International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Providing Kerberos Authentication Using Elliptic Curve Cryptography

Monalisha Mishra[1], G. Sujatha[2]

[1]M. Tech (IT Department) SRM University Chennai, India

[2]Assistant Professor SRM University Chennai, India

Abstract: ***KERBEROS is a key distribution and user authentication service developed at MIT. Kerberos can be described as a trusted third-party authentication system. After a user authenticates with Kerberos, their communications can be encrypted to assure privacy and data integrity. The public key based protocols PKINIT, PKCROSS and PKTAPP help in implementing public key cryptography in different stages of Kerberos framework. The PKINIT extension provides a method for integrating public key cryptography into the initial authentication exchange, by using asymmetric-key encryption algorithm in pre authentication data fields. PKCROSS has been proposed to simplify the administrative burden of maintaining cross realm keys so that it improves the scalability of Kerberos in large multi realm networks. Finally PKTAPP has been proposed to improve the scalability of PKCROSS. The aim of this project is to study the implementation of these Kerberos protocols through a real time application system. In this paper we report how Elliptic curve cryptography provides a strong security by using smaller key size, as well as decreases the memory size and CPU time as compared to other algorithms.***

Keywords: authentication-Kerberos; Public key cryptography; Elliptic curve cryptography; PKINIT; PKCROSS; PKTAPP

1.Introduction

Information security uses cryptography to encrypt and decrypt data. Cryptography enables us to store sensitive information or transmit it across insecure networks so that it cannot be read by anyone except the intended recipient. A cryptographic algorithm, or cipher, is a mathematical function used in the encryption and decryption process. A cryptographic algorithm works in combination with a key—a word, number, or phrase—to encrypt the plaintext. The same plaintext encrypts to different cipher text with different keys. The security of encrypted data entirely depends on two things: the strength of the cryptographic algorithm and the secrecy of the key. The keys used in cryptographic algorithms may be a session key (Symmetric encryption) or a public/private key pair(Asymmetric encryption)[1]. Asymmetric encryption is more safe than symmetric since session key must be securely shared between the client and server**.**

Symmetric-key algorithms are a class of algorithms for cryptography that use the same cryptographic keys for both encryption of plaintext and decryption of cipher text. The keys may be identical or there may be a simple transformation to go between the two keys. The keys, in practice, represent a shared secret between two or more parties that can be used to maintain a private information link. This requirement that both parties have access to the secret key is one of the main drawbacks of symmetric key encryption, in comparison to public-key encryption. Public-key cryptography refers to a cryptographic system requiring two separate keys, one of which is secret and one of which is public. Although different, the two parts of the key pair are mathematically linked. One key locks or encrypts the plaintext, and the other unlocks or decrypts the cipher text. Neither key can perform both functions by itself. The public key may be published without compromising security, while the private key must not be revealed to anyone not authorized to read the messages.

2.Kerberos System

The Kerberos protocol defines how clients interact with a network authentication service. Clients obtain tickets from the Kerberos Key Distribution Centre (KDC), and they present these tickets to servers when connections are established. Kerberos tickets represent the client's network credentials. The Kerberos authentication protocol provides a mechanism for mutual authentication between entities before a secure network connection is established. In a client/server application model, clients are programs acting on behalf of users who need something done. This might be opening and using a file, accessing a mailbox, downloading a file, sharing a file. Servers are programs providing services to clients such as file storage, mail handling, query processing, and print spooling. Clients initiate action and servers respond.

In the Kerberos protocol model, every client/server connection begins with authentication. Client and server, in turn, step through a sequence of actions designed to verify to the party on each end of the connection that the party on the other end is genuine. If authentication is successful, session setup completes and a secure client/server session is established. The Kerberos protocol makes use of:

- Key authentication
- Authenticator messages
- Ticket-granting tickets
- Service-granting ticket

The current version of Kerberos is v5, which was developed in 1993[2]. This is the version on which Microsoft's implementation in Windows 2000/XP/Server 2003 is based. Kerberos was named after Cerberus, the three-headed dog of Greek mythology, because of its three components:

- A Key Distribution centre (KDC), which is a server that has two components: an Authentication Server and a Ticket Granting Service.
- The client (user)
- The server that the client wants to access

Figure1 shows how the logon process works with Kerberos as an authentication method:

1. To log on to the network, the user provides an account name and password.
2. The Authentication Server (AS) component of the KDC accesses Active Directory user account information to verify the credentials.
3. The KDC grants a Ticket Granting Ticket (TGT) that allows the user to get session tickets to access servers in the domain, without having to enter the credentials again .
4. When the user attempts to access resources on a server in the domain, the TGT is used to make the request. The client presents the TGT to the KDC to obtain a service ticket.
5. The Ticket Granting Service (TGS) component of the KDC authenticates the TGT and then grants a service ticket. The service ticket consists of a ticket and a session key. A service ticket is created for the client and the server that the client wants to access.
6. The client presents the service ticket to create a session with the service on the server. The server uses its key to decrypt the information from the TGS, and the client is authenticated to the server.
7. If mutual authentication is enabled, the server also authenticates to the client.

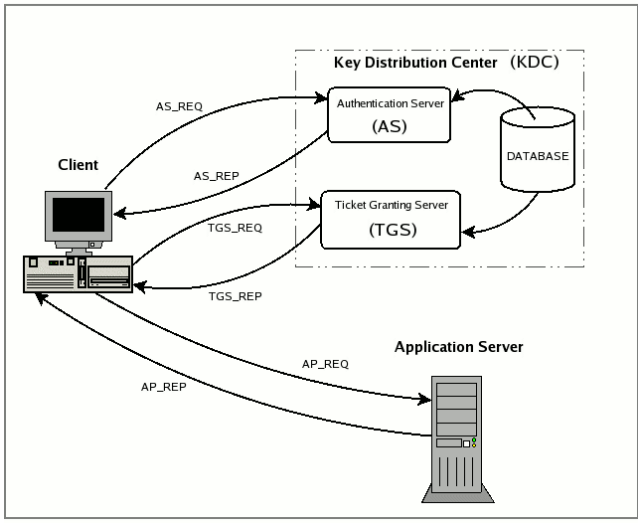
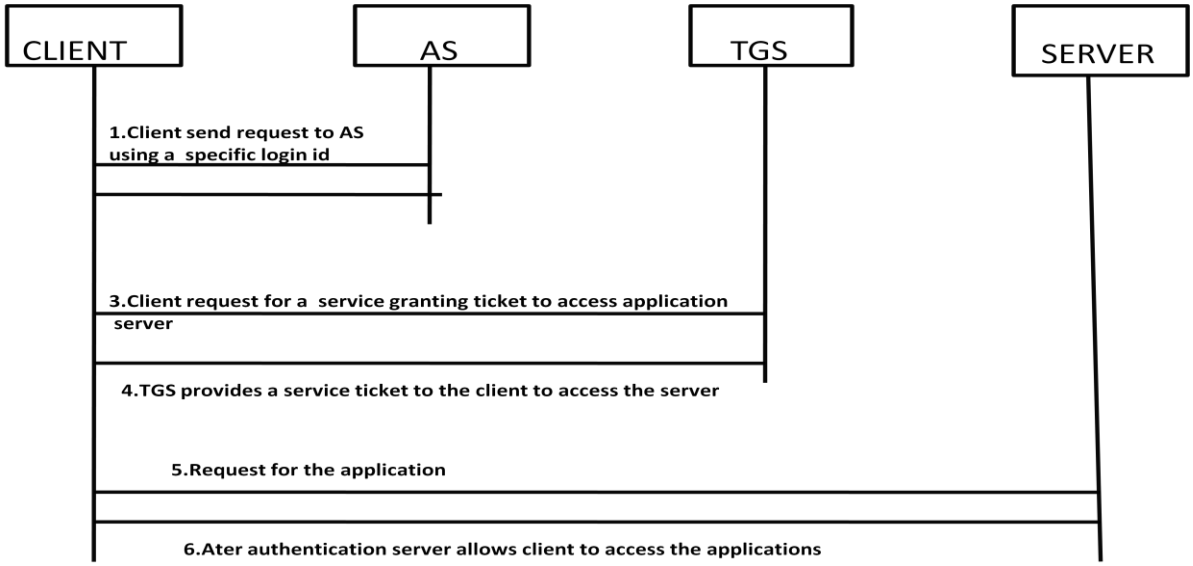


Figure 1: Kerberos Authentication

3. System Implementation

The implementation of the proposed system begins with sending user name and password of client to AS which in turn sends TGT after verification of username. Then the client sends service ticket request to server with received TGT. After the verification of TGT, server sends back SGT to client. If the client wants to access a service in remote KDC then it approaches the local TGS to get the TGT of remote TGS. The local TGS forwards the client's request to remote TGS with some alterations in the packet. Remote TGS verifies the message and replies with TGT. Local TGS forwards it to the client. It is then used by the client to get SGT from Remote TGS. To make the implementation easier it is then divided into three parts PKINIT, PKCROSS and PKTAPP. The key generation for the above three parts are generated using ECC algorithm.

4.Implementation of PKINT

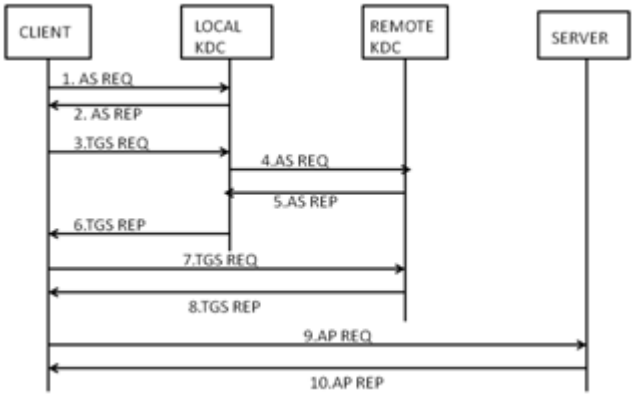


The PKINIT Internet draft specifies that considerable message content must be added to the initial Kerberos Version 5 exchanges to replace the user secret key authentication with public key authentication.

a. verifying the client's digital signature contained in AS_REQ, so that it avoids requests from masquerading intruders.
b. Encrypting the TGT and session key in the server response, so that an eavesdropper cannot obtain the user's credentials.

The client has to register itself to get a valid login id and password. The client then sends a request by using a specific login id (that acts as a signature) for TGT to AS. AS verifies the message and replies with a TGT and a shared key of the client and TGS. At client side, it generates an authenticator and encrypts it with the shared key that was sent by AS in previous message transfer. TGS compares the ticket information with the received authenticator and sends back SGT with a session key that is to be shared between client and the application server. Using the secret key the client will be able to access the data in the application server for example downloading files, uploading files, sharing files with other client.

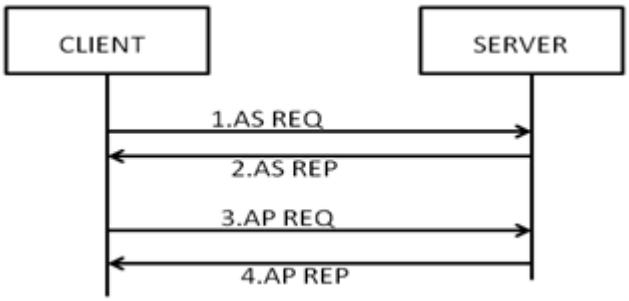
5.Implementation of PKCROSS



While PKINIT addresses the issue of managing secret keys for a large number of clients, it does not address the issue of key management among a large number of realms. A logical extension of PKINIT is the use of public key encryption for multiple-realm KDC-to-KDC authentications. This is the subject of the PKCROSS Internet draft specification.

PKCROSS picks up the multiple realm authentications at the point at which the client has already obtained a TGT. Assume that the client has requested access to a server in a remote realm. Its local KDC initiates a PKCROSS transaction with the appropriate remote KDC. With a few minor variations, the KDC-to-KDC authentication is performed using the PKINIT protocol. Once the client possesses a remote TGT, it may request additional service tickets in the remote realm without involving the local KDC. In case of real time application system the communication takes place between one application server and another mailing server. If a user (client) wants to access the remote mailing server first it will request to the local application server. Then the local application server will get the ticket and a secret key from the remote server and send it back to the local client. Finally with the help of the secret key local client will be able to connect to the remote server directly.

6.Implementation of PKTAPP



PKTAPP is a more efficient protocol than traditional Kerberos from a message exchange perspective. The client may deal directly with the application server. The AS-REQ message, the first message submitted by the client, contains the client's login id and the service ticket requested. The server response, an AS-REP message, contains the server's identity and the session key encrypted with the server's private key. After authentication, the client requests an application service ticket. The entire authentication process is reduced to a total of two message pairs.

7.Implementation Result and Discussion

In symmetric key authentication, the shared key may be hacked by malicious users. If the secret key is known then the whole message can be revealed. Hence asymmetric key cryptography is better than symmetric key cryptography. But there are also some drawbacks in asymmetric cryptography which is based on the algorithm that is chosen to generate key pairs. In the existing system, RSA algorithm is used for the generation of key pairs. The minimum key size of RSA algorithm is 1024bits and it is directly proposed to security. Because of its large key size, more memory space is needed and hence it decreases the performance of the system. Drawbacks in asymmetric cryptography are solved by changing the algorithm for generating key pairs which is used for encryption/decryption. The primary benefit promised by ECC is a smaller key size, reducing storage and transmission requirements—i.e., that an elliptic curve group could provide the same level of security afforded by an RSA-based system with a large modulus and correspondingly larger key—e.g., a 256-bit ECC public key. The system application is developed using ASP.NET in windows environment. The whole application divided into Client application that requests for a particular service, an authentication server that checks the authentication of a client, a ticket granting server that provides a particular service ticket to the client and finally an application server that provides the desired service requested by the client. Hence the total system application divided into 3 modules i.e PKINIT, PKCROSS, PKTAPP that shows the public key extension into traditional Kerberos system.

8.Conclusion

This paper focuses on the public key based extension PKINIT, PKCROSS, PKTAPP. Concerning the security issue, in traditional Kerberos environment, the KDC must remember every user's secret key. Thus, if an unauthorized individual gains even read-only access to the KDC's

database, the security of the KDC is completely compromised. Alternatively, as public key-enabled Kerberos uses public keys for encrypting TGTs, read-only access into a KDC's database would not compromise security at all. The only way for an attacker to violate the public key-enabled Kerberos security is by obtaining a write access to the X.509 Certificate Authority directory. This is much more difficult than just passively reading secret-keys in traditional Kerberos databases. Hence the use of ECC enhanced the security of the cryptosystem by making it difficult for the unauthorized user to decrypt the messages. As ECC provides a strong security by using smaller key size, it decreases the memory size and the CPU time as compared to other cryptosystem. Indeed, this application can be enhanced by using other strong cryptographic algorithm.

References

[1] Sufyan T. Faraj Al-Janabi and Mayada Abdul-salam Rasheed " Public-Key Cryptography Enabled Kerberos Authentication" 2011 Developments in E-systems Engineering
[2] National Institute of Standards and Technology, Entity Authentication Using Public Key Cryptography, Federal Information Processing Standards Publication 196, February 1997
[3] Fabrice KAH GIAC Security Essentials Certification (GSEC) Practical v1.4b Option #1 November 2003 "understanding-kerberos-v5-authentication-protocol_3462_"
[4] Harbitter and D. Menasce, "Performance of public-key-enabled Kerberos authentication in large networks," In Proceedings of 2001 IEEE Symposium on Security and Privacy, Oakland, California, IEEE Computer Society Press, 2001
[5] Downard, "Public-key cryptography extensions into Kerberos," IEEE Potentials, pp. 30-34, December 2002-January 2003.
[6] s. Maria Celestin Vigila1 , K. Muneeswaran "Implementation of Text based Cryptosystem using Elliptic Curve Cryptography" ©2009 IEEE
[7] Moncef Amara 1 and Amar Siad "Elliptic curve cryptography and its application." 2011 7th International Workshop on Systems, Signal Processing and their Applications (WOSSPA)
[8] E. El-Emam, M. Koutb, H. Kelash, and O. Farag-Allah, "A network authentication protocol based on Kerberos," IJCSNS International Journal of Computer Science and Network Security, Vol.9, No.8, August 2009
[9] Ram Ratan Ahirwal, Manoj Ahke "Elliptic Curve Diffie-Hellman Key Exchange Algorithm for Securing Hypertext Information on Wide Area Network" International Journal of Computer Science and Information Technologies, Vol. 4 (2) , 2013, 363 - 368
[10]" A (Relatively Easy To Understand) Primer on Elliptic Curve Cryptography" 24 Oct 2013 by Nick Sullivan.
[11] https://blog.cloudflare.com/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/
[12] BAI Qing-hai1, 2, ZHANG Wen-bo1, JIANG Peng1, LU Xu1 "Research on design principle of Elliptic curve public key cryptography and its implementation." 2012 International Conference on Computer Science and Service System