ISSN 1870-4069

Autenticación para acceso a datos distribuidos basado en Kerberos

Juan Alejandro Ibáñez Ramírez, Francisco de Asís López-Fuentes

Departamento de Tecnologías de la Información
Universidad Autónoma Metropolitana-Cuajimalpa (UAM-C), Ciudad de México, México

flopez@correo.cua.uam.mx

Resumen. Hoy en día la mayoría de datos se encuentran almacenados en sitios remotos que por lo general son sistemas distribuidos (como Dropbox, OneDrive, Google Drive, Outlook). Por lo tanto, surge la necesidad de proteger el acceso a estos datos para garantizar su privacidad e integridad ante el creciente número de ataques a la seguridad, entre ellos los virus informáticos, los troyanos, o el Ransomware [10]. Es por esto que se requieren sistemas que implementen un acceso seguro a datos distribuidos. Uno de los objetivos principales en la seguridad de la información es implementar controles de acceso, los cuales integren políticas y criterios que indiquen bajo qué circunstancias se deberá otorgar acceso a los recursos de un sistema. Este artículo presenta un sistema de autenticación entre múltiples dominios distribuidos basado en Kerberos para tener el control de acceso a datos distribuidos en una red de computadoras, haciendo uso de técnicas de cifrado de mensajes y de los datos.

Palabras claves: autenticación, seguridad, control de acceso, sistemas distribuidos.

Authentication of Access to Distributed Data Based on Kerberos

Abstract. Today most of the data is stored in remote sites that are usually distributed systems (such as Dropbox, OneDrive, Google Drive, Outlook). Therefore, there is a need to protect access to this data to ensure its privacy and integrity in the face of increasing security attacks, including computer viruses, Trojans, or Ransomware [10]. These are important reasons to require implementing secure access to distributed data. One of the main objectives in information security is to implement access controls, which integrate policies and criteria that indicate under what circumstances access to resources of a system should be granted. This article presents a Kerberos-based distributed multi-domain authentication system for controlling access to distributed data in a computer network using data encryption techniques and data.

Keywords: authentication, **s**ecurity, access control, distributed system.

1. Introducción

Las tecnologías de la información y comunicación (TICs), han permitido que las personas actualmente estén conectadas y comunicadas en todo momento sin importar el lugar en donde estén. Esta ubicuidad requiere que los sistemas de información estén distribuidos en diferentes sitios y que los datos puedan ser accedidos también desde cualquier lugar. Sin embargo, los sistemas distribuidos al funcionar dentro de un ambiente abierto de comunicación son susceptibles a diferentes ataques a la seguridad. La seguridad se refiere a las medidas de procedimiento lógicas y físicas orientadas a la prevención, detección y corrección de casos de mal uso, así como a las características que debe tener un sistema de cómputo para resistir a ataques. Los ataques más comunes a la seguridad en un sistema de cómputo son los ataques de intercepción, modificación o fabricación de mensajes. Uno de los objetivos principales en la seguridad de la información es implementar controles de acceso, los cuales integren políticas y criterios que indiquen bajo qué circunstancias se deberá otorgar acceso a los recursos de un sistema. Los mecanismos básicos de control de accesos son integridad, confidencialidad, autenticación y autorización. Las principales características de estos mecanismos son [8, 4, 6]:

- *Integridad* sirve para prevenir cambios impropios o no autorizados sobre el contenido de la información.
- *Confidencialidad* es el mecanismo que permite la ocultación de los recursos a entidades no autorizadas. El uso de la criptografía ayuda en la tarea de preservar la confidencialidad.
- *Autenticación* ofrece mecanismos que permiten una identificación correcta del origen del mensaje, asegurando que la entidad no sea falsa.
- *Autorización* es el mecanismo que determina si una entidad una vez autenticada está autorizada para obtener el acceso al recurso solicitado.

Por otro lado, las políticas de control de acceso en un sistema de seguridad deben ser implementadas para ayudar a establecer quién o quiénes tendrán acceso a los recursos del sistema, quiénes son los dueños de los recursos y qué permisos tendrán los usuarios sobre éstos. De acuerdo con los autores en [7], los actores para una política de acceso son:

- *Autoridades y regímenes:* Las autoridades son responsables de definir los medios de acceso permitidos y clasificar los recursos, determinar las autorizaciones, y especificar los niveles de confianza aplicables, mientras que los regímenes pueden ser entidades relacionadas entre sí.
- *Recursos:* Una política de seguridad debe definir los recursos a los que se aplica: éstos pueden ser elementos intangibles (datos o información), o elementos de hardware.
- *Clasificación de recursos:* Los recursos se pueden clasificar según su nivel de riesgo, costos o las consecuencias asociadas con la gestión de su acceso.
- *Contexto de acceso:* Circunstancias en las que se solicita el acceso o los medios por los cuales se proporciona el acceso, por ejemplo, la hora de la solicitud.

- *Uso permitido:* Usos posibles para un recurso (leer, modificar, crear, eliminar, etc.)
- *Partes:* Pueden ser referidas por identificador (usuario #6718), o por atributo (usuarios del grupo "Administradores").
- *Confianza en la autenticidad:* Representa las reglas de acceso dependiendo de la confianza del sistema en la autenticidad de una parte.

En este trabajo se pretende resolver problemas relacionados al control de acceso para garantizar la seguridad de datos distribuidos en diferentes sitios al integrar el modelo de control de autenticación Kerberos [1], en un ambiente distribuido. Para alcanzar esta meta nuestro modelo propuesto pretende cubrir características como: funcionar dentro de un ambiente de red inalámbrica, resistir la adivinación de contraseñas, resguardar los datos ante una petición falsa y principalmente ofrecer un servicio único de autenticación de usuarios entre varios dominios distribuidos, garantizando el cifrado de mensajes y de datos.

El resto de este artículo tiene la siguiente organización. En la sección 2 se presenta información sobre los sistemas de autenticación, principalmente de Kerberos. La sección 3 presenta el modelo propuesto. Una descripción de la implementación se describe en la sección 4. El artículo concluye en la sección 5.

2. Trabajo relacionado

Algunos de los temas a resolver por los sistemas distribuidos actuales consisten en la necesidad de garantizar un acceso efectivo y seguro a servicios ofrecidos por ciertos proveedores en la nube (infraestructura como servicio, plataforma como servicio y software como servicio). Para este tipo de escenarios, los autores en [3], proponen la utilización del modelo Kerberos. Después de revisar la propuesta anterior, se puede detectar que una posible mejora a este sistema consistiría en la utilización de un algoritmo de cifrado más actual en lugar de DES.

Por otro lado, la empresa estadounidense de desarrollo de software Vandyke [11], plantea el tema de la autenticación de usuarios para la transferencia de archivos usando el protocolo SFTP (Secure Shell File Transfer Protocol), a través de un programa desarrollado por ellos, el cual crea un túnel cifrado de comunicaciones utilizando el modelo cliente-servidor para establecer conexiones remotas. Entre las características de este programa se encuentran que la comunicación se genera mediante el protocolo SSH (Secure Shell), el sistema ofrece cifrado de comunicación SSL (Secure Sockets Layer) y no de archivos en la fuente y que es de pago.

Otro de los temas relevantes para un sistema de seguridad es el control de la autenticación de usuarios. Para lograr este objetivo en [5], los autores plantean usar el protocolo Kerberos para ofrecer un servicio de autenticación que permita acceder a un recurso hospedado en la nube. Además, presentan una modificación al modelo añadiendo un protocolo de Autenticación Distribuida llamado DSA el cual permite al sistema que, mediante una clave de sesión dinámica, realice una autenticación previa antes de que el usuario pueda acceder al servicio solicitado. Esta idea podría ser una posible aportación al presente proyecto.

En relación al tema de la transferencia de archivos usando el modelo Kerberos para el control de acceso a un servidor de descarga de archivos, Al-Ayed y Liu [2], plantean

Tabla 1. Comparativo de los diferentes sistemas de autenticación

Trabajo	Observaciones	Posible mejora
Implementation of Kerberos versión 5 in cloud computing in order to enhance the security issues [3]	- Control de acceso a servicios generales en la nube	- Uso de AES
Secure File Transfer with SSH [11]	- Arquitectura cliente-servidor. - El sistema sólo ofrece cifrado de comunicación y no de archivos en la fuente. - El sistema es de pago y no de uso libre.	- Arquitectura de autenticación distribuida. - Cifrado de archivos.
Distributed Authentication in the Cloud Computing Environment [5]	- Modificación al modelo Kerberos. - Protocolo de autenticación distribuida DSA. - Clave de sesión dinámica.	- Establecer alguna política de control de acceso a un recurso específico en la nube.
Using Kerberos Method to Secure File Transfer Sessions [2]	- Detección de intrusiones basado en el modelo de Markov. - Control de acceso a un servicio FTP.	- Control de acceso a un modelo distribuido mediante autenticación de dominios distribuidos. - Cifrado de archivos.
A lightweight authentication and authorization solution based on Kerberos [9]	- Protocolo ligero. - Resistente a ataques de replicación de mensajes y de adivinación de contraseñas.	- Mejorar las deficiencias del protocolo para hacerlo más robusto.

usar el protocolo FTP (File Transfer Protocol) y no usar SSL para cifrar las conexiones, ya que consideran que el uso de Kerberos es una solución más robusta. También sugieren usar una máquina de aprendizaje basada en el modelo de Markov para detectar intrusiones tomando como entrada la secuencia de posibles estados del modelo Kerberos.

Con el objetivo de ofrecer una versión ligera del modelo Kerberos para resolver el problema de la autenticación y autorización de derechos digitales, Zhang et al. [9], formulan un rediseño del modelo Kerberos con la intención de reducir la carga de cada nodo del sistema. Mencionan que, aunque el protocolo ligero resiste a ataques de replicación de mensajes y de adivinación de contraseñas, aún cuenta con algunas deficiencias.

Después de analizar las propuestas anteriores se puede concluir que el modelo Kerberos es muy flexible a modificaciones y puede integrarse dentro de diferentes tecnologías. Asimismo, el uso de Kerberos se justifica de forma general en todas las propuestas anteriores para responder a las necesidades actuales de autenticación. A continuación, la tabla 1 resume las principales características de cada propuesta, así como las posibles mejoras y aportaciones al presente trabajo.

3. Modelo

El modelo de autenticación propuesto toma como referencia el modelo Kerberos, el cual es un modelo creíble y funcional por las siguientes razones [7]:

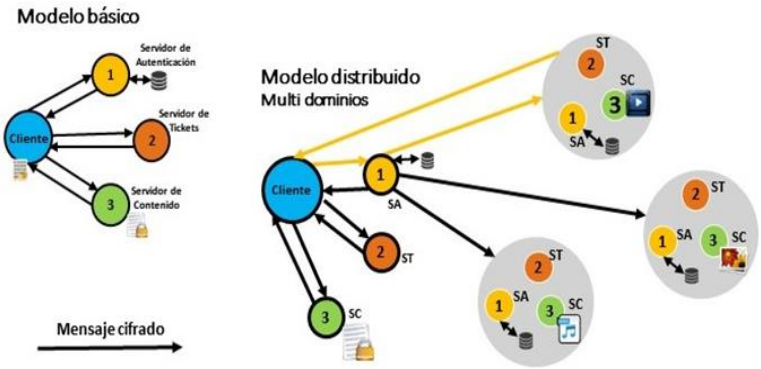

Fig. 1. Modelo básico y distribuido del sistema de autenticación.

- Ha sido ampliamente utilizado, probado, estudiado y está respaldado por una amplia comunidad de desarrolladores.
- Cumple con los requisitos de los sistemas distribuidos modernos, ya que desde el inicio fue concebido para trabajar dentro de entornos de comunicaciones abiertos.
- El modelo arquitectónico es sólido y funcional, lo cual ha permitido la evolución del modelo para una fácil integración con diferentes sistemas.
- El modelo actualmente sigue en funcionamiento e integrado dentro de varios sistemas, como Apache Hadoop, Ad-hoc Networks, Lot o SO Open Source y es una parte integral dentro de la infraestructura de la tecnología de la información actual.

El modelo consiste en tres nodos los cuales podrán estar bajo un solo dominio de red o distribuidos bajo dominios diferentes. El usuario podrá descargar de manera transparente los datos mediante un nodo cliente el cual establecerá conexión con el sistema, y éste a su vez se encargará de realizar las validaciones y enlaces para descargar el contenido distribuido.

Cada dominio tendrá una única base de datos, pero con conocimiento de otros dominios. Todas las comunicaciones o pase de mensajes entre nodos tendrán que ser procesadas mediante una función de cifrado, evitando así el envío de datos en texto claro. La figura 1 muestra de manera gráfica una descripción del modelo básico y distribuido del sistema de autenticación y las etapas de operación. En el modelo básico se muestran los tres servidores: de autenticación, tickets y de contenido, así como los intercambios de mensaje que cada servidor tiene con el cliente. Por otro lado, el modelo distribuido con multi dominios muestra como un cliente puede acceder a contenidos en otros dominios, pero debe hacerlo a través del servidor de autenticación de su propio dominio.

Cada dominio ajeno al del cliente tiene también tres servidores. Sin embargo, la autenticación entre los multi dominios se realiza por medio de los servidores de autenticación de cada dominio. En la figura 2 se puede observar la comunicación y envió de mensajes propuesta en cada etapa de comunicación. Asimismo, se indican las validaciones hechas por cada nodo del sistema.

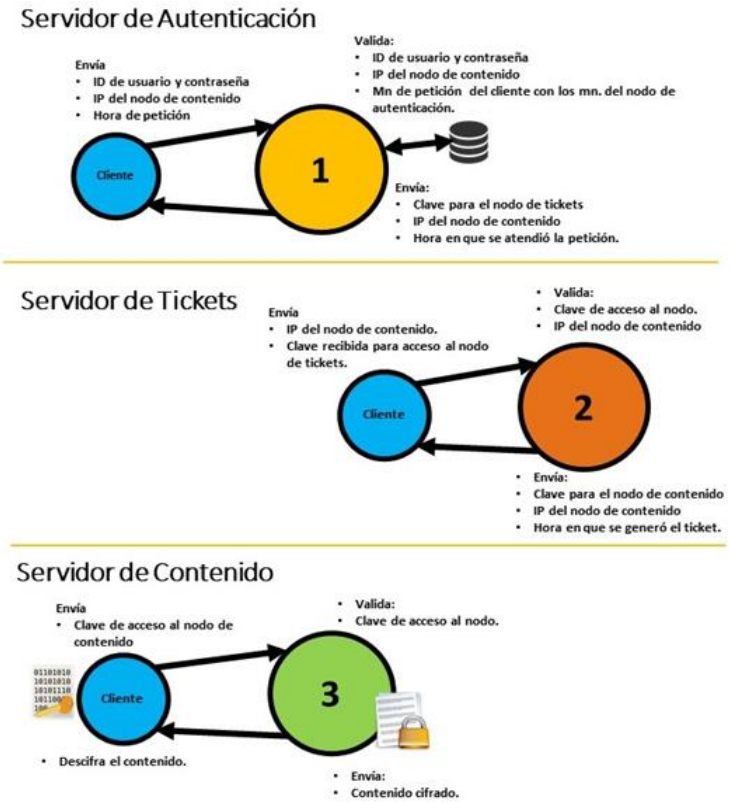

Fig. 2. Modelo de comunicación en cada nodo del sistema.

Un problema a considerar en la autenticación está asociada, con el tiempo de vida de un mensaje con el ticket de concesión, de esta manera [12]:

- Si es demasiado corta, entonces es solicitado repetidas veces por la contraseña.
- Si es demasiado larga, entonces existe una mayor oportunidad para un ataque de repetición.
- La amenaza es que un oponente se robe el ticket y lo use antes de que el tiempo expire.

En el modelo de comunicación de la figura 2, se considera este problema de tiempo de vida para cada ticket que se emite. De tal manera que se pueda determinar que quien presenta el ticket es el mismo cliente para el que se emitió dicho ticket. Una parte importante del funcionamiento de nuestra propuesta es la interoperabilidad entre dominios, ya que esta característica permite que clientes y servidores que pertenecen a diferentes organizaciones o dominios puedan ofrecer servicios entre ellos para usuarios previamente autenticados.

4. Implementación

Se ha puesto en práctica un prototipo básico del esquema de autenticación propuesto usando diferentes servidores en nuestro laboratorio. El sistema consta de 3 nodos los cuales están implementados mediante sockets para el SO Linux usando el lenguaje de programación C. Se usa TCP/IP como el protocolo de comunicación entre los servidores con el propósito de evitar pérdida de paquetes. Cada nodo del sistema es una aplicación independiente la cual puede estar bajo un solo dominio de red (IP), o distribuido bajo un diferente dominio; inclusive, cada uno de los nodos pudieran estar contenidos dentro de varios contenedores (Docker o Linux Container).

En lo que concierne al envío de mensajes entre nodos, se ha implementado una función encargada del cifrado de éstos, dicha función se explicará a detalle más adelante. En esta implementación y para fines de realizar pruebas iniciales, se asume por el momento, que todos los relojes dentro de cada entorno donde se ejecuta cada uno de los nodos están sincronizados. Por lo tanto, no se ha implementado un servicio de sincronización de relojes. Adicionalmente cada nodo fue programado para poder atender a diferentes clientes de manera paralela haciendo uso de la librería "pthread.h" para poder generar un hilo que atienda todas las peticiones de cada cliente. A continuación, se describe el funcionamiento de cada uno de los nodos.

4.1. Servidores

Servidor de autenticación

El servidor es el responsable de recibir la primera conexión del cliente junto con un mensaje compuesto por: ID, contraseña, IP del servidor de tickets y la hora de la petición. Una vez que se recibe el mensaje se validan en un arreglo estático de cadenas el ID, contraseña, IP de servidor de tickets y los minutos en que se atiende la petición. Para validar la hora se extrae como referencia la hora dentro del entorno donde se esté ejecutando el servidor de autenticación (los minutos y la hora deben ser iguales al recibido). Si las validaciones son correctas, se envía la clave para acceder al servidor de tickets y la hora en que se atiende la petición. En caso contrario se cierra la conexión, pero el servidor queda en espera de otras nuevas conexiones.

Servidor de tickets

Cuando se atiende una conexión, este servidor recibe y valida un mensaje con la clave y la dirección del servidor de contenido, si la validación es correcta, se genera un mensaje (ticket) compuesto por la hora en que se atiende la petición, la clave e IP para acceder al servidor de contenido. En caso contrario se cierra la conexión, pero el servidor queda en espera de más conexiones.

Servidor de contenido

En este servidor se encuentra el contenido al que quiere tener acceso el cliente. Cuando se recibe una conexión, el servidor valida si la clave enviada dentro del mensaje es correcta, si la validación es satisfactoria se crea un flujo de lectura de archivo y se envía el contenido bit por bit al cliente, en caso contrario la conexión se cierra y se queda en espera de nuevas peticiones.

4.2. Función de cifrado y descifrado

Para poder enviar mensajes de manera segura, se han programado dos funciones encargadas de cifrar y descifrar cada uno de los mensajes enviados en cada uno de los nodos durante cada una de las validaciones realizadas por el modelo distribuido, dicha función utiliza la librería "mcrypt.h" de uso libre, estas funciones hacen uso del algoritmo Rijndael el cual fue considerado para la especificación AES (Advanced Encryption Standard).

Cabe aclarar que la versión del algoritmo que es utilizada por el sistema es la versión que ofrece soporte para un tamaño de bloque y clave de 256 bits en el modo de cifrado de bloque CBC (Cipher Block Chaining Mode). La razón por la que se decidió utilizar esta versión y no la versión que hace uso de un bloque de 128 bits fue para darle mayor fortaleza contra posibles ataques y a su vez obtener un cifrado rápido y de bajo consumo de memoria.

Adicionalmente y con el fin de realizar pruebas rápidas al sistema, aún no se ha diseñado un servicio para el intercambio de claves. Actualmente el manejo de claves se realiza en tiempo de programación en cada nodo quedando definidas de manera estática, de forma similar, el vector de inicialización para el cifrado de bloques (IV), es establecido de manera estática en tiempo de programación. Sin embargo, no descartamos el poder integrar como una extensión al presente trabajo la integración de una función que haga uso de un algoritmo de cifrado asimétrico o de llave pública en conjunto con la implementación de un servicio de gestión de claves.

Aplicación cliente

Para poder acceder al sistema es necesario disponer de la aplicación cliente, la cual se encarga de realizar las conexiones y recibir las respuestas de las validaciones del sistema, así como del contenido. El nodo cliente está compuesto por 3 funciones principales y una auxiliar encargada de generar una conexión hacia cada nodo del sistema. Cada función es encargada de enviar datos específicos para cada uno de los nodos del sistema, esto dependiendo de la respuesta recibida en cada paso, por el sistema de autenticación. A continuación, se describen cada una de las funciones del nodo cliente.

Conecta

Esta función recibe dos parámetros: la IP para establecer una conexión y el puerto de la aplicación que atenderá la conexión, regresa una instancia de tipo conexión, la cual puede ser usada por otra función para enviar algún tipo de mensaje una vez que la conexión se ha establecido.

Autentifica

Recibe como único parámetro una instancia de una conexión a una dirección IP y puerto específico. En primer lugar, pide al cliente su ID y contraseña para concatenarlos al mensaje que será enviado al servidor de autenticación, adicionalmente a dicho mensaje se le concatena la dirección IP del servidor de tickets y la hora en que se hace la solicitud en la aplicación cliente. Una vez que se han enviado los datos en una sola cadena cifrada y dependiendo de la respuesta recibida, esta función devuelve 0 si la

respuesta de la validación fue incorrecta o 1 si fue satisfactoria, en un caso correcto se recibe una clave para acceder al servidor de tickets y se procede a realizar la conexión con el servidor de tickets, en caso contrario se cierra la conexión y la aplicación cliente.

Solicita_ticket

Cuando se ha pasado la validación del servidor de autenticación, se genera una conexión al servidor de tickets. Esta función recibe como único parámetro una instancia de una conexión a una dirección IP y puerto específico, las cuales corresponden al servidor de tickets. Cuando esta función es llamada se envía un mensaje cifrado el cual contiene la clave de acceso al servidor de tickets y la dirección del servidor de contenido para ser validadas. Si la respuesta es satisfactoria la función devuelve 1 y recibe un mensaje (ticket), compuesto por la hora en que se atendió la petición, la clave de acceso y la IP del servidor de contenido, y continúa con el siguiente paso para descargar el contenido, en caso contrario devuelve 0 y cierra la conexión y la aplicación.

Descarga_contenido

Esta función es ejecutada únicamente cuando las validaciones anteriores han sido satisfactorias y al igual que las anteriores recibe una instancia de una conexión con la dirección del servidor de contenido y su puerto. Una vez establecida la conexión, se envía la clave del servidor de contenido para poder tener acceso, de ser satisfactoria la validación de la clave se genera un flujo para la escritura de un archivo y se recibe bit por bit al igual que el nombre del archivo. Si por el contrario la validación es negativa, la conexión se cierra y la aplicación también.

5. Conclusiones

El rápido desarrollo y la creciente complejidad de las aplicaciones de cómputo que actualmente son desplegadas sobre redes de comunicación han generado una más exigente demanda de cuestiones de seguridad y privacidad de parte de los usuarios. Esto ha generado un gran reto tecnológico y la necesidad de construir sistemas más seguros. En este trabajo se presenta un sistema de autenticación para datos distribuidos basados en el modelo Kerberos. Se ha desarrollado un prototipo básico de nuestro modelo y actualmente se está trabajando en integrar una base de datos relacional al nodo de autenticación para garantizar la persistencia de los datos y brindar una mayor robustez a la gestión de los mismos.

5.1. Trabajo a futuro

Como trabajo a futuro se espera programar una función de cifrado y descifrado de archivos, la cual se integre dentro de los nodos de contenido y cliente, con el objetivo de garantizar la privacidad del contenido mientras viaja en la red. Adicionalmente, y como continuidad del proyecto, contemplamos integrar servicios de sincronización de relojes y de gestión de claves de cifrado para mensajes y archivos, los cuales aporten controles confiables para robustecer el funcionamiento del sistema de manera efectiva sin comprometer su seguridad. Por otro lado, consideramos que el presente trabajo

puede direccionarse hacia diferentes objetivos. Uno de éstos es el poder integrarlo a futuro en una red de internet de las cosas con el fin de poder garantizar la seguridad y privacidad de los datos generados por este tipo de redes dentro de un ambiente abierto.

Referencias

1. Neumann, B. C., Ts'o, T.: Kerberos: An Authentication Service for Computer Networks. In: IEEE Communications Magazine, 32(9), pp. 33–38 (1994)
2. Al-Ayed, F., Liu, H.: Synopsis of Security: Using Kerberos Method to Secure File Transfer Sessions. In: IEEE International Conference on Computational Science and Computational Intelligence, USA (2016)
3. Hojabri, M., Venkat, R. K.: Innovation in cloud computing: Implementation of Kerberos version 5 in cloud computing in order to enhance the security issues. In: IEEE International Conference on Information Communication and Embedded Systems (ICICES), India (2013)
4. López-Fuentes, F. A.: Sistemas Distribuidos. UAM Unidad Cuajimalpa, pp. 1–203 (2015)
5. Liu, Y., Li, Z., Sun, Y.: Distributed Authentication in the Cloud Computing Environment. In: Springer International Publishing Switzerland, LNCS, 9532 (2015)
6. MIT Kerberos Consortium: The Role of Kerberos in Modern Information Systems. pp. 1–53, (2008)
7. MIT Kerberos Consortium: Why is Kerberos a credible security solution?. pp. 1–13 (2008)
8. Bishop, M.: Introduction to Computer Security. Pearson Education, Inc., pp. 2–3 (2005)
9. Zhang, N., Wu, X., Yang, C., Shen, Y., Cheng, Y.: A lightweight authentication and authorization solution based on Kerberos. In: IEEE (2016)
10. Symantec. Informe sobre las amenazas para la seguridad en Internet de 2017. Sitio web: https://www.symantec.com/es/mx/security-center/threat-report (2017)
11. VanDyke: Software Inc. Secure File Transfer with SSH (2008)
12. Stallings, W.: Fundamentos de Seguridad en Redes Aplicaciones y Estándares. Pearson Educación, pp. 28, 31–32, 35, 106, 109 (2004)