**Military Technical College**
**Kobry El-Kobbah,**
**Cairo, Egypt**

**8th International Conference**
**on Electrical Engineering**
**ICEENG 2012**

# A Modified Kerberos Authentication Scheme Using Public Key Cryptography

*By*

Khaled Mohamed Khairy
Egyptian Armed Forces
Kmkkhairy74@hotmail.com

Ahmed Abd Elhafez
Egyptian Armed Forces
aabdelhafez@gmail.com

Essam Abd El Wanees
Egyptian Armed Forces MTC

***Abstract:***

Kerberos is a widely used authentication scheme based on symmetric key cryptography, although Kerberos is a part of MIT's Project Athena it has been adopted by many other organizations for their own purposes. And is being discussed as a possible standard. Despite Kerberos's many strengths, a number of limitations and some weaknesses have appeared due to MIT's environment that needs only user-to-server authentication and others due to deficiencies in the protocol design. In this paper an improved scheme using the Public Key cryptography will be proposed to enhance its security strength to overcome these limitations and weaknesses.

***Keywords:***

Authentication; Kerberos; security analysis; Public Key cryptography

---

## *1. Introduction:*

Modern computer networks provide service to multiple users and require the ability to accurately identify the user making a request. In traditional systems, the user's identity is verified by checking a password typed during login; the system records the identity and uses it to determine what operations may be performed. The process of verifying the user's identity is called authentication, Password-based authentication is not suitable for use on computer networks. Passwords sent across the network can be intercepted and subsequently used by eavesdroppers to impersonate the user.

Stronger authentication methods based on cryptography are required. In which an attacker listening to the network gains no information that would enable it to falsely claim another's identity. Kerberos [3] is the most commonly used example of this type of authentication technology.
Kerberos is a Network Authentication scheme, which was created by MIT for Athena Project based on the deformation of symmetric Needham-Schroeder protocol [4], Since Kerberos was put forward, and it has experienced 5 versions. Among them, versions 1-3 occurred only internally at MIT, Kerberos version 4 was published in the late 1980s by Steve Miller and Clifford Neuman, the primary designers of Kerberos.

With the spread of versions 4, some of its limitations and weaknesses were gradually discovered; MIT expanded and improved the version 4. Now, a more complete version 5 is formed [1] by John Kohl and Clifford Neuman, appeared as RFC 1510 in 1993. Until January 2000, the latest version is krb5 –1.1[2], However, Kerberos is combining with the environment of MIT, there are some limitations to extend it as a standard use in all kinds of existing computer communication network [6].

Aiming at the Kerberos security issues in actual network authentication applications, at home and abroad, many specialists have studied a lot and put forward some improved security measures [7-11]. However, some of these measures will change the deployment structure of Kerberos; some will make the system calculation significantly increase. In this paper, after analyzing the security of Kerberos a proposed scheme based on the Kerberos main structure will be given out, which use the Public Key cryptography beside the symmetric key cryptography to enhance its security strength signally with a small extra computational cost.

## *2. Kerberos Authentication scheme:*

a. The Kerberos's fundamental:

it can be illustrated as shown in figure 1: a KDC (key distribution center) that consists of two logically separate parts: an Authentication Server (AS) and a Ticket Granting Server (TGS) is established to centrally preserve usernames and passwords, and is used to authenticate the users' identity and grant authorization to users, the servers that provide various kinds of service do not directly authenticate the users' identity and not grant any authorization, but provide corresponding services according to granted tickets. The communications between users, KDC and servers are encrypted with DES (data encryption standard) in Kerberos version 4; in version 5 users can identify the algorithm used to encrypt the data.

b. The Kerberos's authentication process:

As shown in Fig.1, the Kerberos's authentication process consists of three stages, including six steps [3, 5]. The process is stated in the following.

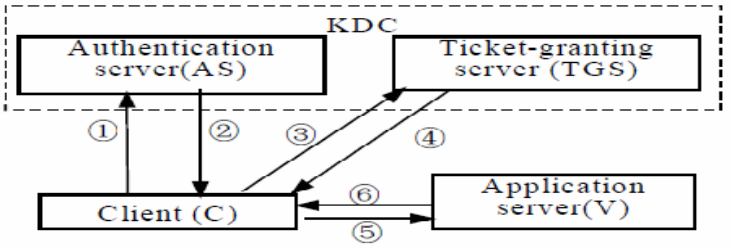


***Figure (1):*** *The Kerberos's fundamental*

**1st** Stage: AS Exchange to obtain TGT

(1) The user asks Authentication Server for a $Ticket_{tgs}$ after checking the client ID and the time stamp.

$$C \rightarrow AS: \quad ID_c \| \ ID_{tgs} \| \ TS_1 \qquad (1)$$

(2) AS generate and transmit $Ticket_{tgs}$ to the valid user.

$$AS \rightarrow C: \quad K_c [K_{c,tgs} \| \ ID_{tgs} \| \ TS_2 \| \ Lifetime_2 \| \ Ticket_{tgs}] \qquad (2)$$

$$Ticket_{tgs} = K_{tgs} [K_{c,tgs} \| \ ID_c \| \ AD_c \| \ ID_{tgs} \| \ TS_2 \| \ Lifetime_2]$$

**2nd** Stage: TGS Exchange to obtain Service Granting Ticket ($Ticket_v$)

(3) The user asks Ticket Granting Server for a $Ticket_v$.

$$C \rightarrow TGS: \quad ID_v \| \ Ticket_{tgs} \| \ Authenticator_c \qquad (3)$$

$$Authenticator_c = K_{c,tgs} [ID_c \| \ AD_c \| \ TS_3]$$

(4) TGT generate and transmit $Ticket_v$ to the user after checking $Ticket_{tgs}$ and the Authenticator.

$$TGS \rightarrow C: K_{c,tgs} ,[K_{c,v} \| \ ID_v \| \ TS_4 \| \ Ticket_v] \qquad (4)$$

$$Ticket_v = K_v [K_{c,v} \| \ ID_c \| \ AD_C \| \ ID_v \| \ TS_4 \| \ Lifetime_4]$$

**3ed** Stage: Client/Server Authentication Exchange to obtain Service

(5) The user asks the Service Server to obtain the service by sending Ticketv and

the Authenticator.

$$C \Rightarrow V: \quad Ticket_v \parallel Authenticator_c \qquad (5)$$

(6) Service Server V verifies the user by checking $Ticket_v \parallel Authenticator_c$ and verify itself by sending back the time stamp added by one and encrypted by the session key $K_{c,v}$.

$$V \Rightarrow C: \quad K_{c,v}[TS_4 + 1] \qquad (6)$$

C can confirm V by decrypting and checking the time stamp.

## *3. Kerberos Security Analysis:*

Security of Kerberos has been analyzed in many works, e.g. [5, 6, 12, 13, 14, 16, 18], in this paper a security analysis of the Kerberos will be performed to identify some of its limitations and weaknesses in the deployed versions of Kerberos to resolve these issues in the proposal scheme.

a. Limitation due to design:

(1) Although Kerberos is a widely used scheme in the identity authentication, but it doesn't provide a strong *mutual* authentication because Kerberos was designed to work with MIT's environment that needs only user-to-server authentication, the user authenticates himself by its user with the password and the server can authenticate itself to the user by encrypting the response with the key derived also from the users' password stored and used in plain inside AS.

(2) The Kerberos authentication scheme doesn't achieve Perfect Forward Secrecy because it doesn't separate between the authentication process and the data transmission, AS use the password to derive the shared key $K_c$ which is used to encrypt the session key $K_{c,tgs}$ used to secure the transmission between user and the TGS, compromising the security of the password will compromise the security of the session key and all the transmitted data.

(3) The Kerberos authentication scheme doesn't provide In-transit confidentiality during Authentication process by sending the authentication request in plain to AS, the response will be encrypted by $K_c$ which is derived from the user password that may be used many times before changing the user password.

(4) The Kerberos authentication scheme doesn't give a solution for avoiding non-repudiation because it doesn't use the digital signatures using the public key cryptography, so any one has the password can simulate the entity.

(5) Kerberos requires that all hosts and servers have clocks accurate within 5 minutes of each other which is the ticket validity time. If a host's clock exceeds this tolerance, then Kerberos denies access to all users and services from that host [16]. The vulnerabilities Kerberos could inherit from subverted time services include Replay attacks of valid tickets and/or reuse of expired tickets [17] and Denial of Service attacks [9] issued by moving a server's clock beyond the clock skew tolerance.

b. weakness against certain attacks

(1) Password Guessing Attacks

Although it does not require the user to transfer plain text passwords, but Password Guessing Attacks [15] are not solved by Kerberos. Though the user can obtain the sharing key with AS by entering his password on client, and avoid the password frequently being used, but after all, the password must be provided, this is not safe enough. From Kerberos authentication process perspective, the AS cannot verify user's identity, but only confirm him by encrypting the messages sent to users in step number 2, session key is encrypted by $K_c$, which is the key derived from user password, and it is not necessary to verify the authenticity of the user when the servers are answering. But it is assumed that only legitimate users have the password. If a user chooses a poor password, it is possible for an attacker to successfully mount an offline dictionary attack by repeatedly attempting to decrypt, with successive entries from a dictionary, messages obtained which are encrypted under a key derived from the user's password. Once the attacker intercepted a response, password attack is easily to be formed.

(2) Brute-force attacks against the KDC.

The Kerberos authentication scheme is vulnerable to brute-force attacks[8] against the KDC (the initial ticketing service and the ticket-granting service) which has unencrypted access to all client and service keys and is thus critical to the security of Kerberos The entire authentication system depends on the trust ability of the KDC(s) which store a hug number of shared keys and user's passwords, so anyone who can compromise system security on a KDC system can theoretically compromise the authentication of all users of systems depending on the KDC.

(3) Ticket stealing and replay attacks

Kerberos was designed for use with single-user client systems. In the more general case, where a client system may itself be a multi-user system, the Kerberos authentication scheme can fall prey to a variety of ticket stealing and replay attacks [8]. Playback attack may take effect during the lifetime of the ticket (5 minutes). The overall security of multiuser Kerberos client systems (file system security, memory protection, etc.) is therefore a limiting factor in the security of Kerberos authentication.

(4) Jacking connection attack.
It's easily suffer jacking connection attack [10], Once the users pass the authentication, as long as $Ticket_v$ tickets are in validity, they can have access to any services, once the attackers intercept a cipher text, and get to know the plaintext, they can easily get $K_{c,v}$, thus to steal server information by disguising as a legitimate user.

## 4. ***The Proposed Scheme:***

a. The idea of the Proposal.
The proposal keeps the main structure of the original Kerberos authentication scheme, it only used some concepts like public key cryptography, hash function, Certificates and Nonces to overcome the shortcomings due to the design of the original Kerberos based on symmetric key cryptography and also to strengthen the scheme against different attacks as shown in figure 2.

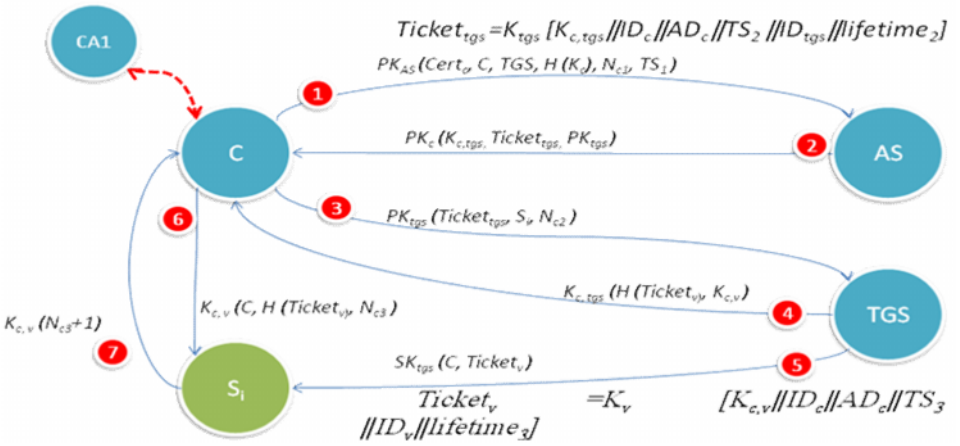


***Figure (1):*** *The proposed scheme diagram*

b. The process of authentication of the proposed scheme.

1st Stage: AS Exchange to obtain TGT

The user asks for a TGT (Ticket Granting Ticket) at a client:

The client will send a message to the AS encrypted using the public key of AS requesting the TGT ticket, the message includes the user certificate issued by CA, the user ID and TGS ID, the hash of the password, Nonce and the current time.

$$C \Rightarrow AS: \quad PK_{AS}\,(Cert_c, C, TGS, H(K_c), N_{c1}, TS_1) \qquad (p\text{-}1)$$

The AS grants a TGT to user:

- AS will decrypt the message using his private key to get its contents.
- It will decrypt the user Certificate using his public key of the CA to authenticate the user and to get its public key $PK_c$ and to check the validity time of the certificate.
- It will compare the stored hash of the user password with the received one.
- It will use the nonce $N_{C1}$ to generate a session key $K_{c,tgs}$.
- It will use the time stamp $TS_1$ to check the validity time of the request.
- After AS verify the user it will generate the requested $Ticket_{tgs}$ includes (session key between user and TGS, user ID, user address, current time, TGS ID and the ticket lifetime) encrypted by the shared session key between AS and TGS $K_{tgs}$.

$$Ticket_{tgs} = K_{tgs}\,[K_{c,tgs} \| ID_c \| AD_c \| TS_2 \| ID_{tgs} \| lifetime_2]$$

- Then it will construct the response to the user includes (session key between user and TGS, the encrypted ticket, TGS public key) encrypted using the user public key $PK_c$.

$$AS \Rightarrow C: \quad PK_c\,(K_{c,tgs}, Ticket_{tgs}, PK_{tgs}) \qquad (p\text{-}2)$$

2ed Stage: TGS Exchange to obtain Service Granting Ticket ($Ticket_v$)

Asking for the credential $Ticket_v$.

- The user will decrypt the message using his private key to get its contents *($K_{c,tgs}$, $Ticket_{tgs}$ and $PK_{tgs}$)*.
- The user will construct the credential $Ticket_v$ request from the service server V includes (ticket $Ticket_{tgs}$, the service server name *V* and nonce $N_{c2}$) encrypted by the public key of TGS.

$$C \Rightarrow TGS: \quad PK_{tgs}\,(Ticket_{tgs}, S_i, N_{c2}) \qquad (p\text{-}3)$$

TGS grants C to user.

- TGS will decrypt the message using his private key to get its contents *($Ticket_{tgs}$, V and $N_{c2}$)*.
- TGS will decrypt the ticket $Ticket_{tgs}$ using the shared secret key $K_{tgs}$ to get the session key $K_{c,tgs}$, *user ID, user address,* current time $TS_2$, *TGS* ID and the ticket lifetime.

- TGS will check the time stamp to prevent replay attack.
- TGS will use the nonce $N_{C2}$ to generate a session key $K_{c,v}$.
- TGS will generate the requested ticket $Ticket_v$ includes (session key between user and service server V, user ID, user address, current time, service server ID and ticket lifetime) encrypted by the shared session key between TGS and V.

$$Ticket_v = K_v\,[K_{c,v} \| ID_c \| AD_c \| TS_3 \| ID_v \| lifetime_3]$$

- TGS will send the message includes (the hash of the ticket $Ticket_v$ *and the session key* $K_{c,v}$) encrypted using the session key $K_{c,\,tgs}$.

$$TGS \Rightarrow C: K_{c,\,tgs}\,(H\,(Ticket_v), K_{c,v}) \qquad (p\text{-}4)$$

- TGS will also send the message includes (the user name and the ticket $Ticket_v$) signed using its private key to authenticate himself to the service server.

$$TGS \Rightarrow V: SK_{tgs}\,(C, Ticket_v) \qquad (p\text{-}5)$$

3ed Stage: Client/Server Authentication Exchange to obtain Service

User requests the service.

- User will decrypt the message delivered from TGS using the session key $K_{c,tgs}$ to get its contents (the hash of the ticket $Ticket_v$ *and session key* $K_{c,v}$).
- User will generate the requests includes (the user name, the hash of the ticket $Ticket_v$ *and nonce* $N_{c3}$) encrypted using the session key $K_{c,v}$.

$$C \Rightarrow V: \quad K_{c,\,v}\,(C, H\,(Ticket_v), N_{c3}) \qquad (p\text{-}6)$$

Service server response.

- Service server V will decrypt the signed message delivered from TGS using TGS public key.
- It will decrypt the ticket $Ticket_v$ using the shared key $K_v$ to get its contents (session key between user and service server V, user ID, user address, current time, service server ID and ticket lifetime).
- It will ensure the validity time of the ticket by checking the time stamp and the ticket lifetime.
- It will use the session key $K_{c,v}$ to decrypt the message received from the user to get its contents *(C, H (*$Ticket_v$*) and* $N_{c3}$*)*.
- It will compute the hash of the ticket $Ticket_v$ and compare it with the received one to ensure the message integrity.
- If the comparison was true it will generate the response to the user includes ($N_{c3}+1$) encrypted using the session key $K_{c,\,v}$.

$$V \Rightarrow C: \quad K_{c,\,v}\,(N_{c3}+1) \qquad (p\text{-}7)$$

## 5. ***Security Analysis of the Proposed Scheme:***

Comparing the proposed scheme with to the original one, The proposed scheme has helped to overcome the shortcomings due to the design of the original Kerberos

based on symmetric key cryptography and also to strengthen the scheme against different attacks as follow:

a. Providing in-transit confidentiality to prevent Eavesdropping attack by encrypting the entire message between the user and servers using both asymmetric key cryptography and symmetric key cryptography during the authentication process and also during data transitions.
b. Providing strong mutual authentication using the Public Key cryptography beside the Symmetric key.
c. Avoiding non-repudiation through the usage of digital signatures.
d. Reducing the effectiveness of Man-in-The-Middle Attacks by using the public key cryptography because the opponent needs to know the private key of each entity.
e. Preventing the opponent from applying the Off-Line Guessing Attacks by using the Nonces with time stamps by increasing the computation needed by the opponent to apply the attack.
f. Providing data integrity of the ticket $ticket_v$ in step 4 by using the hash function.
g. Providing a secure way to transmit the user password online hashed by a one way function so even if the opponent can intercept the hash of the password he can't recover the password
h. Achieving Perfect forward secrecy (PFS) by separating the authentication process and the data transmission, the key used in the authentication (public key cryptography) doesn't use to generate the session keys used to protect transmission of data between the user and servers, and also the keys used to protect transmission of data doesn't used to derive any additional keys.
i. Using the certificates issued by the trusted Certificate Authorities to authenticate the user to the Key Distribution System KDS, storing the users password hashed and using short life sessions keys can mitigate the risk of compromising the KDC which is vulnerable to brute-force attacks from the unauthorized users.

## *6. Conclusions:*

Although Kerberos Authentication scheme is a widely used scheme in the identity authentication but still has some limitations like it doesn't provide a strong *mutual* authentication, it doesn't achieve Perfect Forward Secrecy; it doesn't provide In-transit confidentiality during Authentication process and also couldn't avoid the non-repudiation of both client and server.

Kerberos Authentication scheme also has some weaknesses against some attacks like Password Guessing Attacks, Brute-force attacks against the KDC, Ticket

stealing and replay attacks and Jacking connection attack.

The proposed Authentication scheme used public key cryptography hash function, certificates and nonce beside the symmetric key cryptography to avoid these limitation and weaknesses.

## ***References:***


[1] John T. Kohl, C. Neuman : "*The Kerberos Network Authentication Service (V5)*",RFC 1510, September 1993

[2] Neuman C., *The Kerberos network authentication Service (v5)*, RFC 4120, 2005.

[3] Jennifer G. Steiner, "*Kerberos: An Authentication Service for Open Network Systems*", Project Athena, Massachusetts Institute of Technology Cambridge, MA 02139 steiner@ATHENA.MIT.EDU, March 30, 1988

[4] Roger Needham and Michael Schroeder "*Needham Schroeder Symmetric Key*", 1978, Last modified November 8, 2002

[5] B. Clifford Neuman and Theodore Ts'o "*Kerberos: An Authentication Service for Computer Networks*", , ISI/RS-94-399 September 1994

[6] Bellovin SM, Merrittm., " *Limitations of the Kerberos authentication systems*", Journal of ACM SIGCOMM Computer Communication Review,1990, 20 (5) :119-132.

[7] Ravi Ganesan " *Yaksha' : Augmenting Kerberos with Public Key Cryptography*", 7027-4/9$54 .000 1995 IEEE ,

[8] GIAC directory of certified professionals 2002 " *Network Security: Authentication Applications Kerberos and Public Key Infrastructure*".

[9] Allen Householder "Managing *the Threat of Denial-of-Service Attacks*", CERT® Coordination Center, In collaboration with: Rob Thomas v10.0 October 2001.

[10] Wen Lei, Hai Cao "*An Improved Kerberos Scheme Based on Dynamic Password*" Published Online December 2010 in MECS (http://www.mecs-press.org/)

[11] Chandrasekar, V.R. Rajasekar & V. Vasudevan," *Improved Authentication and Key Agreement Protocol Using Elliptic Curve Cryptography*". International Journal of Computer Science and Security (IJCSS), Volume (3): Issue (4), 2003

[12] Hany M. Harb, Yousef B. Mahdy, Montaser M. Mohammed, Yasser F. Uthman " *Overcoming Kerberos Structural Limitations"*, 2004

[13] Thomas Wu "*A real world analysis of password Kerberos security*", tjw@cs.stanford.edu, 2000

[14] Giampaolo Bella "*Formal Analysis of the Kerberos Authentication System*", Journal of Universal Computer Science, vol. 3, no. 12 (1997), 1337-1381, submitted: 20/11/96, accepted: 22/6/97, appeared: 28/12/97 ã Springer Pub. Co.

[15] Teodor Sommestad," *Password authentication attacks: a survey of attacks and when will succeed*" teodors@ics.kth.se , Technical report, TRITA-EE 2011:067.

[16] Ian Downard and Dr. Ann Miller, "*An Analysis of the Kerberos Authentication System*" Paper presented at the RTO IST Symposium on "*Real Time Intrusion Detection*", held in Estoril, Portugal from 27-28 May 2002, and published in RTO-MP-101.

[17] Emmanuel Bouillon, "*Taming the beast : Assess Kerberos-protected networks*", Black Hat EU 2009.

[18] Alexandra Boldyreva and Virendra Kumar, "*An Provable-Security Analysis of Authenticated Encryption in Kerberos*", A preliminary shortened version of this paper appears in 2007 IEEE Symposium on Security & Privacy Proceedings.