Preparación de Artículos revista Vínculos: Ciencia, Tecnología y Sociedad
Fecha de envío: 27/08/2021
Fecha de recepción: 19/09/2021
Fecha de aceptación: 13/01/2022

Proceso de cifrado con algoritmo idea y ofuscamiento de código en servicios web

Encryption process with idea algorithm and code obfuscation in web services

Diego Fernando Camargo-Ruiz [1], Edwar David Tovar-Zambrano [2], Julián David Niño-González [3]

Resumen: La rápida transformación de la tecnología actual y la interconexión del mundo, ha hecho que las personas estén conectadas a internet a través de múltiples dispositivos móviles, los cuales funcionan principalmente a través de servicios web que facilitan múltiples tareas del día a día, sean personales o empresariales, las soluciones tecnológicas basadas en servicios web son una realidad muy rentable y bastante aplicada por las empresas, teniendo presente la evolución que han tenido estos tipos de servicios surge la necesidad de aplicar estrategias para protección de la información que transportan y la propiedad intelectual, es ahí donde el ofuscamiento de código fuente ofrece una muy buena alternativa para la propiedad intelectual y el cifrado de la información en lo9s servicios web nos ayuda en la protección de la información que viaja a través de la web, y para poder conseguir este ofuscamiento existen diversas técnicas que se pueden aplicar al código fuente, e incluso a código compilado para mitigar el acceso no autorizado a los desarrollos implementados,

[1] Ingeniero en Telemática, Universidad Distrital Francisco José de Caldas, Colombia, Bogotá D.C., Colombia. dfcamargor@correo.udistrital.edu.co
[2] Ingeniero en Telemática, Universidad Distrital Francisco José de Caldas, Colombia, Bogotá D.C., Colombia. edtovarz@correo.udistrital.edu.co
[3] Ingeniero en Telemática, Universidad Distrital Francisco José de Caldas, Colombia, Bogotá D.C., Colombia. judninog@correo.udistrital.edu.co

e incluso dificultar la tarea de aplicar ingeniería inversa a programas que serán utilizados por personas no pertenecientes a la organización. para el proceso de cifrado existen muchas alternativas, entre ellas es la implementación de dos pasos adicionales en la interconexión de servicios web que nos cifre y descifre los datos que viajan en servicios web con ayuda del algoritmo simétrico IDEA, de esta manera se logra que, en procesos de ingeniería inversa, sea más difícil el descifrar la información que viaja a través de estos servicios.

Palabras clave: Cifrado, Cloud, IDEA, Ofuscamiento, Rest, Servicios web, Software.

Abstract*:* The rapid transformation of current technology and the interconnection of the world, has made people connected to the Internet through multiple mobile devices, which work mainly through web services that facilitate multiple tasks of daily life, whether personal or business, technology solutions based on web services are a very profitable reality and quite applied by companies, bearing in mind the evolution that these types of services have had the need to implement strategies to protect the information they carry and intellectual property, is where source code obfuscation offers a very good alternative for intellectual property and encryption of information in web services helps us in protecting the information that travels through the web, and to achieve this obfuscation there are several techniques that can be applied to the source code, and even compiled code to mitigate unauthorized access to the developments implemented, and even hinder the task of reverse engineering programs to be used by people outside the organization. For the encryption process there are many alternatives, among them is the implementation of two additional steps in the interconnection of web services that encrypt and decrypt the data that travels in web services with the help of the symmetric algorithm IDEA, thus achieving that in reverse engineering processes, it is more difficult to decipher the information that travels through these services.

Keywords: Encryption, Cloud, IDEA, Obfuscation, Rest, Web services, Software.

1. Introducción

La seguridad y la protección de la propiedad intelectual y los datos que viajan a través de la red es una de las principales constantes a evaluar en los servicios web que permiten a las personas disfrutar de diversos beneficios que faciliten las tareas del día a día, para poder lograr esta protección y seguridad es necesario la implementación de técnicas y procedimientos que ayuden a la mitigación de estas posibles vulnerabilidades que puedan afectar el sistema [6], también teniendo presente la seguridad de los datos, se buscan algoritmos complejos y eficientes que ayuden a garantizar la confidencialidad de la información [13], pero para poder aplicarlos a los desarrollos que son implementados en las casas de software, se necesita conocer los estándares existentes en la industria y, así mismo, darles un valor agregado que apoyen en la misión de asegurar suplir las vulnerabilidades asociadas a la aplicación de la ingeniería inversa.

para organizaciones que manejan información sensible es en muchos casos obligatorio que implementen controles y revisiones periódicas que garanticen el cumplimiento de las políticas de seguridad establecidas por la propia organización o por la asociación con terceros, ello hace necesario el análisis e investigación de nuevas tecnologías o procedimientos que ayuden a complementar la seguridad presente en los procesos de desarrollo de software. Para lograrlo se deben analizar algoritmos de cifrado de datos que sean óptimos y permitan un fácil acople teniendo en cuenta el rápido avance tecnológico que está presente en la época actual [3].

Teniendo presente que el desarrollo de la tecnología va encaminada a servicios instantáneos en dispositivos móviles que faciliten el acceso a procesos seguros y eficientes, que son necesarios en la cotidianidad de las personas, enfocamos el desarrollo del presente artículo en los desarrollos

de software principalmente relacionados con servicios web, para poder proteger la propiedad intelectual mediante técnicas de ofuscamiento y ayudar con un proceso de cifrado basado en algoritmos [8] en el envío y recepción de las peticiones presentes en este tipo de servicios que se apoyen en la arquitectura y el hardware para su desarrollo [5]

2. Evolución de la seguridad en servicios web

La seguridad de la red es una de las estrategias en cuanto a protección más usada en los últimos tiempos, ya que su facilidad para adaptarse y/o integrarse en cuanto a combatir amenazas.

Cuando se habla de seguridad asociamos la falta de riesgos o inconvenientes que se lleguen a presentar en un sistema. Podemos entender la palabra seguridad informática, con un conjunto de técnicas encargadas y encaminadas a siempre brindar altos niveles de confiabilidad en la web.

Otro aspecto importante a la hora de hablar de seguridad es cuando se menciona los servicios de seguridad, ya que estos mejoran y ayudan a disminuir las vulnerabilidades que se lleguen a presentar, los servicios se dividen en seis, disponibilidad, autenticación, integridad, no repudio, control de acceso y disponibilidad.

Figura 1. Servicios de Seguridad.

Fuente: Planificación de las necesidades de seguridad [15].

- Disponibilidad

Se encarga de asegurar que el acceso que se tenga a la información alojada en un sistema solamente lo puedan visualizar los usuarios autorizados, evita se pueda leer, copiar, o modificar la información a menos que se tenga autorización.

- Autenticación

Es la forma en la que un sistema compruebe la identidad de quien pretende ingresar, verificando la autenticación para así mismo conceder el permiso de acceder a la información solicitada, son las medidas que se deben tomar como protección a la información que se pretende acceder.

- Integridad

Es la parte encargada de que se verifique y se garantice que la información transmitida por cualquier canal o medio sufra algún tipo de modificaciones sin que estas cuenten con la autorización correspondiente.

- No repudio

Este servicio previene que tanto el receptor como el emisor reciba algún tipo de información sin la veracidad pertinente ya que permite que este pueda negar dicho mensaje trasmitido, el no repudio permite u ofrece protección frente al usuario ya que permite aceptar o no los mensajes transmitidos.

- Control de acceso

Controla el acceso de los sistemas y aplicaciones mediante cualquier medio de comunicación, el cual se cerciora de la correcta identificación al generar intento de ingreso o acceso a la información, también da la facultad al propietario de qué tipo de acceso brinda.

2.1 Seguridad en la nube

La computación en la nube es uno de los términos más importantes en la actualidad, ya que es una tecnología que facilita y mejora los costos que se tienen en una organización, debido a que manejan técnicas, donde se realizan múltiples procesos para reducir el tiempo de actividad y así mismo ofrecer beneficios de seguridad a los distintos consumidores de tecnología [4, 11]. Cloud brinda una distribución más completa y eficaz en cuanto al procesamiento de datos a los diferentes activos de una organización [9].

2.2 Modelos de servicios

Modelos de servicios	Características
Saas (Software como servicio)	No necesita instalación alguna ya que se accede al sistema a través de la web. Movilidad, accesibilidad y bajos costos. Datos alojados en servidores del proveedor.
PaaS (Plataforma como Servicio)	Permite entregar sus servicios desde aplicaciones livianas. Evita gastos en administración y compras de licencias de Software. Agregar más funcionalidad de desarrollo sin incorporar más personal. Usar herramientas sofisticadas a un precio asequible.
IaaS (Infraestructura como servicio)	Se contrata dispositivos virtuales de infraestructura. El pago se realiza de acuerdo al uso empleado. Informática de alto nivel.

Figura 2. Modelo de servicios en la nube.

Fuente: Elaboración propia.

Modelo simple de proceso de seguridad en la nube

Existe un proceso sencillo y con un gran nivel para para realizar la administración de una nube.

- Identificar los requisitos de seguridad y cumplimiento necesarios y cualquier control existente.

- Seleccione su proveedor de servicios en la nube, servicio y modelos de implementación.
- Definir la arquitectura.
- Evaluar los controles de seguridad.
- Identificar las lagunas de control.
- Diseñar e implementar controles para llenar los vacíos.
- Gestionar cambios a lo largo del tiempo

Figura 3. Modelo de proceso de la seguridad en la nube.

Fuente: GUÍA DE SEGURIDAD DE Áreas Criticas Para la Computación en la Nube V4.0 [14].

3. Ofuscamiento de código

Uno de los principales inconvenientes presentes en los programas de software, principalmente los ejecutados en cliente, es que el código completo encargado de alojar la lógica de ejecución debe ser descargado para ser compilado y ejecutado, lo que inevitablemente ocasiona que todo el trabajo y desarrollo del personal técnico encargado de implementar el programa, quede al descubierto por cualquier persona que ejecute la aplicación en su dispositivo cliente, teniendo esto presente, se exponen los siguientes puntos importantes a considerar cuando se quiere implementar ofuscamiento en el código:

1. *Propiedad intelectual:* se da principalmente cuando la organización o el equipo técnico encargado de realizar la implementación de código de software, no quiere que lo copien, modifiquen o utilicen en otros sistemas sin previo consentimiento.

2. *Seguridad de la información:* si el código desarrollado está ofuscado y adicional se utilizan herramientas que ayuden a la seguridad de la información, hace difícil el analizar el código en búsqueda de vulnerabilidades del sistema, dificultando la tarea de aplicar ingeniería inversa al software desarrollado, mitigando posibles ataques y vulnerabilidades que pudieran estar más evidentes en un código sin ofuscar.

3. *Rendimiento:* dependiendo de la técnica de ofuscamiento aplicada, se pueden llegar a notar ventajas del ofuscamiento como el rendimiento del software, teniendo en cuenta que algunas de estas técnicas, hacen que el código pese menos y sea más fácil que el dispositivo de cómputo ejecute el aplicativo.

Si se observa los puntos anteriormente mencionados, se tienen distintos motivos los cuales impulsan a la industria del desarrollo de software a incluir en sus programas, la fase de ofuscamiento del código, para ello, existen técnicas que pueden ser aplicadas a esta práctica, entre ellas tenemos:

1. *Ofuscamiento de diseño:* para este ofuscamiento, lo que se busca es cambiar las clases implementadas en la arquitectura del programa, uniéndolas o dividiéndolas de tal manera que sea más complejo realizar un entendimiento de la estructura del desarrollo.

2. *Ofuscamiento de datos:* Esta técnica de ofuscamiento es basado principalmente en las variables, constantes y estructuras que forman parte de la codificación, de tal forma que se cambien, agreguen o modifiquen atributos del desarrollo de manera que, yendo al detalle,

sea más confuso entender estas variables y constantes del código. hay diferentes prácticas para implementar esta técnica como lo es la separación de variables, o el cegamiento de las constantes.

3. *Ofuscamiento de flujos de control:* Con este ofuscamiento se busca modificar el entendimiento del flujo de proceso que sigue el desarrollo, de tal manera que no sea fácilmente perceptible aplicando ingeniería inversa. este ofuscamiento se logra por varios métodos como la adición de código muerto o la transformación de tablas de interpretación.
4. *Ofuscamiento de instrucciones:* Esta consiste en sustituir la lógica implementada en las funciones que realizan el flujo de los procesos, sustituyéndolas por instrucciones más complejas que finalmente realicen el mismo proceso final.
5. *Ofuscamiento con virtualización de código:* En los últimos años, la virtualización de código está emergiendo como un método prometedor para implementar la ofuscación de código [1], este método está enfocado principalmente a reemplazar las instrucciones del programa en un nivel medio con instrucciones con las cuales no se estén muy familiarizados para el tipo de lenguaje escogido.
6. *Ofuscamiento de código intermedio:* esta técnica va orientada a aquellos lenguajes que son compilados para su ejecución, una vez compilados pueden ser fácilmente descompilados, si aplicamos este ofuscamiento, lo aplicamos al código compilado de tal manera que no cambie su funcionalidad, pero sí dificulta un adecuado descompilamiento del programa.

Para el ofuscamiento de código fuente, existen múltiples herramientas que nos ayudan a implementarla en los desarrollos el ofuscamiento del código desarrollado. una de ellas es JSPro, una herramienta para ofuscación de código JavaScript basada en el estándar emergente Web

Assembly la cual está basada en el ofuscamiento de código por virtualización, como se observa en la Figura 4, el ofuscamiento de código permite aumentar el rendimiento de acuerdo con la técnica y método implementado [1].

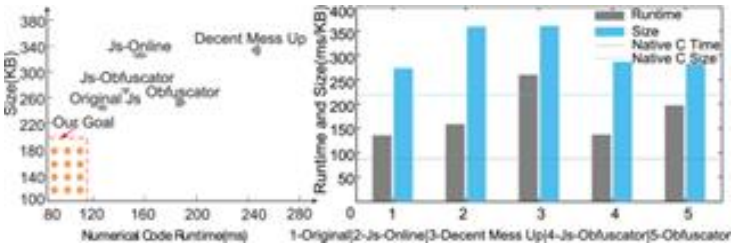

Figura 4. Tiempo de ejecución de código JavaScript ofuscado.

Fuente: Leveraging WebAssembly for Numerical JavaScript Code Virtualization [1].

4. Cifrado de datos en servicios web con el algoritmo idea

Para la implementación de cifrado de datos en los servicios web Front End y Back End vamos a analizar el algoritmo de cifrado IDEA, para este punto nos centramos en los servicios web tipo REST cuyo tipo de contenido es en formato JSON. Por lo general, este tipo de servicio está basado en programación orientada a objetos, en los cuales el cuerpo recibido en el servicio web, corresponde a la representación de un objeto en el lenguaje de programación utilizado, de esta forma, se hace más simple el trabajar durante el desarrollo de software con estos objetos, apoyándose en distintos frameworks para el mapeo de campos correspondiente al envío y recepción de las peticiones a los servicios web [12].

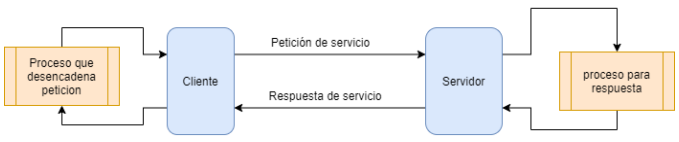

Figura 5. Esquema funcionamiento de servicio REST.

Fuente: Elaboración propia.

Como se aprecia en la Figura 5 el funcionamiento de los servicios REST web es basado en peticiones a un servidor expuesto en la red, para ello, existe un proceso en el cliente que desencadena la petición hacia el servidor, donde esta petición es recibida, y desencadena en el servidor un proceso de respuesta la cual es enviada al cliente, y recibida por el proceso en cliente que inició la petición REST. Es en los procesos que desencadenan la petición en el cliente y el proceso de la respuesta que estará presente el cifrado con el algoritmo idea.

El propósito general de implementar este algoritmo en los servicios web es por motivos de seguridad, ya que actualmente en la industria de los sistemas de información se manejan datos sensibles que viajan a través de internet, y que generalmente pueden ser observados por las personas encargadas de mantener el sistema, e incluso, por aquellos clientes que desean aplicar ingeniería inversa a los servicios web en busca de vulnerabilidades que se puedan explotar. Por estos motivos, se decidió desarrollar un modelo de cifrado de datos para los servicios web basados en un sistema con una modelo vista controlador, en el cual viaja información sensible de datos personales, contraseñas e información financiera.

Para este sistema de seguridad en los servicios web, se propone dos métodos genéricos que puede ser utilizado por cualquier servicio web, de tal forma que tanto el envío y recepción de datos, se reciba el mensaje cifrado con ayuda del algoritmo IDEA, para ello, los dos métodos cuentan con 4 pasos básicos cada uno como se muestra en la Figura 6

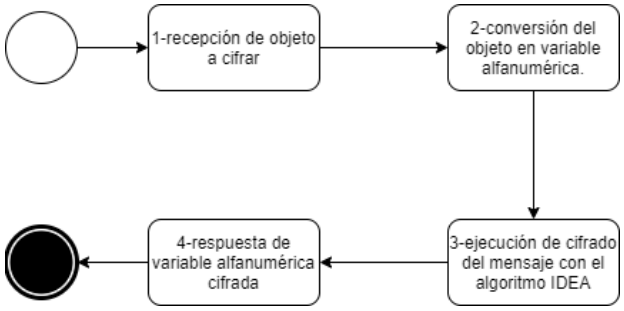

Figura 6. pasos para cifrado de datos IDEA.

Fuente: Elaboración propia.

Los pasos básicamente consisten en:

1. *Recepción del objeto a cifrar:* en este paso, se tiene en cuenta un objeto genérico el cual contendrá la información que se busca enviar, y que se busca cifrar para su posterior envío en el servicio web.
2. *Conversión del objeto en variable alfanumérica:* para el manejo más asertivo y simple de implementar, pasamos todo el objeto a una variable en el formato alfanumérico que se enviará en el cuerpo del servicio, de esta forma no viajará el objeto como tal, en su lugar solo se tendrá una única respuesta de variable alfanumérica.
3. *Ejecución de cifrado del mensaje con el algoritmo IDEA:* para la implementación del cifrado con este algoritmo, se necesita una llave que está previamente definida de 128 bits [2] con la cual se realizará el proceso de cifrado del mensaje, el resultado de este proceso se almacena en una variable que responderá este método.
4. *Respuesta de variable alfanumérica cifrada:* del resultado del proceso de cifrado con el algoritmo IDEA, se devuelve la variable que tiene almacenado los datos del objeto cifrados con la llave de 128 bits y el proceso del algoritmo.

Se debe tener en cuenta que para el proceso de cifrado y descifrado de datos, se debe utilizar la misma llave ya que el algoritmo IDEA es un algoritmo simétrico [10]. Partiendo de esta idea, el proceso de descifrado del algoritmo ilustrado en la Figura 7 se realiza utilizando la misma llave

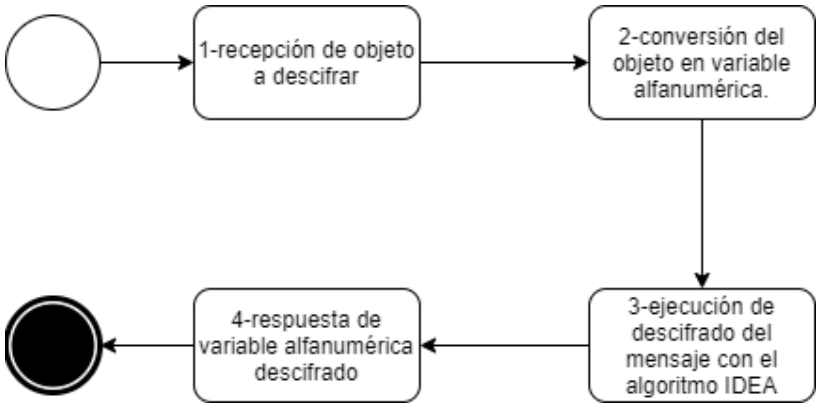

Figura 7. Pasos para descifrado de datos IDEA.

Fuente: Elaboración propia.

Teniendo presente cómo van a funcionar los métodos de cifrado y descifrado para los servicios web, ubicamos estos métodos en los procesos correspondientes al cliente y al servidor en el esquema del funcionamiento de los servicios REST.

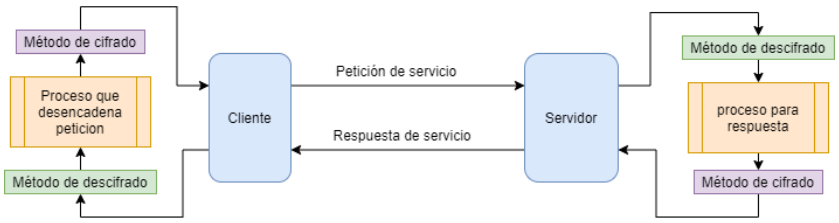

Figura 8. Esquema funcionamiento de servicio REST con la implementación del algoritmo IDEA.

Fuente: Elaboración propia.

Como se aprecia en la Figura 8 los métodos de cifrado y descifrado deben estar tanto en los procesos del cliente, como en los del servidor, y se implementan justo antes de su envío y recepción,

de esta forma se logra trabajar en los pasos de petición del servicio y respuesta del servicio, con la información cifrada, esto ayuda principalmente a los logs de los servicios, para que de esta forma se pueda ver reflejado el flujo del proceso, mas no el contenido de su información.

5. Conclusiones

Aunque existen muchos tipos de algoritmos estandarizados en procesos para el cifrado de datos en las distintas fases de un flujo normal de software, en el presente artículo se muestra una fase adicional que puede ser implementada para la ayuda en la confidencialidad de la información en aplicativos que son principalmente de arquitectura separada en la vista y el controlador de la lógica de negocio, si se combina con el ofuscamiento de código puede ayudar en gran medida a mitigar la explotación de vulnerabilidades en aplicativos web, ya que se busca combinar protocolos y técnicas establecidas con procesos eficientes e independientes de estándares que ayuden a la protección de la propiedad intelectual a nivel del desarrollo y de los datos procesados [2, 7].

Referencias

[1] S. Wang et al., "Leveraging WebAssembly for Numerical JavaScript Code Virtualization", in IEEE Access, vol. 7, pp. 182711-182724, 2019. https://doi.org/10.1109/ACCESS.2019.2953511

[2] R. Modugu, Y. Kim and M. Choi, “Design and performance measurement of efficient IDEA (International Data Encryption Algorithm) crypto-hardware using novel modular arithmetic components”, 2010 IEEE Instrumentation & Measurement Technology Conference Proceedings, pp. 1222-1227, 2010. https://doi.org/10.1109/IMTC.2010.5488049

[3] S. J. Pirzada, A. Murtaza, T. Xu, and L. Jianwei, “Architectural Optimization of Parallel Authenticated Encryption Algorithm for Satellite Application”, in IEEE Access, vol. 8, pp. 48543-48556, 2020. https://doi.org/10.1109/ACCESS.2020.2978665

[4] Kwangsu Lee, Dong Hoon Lee, Jong Hwan Park, Moti Yung, “CCA Security for Self-Updatable Encryption: Protecting Cloud Data When Clients Read/Write Ciphertexts”, The Computer Journal, vol. 62, no. 4, pp. 545-562, 2019. https://doi.org/10.1093/comjnl/bxy122

[5] R. F. Mirzaee, M. Eshghi, “Design of an ASIP IDEA crypto processor”, 2011 IEEE 2nd International Conference on Networked Embedded Systems for Enterprise Applications, pp. 1-7, 2011. https://doi.org/10.1109/NESEA.2011.6144954

[6] A. Bozesan, F. Opritoiu, M. Vladutiu, “Offline error-detection strategies for the IDEA NXT crypto-algorithm”, 2014 18th International Conference on System Theory, Control and Computing (ICSTCC), pp. 37-42, 2014. https://doi.org/10.1109/ICSTCC.2014.6982387

[7] K. Shahbazi, M. Eshghi, R. Faghih Mirzaee, “Design and implementation of an ASIP-based cryptography processor”, Engineering Science and Technology, an International Journal, vol. 20, no. 4, pp. 1308-1317, 2017. https://doi.org/10.1016/j.jestch.2017.07.002

[8] S. Afzal, M. Yousaf, H. Afzal, N. Alharbe, M. Mufti, “Cryptographic Strength Evaluation of Key Schedule Algorithms”, Security and Communication Networks, 2020. https://doi.org/10.1155/2020/3189601

[9] R. Petrasch, “Model-based Engineering for Microservice”, 14th International Joint Conference on Computer Science and Software Engineering (JCSSE). Bangkok, 2017.

[10] P. Saraswat, K. Garg, R. Tripathi, A. Agarwal, “Encryption Algorithm Based on Neural Network”, 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), pp. 1-5, 2019. https://doi.org/10.1109/IoT-SIU.2019.8777637

[11] K. Chennam, L. Muddana, R. Aluvalu, “Performance analysis of various encryption algorithms for usage in multistage encryption for securing data in cloud”, 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), pp. 2030-2033, 2017. https://doi.org/10.1109/RTEICT.2017.8256955

[12] X. Fu, B. Liu, Y. Xie, W. Li, and Y. Liu, “Image Encryption-Then-Transmission Using DNA Encryption Algorithm and The Double Chaos”, in IEEE Photonics Journal, vol. 10, no. 3, pp. 1-15, 2018. https://doi.org/10.1109/JPHOT.2018.2827165

[13] Y. Zhou, B. Yang, T. Wang, Z. Xia, H. Hou, “Continuous Leakage-Resilient Certificate-Based Encryption Scheme Without Bilinear Pairings”, The Computer Journal, vol. 63, no. 4, pp. 508-524, 2020. https://doi.org/10.1093/comjnl/bxz085

[14] R. Mogull, J. Arlen, F. Gilbert, A. Lane, D. Mortman, “Guía de seguridad de Áreas Criticas Para la Computación en la Nube”, V4.0., 2017.

[15] M. Ángel Barrera Pérez, N. Y. Serrato Losada, E. Rojas Sánchez, G. Mancilla Gaona, “State of the art in software defined networking (SDN)”, Visión electrónica, vol. 13, no. 1, pp. 178–194, 2019. https://doi.org/10.14483/22484728.14424