ISSN:2320-0790

COMPUSOFT
An International Journal of Advanced Computer Technology

ENHANCED KERBEROS AUTHENTICATION FOR DISTRIBUTED ENVIRONMENT USING TWO PHASES SECURITY

Dr.Mahmoud Khalifa
Department of computer science, College of Art & Science
Univrsity of Bisha, Bisha, Saudia Aribia
mkhalifa@ub.edu.sa

Abstract: There are many ways to detect , guess ,extract and compute password for online attack, there for using Password protect models are not enough safe to provide the security to the users specially in financial services to restrict unauthorized access to the system like password online guessing attacks which is mainly brute force and dictionary attacks are achieved by limiting the number of attempts made during login [1].

To secure the various systems for the provision of customer services from intrusion common types and specifically Replay attack, Password guessing attack, screen shot, key logger attack we used a model of two phases.
First phase used Kerberos model (KDC) as a trusted third party between client and server. So several possible goals accomplished through our study and are summarized as follows

Kerberos uses cryptographic tickets in order to avoid transmitting plain text passwords over the network [2]. To eliminate a number of problems experienced by the Kerberos protocol this is based on the basis of this model. We used RSA encryption to secure the keys session contain username and password concatenated with id code read from external device in order to avoid transmitting plain text which can be detected by key logger attack or screenshot attack .

In the second phase we coded all session by apply effectively the CRC algorithm to perform secure communication on an open network, using cryptographic tickets.

Keywords— Kerberos, Screenshot, Key logger, Replay; guessing, RSA, CRC .

I. INTRODUCTION

The information security can be defined as the science that works to provide protection for the information from internal or external risks. And it is also known as an information security standards and procedures taken to prevent access to the hands of unauthorized persons across communications to ensure the authenticity and validity of these communications.

The protection of information is old, but began to be used effectively since the beginning of the development of technology and information security is based on "protection systems for operating systems, software applications, and database systems as well as the protection of access to the systems".

Since more than twenty years, information security has identified confidentiality, Integrity and availability.

in our project we are working on designing improved Authentication model of users identity based on Kerberos environment.
Kerberos is a system of authentication developed at MIT as part of the Athena project. Kerberos uses encryption technology and a trusted third party, an arbitrator, to perform secure authentication on an open network. Specifically, Kerberos uses cryptographic tickets in order to avoid transmitting plain text passwords over the wire[2]. As shown in the Figure (1"a,b").
In this study also Rivest, Shamir &Adleman(RSA algorithm[3] was used to secure the keys session stage and cyclic redundancy check (CRC)algorithm[ ] will used as a second phase to complete the process of granting access authorization Ticket Granting Ticket(TGT).
Risks and types of attack Will be analyzed that would be inflicted proposed authentication model as shown in Figure (2)

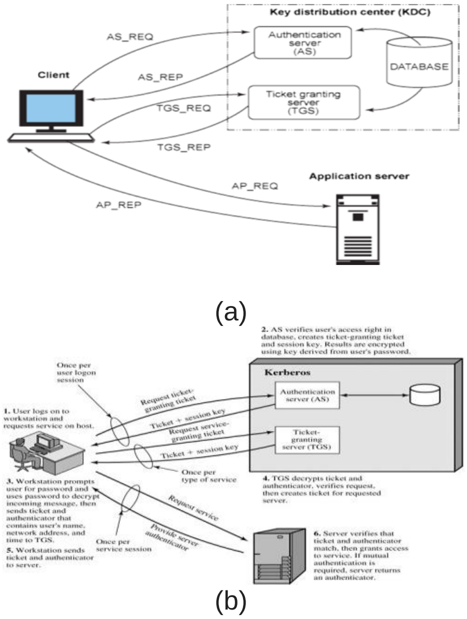
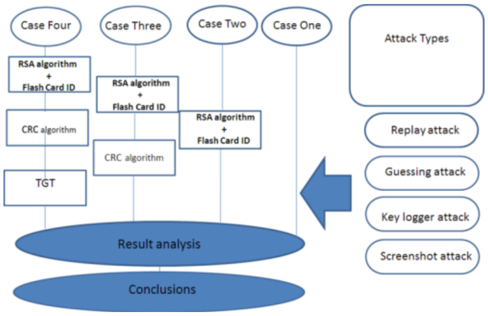

Key distribution center (KDC)
Authentication server (AS)
DATABASE
Ticket granting server (TGS)
Client
AS_REQ
AS_REP
TGS_REQ
TGS_REP
AP_REQ
AP_REP
Application server
(a)
2. AS verifies user's access right in database, creates ticket-granting ticket and session key. Results are encrypted using key derived from user's password.
Kerberos
Authentication server (AS)
Ticket-granting server (TGS)
Once per user logon session
1. User logs on to workstation and requests service on host.
Request ticket-granting ticket
Ticket + session key
Request service-granting ticket
Ticket + session key
Once per type of service
4. TGS decrypts ticket and authenticator, verifies request, then creates ticket for requested server.
3. Workstation prompts user for password and uses password to decrypt incoming message, then sends ticket and authenticator that contains user's name, network address, and time to TGS.
5. Workstation sends ticket and authenticator to server.
Once per service session
Request service
Provide server authenticator
6. Server verifies that ticket and authenticator match, then grants access to service. If mutual authentication is required, server returns an authenticator.
(b)

Figure (1): (a)The Key Distribution CenterSteps
(b)The Protocol Kerberos Steps



Figure(2) :The Proposed Authentication Model using Kerberos protocol

II. Research Importance and Justification

The proposed study introduces enhanced authentication model based on Kerberos environment.

Through improved design of this model has been an effective solution to eliminate a number of problems experienced by the Kerberos protocol is based on the basis of this model. "Knowing that this protocol is characterized by several features previously been referred to in item provided the beginning of this study, the proposed ". .

- And we thus contribute to the increase of its quality as a model ascertain the possibility of its use in a variety of applications the reliability of a high degree with remarkable resilience in the ways of dealing safely to its users in various computer areas to ensure the confidentiality of data (medical, financial and commercial ones) and keeping it from intrusion and specifically piracy sensitive sites, such as online banking or perhaps when you send an e-mail or even when entering the social networking sites. In addition to offering new mechanisms to open Hobo future solutions for researchers to develop methods of optimization with this more comprehensive safety flexibly.

III. Objectives

This study aims to use the following:
1-Rivest, Shamir &Adleman(RSA) algorithm"Session key "
2-cyclic redundancy check algorithm
3-Subscription card (external memory)

To secure the various systems for the provision of customer services from intrusion common types and specifically
1- Replay attack
2- Password guessing attack
And we can say that several possible goals accomplished through this study and are summarized as follows:
1. identify (Kerberos protocol) and its history .
2. Find out how this protocol is used in the construction of some systems and secured, through which we will cite previous studies summarized in its item in addition to our use of this protocol in this that are under study.
3. To understand how the RSA algorithm and application to secure the keys as (FIRST PHASE).
4. How to employ effective CRC algorithm used for the first time in secure
systems which have been built using Kerberos protocol to complement the
first securing phase (SECOND PHASE).
5. To test different types of attacks as input to the designer model under study
that the model suffered from them and how to implement .

IV. Problem Statement

It is well known recently used Kerberos protocol to authenticate Parties associated with each other.
[Client, Application server, Authentication server]
And the presence of deficiencies in the model used to provide financial services in locations such as banks, government institutions, medical facilities, military organizations, and educational institutions ... etc.
Has been seeking to improve its performance (this model) in various forms to discuss anticipated problems mentioned as follows:
1. Replay attack
2. Password guessing attack
3. Key logger attack
4. Screenshot attack
And which is still under discussion as we will explain later and the presence of any of the weaknesses of the above (gaps) or Inadequate solutions. systems exposed to be secured to the loss of rights , the loss of economic actions of individuals , groups, companies (organizations). so we are in the process of suggest the best solutions which will come in the proposed solution below

V. Proposed Solution

The verification (ascertained) using (Kerberos protocol) in the science of network security is one of the most important topics of focus these days and that since he
has this protocol, the ability to perform the work steps are strongly built and lack the ability to penetrate in addition to the flexibility that the advantage of this protocol to respond hand to develop and improve its performance byeffective entrustedmethod within the authentication model.

In The item(PROLEM STSTEMENT) we mention some of the weaknesses suffered by this protocol , which is currently under treatment in many research Centers and scientific papers presented [as will be indicated to them two studies are:

1- REPLAY ATTACKPREVENTION IN KERBEROS AUTHENTICATION PROTOCOL USING TRIPLE PASSWORD No.2, March 2013
2- ENHANCD KERBEROS AUTHENTICATION FORDISTRIBUTED ENVIRONMENT20 November 2014

In The item (Previous Studies) isa summary of what has to know andwhat theoutcomes of the development of this protocol]And specifically it was:
1. Replay attack
2. Password guessing attack
3. Key logger attack
4. Screenshot attack

Our study also includes improved solutions.andfor design authenticate based on (KERBEROS PROTOCOL) model usingmorethan the beginning of the process of securing:
-use RSA algorithm to secure the keysasFirst Phaseincluded machine-readable card "external memory" in order to prevent guessing attack.
- It was the employment of CRC algorithm (for insurance) complementary to the first stage of the process " Session Key". insurance can be considered as the
(second phase.)
- It should be noted that this study use the CRC algorithm for the first time to secure systems has been to rely on (Kerberos protocol) in their construction which for so to speak was inspired by the idea of this study:

"ENHANCD KERBEROS AUTHENTICATION FORDISTRIBUTED ENVIRONMENT 20 November 2014"

The starting point depending on what has been reached satisfactory results in the conclusion of this published paper.
- As they proceeded to use the RSA algorithm to secure the keys with an external memory (card) reading is to add to use the CRC algorithm as a complement to the process . and moreover full insurance session where is the third stage In that weresimilar in form and differed in the mechanism of the application and the means used with the this study:

"REPLAY ATTACK PREVENTION IN KERBEROS AUTHENTICATION PROTOCOL USING TRIPLE PASSWORDNo.2, March 2013"

VI. Methodology

The methodology of the paper can be summarized in the following steps:

1. will be shown how the model routinely in the case of a service request
2. Two types of attack (Replay attack , Guessing password attack) will be shown how to implement each of them and how to use them was an attempt to break through the model, which was designed as follows:
   a- when the insurance model using RSA algorithm included with extra data from external memoryas machine-readable(Identity data) .
   b- When you add the second insurance using the CRC algorithm developed to secure the full session.
   C- compared to the results obtained in both paragraphs (a, b) above to see how effective the model designer in repelling all of Replay attack, as well as Guessing password attack is initially
3. The proposed algorithm and related manipulations will be programmed using Java software.
4. The results are analyzed and discussed.
5. Conclusions are withdrawn.

VII. Previous studies

Several Papers and scientific studies have discussed the design and development (authentication model) based on Kerberos environment and the extent of its importance in obtaining effective and efficient insurance in various service institutions through building authentication model. this model can be access to a service while preserving the safety and confidentiality of data physically and programmatically.

Our study highlighted several scientific contributions to its importance in the development and improvement access to the full insurance model.

The following summary of the two studies were the closest to our topic at hand under discussion in terms of working to improve the performance of Kerberos protocol.

to follow a specific mechanism to secure data to model the user service. knowing that the mechanism of insurance for both studies differ each other in form and content , even though they aim to the same end .
that we will try hard to reach secure authentication model with high efficiency against the types of attack. currently there are weaknesses in Kerberos protocol are still being processed and can be manipulated by unauthorized to try to break through .but we will follow a method in which the mechanism of the application differ soft and hard ware down to the same goal.
The first study was:

"REPLAY ATTACK PREVENTION IN KERBEROS AUTHENTICATION PROTOCOL USING TRIPLE PASSWORDNo.2, March 2013"

This study developed a method to prevent or prevention of :

(Replay attack, Password guessing attack) By using the "Triple Pass Word scheme"

Familiarized themselves with the Content of " RELATEDWORK" of this paper has many of the relevant studies which proposed and mentioned for several systems
to prevent the replay attack in Kerberos authentication protocol with a detailed explanation as in this examples:
1-Yang Jian, An Improved Scheme of Single Sign-on Protocol, Fifth International Conference on Information Assurance and Security, PP. 495-498, IEEE 2009
1-Yang Jian, An Improved Scheme of Single Sign –on Protocol Based on Dynamic Double Password International Conference on Environmental Science and Information Application Technology, IEEE2009. PP. 572-575
an overview of the Kerberos protocol is then displayed reading it is clear that using this protocol to authenticate the environment multiple services , the Kerberos protocol performs the tasks entrusted to him in three phases each of which is explained inuseful detail .
It is by looking at the Figure[1][b] that shows the Basic Kerberos Architecture in this paper we can identify the six steps mentioned respectively to explain the mechanism of action of Kerberos and we find that the basic problem with this protocol is vulnerable and significantly to replay attack and can be known replay attack as below:
Replay attack is an attack in which attacker captures messages transmitting through the channel, modifies it and replay back on the transmitting channel So, it is necessary to prevent the replay attack especially when two parties need secure communication over the internet.
In the item PROPOSED MODIFICATIONS TO THE KERBEROS PROTOCOL It says :
(The main problem with the Kerberos Authentication Protocol is that of replay and password attack. Problem arises when Authentication Server (AS) sends Ticket-Granting-Ticket (TGT) to the client process running in the user. Kerberos V5 even can't avoid the replay attack. An attacker can capture all the messages transmitting from the Authentication Server (AS) to the user and apply all possible combination on the messages that he has captured. After applying all the possible combination of the captured messages, an attacker presents TGT to the Ticket-Granting-Server (TGS). TGS checks that this is a valid authenticator, so it passes Service-Granting-Ticket to the attacker and attacker may gain unauthorized access to the services stored on the Application Server .
In this modified Kerberos protocol, Ticket-Grating-Server (TGS) presents Session Key to the server by encrypting them with the hash of user password.AS passes two passwords to the TGS and TGS further passes one password to the Application Server.)
With respect to the steps involved in the proposed architecture, there will find a detailed explanation includes architectural form and a description of the work in every step.
The advantages of the Proposed Architecture are Provides protection against replay attacks, It provides triple layer of protection, Password attack protection but The Limitation of the Proposed When a new user logs on the workstation, then he will have to enter three passwords and these passwords will be stored at the TGS.
Finally, the conclusion, which stated the following:
"The approach used in this paper attempts to prevent replay attack by using three passwords, a new user must enter these passwords that will be stored on the Authentication Server. If an attacker gains access to TGT, then he can easily replay them to the TGS, but not to the Application Server (V). The reason for this is that attacker does not know the password to get session key used for communication with the Server V. So, we have to prevent attacks from taking unauthorized take control from system even if he has gain access to session key and the ticket. The approach used in our proposed architecture provides protection against replay and password attack".
The second study was:

"ENHANCED KERBEROS AUTHENTICATION FOR DISTRIBUTED ENVIRONMENT
20November 2014"

See Through this paper, we found that the most important goals was :
"provide unique and enhanced authentication model based on Kerberos environment".
It has the ability to provide protection for authentication system (Kerberos environment) From password - guessing attack and replay attack.
Additionally, insists the need for an additional Session Key and a nonce to be used between the Authentication Server (AS) and Client .
And the final result , in this paper from this enhanced model making the security stronger , helps Kerberos environment to prevent against replay attack and password-guessing attack.

We can sum up The Implementation of " PROPOSEDWORK " as noted in this paper:
(Traditionally, the banks and financial institutions sends only OTP to the client ' s mobile to verify the identity of the request. If the SIM card is hacked, the secret is revealed and the identity can be compromised. In that proposed work, a new and unique idea to use computed final nonce value has been implemented and hence it cannot be hacked and the identity cannot be compromised. Because when AS sends the nonce value, if it is hacked also It will not be useful to the hacker. Whereas in the previous case, if the information sent by the bank is hacked, then the identity is lost.
Though, session key can be used to identify the identity of the Client, the final nonce value provides not only additional

verification of the identity but also helps to avoid replay attack. The final nonce value must be in multiples of 5 within the range of 105 to 150. The TGT request must include the final nonce value that is sent from the Client to the AS).

VIII. Project Requirements

The requirements of this project areas mentioned below:

1. Previous studies on the design model to verify the user's identity based on the Kerberos protocol.
2. Connect the external memory of the first phase of the authentication model(AM).
3. Know the mechanism of action of RSA algorithm in securing the keys.
4. Learn how to use the CRC algorithm-to secure data.
5. Service to be accessed by multi-user.
6. Operations model steps with Replay attack.
7. Operations model steps with Password guessing attack.
8. Java Software (NetBeans 8,0.2),SQL ServerandASP.net

IX. RESULT AND DISCTION

Login procedures:

1) The user enters a username and a password. If his computer contains a cookie stored by the login server then the cookie is retrieved by the server.

2) The server checks whether the username is valid and whether the password is correct for this username.

3) If the username/password pair is correct, then

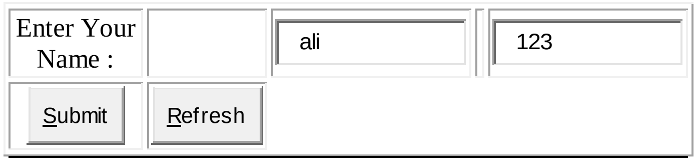
(a) If the cookie is correctly authenticated and has not yet expired, and the user identification record in the cookie agrees with the entered username, then the user is granted access to the server.

(b) Otherwise, the server generates and RTT and sends it to the user. The user is granted access to the server only if he answers the RTT correctly.

4) If the username/password pair is incorrect, then

(a) The user is asked to answer an RTT with
Probability P (0<P<=1). When his answer is received he is
Denied access to the server, regardless of whether it is
Correct or not.

(b) With probability 1-P, the user is immediately
Denied access to the server.



Enter Your Name :		ali	123
Submit	Refresh		

Figure[3] : login page with ID read from external flash 2000024913786

VS Protocol

VS protocol referred to as Van Oorschot and Stubblebine protocol proposed modifications to the previous protocol which track failed logins per username to impose ATT challenges after exceeding a configurable threshold of failures. In addition, upon entering correct credentials in the absence of a valid cookie, the user is asked whether the machine in use is trust worthy and if the user uses it regularly. The cookie is stored in the user's machine only if the user responds yes to the question.

Procedures:

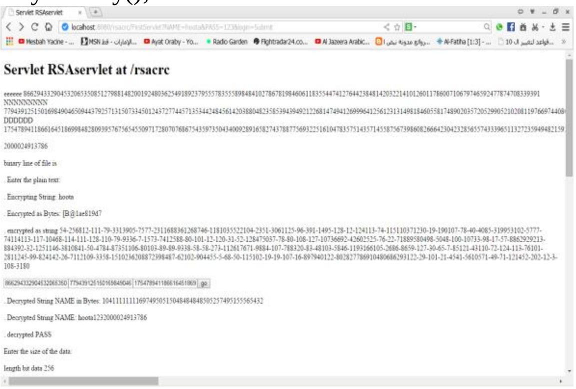
1) User logins with username and password.
2) If there is a cookie in user device then server
Retrieves it.
3) If username/password is correct then
if the cookie is present and valid then login
Successfully
4) else
if Owner Mode(username) or Failed Logins(username)
1 ≥ threshold value then
Send a Turing test to user, login successfully if answer correctly
5) Else login successfully
6) Else
7) Set decision function to TRUE with probability P
8) If decision function or Failed Login(username) 2 ≥ threshold value then
9) Ask a Turing test, wait for answer. Say login
Fails.
10) Else say login fails immediately.

These protocols involve large number of Turing test which an valid user also must undergo which reduces the convenience of the user.

SI as primary units. English units may be used as secondary units (in parentheses). Use a zero before decimal points: “0.25”, not “.25”. Use “cm$_3$”, not “cc”.

```
BigInteger ee= rsa.e;
             BigInteger dd= rsa.d ;
               BigInteger nn= rsa.N ;
              out.println("eeeeee " +    ee);
          out.println("NNNNNNNNNN " +    dd);
          out.println("DDDDDD " +   nn);
      String fileName = "d:\\myid.txt";
          String line = null;
       try {
             FileReader fileReader =
               new FileReader(fileName);
             BufferedReader bufferedReader =
               new BufferedReader(fileReader);
           while((line = bufferedReader.readLine()) != null) {
            out.println("<P>"+line);
          out.print("<P> binary  line of file is " );
        out.println("<p> . Enter the plain text: \n ");
      out.println("<p> . Encrypting String:   " +
request.getParameter("NAME"));
       String  namepass
=request.getParameter("NAME")+request.getParameter("PAS
S")+ line ;
      byte[]  pp =namepass.getBytes();
```

```
        byte[] encrypted = new BigInteger(pp).modPow(ee,
nn).toByteArray();
```



```
Servlet RSAservlet at /rsacrc

eeeeee 866294332904532065350851279881482001924803625491892379555783555898484102786781984606118355447412764423848142032214101260117860071067974659247787470833939I
NNNNNNNNN
77943912515016984904650944379257131507334501243727744571353442484561420388048235853943949212268147494126999641256123131498184605581748902035720529905210208119766974408
DDDDDD
17547894118661645186998482809395767565455097172807076867543597350434009289165827437887756932251610478357514357145587567398608266642304232856557433396511327235949482159

2000024913786

binary line of file is

. Enter the plain text:

. Encrypting String: hoota

. Encrypted as Bytes: [B@1ae819d7

. encrypted as string 54-256812-111-79-3313905-7577-2311688361268746-1181035522104-2351-3061125-96-391-1495-128-12-124113-74-115110371230-19-190107-78-40-4085-319953102-5777-74114113-117-10468-114-111-128-110-79-9336-7-1573-7412588-80-101-12-120-31-52-128475037-78-80-108-127-10736692-42602525-76-22-71889580498-5048-100-10733-98-17-57-8862929213-884392-32-1251146-3810841-50-4784-87351106-80103-89-89-9338-58-58-273-112617671-9884-107-788320-83-48103-5846-1193166105-2686-8659-127-30-65-7-85121-43110-72-124-113-76101-2811245-99-824142-26-7112109-3358-1510236208872398487-62102-904455-5-68-50-115102-19-19-107-16-897940122-80282778691048068629312 2-29-101-21-4541-5610571-49-71-121452-202-12-3-108-3180

866294332904532065350 | 779439125150165849046 | 1754789411866164518 69 | go

. Decrypted String NAME in Bytes: 104111111116974950515048484848505257495155565432

. Decrypted String NAME: hoota1232000024913786

. decrypted PASS

Enter the size of the data:

length bit data 256
```

Every modern communication protocols uses one or more error-detection algorithms to achieve reliable communication between source and destination[1a]. CRC is by far the most popular. The sender and receiver agree on a certain fixed polynomial called the generator polynomial.CRC properties are defined by length and coefficients of the generator polynomial.. The protocol specification generally defines CRC in hexadecimal or polynomial notation.

```
        byte[] decrypted = rsa.decrypt(encrypted);
        out.println("<p> . Decrypted String NAME in Bytes:   "
+ bytesToString(decrypted));
        out.println("<p> . Decrypted String NAME:    " + new
String(decrypted));
        out.println("<p> .  decrypted PASS ");
```

In the RSA cryptosystem, the public modulus N = pq is a product of two primes of the same bit size. The public and private exponent e and d satisfy ed = 1 mod (p −1)(q −1).
In many applications of RSA, either e or d is chosen to be small, for efficient modular exponentiation in the encryption/verifying or in the decryption/signing phase. It is well-known that it is dangerous to choose a small private exponent, since Wiener [8] showed that the RSA scheme is insecure if d < N0.25, which was extended to d < N0.292 by Boneh and Durfee [9].

X. CRC SOFTWARE IMPLEMENTATION

Following are steps for implementing a CRC in software. The steps for CRC computation are followed at transmitter side and that for CRC Checking are followed at receiver side.

To compute an n-bit binary CRC, line the bits representing the input in a row, and position the (n+1)-bit pattern representing the CRC's divisor (called a "generator polynomial") underneath the left-hand end of the row as in figure[4].

```
out.println("<p> Enter the size of the data:");
                  out.println("<p>   length   bit   data   "   +
encrypted.length );
out.println("<p>   bit data " +  bytesToString(encrypted));
                        byte divisor[] = {1,0,0,0,0,0,1,1,1}; ;
                        out.println("<p> length bit divisor " +
divisor.length  );
                   out.println("<p> then  The CRC code generated
is: ");
                        byte remainder[] = divide(encrypted ,
divisor);
                        for(int i=0 ; i < remainder.length-1 ; i++) {
                                   out.print(remainder[i]);
             }
                    byte[] rem_data = new byte[remainder.length-
1];
                                   System.arraycopy(remainder, 0,
rem_data, 0, remainder.length-1);
                     out.print(" the remain_data is ." +
bytesToString(rem_data));
                  out.println("<p> generated   data is:");
                       for(int i=0 ; i < encrypted.length  ; i++) {
                                   out.print(encrypted[i]);
                        }
             out.println("<p> then  The CRC code generated with
data is:");
……………………………………………………………….
out.println("<p> Enter the size of the data:");
                   out.println("<p> length bit data " +
encrypted.length );
                        out.println("<p>   bit data " +
bytesToString(encrypted));
                        byte divisor[] = {1,0,0,0,0,0,1,1,1}; ;
                        out.println("<p> length bit divisor " +
divisor.length  );
                   out.println("<p> then  The CRC code generated
is: ");
                        byte remainder[] = divide(encrypted ,
divisor);
                        for(int i=0 ; i < remainder.length-1 ; i++) {
                                   out.print(remainder[i]);
             }
                    byte[] rem_data = new byte[remainder.length-
1];
                                   System.arraycopy(remainder, 0,
rem_data, 0, remainder.length-1);
                     out.print(" the remain_data is ." +
bytesToString(rem_data));
                  out.println("<p> generated   data is:");
                       for(int i=0 ; i < encrypted.length  ; i++) {
                                   out.print(encrypted[i]);
                        }
             out.println("<p> then  The CRC code generated with
data is:");
                         for(int i=0 ; i < encrypted.length   ; i++) {
                                   out.println(encrypted[i]);
                        }
             for(int i=0 ; i < remainder.length-1 ; i++) {
                                   out.println(remainder[i]);
                        }
                        out.println( "<p>");
                        byte[] sent_data = new
byte[encrypted.length  + remainder.length-1];
                            System.arraycopy(encrypted, 0,
sent_data, 0, encrypted.length);
                System.arraycopy(remainder, 0, sent_data,
encrypted.length, remainder.length-1);
                 out.println("<p> sent data length  is " +
(sent_data.length) + ":");
                 out.println("<p> sent data is " +
bytesToString(sent_data) + ":");
          out.println("<p>");
           byte remainde[] = divide(sent_data, divisor);
         for(int i=0 ; i < remainde.length-1 ; i++) {
                          if(remainde[i] != 0) {
                                   out.println("<p> There is
an error in received data...");
                                  return;
                          }
                }
                 out.println("<p> Data was received without
any error." + Arrays.toString(remainde));
           out.println("<p>");
```

XI. CONCLUSION AND FUTURE WORK

The proposed work focuses mainly on replay attack and password-guessing attack. It can also prevent from key logger attack and screenshot attack when the communication between the client and the server is implicit. It provides mutual authentication between the client and the server and thus, OTP cannot be trapped. This concept can be implemented in banks and financial institution where financial transactions are frequently made. Even though the malicious user possesses a duplicate SIM card, by using this concept, the OTP cannot be used to steal one's identity. The paper does not insist that Kerberos has many flaws. In fact, there is no doubt that Kerberos is the most popular and highly efficient network authentication protocol. The proposed work seeks to focus on how it can be made hack-proof. Like the proverb "a fruitful tree is often stoned" says, it aims to strengthen the Kerberos protocol and not to stone it. Though the proposed work has many advantages, the replay attack is possible if the malicious user captures the first TGT request. The CRC value marked against the user database becomes invalid only after first TGT request. So if the first TGT request itself is replayed, it may not be possible to identify the malicious user

Types of Attacks	Kerberos	Reason for loophole	Proposed Work	Justification
Replay Attack	Possible	Does not prove identity & Tokens reusable	Not Possible	CRC Mechanism
Password-Guessing Attack	Possible	Initial handshake is not protected	Not Possible	Session Key RSA
Keylogger Attack	Possible	Password is entered via keyboard	Not Possible	ID is entered via external device
Screenshot Attack	Not possible	Internal Communica-tion	Not Possible	Internal Communica-tion

TABLE 1. COMPARISON OF KERBEROS AND PROPOSED WORK

ACKNOWLEDGMENT

I Would like to send my great appreciation to my teachers in all stages primary and secondary school, Bsc, Msc, PHD . who taught me and never I forget their care. thanks to my family, mother, father ,wife and children.

XII. REFERENCES

[1] J. Jayavasanthi Mabel1, Mr. C. Balakrishnan2,

RESISTING PASSWORD BASED SYSTEMS FROM ONLINE GUESSING ATTACKS, International Conference on Information Systems and Computing (ICISC-2013), INDIA.

[2] William Stallings, Cryptography and Network Security Principles and Practices, Fourth Edition, Prentice Hall ,November 16, 2005

[3] GaganDua 1, NitinGautam 2, Dharmendar Sharma 3,Ankit Arora 4,REPLAY ATTACK PREVENTION IN KERBEROS AUTHENTICATION PROTOCOL USING TRIPLE PASSWORD, (IJCNC) Vol.5, No.2, March 2013

[4] A. JESUDOSS #1, N.P. SUBRAMANIAM*2, ENHANCED KERBEROS AUTHENTICATION FOR DISTRIBUTED

[5] ENVIRONMENT,Journal of Theoretical and Applied Information Technology (JATIT)20th November 2014. Vol. 69 No.2

[6] [1a] Jyoti Wadhwani1, Prof. Nitin Narkhede2, Implementation of communication using Cyclic Redundancy Check, ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 7, July 2013

[7] Ellen Jochemsz1,_ and Alexander May2, A Polynomial Time Attack on RSA with Private CRT-Exponents Smaller Than *N*0.073

[8] Wiener, M.: Cryptanalysis of Short RSA Secret Exponents. IEEE Transactions on

Information Theory 36, 553–558 (1990)

[9] Boneh, D., Durfee, G.: Cryptanalysis of RSA with Private Key *d* Less Than *N*0.292. IEEE Transactions on Information Theory 46, 1339–1349 (2000)