COMPARISON OF X.509, KERBEROS 5 AND PKINIT FOR OPEN DISTRIBUTED NETWORK

Monika Gogna
Research Scholar, IKGPTU, Jalandhar, India

Abstract: Communication, Availability, Control, Computation and Security are the major parameters that the network environment is working upon since very long. The most significant advantages of distributed networks includes reliability, performance and extensibility. But when it comes to open distributed network, many challenges arise due to vast number of concurrent users seeking access to the data at the same time and instantly without bothering about the different platforms followed by different servers in the distributed environment. In this paper, the comparison of three protocols X.509, Kerberos 5 and PKINIT have been presented in a distributed network to analyse whether mapping between these protocols is possible or not.

***Keywords**:* **X.509, Kerberos, PKINIT, Authentication.**

I-Introduction

In an Open Distributed Network, the computers can interact with other open systems while conforming to the standards and they seems to be single coherent system to the users. The main objective of this framework is to provide scalability, connectivity, openness and distribution transparency [1]. There are many protocols that are supporting this environment but three of them which are widely used are: X.509, Kerberos and PKINIT. These protocols allow heterogeneous components to interoperate and also sharing of their resources with ease. Also, they concentrate on the fault tolerance whereby it assures that partial failure of any component does not affect the whole system. Each of the protocols have their own formats. Therefore, while working in an open environment, one need to consider its technicalities and streamline communication.

1.1 Outline of the paper

The purpose of this paper is to compare how the initial authentication process differs across three protocols, namely, X.509,

Kerberos 5 and Public Key Cryptography for Initial Authentication (PKINIT) while working in the distributed environment. In Section 2, the process of issuing and revoking X.509 certificates and the list of some protocols that support these certificates are discussed. In Section 3 Kerberos basics and its authentication process to access the services of the application server is discussed. Section 4 provides how the authentication process changes from Kerberos 5 to PKINIT and in Section 5, the comparative analysis of these protocols is presented.

II-X.509 Certificates basics

X.509 is an ITU-T (ITU Telecommunication Standardization Sector) standard for PKI in cryptography. It defines specific formats for PKC (Public Key Certificates). X.509 certificates uses Abstract Syntax Notation One (ASN.1) which is an interface description language for defining the format of data to be transferred between the Open System Interconnection (OSI) and also this is International Standards Organization (ISO) data representation used to achieve interoperability between platforms [2,3]. Infact, it is the uniform language that helps sending and receiving computers to exchange data.

2.1 Versions of X.509

There are three versions for the X.509 Certificate and the fields defined for the versions vary and depicted in Figure 1.

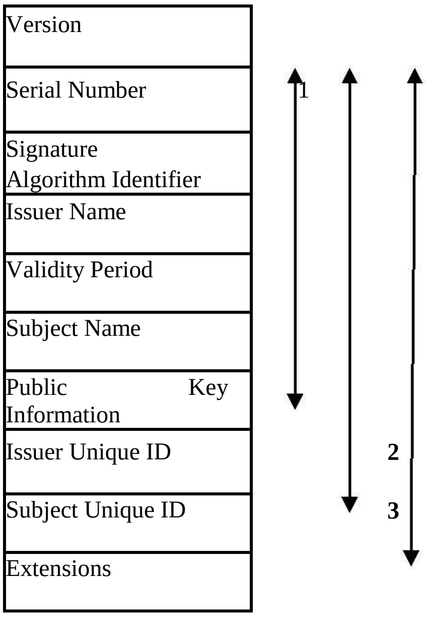


Figure 1: Certificate Structure [4]

SSL was the first protocol that was tested against X.509 certificates. Other protocols that can be used with the X.509 includes TLS, IPSec, S/MIME, HTTPS, LDAPv3 and EAP and their working format with X.509 is defined in the RFC 5280 and RFC 4945.

2.2 Digital Signatures

CA when issuing the certificate has the authority to digitally sign the certificate. Digital Signature actually refers to the encryption of the hashed message with the private key of the CA. The public key of the CA is known to the public, so anyone can use that to verify the message and the digital signature. The benefit of using the digital signatures are authenticity, integrity and non-repudiation to electronic documents. The digital signature are valid in India after the enactment of Information Technology Act 2000. The signature algorithms that are used for signing by the CA is listed in [5,6,7].

2.3 Revoking a Certificate

Version1 was released in 1988 that assumed the hierarchical system of certification authorities. Later on, in version 2 to support directory access control two more fields Issuer unique ID, Subject Unique Id were added in 1993. Finally, version 3 came into existence which defines the formatting used for certificate extensions in 1996.

The main fields of X.509 certificates include subject name and the public key. In this protocol, CA is the Trusted Third Party (TTP). The certificates are issued by CA according to the policy defined at the root level. CA also uses the public/private key pair and this public key can be identified by using the Authority Key Identifier extension. The operations that public key can perform is restricted by the extension KeyUsage also defined in version 3. Policy Constraints extensions helps the CA to set the constraints path validation by prohibiting policy mapping.

CA also maintains and publishes a list of certificates that can no longer be used. He has the authority to revoke the certificates where the reason could be the certification expiration or private key might get compromised and many more. The status of the certificates can be checked by the CA in two ways:

a) Through Certificate Revocation List (CRL).

b)Through Online Certificate Status Protocol (OCSP) which provides more up-to-date status than CRL.

2.4 Process of Issuing a Certificate

The process of issuing the X.509 certificate is shown in Figure 2.

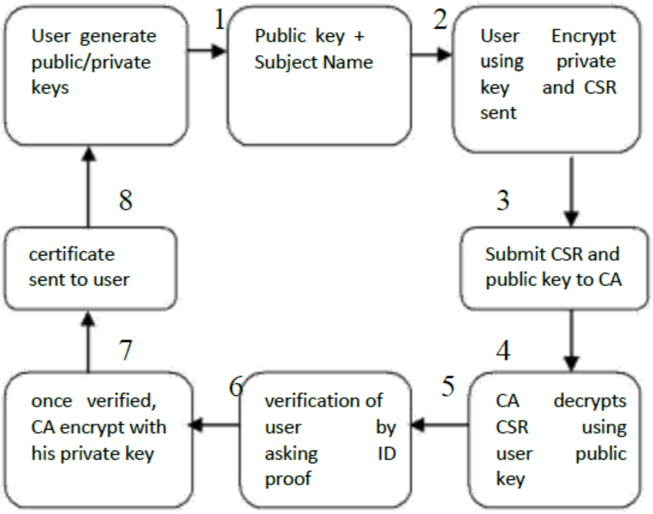

User generate public/private keys
1
Public key + Subject Name
2
User Encrypt using private key and CSR sent
3
Submit CSR and public key to CA
4
CA decrypts CSR using user public key
5
verification of user by asking ID proof
6
once verified, CA encrypt with his private key
7
certificate sent to user
8

Figure 2: Steps for Issuing X.509 Certificate

These certificates can be issued either by using keytool in java or using OpenSSL. The main applications of digital certificates include secure email, secure web communications, digital signing of software files, IPSec authentication and many more.

III-Basics of Kerberos 5

X.509 certificates are based on the concept of asymmetric encryption where private key is owned by the CA and the public key is distributed to the user who wish to own the certificate from the CA. The problem in this protocol is to maintain the database of these keys. Kerberos is the protocol which eliminates this need of possessing and securing the private keys. It allows the users to authenticate themselves without transmitting their passwords on the network. It is the network authentication protocol and the primary version was released at MIT by Steve Miller and Clifford Neuman in late 1980s with the RFC 4120. Version 5 was released by John Kehl and Clifford Neuman in 1993 with the RFC 1510. It was based on Needham-Schroeder symmetric-key protocol. In practice, the cryptographic scheme used within Kerberos is Data Encryption Standard (DES).

The name Kerberos was taken from the three-headed dog who guards the entrance of the Hades. Here, these three heads resembles the Authentication, Authorization and Auditing. Also, there are three servers namely Key Distribution Center (KDC), Authentication Server (AS) and the Ticket Granting Server (TGS).

The main aim behind Kerberos is to allow the nodes to communicate over the insecure network where the passwords will never travel over the network and also will be discarded immediately from the client machine once it is used. Moreover, these passwords never get stored on the client as well as on server machine.

3.1 Need of Kerberos in Distributed environment

While accessing services in an open distributed environment, there is possibility of many types of threats which could be:

a) Network address can be altered.

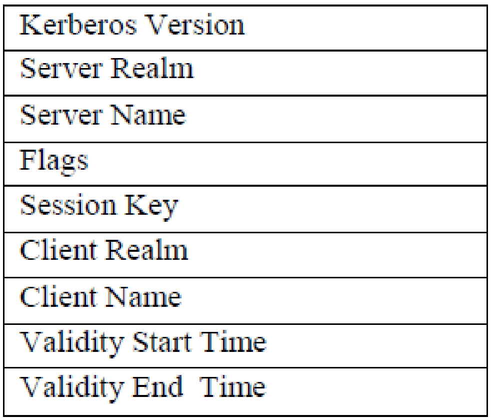
b) Replay attack to disrupt operations.

c) A hacker may get access to any workstation and pretend to be an authorized user.

Therefore, workstations including client and server which is considered to be in the Kerberos realm need to do mutual authentication before starting any communication. All the workstations need to be registered with the Key Distribution Center (KDC) and workstations beyond the domain which are not registered cannot do any kind of communication. Every client needs to share secret with KDC also.

3.2 Ticket Structure of Kerberos 5

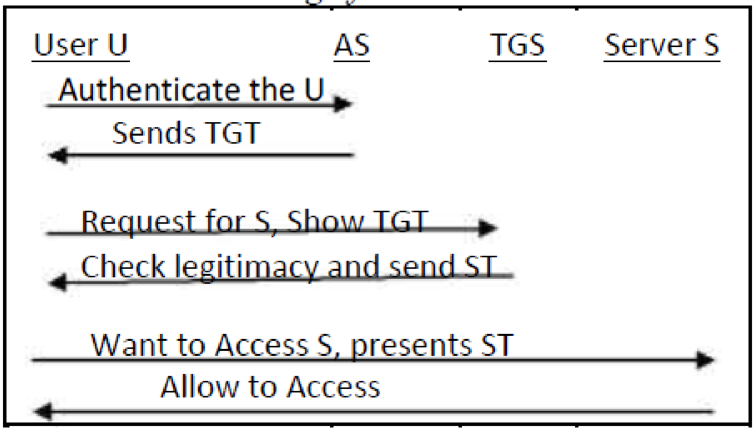
The structure of ticket that is generated in Kerberos is shown in Figure 3.



Kerberos Version
Server Realm
Server Name
Flags
Session Key
Client Realm
Client Name
Validity Start Time
Validity End Time

Figure 3: Ticket Structure [4]

3.3 Working of Kerberos 5



3. Now, client interacts with the TGS and after proving its authenticity, requests for a ticket to access services of a application server in the domain and also send the name of the application server with which it wants to initiate communication.

4. Once TGS is assured about the legitimacy of the user, it sends back client, a service ticket ST to access the Application Server, name and a timestamp.

5. Now, with the Service ticket and the authenticator, the client can communicate application server .

Whenever a client wants to initiate a new session, this exchange of messages is mandatory.

The working of Kerberos 5 starts with the authentication process between the client machine and Authentication Server. The process is shown in the Figure 4 and works as follows:

1. A nonce is generated at the client machine which along with the user name is send to Authentication Server (AS) where there is a database of already registered users.

2. If the name matches with the user name in the database, then the AS sends back two encrypted components to the client. First component consists of freshly generated Authentication Key (AK), Client name (C) and timestamp and all these are encrypted with the key which is shared between AS and Ticket Granting Ticket (TGT). The second component consists of AK,

nonce which binds this response to Client's original request, timestamp and name of Ticket Granting Server (TGS) and here these all are encrypted with a key derived from user password.

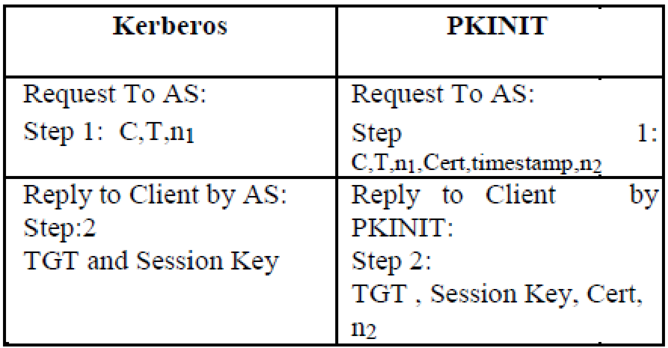
It is an extension to Kerberos 5 [RFC 4120]. Infact, it can be considered as the combination of X.509 and Kerberos 5 as it supports both of them. This protocol uses public key cryptography so that shared secrets can be avoided between the client and the Authentication Server (AS) that is required in the Kerberos 5. The major drawback of Kerberos 5 is that database of shared secrets need to be maintained by the KDC. PKINIT works similar to Kerberos but modifications has been done in the first step when authentication process starts with the AS. Therefore, it relies on asymmetric encryption and digital certificates in the first round and retains symmetric encryption in the later rounds. The efforts to maintain long-term keys as in Kerberos can then be ignored. In this way, pre-authentication can be avoided or may be used as per the domain needs. It supports Diffie-Hellman Key Exchange method and Public Key Encryption (RSA) method to encrypt the reply to the client. Although, RSA is less commonly used [9].

PKINIT [RFC 4556] allows authenticating the user with X.509 certificate. A

Distinguished Name (DN) is the field that is used in the X.509 which determines the name of the client. To do the mapping, this DN can be hashed and can be used as a client name in Kerberos. The Public key is used for Signature Verification and the private key is used for Signature generation key. Also, public keys are signed by some Certification Authority (CA).

4.1 Extensions in the Initial Authentication

The difference in the initial authentication steps of both Kerberos 5 and PKINIT is shown in Figure 5.



Kerberos	PKINIT
Request To AS: Step 1: C,T,n_1	Request To AS: Step 1: C,T,n_1,Cert,timestamp,n_2
Reply to Client by AS: Step:2 TGT and Session Key	Reply to Client by PKINIT: Step 2: TGT , Session Key, Cert, n_2

Figure 5: Initial Step of Kerberos 5 and PKINIT

REQUEST: In Kerberos 5, during the first step of request client name, server ID and a nonce n_1 is sent to AS whereas when using PKINIT, along with Client name, server ID, nonce, a client certificate, his signature on timestamp and another nonce n_2 is send to AS.

REPLY: In the reply to client , the AS sends TGT and session key where TGT is encrypted with KDC private key and both TGT and session key are encrypted with the clients secret key. It also contain nonce n_1, matching of this nonce with the message request nonce is mandatory. Reply from PKINIT to client includes TGT, Session key along with client certificate as well as its signature using its secret key and the nonce n_2 from client's request. All of this is encrypted with client's public key.

V-Comparison of X.509, Kerberos 5 and PKINIT

Comparison of the protocols depicted in Figure 6.

Protocol	**X.509**	**Kerberos 5**	**PKINIT**
Channel	Many to One	One to one	Many to one
Encryption	Asymmetric	Symmetric	Both
Algorithm	RSA	DES	Diffie-Hellman/ RSA
RFC	5280	4120	4556
Generate	Certificate	Ticket	Certificate/Ticket
Storing	Public key	Private key	Nothing
Use of Public Key Cryptography	Yes	No	Yes
Pros	Authenticity, Integrity and Non-Repudiation	Single-Sign on, non-transmission of passwords, strong authentication	Simplifies key sharing, scalability and improved security
Technical Deficiencies	Compromise of private keys	Double encryption, session keys, password attacks	Complex computations as larger keys are required.
Trusted Third Party	CA	KDC	CA and KDC

Figure 6: Comparison of Protocols

VI-Conclusion

In distributed systems, computers at remote locations communicate with each other through message passing. Network protocols plays a vital role during the first phase i.e. Authentication. Each protocol have their own formats and procedure to initiate the communication. To meet the requirements of open distributed network, comparison of the three protocols X.509, Kerberos 5 and PKINIT is done. These network protocols helps heterogeneous systems in interoperability and data sharing. X.509 and PKINIT supports public key cryptography whereas Kerberos does not because it is based on the scheme of shared secret. After the comparison of these protocols, it has been observed that PKINIT is intended to add more flexibility, security and administrative convenience as compared to Kerberos 5 and X.509 as it supports features of both as well as provides its own. It has been found that there is possibility of mapping between these protocols which will be done in future and implementation will be tested in the cloud environment.

References:

[1] http://slideplayer.com/slide/5154894/, Dr. Rajkumar Buyya.

[2] https://en.wikipedia.org/wiki/X.509, (Accessed on: 9 November,2017).

[3]]https://en.wikipedia.org/wiki/Abstract_Syntax_Notation_One, (Accessed on: 24 June,2017).

[4] J. Marasinghe, S. Gadawala, "Critical Analysis of X.509 and Kerberos for Distributed Authentication," 2007, pp.1-9.

[5] W. Polk, "RFC 3279: Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," 2002, pp.1-27.

[6] J. Schaad, "RFC 4055: Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," 2005, pp.1-25.

[7] S. Leontiev, D. Shefanovski, "RFC 4491: Using the GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 Algorithms with the Internet X.509 Public Key Infrastructure Certificate and CRL Profile," 2006, pp.1-20.

[8] F.Butler, I.Cervesato, Aaron D.Jaggard, A.Scedrov, C.Walstad, "Formal analysis of Kerberos 5," Theoretical Computer Science 367, Elsevier, 2006, pp.57-87.

[9] S.Moore, P.Miller, "RFC 8070: Public Key Cryptography for Initial Authentication in Kerberos (PKINIT) Freshness Extension," 2017, pp.1-9.