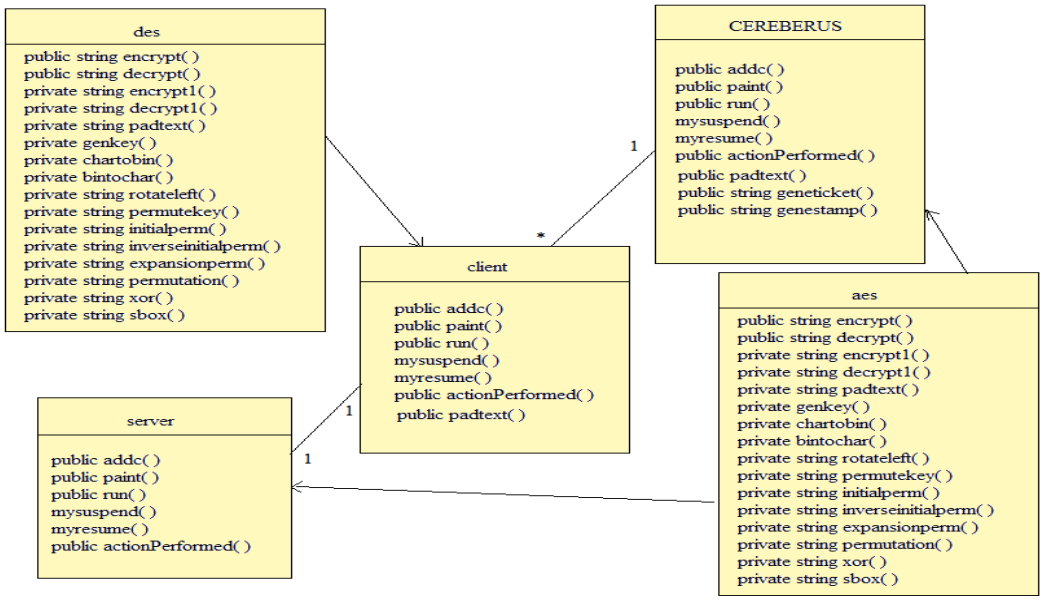


Fig 4. Class Diagram

6). Database Design

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make database access easy, quick, inexpensive and flexible for the user. Relationships are established between the data items and unnecessary data items are removed. Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability. This ensures minimizing data storage required, minimizing chances of data inconsistencies and optimizing for updates. The MS Access database has been chosen for developing the relevant databases using Java[9]. The sample of database of user name and password is shown in Table 1.

Table 1. Database table of usernames and passwords

info1		
ID	**username**	**password**
1	PANKAJ	PANKAJ
2	ANKIT	ANKIT
3	BHARAT	BHARAT

IV. RESULTS AND APPLICATION INTERFACE

First three command prompts have to be opened to set the path of the file to run the server, client and the Kerebros, by which GUI based window will open, out of which enter the username of client in the client window as shown in Fig 5.

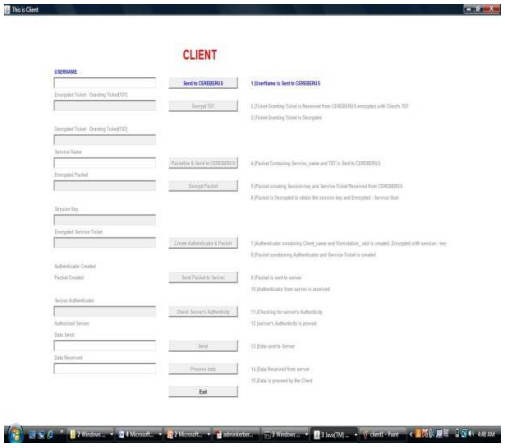


Fig 5. GUI window for Client

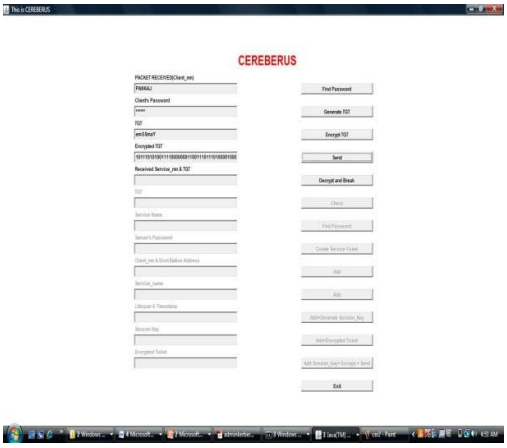
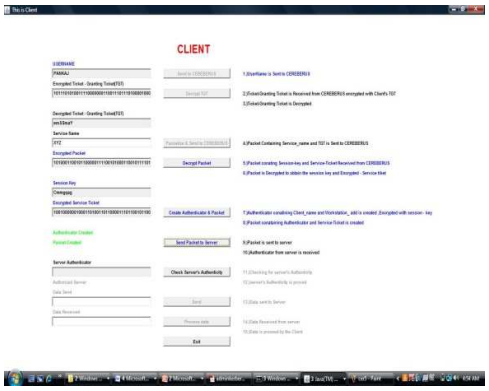


Fig 6. GUI window for Kerberos to generate, encrypt and send the authentic ticket to the client

Then click on find password in the cereberus window to generate, encrypt and send the ticket to the client for authentication as shown in Fig 6.

Then decryption of the the authentic ticket send by the cereberus in the client window is done along with the ticket send the service name to the cereberus. Cereberus decrypts and verifys the ticket, service name and the password send by client. It then generate a service ticket for the client to access the services of the server and also encrypts it with client's name, client's service name, client's IP address and session key. It sends it all to the client. Fig 7 shows that client decrypts the packet sent by the cereberus, obtain the session key and authenticate the packet. It then sends the authenticated packet to the server.



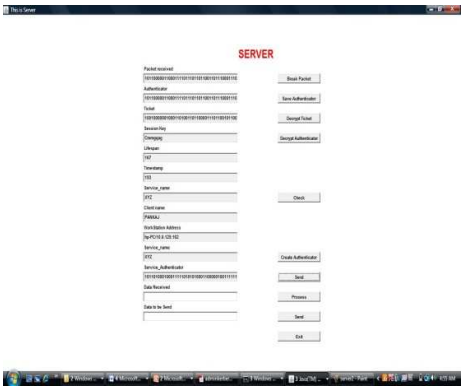


Fig 7. Client sending the authentic packet to server
Fig 8. Serve checking the client and establishing connection with it

At this stage the server becomes active and breaks the packet sent by the client. It then saves the authenticator for future aspect by decrypting the ticket and the authenticator. It then checks the details of the client with the existing database and establishes connectivity with the client as shown in Fig 8.

Now the work of Kerberos to provide network authentication is finished and the client and the server are now free to communicate with each other.

Fig 9. Application GUI

Fig 10. Cryptography Module GUI

Fig 11. Steganography: Embedding Audio File

Fig 12. Steganography: Embedding Image File

Fig 13. Steganography: Retrieving Image File

Fig 8 and Fig 9 shows the main GUI of application and cryptography module. Fig 11 shows the embedding of audio file in a message in steganography module and embedding of Image file can be seen in Fig 12. Then the retrieving of image file from the message can be seen in Fig 13.

V. CONCLUSION

With the help of this Software cum Remote System Controller any one can get to know what goes at the back of any Network with the help of Kerberos Protocol Implementer and also one will be able to Encrypt any file which can be send securely anywhere through-out the Globe with the help of Cryptography Concept using DES Algorithm. One can even hide any message which can be Text/Audio and Video (Steganography) behind any file to make it secured and inaccessible. This complete software cum application i.e. Remote System Controller has many Benefits.

REFERENCES

[1] Aditya Harbola, Deepti Negi and Deepak Harbola, A new A3 Kerberos Model, International Journal of Advanced Research inComputer Science and Software Engineering, Volume 2, Issue 3, 2012, pp. 290-293.
[2] Prof R.P. Arora and Garima Verma, Implementation of Authentication and Transaction Security based on Kerberos, International Journal of Innovative Technology & Creative Engineering, 2011, pp. 10-19.
[3] Alex Biryukov, Christophe De Cannière, Michaël Quisquater, On Multiple Linear Approximations, Advances in Cryptology – CRYPTO 2004, Lecture Notes in Computer Science Volume 3152, 2004, pp 1-22.
[4] Don Coppersmith, The Data Encryption Standard DES and its strength against attacks, IBM Journal of Research and Development, 1994, pp. 243-250.
[5] B. Clifford Neuman and Theodore Ts'o, Kerberos: An Authentication Service for Computer Networks, IEEE Communications, 32(9):33-38. September 1994.
[6] Neuman, B.C. , Kerberos: an authentication service for computer networks, Communications Magazine, IEEE, 1994, pp. 33-38.
[7] (2013), "Kerberos Overview- An Authentication Service for Open Network Systems", [Online] Available: http://www.cisco.com/en/US/tech/tk59/technologies_white_paper09186a00800941b2.shtml John T. Kohl, B. Clifford Neuman, and Theodore Y. T'so, The Evolution of the Kerberos Authentication System. In Distributed Open Systems, pages 78-94. IEEE Computer Society Press, 1994.
[8] Michael J. Wiener and Keith W. Campbell, "DES is not a Group", Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, pp. 512-520.
[9] (2013), "Using the Java Service Engine in a Project", [Online] Available: http://docs.oracle.com/cd/E19509-01/821-0237/book-info/index.html
[10] Kerberos: "The Network Authentication Protocol", [Online], Available: http://web.mit.edu/kerberos/
[11] (2013), "Kerberos", [Online], Available: http://www.giac.org/cissp-papers/47.pdf
[12] (2013), "Data Encryption Standard", [Online], Available: http://www.springerlink.com/content/16udaqwwl9ffrtxt/

