Implementación de un sistema de encriptación a una base de datos mediante el método de curva elíptica.

Toro Demera, Erika Andrea

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica y Telecomunicaciones

Trabajo de titulación, previo a la obtención del título de “Ingeniera en Electrónica y Telecomunicaciones”

MSc. Bernal Oñate, Carlos Paúl

29 de junio del 2020

URKUND

Urkund Analysis Result

Analysed Document:	TORO_DEMERA_TESIS JUNIO.pdf (D76123482)
Submitted:	7/8/2020 12:52:00 AM
Submitted By:	mgutierrez@difusion.com.mx
Significance:	3 %

Sources included in the report:

258.pdf (D26291098)
https://medium.com/articulos-de-la-comunidad/criptograf%C3%ADa-de-curva-el%C3%ADptica-99b8c8c1657c
https://www.oroyfinanzas.com/2014/01/criptografia-curva-eliptica-bitcoin-por-que-utiliza-ecdsa/
https://www.thinknetworks.pe/plan-gestion-seguridad-informacion/
https://www.oracle.com/ve/database/security/
https://guiabit.win/que-es-el-algoritmo-de-firma-ecdsa/
https://www.oscarblancarteblog.com/2018/11/30/data-transfer-object-dto-patron-diseno/
https://aprendiendoarduino.wordpress.com/category/api-rest/
https://www.ionos.es/digitalguide/servidores/herramientas/instala-tu-servidor-local-xampp-en-unos-pocos-pasos/
https://www.isotools.org/pdfs-pro/iso-27001-sistema-gestion-seguridad-informacion.pdf
https://www.json.org/json-es.html
https://www.lr.org/es-es/iso-27001/
https://tools.ietf.org/html/rfc5639#section-2.1

Instances where selected sources appear:

16

20.07.2020

ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

CERTIFICACIÓN

Certifico que el trabajo de titulación, **"Implementación de un sistema de encriptación a una base de datos mediante el método de curva elíptica"** fue realizado por la señorita **Toro Demera, Erika Andrea** el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 20 de julio del 2020

Firma:

Bernal Oñate, Carlos Paúl

C. C 1709775637

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

RESPONSABILIDAD DE AUTORÍA

Yo, **Toro Demera, Erika Andrea**, con cédula de ciudadanía n° 0850033242, declaro que el contenido, ideas y criterios del trabajo de titulación**: "Implementación de un sistema de encriptación a una base de datos mediante el método de curva elíptica."**, es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 20 de julio del 2020

Firma

Erika Toro

Toro Demera, Erika Andrea

C.C.: 0850033242

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

AUTORIZACIÓN DE PUBLICACIÓN

Yo, **Toro Demera, Erika Andrea**, con cédula de ciudadanía n° 0850033242 ,autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **"Implementación de un sistema de encriptación a una base de datos mediante el método de curva elíptica.",** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 20 de julio del 2020

Firma

Erika Toro

Toro Demera, Erika Andrea

C.C.: 0850033242

Dedicatoria

Deseo dedicar mi tesis a:

Mis padres Manuel Toro, Consuelo Demera y a mis hermanos, Nicole y Diego, quienes con su amor, sacrificio y paciencia me apoyaron a lo largo del desarrollo de la carrera.

A mi tía María Demera y a mi querido primo Alan Garcés, por su apoyo incondicional y su cariño sincero.

En especial a mi fallecida abuelita Teresa Irene Gracia Chica por cuidarme con sus oraciones desde lejos pidiendo a Dios y a la Virgen me acompañen siempre.

Agradecimiento

A mi familia por confiar en mi en todo momento, no soltar mi mano aún en los momentos más difíciles y recibirme siempre con un amoroso abrazo.

A David Sánchez por ser mi familia lejos de casa y Alex Verdugo por ser mi amiga incondicional.

A mi tutor de tesis, por haberme guiado en la elaboración del presente trabajo de titulación brindándome un apoyo cercano para culminar mis estudios superiores.

Un agradecimiento infinito a la Universidad de las Fuerzas Armadas ESPE, por haberme dado la oportunidad de educarme y aprender a desarrollar habilidades profesionales.

Índice de Contenidos

Índice de Tablas

Índice de Figuras

Resumen

El proceso de encriptación es muy importante para diversas aplicaciones que abarque el tratamiento de información sensible ya que permite proteger de agentes externos la información almacenada en bases de datos, brindando seguridad y confianza al usuario. El modelo de encriptación de curva elíptica presenta como ventaja una generación de claves más cortas pero al mismo nivel de seguridad que otros métodos con longitudes de claves mucho más extensas. El proyecto implementa un sistema de encriptación utilizando el algoritmo de curva elíptica para protección de la información de una base datos ante ataques desde un dispositivo móvil. Se diseñó la interfaz de ingreso de información a la base de datos desde donde se realiza proceso de encriptación y desencriptación, de tal manera que en el gestor de base de datos se almacene la información de usuario encriptada. La aplicación de la norma ISO/IEC 27001 permite establecer una correcta gestión de la seguridad de la información, mediante controles y estrategias basados en el ciclo de mejora continua. Finalmente, el sistema fue sometido a pruebas de validación de seguridad de la información cuando se accede a ella desde un dispositivo móvil y se transmite la información por un canal inseguro, realizando el número necesario de pruebas para garantizar que la información viaja a través de la red está encriptada, siendo ilegible para el intruso incluso si logra tener acceso a ella.

PALABRAS CLAVES:

- **ENCRIPTACIÓN DE CURVA ELÍPTICA**
- **GESTOR DE BASE DE DATOS**
- **ISO/IEC 27001**

Abstract

The encryption process is very important for various applications that cover the treatment of sensitive information since it allows the protection of information stored in databases from external agents, providing security and trust to the user. The elliptic curve encryption model has the advantage of generating shorter keys but at the same level of security as other methods with much longer key lengths. The project implements an encryption system using the elliptic curve algorithm to protect the information in a database against attacks from a mobile device. The interface for entering information into the database from which the encryption and decryption process is carried out was designed, so that the encrypted user information is stored in the database manager. The application of the ISO / IEC 27001 standard allows the correct management of information security to be established, through controls and strategies based on the continuous improvement cycle. Finally, the system was subjected to information security validation tests when it is accessed from a mobile device and the information is transmitted through an insecure channel, performing the necessary number of tests to ensure that the information travels through the network is encrypted, being unreadable to the intruder even if you can access it.

KEYWORDS:

- **ELLIPTIC CURVE CRYPTOGRAPHY**
- **DATABASE MANAGERX**
- **ISO/IEC 27001**

Introducción del Proyecto de Investigación

Antecedentes y Justificación del Proyecto

La criptografía o encriptación de información genera privacidad y confidencialidad, mediante la conversión de información a datos ilegibles accediendo a ella sólo con la clave que la resguarda. Una de las primeras grandes apariciones de la criptografía como protagonista en la historia fue durante la segunda guerra mundial, cuando el matemático Turing reveló los mensajes de la famosa, hasta entonces indescifrable máquina Enigma. Actualmente, el proceso de encriptación es muy importante para un entorno de comercio electrónico, por ejemplo, donde la confidencialidad es la clave de transacciones comerciales exitosas y confiables al mismo tiempo que se mantiene una reputación sobre sus competidores (Dante Ramírez, 2018).

Para operaciones comerciales que requieren usar métodos de encriptación con claves más cortas a comparación de los métodos antiguos y al mismo tiempo proporcionar un nivel de seguridad superior, se recomienda el uso de la criptografía de curva elíptica. Debido a que su principal ventaja es la posibilidad de reducir requisitos de almacenamiento y transmisión, generando claves más cortas. Una clave basada en la criptografía de curva elíptica puede dar el mismo nivel de seguridad con un clave de 256 bits como un algoritmo RSA con una clave de 2048 bits (Preukschat, 2014).

La generación de dichas claves conlleva un conjunto de procesos y conceptos matemáticos extensos. Organizaciones sin fines de lucro como Legión de Bouncy Castle, generan librerías y subrutinas que permiten dependiendo la aplicación,

conseguir implementar algoritmos de encriptación con distintos métodos y en distintos lenguajes de programación (Legion of Bouncy Castel Inc., 2013).

Toda la información que se genera digitalmente en el mundo es almacenada en su mayoría dentro de sistemas gestores de bases de datos, en el documento “Base de datos y sus vulnerabilidades más comunes” (Telefónica Company, 2015), se menciona a la falta de cifrado en información sensible como una de las principales vulnerabilidades. Por ello Oracle desarrolla soluciones como *Oracle Database Security*, que proporcionan servicios de: cifrado de datos transparentes y gestión de claves de cifrado (Oracle, 2017).

En ese contexto, la gestión de la seguridad de la información en una empresa proporciona una ventaja competitiva al demostrar a los clientes que la seguridad de su información es primordial (Segovia, 2017). Esto se hace tangible mediante la aplicación de la norma ISO/IEC 27001, con un proceso sistemático y estructurado para garantizar que los riesgos de la seguridad de la información sean conocidos, asumidos, gestionados y minimizados (ISOtools Excellence, s.f.).

Ante lo mencionado sobre la importancia de la encriptación de curvas elípticas en el comercio por medios electrónicos, gracias a que garantiza un alto grado de confidencialidad en el transporte de la información, se han realizado implementaciones de sistemas de encriptación utilizando este algoritmo. Por ejemplo, el trabajo propuesto por Camilo A. Barbosa desarrolla la implementación del criptosistema de curva elíptica en una aplicación móvil y en una aplicación Web, en el que se realiza el cifrado y descifrado del flujo de datos entre las dos aplicaciones que opera bajo un entorno *e-commerce* (Camilo A. Barbosa, 2011). También, Sergio Navas presenta un modelo de

seguridad para Hadoop, que permite a la fuente de los datos controlar quien tiene acceso evitando así un uso no autorizado de manipulación de información (Sergio Navas, 2016)

Aunque los trabajos anteriores realizan la implementación de un sistema de encriptación de curvas elípticas para información sensible, no presentan una comprobación de la seguridad de la información cuando se accede a una base de datos desde un dispositivo móvil.

Objetivos de la Investigación

General

Implementar un sistema de encriptación de curva elíptica a una base de datos para garantizar la seguridad en ataques desde dispositivos móviles.

Específicos

- Definir las funciones o subrutinas necesarias para realizar la encriptación de curva elíptica a la información.
- Diseñar un algoritmo experimental en base al modelo matemático para la generación de claves de encriptación y desencriptación de curva elíptica.
- Realizar una interfaz que permita el ingreso de información encriptada a la base de datos.
- Implementar una base de datos que contenga la información encriptada.
- Evaluar el sistema criptográfico de curva elíptica desde un dispositivo móvil con el fin de determinar el nivel de seguridad ante ataques.

- Realizar el un número adecuado de pruebas para evaluar los resultados con un análisis que permita validar la seguridad.
- Demostrar la garantía en la seguridad de la información mediante la aplicación de la norma ISO/IEC 27001.

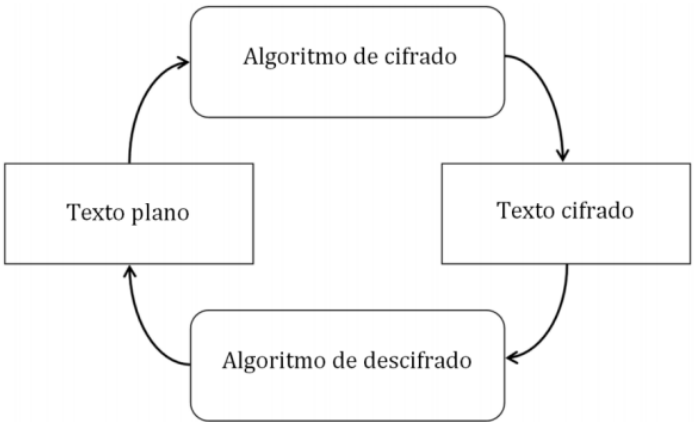
Marco Teórico

Encriptación

La encriptación es el arte y la ciencia de lograr seguridad mediante la codificación del mensaje con el fin de convertirlo en información no legible (Harinath, 2015). Se ha convertido en una disciplina importante, objeto de investigación que proporciona una base para la seguridad de la información en muchas aplicaciones en especial comunicaciones de datos (Alfred J. Menezes, 2018).

Figura 1.

Proceso de cifrado y descifrado de un texto.



Nota. El gráfico describe el ciclo de cifrado y descifrado de un mensaje. Tomado de Análisis de las ventajas de la criptografía de curva elíptica, frente a los sistemas criptográficos asimétricos actuales por Universidad Nacional Autónoma de México, 2012.

Para el proceso de encriptación o cifrado que se observa en la Figura 1, se requiere un algoritmo que haga que los datos sean incomprensibles y una llave secreta sin la cual no es posible realizar el proceso inverso, la desencriptación. Los modelos de ataque y los objetivos de seguridad deben ir de la mano, se considera un algoritmo de

encriptación seguro si de acuerdo al modelo de ataque no se logra vulnerar el objetivo de seguridad (Aumasson, 2018).

El mensaje puede ser cifrado en bloque, donde se toma un número determinado de bits del mensaje a encriptar y se cifra dicho bloque, el tamaño de los bloques depende del algoritmo particular que se utilice y de los diversos modos en los que el algoritmo podría funcionar. El cifrado del mensaje también puede ser en flujo, el cual encripta cada bit en el mensaje 1 bit a la vez. También es posible que un cifrado de bloque actúe como un cifrado de flujo configurando un tamaño de bloque de 1 bit.

Aunque los cifrados de bloque son a menudo más lentos que los cifrados de flujo, tienden a ser más eficientes, aunque utilicen más recursos y sean más complejos de implementar en hardware o software. Sin embargo, un error en el proceso de cifrado de bloque puede dejar inutilizable un segmento de datos más grande que el que encontraríamos en un cifrado de flujo, ya que el cifrado de flujo solo funcionaría con 1 bit en particular.

Encriptación con Clave Simétrica

En la encriptación simétrica, la llave usada para encriptar información es la misma que se usa para realizar la desencriptación de información cifrada, como se observa en Figura 2. El uso de una sola llave es una de las principales debilidades de este tipo de criptografía debido a que, si la clave queda expuesta a un atacante este podrá descifrar o modificar el mensaje proporcionando confidencialidad y no integridad.

Elvira Mifsuf Talón en el libro Apache, indica que el tamaño de la clave es proporcional a la robustez del algoritmo de encriptación simétrica y debe ser mayor de

40 bits para ser inalterable frente a ataques. Menciona que por su rapidez, este tipo de algoritmo resulta eficiente para el cifrado de extensos volúmenes de información.

Figura 2.

Cifrado de claves simétricas.

Nota. El gráfico representa el uso de la clave simétrica, la cual se usa en el cifrado y descifrado de un mensaje. Tomado de Conceptos de cifrado por IBM Knowledge Center, 2018.

Algunos de los algoritmos criptográficos de clave simétrica, como DES y AES, son o han sido utilizados regularmente por el gobierno de EE. UU. Y otros como algoritmos estándar para proteger datos altamente confidenciales (Andress, 2014).

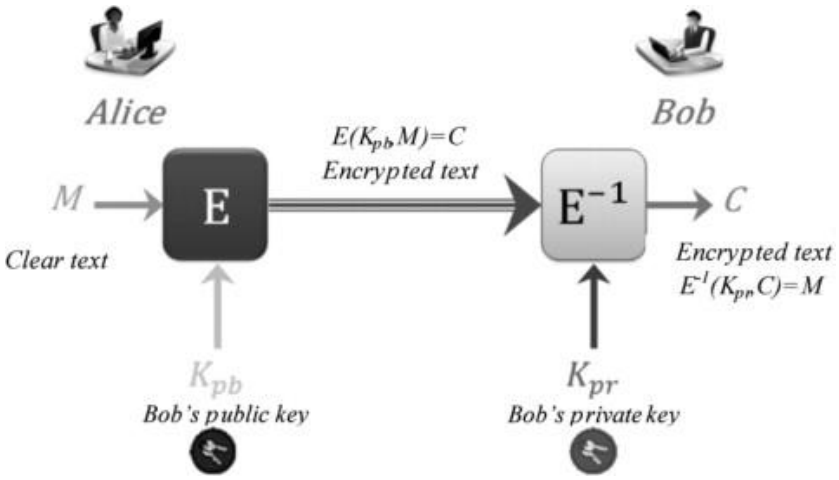
Encriptación con Clave Asimétrica

En la encriptación asimétrica, cada usuario tiene un par de claves: la primera clave es pública, accesible para todos los usuarios que participan de la comunicación, y la otra es privada, solo debe ser conocida por el titular legítimo. Estas dos claves son específicas de cada algoritmo y están relacionadas de una manera muy específica; esencialmente, si una clave se usa para cifrar, la otra se usará para descifrar (Kiennert, 2015).

Si Alice desea enviar un mensaje encriptado a Bob, debe usar la clave pública de Bob para encriptar el mensaje y luego Bob descifra el mensaje utilizando su propia clave privada, como se muestra en la Figura 3.

Figura 3.

Comunicación con clave asimétrica



Nota. El gráfico representa texto plano cifrado con la clave pública del receptor y descifrado con la clave privada del receptor. Tomado de Hacking the Code por Burnett, 2004, Syngress.

La criptografía asimétrica aborda los problemas de escalabilidad e intercambio de claves de la criptografía simétrica mediante el uso de un modelo de clave pública y privada. La idea es que puede distribuir libremente su clave pública, que otros pueden usar para cifrar datos a los que solo usted puede acceder con su clave privada.

Requiere de un procesador intensivo y, por lo tanto, rara vez se usa para cifrar toda la comunicación. En cambio, la mayoría de las aplicaciones usan criptografía

asimétrica para intercambiar una clave de sesión y luego usan un cifrado simétrico para encriptar el tráfico durante esa sesión (Burnett, 2004).

Aunque la criptografía asimétrica proporciona seguridad al utilizar menos recursos, la distribución de claves es un problema importante. Los algoritmos de criptografía asimétrica como RSA, utilizan dos claves diferentes, una clave para el cifrado (remitente) y otra clave y otra diferente para descifrar (destinatario). De todos modos, eso requiere muchos recursos computacionales debido a su gran tamaño en las claves (Centro Criptológico Nacional, 2012).

Criptografía de Curva Elíptica

Es una criptografía asimétrica basada en la estructura algebraica de curvas elípticas creada por Victor Miller and Neal Koblitz en 1985, donde calcular la llave privada a partir de la llave pública significa resolver un problema computacional complejo como el logaritmo discreto.

En Tabla 1, se muestra una aproximación del tamaño en bits de las longitudes de las claves para diferentes algoritmos de encriptación y su respectivo nivel de seguridad. La encriptación de curva elíptica permite obtener un nivel de seguridad elevado reduciendo la longitud de claves a comparación del algoritmo de encriptación RSA.

Tabla 1

Comparación de nivel de seguridad y longitud de clave entre algoritmos de encriptación.

Nivel de Seguridad	Longitud de clave RSA	Longitud de clave ECC
80	1024	160
112	2048	224
128	3072	256
192	7680	384

256	15360	512

Nota. Recuperado de Criptología de empleo en el esquema nacional de seguridad guía/norma de seguridad de las tic (ccn-stic-807). Copyrigth 2012 del Centro Criptológico Nacional.

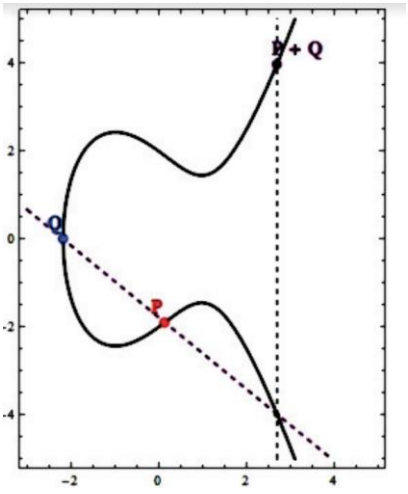
Las propiedades matemáticas de las curvas elípticas son la base para el desarrollo del algoritmo de encriptación, la ecuación de curva elíptica sobre campo finito es representado por la ecuación (1).

$$y^2 = \{x^3 + ax + b\} \, mod \, \{p\} \tag{1}$$

Dentro de las operaciones básicas de curva elíptica, la suma de los puntos de la curva $P(x_1, y_1)$ y $Q(x_2, y_2)$ donde P y Q son diferentes, se muestra en Figura 4.

Figura 4.

Suma de dos puntos en una curva elíptica



Nota. El gráfico muestra la ubicación del punto Q resultado de la suma de los puntos P y Q. Tomado de Implementation of Text Encryption using Elliptic Curve Cryptography por Singh, 2015, ScienceDirect.

$P + Q = R(x_3, y_3)$ está dado por el siguiente cálculo que se muestra en las ecuaciones

$$x_3 = \{\lambda^2 - x_1 - x_2\} \, mod \, \{p\} \quad (2)$$

$$y_3 = \{\lambda(x_1 - x_3) - y_1\} \, mod \, \{p\} \quad (3)$$

Donde

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \, mod \, \{p\} \quad (4)$$

La ventaja más atractiva para usar la encriptación de curva elíptica se da en los entornos restringidos donde se debe controlar la potencia de procesamiento, almacenamiento y ancho de banda (Reyad, 2018).

Problema del Logaritmo Discreto

La seguridad de la encriptación de curva elíptica se basa principalmente en nivel de dificultad del problema del logaritmo discreto. Sea P y Q dos puntos que se encuentran en una curva elíptica tal que (5), donde k es un escalar.

$$\mathrm{k} * \mathrm{P} = \mathrm{Q} \quad (5)$$

Dado P y Q, es computacionalmente inviable obtener el valor de k, si k es lo suficientemente grande se denomina logaritmo discreto de Q a la base P. Si bien el cálculo de la multiplicación de un escalar por un punto se considera primaria, las operaciones de suma, duplicación y multiplicación son requeridas para encontrar la pendiente (Singh, 2015).

Curva Elíptica

Todos los parámetros de dominio de curva elíptica recomendados sobre F_p usan números primos aleatorios ya que facilitan implementaciones eficientes. Este proceso asegura que los parámetros no puedan ser predeterminados. Por lo tanto, es

extremadamente improbable que los parámetros sean susceptibles a futuros ataques de propósito especial.

Los parámetros de las curvas elípticas normalmente se encuentran normalizados, los parámetros recomendados al azar fueron elegidos seleccionando repetidamente una semilla aleatoria y contando el número de puntos en la curva correspondiente hasta encontrar los parámetros apropiados.

Las curvas Brainpool se usan para la autenticación y el intercambio de claves en el protocolo de Seguridad de la capa de transporte (TLS),las curvas que se describen en tabla 2 fueron generadas de una manera pseudoaleatoria verificable y cumplen con los requisitos de seguridad de los estándares relevantes de ISO (Organización internacional para la estandarización), "Tecnología de la información - Técnicas de seguridad - Digital Firmas con el Apéndice - Parte 3: Logaritmo discreto - Mecanismos Basados " , "Tecnología de la información - Técnicas de seguridad -Técnicas criptográficas basadas en curvas elípticas Parte 2: firmas digitales ", y el Instituto Americano de Estándares Nacionales, "Clave pública Criptografía para la industria de servicios financieros: Algoritmo de firma digital de curva elíptica (ECDSA) ".

El nivel de seguridad proporcionado por los cifrados simétricos y las funciones hash utilizadas junto con los parámetros del dominio de la curva elíptica deberían coincidir o con el nivel proporcionado por los parámetros del dominio. En Tabla 2, se indican los tamaños mínimos de clave para cifrados simétricos y funciones hash que proporcionan una seguridad comparable (Lochter, 2010).

Tabla 2

Comparación tamaños mínimos de clave para cifrado de curva elíptica y funciones hash.

Parámetro de Dominio	Longitud mínima clave simétrica	Función Hash
brainpoolP160r1	80	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512
brainpoolP192r1	96	SHA-224, SHA-256, SHA-384, SHA-512
brainpoolP224r1	112	SHA-224, SHA-256, SHA-384, SHA-512
brainpoolP256r1	128	SHA-256, SHA-384, SHA-512
brainpoolP320r1	160	SHA-384, SHA-512
brainpoolP384r1	192	SHA-384, SHA-512
brainpoolP512r1	256	SHA-512

Nota. Recuperado de Elliptic Curve Cryptography (ECC) Brainpool Standard. Copyrigth 2010 de Lochter, M.

Generación de Llaves

Para la generación de llaves de un criptosistema de curva elíptica el usuario debe seguir los siguientes pasos:

- Generar una curva elíptica E, sobre el campo finito F_q.
- Seleccionar un punto generador G, de la curva elíptica y su orden debe ser primo, n con $n \approx q$.
- Generarla clave privada d, un número aleatorio tal que $1 < d < n - 1$, para calcular la clave pública P_b.

$$\boldsymbol{P_b = d * G} \quad (6)$$

ECDH

El esquema de acuerdo de clave anónima Elliptic Curve Diffie – Hellman Key Exchange, permite a dos partes, cada una con un par de claves pública-privada de curva elíptica, establecer un secreto compartido sobre un canal inseguro (Svetlin Nakov, 2018). ECDH se basa en la siguiente propiedad de los puntos EC:

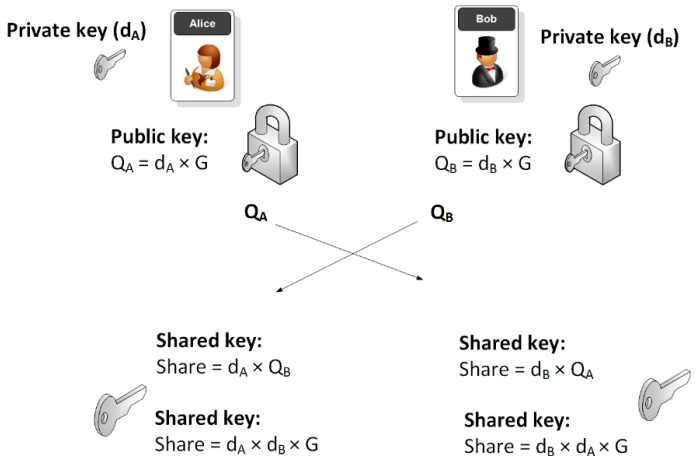
$$(\boldsymbol{a} * \boldsymbol{G}) * \boldsymbol{b} = (\boldsymbol{b} * \boldsymbol{G}) * \boldsymbol{a} \quad (7)$$

alicePubKey * bobPrivKey = bobPubKey * alicePrivKey = *secret* (8)

Si tenemos dos números secretos a y b (dos claves privadas, pertenecientes a Alice y Bob) y una curva elíptica ECC con el punto generador G, podemos intercambiar sobre un canal inseguro los valores (a * G) y (b * G) (las claves públicas de Alice y Bob) y luego podemos derivar un secreto compartido como se observa en la ecuación 7 y de manera gráfica en Figura 5.

Figura 5.

Esquema de acuerdo de clave anónima.



Nota. El gráfico describe el proceso de generación del secreto compartido a partir de las claves públicas y privadas de ambos usuarios. Tomado de Elliptic Curve Diffie Hellman (ECDH) with differing elliptic curves por Security Site, 2018.

El algoritmo ECDH (Elliptic Curve Diffie – Hellman Key Exchange) realiza el siguiente proceso:

- Alice genera un par de claves ECC aleatorias: {alicePrivKey, alicePubKey = alicePrivKey * G}
- Bob genera un par de claves ECC aleatorias: {bobPrivKey, bobPubKey = bobPrivKey * G}
- Alice y Bob intercambian sus claves públicas a través del canal inseguro (por ejemplo, a través de Internet)
- Alice calcula sharedKey = bobPubKey * alicePrivKey
- Bob calcula sharedKey = alicePubKey * bobPrivKey

Ahora, tanto Alice como Bob tienen el mismo secreto compartido como se describe en la ecuación 8.

Encriptación del Mensaje

La misión de proteger la confidencialidad, integridad y autenticidad del mensaje a través de varias comunicaciones se ha convertido en una preocupación importante especialmente con el uso creciente de técnicas digitales para transmitir y almacenar los mensajes.

En la mayoría de los sistemas criptográficos, se debe tener un método para mapear el mensaje en un valor numérico sobre el cual podemos realizar operaciones matemáticas. Para usar curvas elípticas, se requiere de un método para mapear el mensaje en un punto en una curva elíptica y obtener un nuevo punto, lo que será presentado como el texto cifrado.

El problema de codificar mensajes como puntos en una curva elíptica no es tan simple como se supone. En particular, no hay un algoritmo determinista que permita escribir puntos en una curva elíptica arbitraria (Reyad, 2018).

Seguridad

El objetivo de un problema computacional es encontrar el algoritmo más eficiente para resolverlo. La teoría de la complejidad computacional es una rama de la teoría de la computación, que estudia de manera teórica la complejidad referente a la resolución de un problema computable (Gustavo, 2012).

El problema de logaritmo discreto sobre el que se basa la encriptación de curvas elípticas requiere que los ataques sean sofisticados y que tengan conceptos matemáticos cada vez más complejos. El ataque por fuerza bruta consiste en probar

todos los posibles valores de un entero tal que sea igual a la clave privada (Centro Criptológico Nacional, 2012).

La criptografía de curva elíptica, al igual que los sistemas criptográficos asimétricos y basa su seguridad en un problema matemático que con los conocimientos y tecnologías actuales no puede ser resuelto en un tiempo razonable.

Bouncy Castel

Es una colección de interfaces de programación de aplicaciones enfocados en la criptografía desarrolladas por la organización benéfica “Legion of the Bouncy Castle”. Es de libre distribución y tiene versión para su ejecución en java, también se enfoca en el desarrollo de aplicaciones criptográficas para Android con la versión Spongy Castel.

Java, proporciona un soporte de cifrado, pero no es completo, por ello se requiere el uso de librerías complementarias (Candel, 2018).Facilita la aplicación de operaciones propias de la curva elíptica y los puntos que la conforman, como creación de puntos de la curva, suma de puntos y obtención de coordenadas de cada punto.

Base de datos

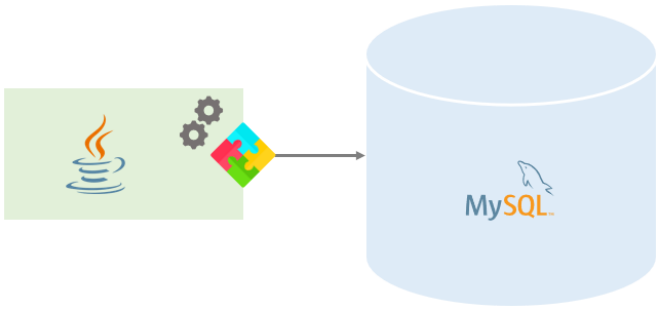
Una base de datos es una colección organizada de información estructurada, almacenada electrónicamente en un sistema informático y generalmente está controlada por un sistema de gestión de bases de datos lo cual se denomina un sistema de base de datos. Los datos pueden ser gestionados, modificados, actualizados, controlados y organizados fácilmente. La mayoría de las bases de datos utilizan lenguaje de consulta estructurado para escribir y consultar datos (Oracle, 2018).

MySQL

Matias Fossati menciona en el documento “Todo sobre MySQL”, MySQL es una base de datos ideal para aplicaciones donde hay baja concurrencia en la modificación de datos y el entorno es intensivo en lectura de datos. Además de trabajar con el lenguaje de consulta estructurado SQL, como se observa Figura 6, tiene compatibilidad entre sistemas y dispone de conjunto de funciones con el fin de ser utilizadas por otro software en diferentes lenguajes de programación.

Figura 6.

Java proporciona una interfaz estándar para interactuar con bases de datos.



Nota. El gráfico describe la versatilidad de MYSQL para trabajar con diversos sistemas operativos por medio de la interfaz de Java. Tomado de MySQL and Java Developer's Guide por Matthews, 2003, Wiley Publishing Group.

Xampp

Es una distribución de Apache que incluye software libre, el servidor web Apache y el lenguaje PHP permite configurar la base de datos en MySQL y el almacenamiento de datos para servicios web. Para cambiar la configuración usando Xampp se debe encontrar el archivo my.cnf que se encuentra en la carpeta myql\bin donde el programa se encuentra instalado (Digital Guide IONOS, 2019).

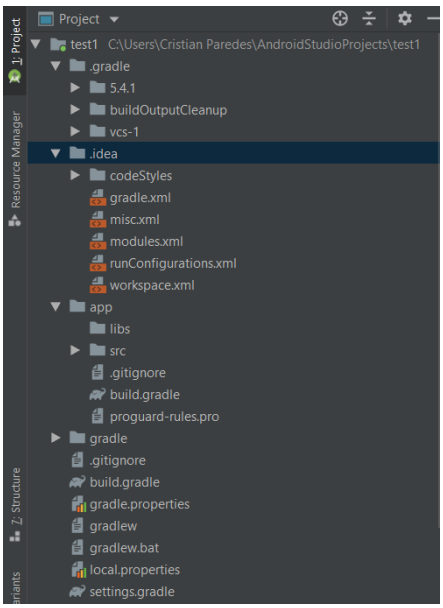
Android Studio

Android Studio en el documento “Guía de usuario” en su página web menciona que es el entorno de desarrollo integrado oficial para el desarrollo de apps para Android. Además del potente editor de códigos y las herramientas para desarrolladores ofrece otras funciones como: insertar cambios de códigos y recursos a la aplicación en ejecución sin reiniciar la aplicación, un emulador rápido y cargado de funciones, un sistema de compilación flexible basado en el sistema de compilación Gradle.

Cada proyecto de Android Studio incluye uno o más módulos con archivos de código fuente y archivos de recursos, además muestra los archivos de tu proyecto en la vista de proyecto de Android de manera organizada, como se ve en la Figura 7.

Figura 7.

Archivos de proyecto en Android Studio.



Nota. El gráfico muestra cómo se visualizan los elementos en un proyecto en Android estudio.

Java Enterprise Edition

Según Groussard T. en su libro “Desarrollo de aplicaciones web con JEE 6”, indica que la tecnología JEE, contribuye con el desarrollo de aplicaciones distribuidas y es considerada una normativa que describe los elementos que las constituyen su funcionamiento, por ejemplo:

- La manera de desarrollarse los componentes de una aplicación
- La comunicación entre componente o con otras aplicaciones
- La forma de organización de os componentes para construir una aplicación.
- Las restricciones que deben respetar los servidores encargados de almacenar las aplicaciones.

La ventaja de Java Enterprise Edition con relación a otras tecnologías radica en que, en el caso de requerir evolucionar hacia otro servidor con más rendimiento no se deben realizar grandes cambios a la aplicación.

Protocolo HTTP

Es un protocolo cliente servidor que gestiona las transacciones web, es el que nos permite navegar mediante la introducción de direcciones web y el seguimiento de enlaces. El navegador web es una aplicación cliente HTTP que nos permite conectar con servidores que alojan páginas web y que poseen software con servidor HTTP instalados.

HTTP es un protocolo sin estados, es decir no guarda ninguna información sobre conexiones anteriores, una vez se ha producido una solicitud y una respuesta tanto cliente como servidor olvidan los datos. Este problema se resuelve mediante el uso de

cookies, los cuales son archivos que mantiene información que un servidor almacena como lo son los datos de autenticación por ello, se mantiene de forma virtual la conexión manteniendo el concepto de sesión (Rules, 2016).

Servicios Rest

Su nombre corresponde a las siglas para representar "Representational State Transfer", cuya característica principal es no tener estado es decir no recordar peticiones anteriores a la que se está ejecutando. Por ello, el cliente mantiene el estado anterior y lo transfiere en cada llamada, para que un servicio REST me recuerde un estado debo identificarme en cada llamada que realice enviado usuario, contraseña o un token como credencial.

La principal ventaja de no poseer estados es que no se requiere recursos como memoria para almacenar los estados de cada usuario, la escalabilidad permite que los servicios REST no se saturen con todo el volumen de información que implicaría guardar muchos estados de incontables clientes (Tomás, 2014).

Api Rest.

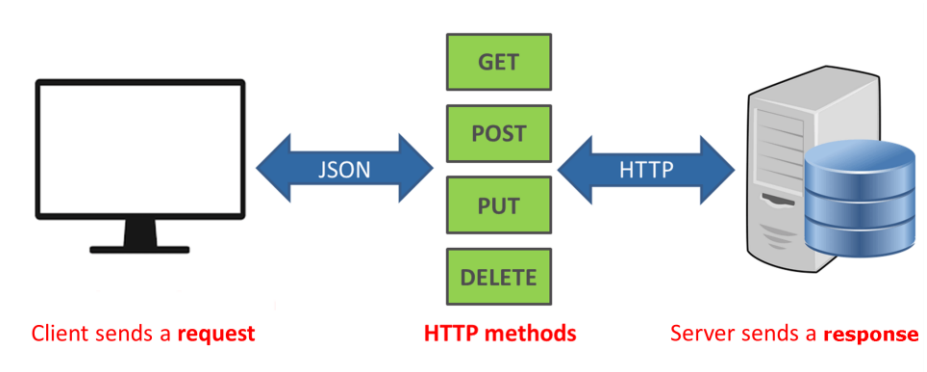
Consiste en una combinación entre la interfaz de programación de aplicaciones y los servicios REST que producen un estándar lógico y eficiente para la creación de servicios web (Crespo, 2018). En Figura 8 se observa el proceso de las operaciones más importantes con datos tipo JSON o XML, son cuatro y se detallan a continuación:

- GET: Recupera información de recursos únicamente y no permite modificarla de ninguna manera.

- POST: Permite crear un nuevo recurso subordinado en la colección de recursos.
- PUT: Permite actualizar el recurso existente, las solicitudes PUT se realizan en un recurso individual.
- DELETE: Permite eliminar recursos.

Figura 8.

Estructura de comunicación de API REST



Nota. El gráfico muestra el proceso de comunicación entre el cliente y el servidor al momento de realizar una petición desde la interfaz generada en Android Studio. Tomado de API REST por Enrique Crespo, 2018.

JSON

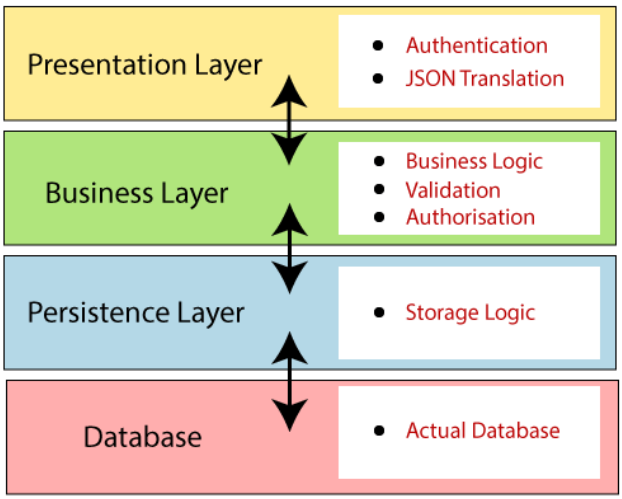
JSON es un formato de texto independiente del lenguaje de programación, pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades convierten al formato JSON en un lenguaje ideal para el intercambio de datos (JSON ORG, 2012).

Spring Boot

Es un módulo de Spring Framework que se usa para el desarrollo de aplicaciones de producción con un mínimo de trabajo al momento de realizar configuraciones. Tiene una arquitectura por capas en donde cada capa se comunica únicamente con la capa superior o inferior, en forma jerárquica como se observa en la Figura 9.

Figura 9.

Arquitectura por capas del módulo Spring Boot.



Nota. El gráfico muestra el proceso de comunicación entre capas dentro del módulo Spring Boot el cual nos permite configurar de manera efectiva aplicaciones móviles. Tomado de JavaTpoint por Sonoo Jaiswal , 2015.

La capa de presentación se encarga de las peticiones HTTP, convierte los parámetros JSON enviados por el cliente en objetos, autentica las peticiones y las transfiere a la capa de negocio. La cual contiene toda la lógica del negocio y está

formada a su vez en clases correspondientes a servicios que utiliza los cuales son proporcionados por la capa de acceso a datos, también se encarga de validaciones y autorización de ingreso.

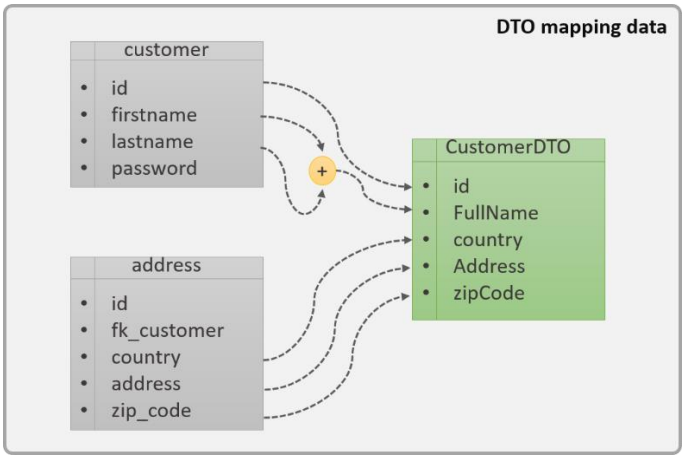
La capa de persistencia contiene toda la lógica de almacenamiento y se encarga de convertir objetos del negocio en registros de la base de datos en forma bidireccional. Finalmente, la capa de base de datos lleva a cabo las operaciones del CRUD (create, retrieve, update, delete) en la base de datos (Jaiswal, 2015).

DTO

El patrón de diseño DTO gestiona la forma en que la información debe viajar desde la capa de servicios a las aplicaciones, es decir realiza un intercambio de información entre cliente y servidor con una serie de atributos que pueden ser enviados o recuperados del servidor en una sola invocación desde varias fuentes o tablas en una única clase (Blancarte, 2018).

Figura 10.

Mapeo de información del cliente por parte del servidor mediante DTO.



Nota. El gráfico muestra como el patrón de diseño gestiona la información de la base de datos de manera eficiente, realizando el envío de información de un mismo cliente en una sola petición. Tomado de Software Arquitect por Oscar Blancarte, 2018.

En figura 10, se puede observar que la información que obtiene el servidor proviene de dos tablas diferentes pertenecientes al mismo cliente, y puede seleccionar únicamente los datos que requiera o incluso formar un nuevo dato a partir de otros.

Las características necesarias para considerar que se utiliza DTO para transmitir información son:

- Lectura: La característica de transmisión de información implica que no se realicen operaciones con los datos obtenidos en la clase DTO, ya que en esta debe constar únicamente los métodos GET y SET.
- Serializable: Para que un objeto pueda viajar a través de la red debe convertirse en un conjunto de bytes representativos para después ser reconstruido.

ISO 27001

Es un estándar de seguridad internacional que describe los requisitos sugeridos para monitorear y mejorar un sistema de gestión de seguridad de la información (SGSI). Un SGSI es un conjunto de políticas para proteger y administrar la información confidencial de una empresa.

Los beneficios de trabajar con la certificación ISO 27001 incluye el control de acceso a la información dentro de una organización, reduciendo el riesgo de que dicha información pueda ser robada, además de trabajar con protocolos de gestión de la información que detallan cómo deben manejarse y transmitirse ciertos datos (IMPERVA, 2017).

Ofrece un enfoque sistemático con la implantación de controles de seguridad de la información eficaces, lo cual permite a una organización evaluar permanentemente los riesgos y las amenazas a las que está expuesta e impulsa las acciones necesarias para gestionarlos (Lloyd's Register, 2019).

La norma ISO 27002 trabaja de manera complementaria con ISO 27001 ya que, es una guía de buenas prácticas para la implantación de un sistema de gestión de seguridad de la información, donde todo lo relacionado a criptografía tiene un dominio de control propio en la sección 10 del anexo A y la gestión de claves es un aspecto que se debe desarrollar (Álvarez, 2017).

Para un conocimiento más extenso sobre la implementación o certificación de la norma ISO 27001 se recomienda contactarse con auditores o expertos, también puede adquirir la norma en su página oficial.

Gestión de Claves

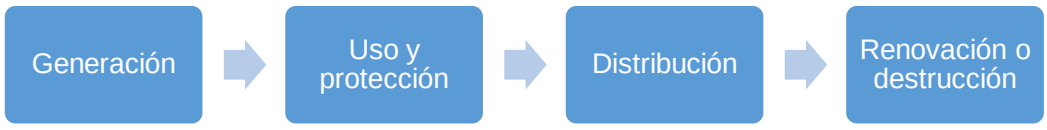
Los controles son obligatorios en la medida en que tengan aplicabilidad, por ello se debe definir qué medida de seguridad dentro del anexo A de la norma ISO 27001 se puede implementar.

Dentro de este contexto se tomará como guía la sección 10.1.2 “Gestión de Claves”, la cual indica que se debe desarrollar y aplicar una política sobre el uso, protección y tiempo de vida de las claves criptográficas.

La gestión de claves implica establecer políticas que tengan en cuenta el ciclo de vida completo de la clave, como se muestra en la Figura 11, se debe determinar fechas de activación y desactivación para reducir riesgos.

Figura 11.

Ciclo de vida de claves criptográficas



Nota. El gráfico describe el ciclo de vida da una clave generada por el algoritmo de curva elíptica.

Metodología del Proyecto de Investigación

Descripción general del proyecto de investigación

El presente trabajo de investigación tiene la intención de realizar la encriptación de información en una base de datos mediante la criptografía de curvas elípticas. A través de una aplicación móvil en Android, la información puede ser ingresada a la base de datos y ser desencriptada por el usuario que se encuentra registrado y autenticado.

Incluso si un intruso logra acceder a la base de datos remotamente mediante un dispositivo móvil y visualizar la información esta será incomprensible, ya que se encuentra encriptada. El usuario que posee la llave privada del método de encriptación es el único capaz, mediante la interfaz del dispositivo móvil, de obtener la información de manera legible.

La encriptación de la información sensible se realizará mediante la generación del secreto compartido propio del esquema ECDH, con la clave privada y pública del usuario las cuales se encuentran en el teléfono y en la base de datos respectivamente.

La interfaz permite el ingreso de la información a la base de datos de manera que, se ingrese de manera legible y que una vez la información se encuentre en la base sea no interpretable y se mantenga almacenada de esa manera hasta que sea desencriptada por el usuario mediante la interfaz que utiliza el algoritmo.

Los procesos de seguridad de la información son controlados por la norma ISO27001 mediante el establecimiento de políticas para la gestión de claves en el cual se gestiona todo el ciclo de vida de las claves para minimizar riesgos.

Finalmente, se evaluará la seguridad de la información en la base de datos cuando se accede a ella desde un dispositivo móvil mediante el programa Wireshark al

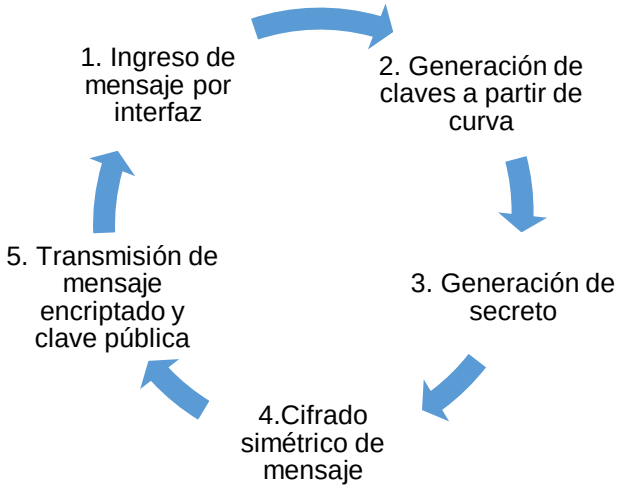
enviar paquetes de datos encriptados con información sensible, de esta manera se validará que la información viaja desde la interfaz hasta la base de datos de manera segura.

Proceso

Para realizar el cifrado de un mensaje por medio del esquema ECDH se realizó el proceso que se muestra en figura 12 y 13, donde las propiedades de la curva elíptica seleccionada es la base para la generación de claves pública y privada, y posterior generación secreto compartido.

Figura 11.

Proceso de encriptación de mensaje



Nota. El gráfico describe el proceso para realizar la encriptación de un texto plano mediante el algoritmo de curva elíptica para que la información sea ilegible.

El secreto es usado como una llave para realizar una encriptación simétrica, es decir con la misma clave que se encripta se desencripta, con el mensaje ingresado y de esa manera transmitirlo a la base de datos para ser almacenado.

Figura 12.

Proceso de desencriptación de mensaje

Nota. El gráfico describe el proceso para realizar la desencriptación de un texto plano mediante el algoritmo de curva elíptica para que la información pueda ser interpretada.

Generación de Curva Elíptica

Se trabaja con la curva brainpoolp256r1 recomendada por NIST (National Institute of Standards and Technology), cuyos parámetros son mostrados en Tabla 2, que proporciona una longitud mínima de claves de 128 bits con una computadora HP Pavilion con procesador de Intel(R) Core (TM) i5 @ 2.6 GHz y 8 GB de memoria RAM.

Generación de llaves

La generación de las claves pública y privada se realiza mediante la clase KeyPairGenerator de manera aleatoria con la clase SecureRandom para cumplir el criterio $d \in [1, n-1]$. Donde n es un número primo que representa el orden del punto base de la curva, dicho parámetro es propio de cada curva elíptica.

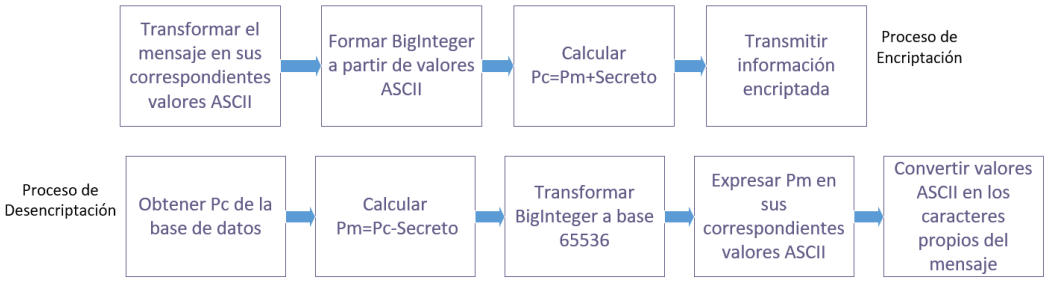
Cifrado de datos

Los datos deben ser tipo puntos de la curva elíptica para poder realizar operaciones propias de este tipo de datos como la suma de puntos. Esto implica que se debe realizar una conversión al tipo de datos “punto de la curva” o en Java conocido como ECPoint. El método de representación de datos como puntos de la curva utilizado abarca el cifrado en bloques lo significa que se va a generar un punto representativo por cada información ingresada en los campos de la base de datos y no un punto por cada letra.

El proceso de representación de la información en puntos de la curva elíptica se muestra en Figura 14, el diagrama de bloques indica además la desencriptación del mensaje obtenido desde la base de datos.

Figura 13.

Diagrama de bloques del proceso de representación del mensaje.



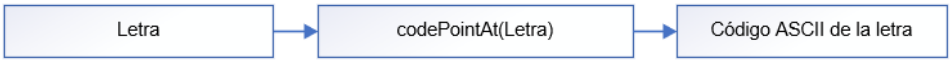
Nota. El gráfico describe el proceso para convertir el texto plano ingreso por usuario a un punto en la curva elíptica generada.

Conversión a ASCII

Los valores ASCI permiten asignar un valor numérico a una letra o símbolo de manera que se pueda realizar cálculos matemáticos con dichos valores. Por ello, como se muestra en la figura 14, el primer paso después de la obtención del texto ingresado, es obtener los correspondientes valores ASCII de cada letra con la función codePointAt() de tipo String propia de Java.

Figura 14.

Diagrama de bloques del proceso de conversión a código ASCII.



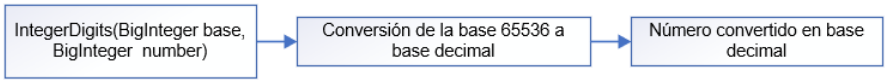
Nota. El gráfico describe el dato de entrada, la función que ejecuta el proceso de conversión a código ASCII y lo que se obtiene de este proceso.

Convertir a BigInteger

Los vectores formados se convierten en un número tipo BigInteger de base decimal obtenido de la conversión de la base 65536 formado por 15 valores ASCII que corresponden a la información ingresada en cada campo de la interfaz. Este proceso está desarrollado en la función todecimal la cual como se muestra en la figura 15, recibe como argumento un número BigInteger y la base en la que encuentra el número y devolverá un arreglo con el número convertido a base decimal.

Figura 15.

Diagrama de bloques del proceso de conversión de base 65536 a base decimal



Nota. El gráfico describe los datos de entrada, la función que ejecuta el proceso de conversión de base 65536 a base decimal y como dato de salida el número convertido a base decimal.

Encriptación de Curva Elíptica

Una vez representada la información ingresada en la interfaz en puntos de la curva se codifica con la clave pública calculada a partir de la clave privada, ver ecuación 6. La ecuación 8, representa la encriptación del mensaje (Pm) mediante el algoritmo de encriptación de la curva elíptica.

$$\boldsymbol{Pc = Pm + Secreto} \quad (8)$$

De esta manera, para saber desencriptar la información se requiere conocer la llave pública para realizar la generación del secreto. Esta operación se describe en la función encryptstring de la librería BouncyCastel, en la que se realizan las operaciones

definidas en las propiedades matemáticas de las curvas elípticas descritas en la sección 2.2, de manera que ambos tipos de datos puedan ser procesados.

Desencriptación de Datos

Obtenido el punto de la curva Pc desde la base de datos, se realiza la operación inversa para obtener Pm, como se muestra en la ecuación 9.

$$\mathbf{Pm} = \mathbf{Pc} - \mathbf{Secreto} \quad (9)$$

$$\mathbf{Pm} = \mathbf{Pm} + (\boldsymbol{Clave\ Privada} * \boldsymbol{Clave\ P\acute{u}blica}) - (\boldsymbol{Clave\ Privada} * \boldsymbol{Clave\ P\acute{u}blica}) \quad (10)$$

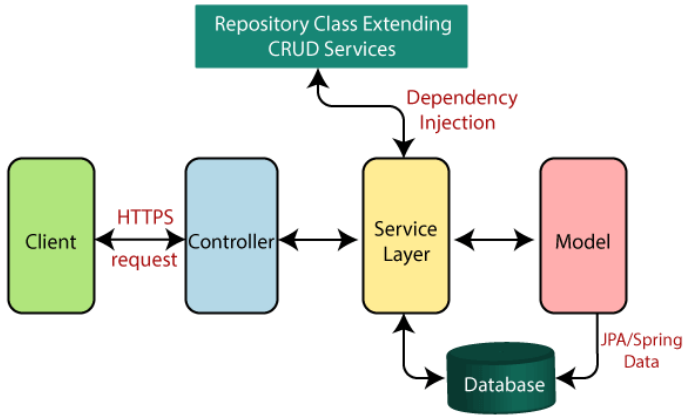
De la resta del secreto generado a partir de las claves pública y privada se obtiene Pm, el punto de la curva que representa la información ingresada por el usuario a la base de datos. La función que permite obtener el mensaje encriptado es decryptstring de la librería BouncyCastel, la cual es el proceso contrario a la función mencionada en la sección anterior.

Arquitectura del Sistema

El cliente Android mediante la interfaz de la app móvil hace una petición HTTP que es recibida por la capa de servicio del servidor Java. En este caso ya que son web services no existen controladores como tal como se observa en la figura 16, sino que la petición pasa directamente a la capa de servicio.

Figura 16.

Flujo de Arquitectura



Nota. El gráfico muestra el flujo que sigue la arquitectura del módulo Spring Boot.Tomado de JavaTpoint por Sonoo Jaiswal , 2015.

En las clases de servicios por inyección de dependencias se instancian las clases de repositorio necesarias, que son clases que contienen las operaciones del CRUD y otras operaciones desarrolladas. Las clases de repositorio se comunican con la base de datos a través de las clases de modelo y la capa de persistencia se maneja con JPA que es un Api de Java para manejar estructuras de datos relacionales (Jaiswal, 2015).

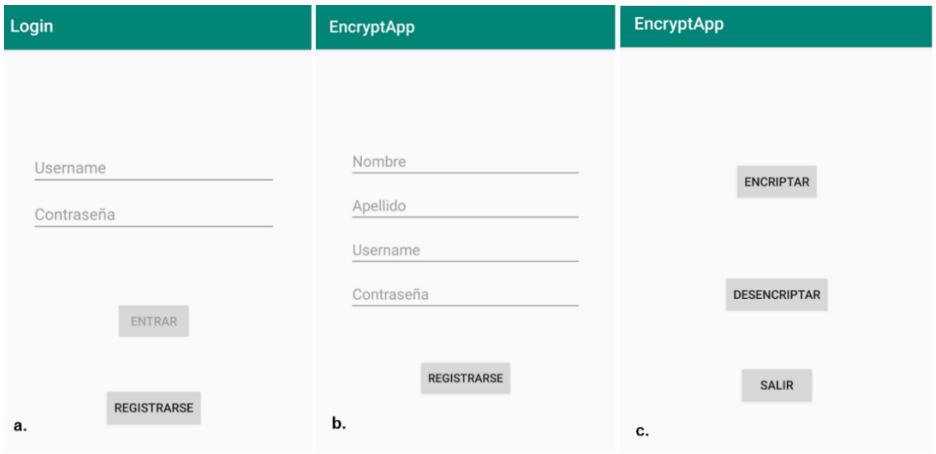
Interfaz de Aplicación Móvil

La aplicación móvil presenta la interfaz que se observa en figura 17, en la sección a. se tiene la opción de ingresar con un usuario previamente registrado o registrarse en caso de ser un nuevo usuario. En caso de ser usuario nuevo se puede registrar ingresando los campos que especificados en la sección b. y en caso de autenticarse con la clave propia de la aplicación.

Una vez autenticado el usuario, se puede acceder a realizar el ingreso de las contraseñas para ser almacenadas en una base de datos encriptada o visualizar la contraseña de una determinada plataforma.

Figura 17 .

Interfaz de aplicación móvil



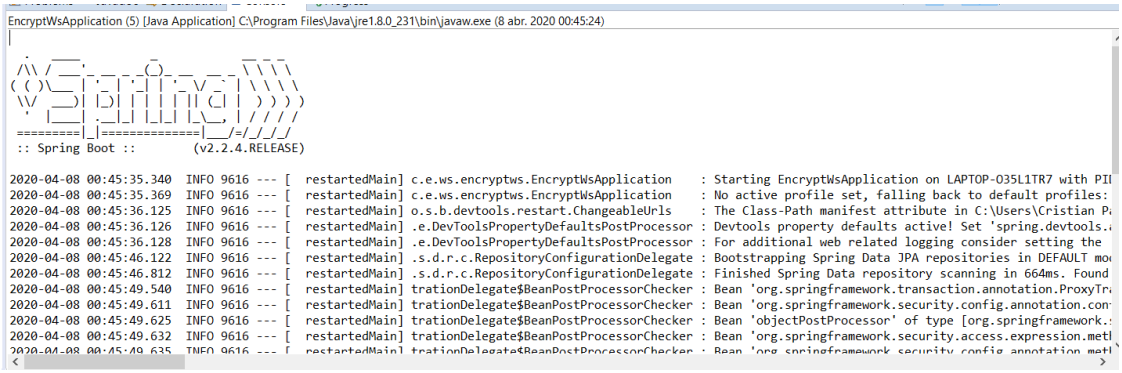
Nota. El gráfico los escenarios con los que interactúa el usuario en su teléfono. a. Pantalla de inicio, b. Pantalla de registro, c. Selección de actividad encriptar/desencriptar o salir.

Ingreso de usuarios

El proceso interno de la funcionalidad de la aplicación implica primero levantar el servidor de la base de datos por medio del programa Xampp como se observa en la figura 24. Siguiendo la estructura de la arquitectura del proyecto se levanta el servidor Java, como se observa en la figura 18, que permitirá gestionar las peticiones del cliente desde la aplicación en Android por medio del protocolo HTTP.

Figura 18.

Servidor Java Spring Boot



Nota. El gráfico muestra la ejecución del servidor Java para levantar los servicios de comunicación con la base de datos.

Después de esto es posible acceder a la información en la base datos o registrar nueva información en ella. Para realizar el registro de usuario en la interfaz de la aplicación, se registra el nombre, apellido, usuario y clave para acceder a la aplicación como se muestra en la figura 19.

Figura 19.

Registro de usuario

Nota. El gráfico muestra el registro de un nuevo usuario mediante la interfaz móvil.

En figura 20 se observa que el registro se ha creado satisfactoriamente y se almacena en la base datos encriptado con el fin de que si un intruso accede a la información del servidor de la base de datos no pueda comprender la misma.

Figura 20.

Registros en base de datos.

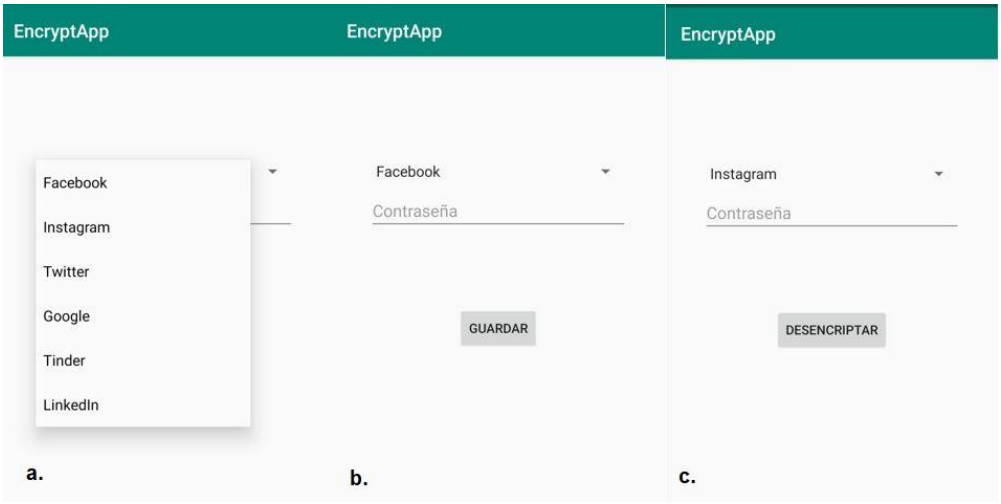
Nota. El gráfico muestra el registro de los datos encriptados del usuario en la base de datos.

Una vez autenticado el usuario registrado, al escribir un dato a ser encriptado en este caso contraseñas de plataformas digitales en la figura 21 sección a., se evidencia que el usuario puede almacenar de manera segura las contraseñas de Facebook, Instagram, Twitter, Google, Tinder y LinkedIn.

De igual manera en la sección b. y c. se observa que despliega el menú de plataformas a seleccionar tanto para encriptar y guardar o, para desencriptar y mostrar.

Figura 21.

Encriptación y Desencriptación en aplicación móvil.



Nota. El gráfico los escenarios con los que interactúa el usuario cuando guarda una clave mediante la aplicación. a. Despliegue de plataformas, b. Registro de clave de cuenta de Facebook, c. Registro de clave de cuenta de Instagram.

Base de datos

La base de datos está conformada por la información de claves personales del usuario en las diferentes plataformas digitales y los datos de usuario, las tablas que la conforman son los siguientes:

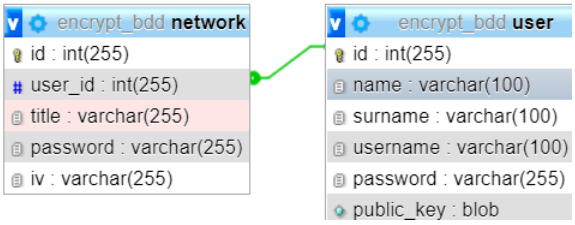
- User, contiene la información propia del usuario como: nombre, apellido, nombre de usuario, contraseña y clave pública única para cada usuario, id número de usuario que se encuentra registrado.
- Network, contiene la información propia de cada plataforma como: id número de la plataforma digital, nombre de la plataforma, contraseña guardada por el usuario para

ingresar a esa plataforma, vector de inicialización el cual permite generar distintos cifrados de información para un mismo texto.

En la figura 23, se presenta la estructura de la base de datos la cual contiene tablas relacionadas ya que de acuerdo con el id del usuario le corresponde diferentes claves de cada plataforma.

Figura 22.

Estructura de base de datos.

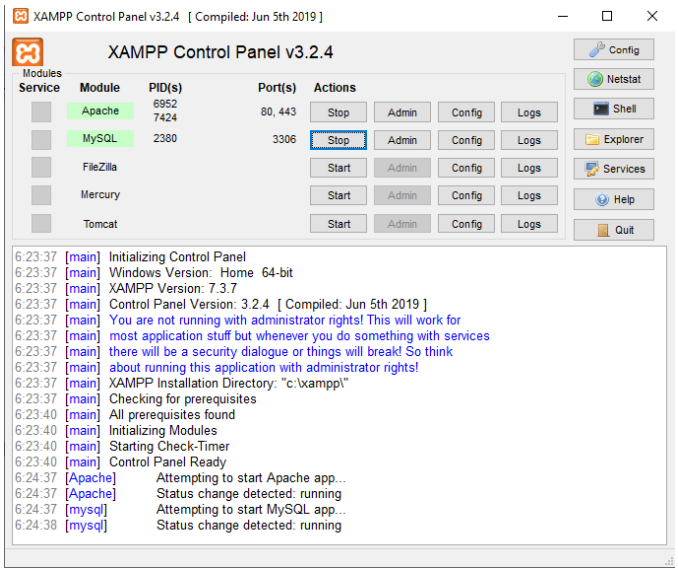


Nota. El gráfico muestra la estructura y relación de las tablas que conforman las bases de datos de usuarios.

Para acceder a la base de datos MySQL se debe ejecutar el servidor Xampp e inicializar los módulos Apache y MySQL, en la figura 23 se observa la interfaz propia del programa. En el botón “Admin”, se ejecuta phpMyAdmin para manejar la administración de MySQL. En phpMyAdmin se puede crear bases de datos, examinar la información y definir la estructura de la tabla donde se almacena la información encriptada.

Figura 23.

Ejecución de servidor Xampp.



Nota. El gráfico muestra la interfaz del programa Xampp que permite establecer la conexión con la base de datos MYSQL desde el servidor.

Servidor Java

El objetivo de este recurso es levantar la capa de persistencia, para que los datos se transmitan desde la aplicación hacia la base de datos de manera que se mantengan las relaciones entre valores de tablas relacionadas y a su vez gestionar las peticiones del usuario.

- Capa de modelo, realiza la abstracción del modelo de negocio es decir entidades, usuarios, se crean las entidades con sus respectivas relaciones
- Controladores, son clases que acceder a la capa de modelo para realizar acciones de lectura, escritura, modificación y borrado.

- Vista, es donde se generan las apis para que sean visibles se usa modelo vista controlador, singlenton, dto.
- ORM, está en modelo realiza un mapeo del modelo de clases a un modelo entidad relación.

Gestión de claves

Para el desarrollo de este trabajo y para fines aplicativos al mismo se establecen las siguientes políticas en cumplimiento a la norma ISO 27001, con el objetivo de minimizar riesgos en la seguridad dentro de ese escenario:

- La generación de la clave privada debe ser aleatoria siguiendo las propiedades matemáticas de las curvas elípticas que se explican en la sección 2.2.3 de este documento.
- La clave privada generada no será compartida, una vez generada en el dispositivo móvil se almacenará encriptada en la base de datos interna del dispositivo SQLite.
- Toda información del usuario registrada en la aplicación debe ser guardada en la base de datos encriptada mediante el algoritmo criptográfico.
- La información que se almacenará en la base de datos no debe visualizarse de manera legible en el canal de transmisión hacia la base de datos.
- La contraseña personal de cada usuario no deberá ser transmitida por ningún medio digital para evitar ser víctima de atacantes por medio de la ingeniería social.
- La contraseña personal de cada usuario en la interfaz debe tener mínimo 8 caracteres y contener al menos un carácter especial.

- La clave privada se deberá cambiar en un lapso mínimo de 8 meses desde el registro del usuario y una vez realizado el primer cambio, el segundo cambio se realizará en un periodo de 1 año.

Análisis de Resultados

El proceso de encriptación se realiza a toda la información que el usuario ingresa a por medio de la aplicación móvil, entre ellos campos como nombre completo, usuario, contraseña de ingreso y la contraseña propia de cada plataforma dependiendo del tipo de red social que seleccione.

La información encriptada es almacenada en una base de datos MySQL mediante el servidor Xampp con la administración de phpMyAdmin de manera gráfica. El proceso que se describe fue aplicado para cada campo correspondiente al nombre del usuario y los parámetros de la curva son los siguientes:

$$\boldsymbol{n} = 115792089210356248762697446949407573529996955224135$$

$$760342422259061068512044369$$

$$\textbf{EC Private Key} = [\text{c9: 28: f7: a8: a7: e6: f8: 98: 85: d5: 57: 8a: c3: e1: cc: 78: db: 7e: 40: 55}]$$

$$\textbf{EC Public Key} = \text{X: 5ca0ae81791b1784a3fae33553cc5314373daabafc7fe896e82efab83d0f17b6}$$

$$\text{Y: e63845e93d42635497b59e12b0cb64d620a3042a327daa85cc4a6ec72b3978d}$$

$$\boldsymbol{G} = 6b17d1f2e12c4247f8bce6e563a440f277037d812deb33a0f4a13945d898c296,$$

$$4fe342e2fe1a7f9b8ee7eb4a7c0f9e162bce33576b315ececbb6406837bf51f5$$

Donde cada uno representa respectivamente:

- n, número primo que representa el orden del punto base de la curva.
- EC Private Key, clave privada generada.
- G, punto generador de la curva elíptica.
- Ec Public Key, clave pública generada.

Una vez establecidos los parámetros de la curva, se procede a la encriptación de datos registrados los cuales se desarrollan en la siguiente sección.

Encriptación

A continuación, se mostrarán los pasos que se ejecutan cuando se realiza la encriptación elíptica de una información determinada en este caso, el nombre del usuario.

- Insertar nombre de usuario: kcastillo
- Los valores ASCII equivalentes son: $\{107,99\ ,97,115,116,105,108,108,111\}$
- Convertir a base decimal el número 107997115116105108108111 formado por los valores ASCII en base 65536, que da como resultado el número $\{113999290923567984853125612857907836245105850253422\}$.

Paralelamente, se realiza el cálculo del secreto compartido con la clave pública y privada del usuario, para la aplicación en este proyecto no se generará la clave de un usuario B ya que en la transmisión de información solo se implica a un usuario.

$$Secreto\ Compartido = javax.crypto.spec.SecretKeySpec@17458$$

El punto de la curva Pc es enviado a la función encryptstring de la librería BouncyCastel, la cual realiza el proceso de encriptación con el secreto compartido generado de la clave pública y privada, teniendo como resultado el nombre de usuario encriptado y listo para ser transmitido a la base de datos.

$$54CF528BA53D29461493FC1370089868DC9952460158E9EB64$$

Este proceso se realiza para la encriptación de toda la información del usuario ya que toda es transmitida por un canal inseguro. En la Tabla 3 del documento se registra el nombre de usuario encriptado para diferentes longitudes de clave privada, con el fin de analizar si este influye directamente en la longitud del texto encriptado.

Tabla 3

Comparación de longitud de texto encriptado entre claves de diversa longitud.

Parámetro de Dominio	Longitud mínima clave privada[bits]	Texto encriptado
brainpoolP160r1	80	760601246E21A56EA0272CD164E8DB9664611ED901A35ABD8B
brainpoolP192r1	96	D2ABD69F07AA687119E7B304FC83C6B23F88037D2041EE385D
brainpoolP224r1	112	F30AB69040C19539F84EDC96BDAB6424345117A3BCA9CC7499
brainpoolP256r1	128	0C5298EBE3E7AF6A10B9069711327300177 9D0AE070C623188
brainpoolP320r1	160	3228BE24DBAF02C54577AC79436ED428F9EB0DA43DFAE3C718
brainpoolP384r1	192	2FB101EAFF94C00E1594247D37C8CB6710760F8AB7F0B9A274
brainpoolP512r1	256	1399F3C6E6354A2FCD9827382A545514AEDE7DF75D2419DA4F

Nota. Esta tabla describe el texto encriptado dependiendo del tipo de curva elíptica y la longitud de su clave privada.

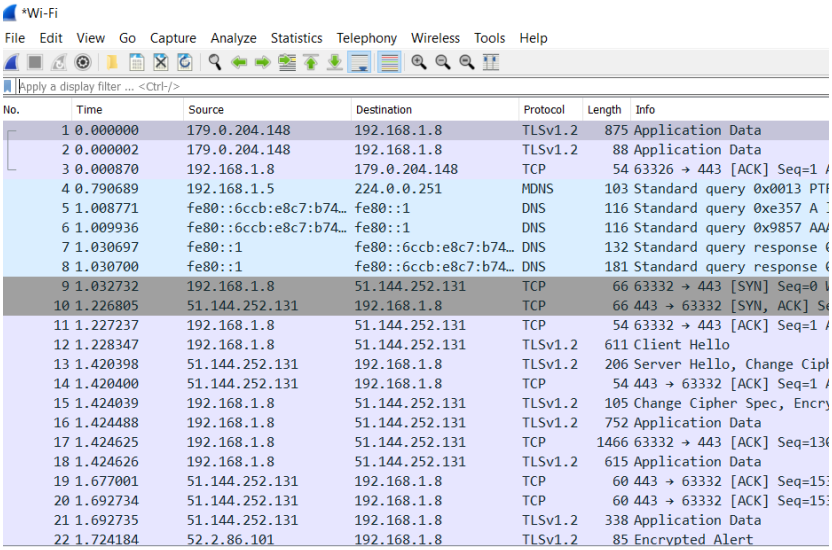
Se visualiza que el tamaño de caracteres que conforman el texto encriptado no depende del tamaño de bits de la clave privada, ya que la longitud del texto permanece constante ante la variación de la longitud de la clave privada.

Transmisión de Información

Para el análisis de los paquetes que se generan en la transmisión de información desde el cliente, es decir la aplicación móvil hasta el servidor de la base de datos se utilizó el programa Wireshark. En el cual filtramos el canal de transmisión que

deseamos escuchar, en este caso la red inalámbrica local a la cual se encuentra conectado el cliente y el servidor.

Figura 24. *Interfaz de transmisión de paquetes en Wireshark*



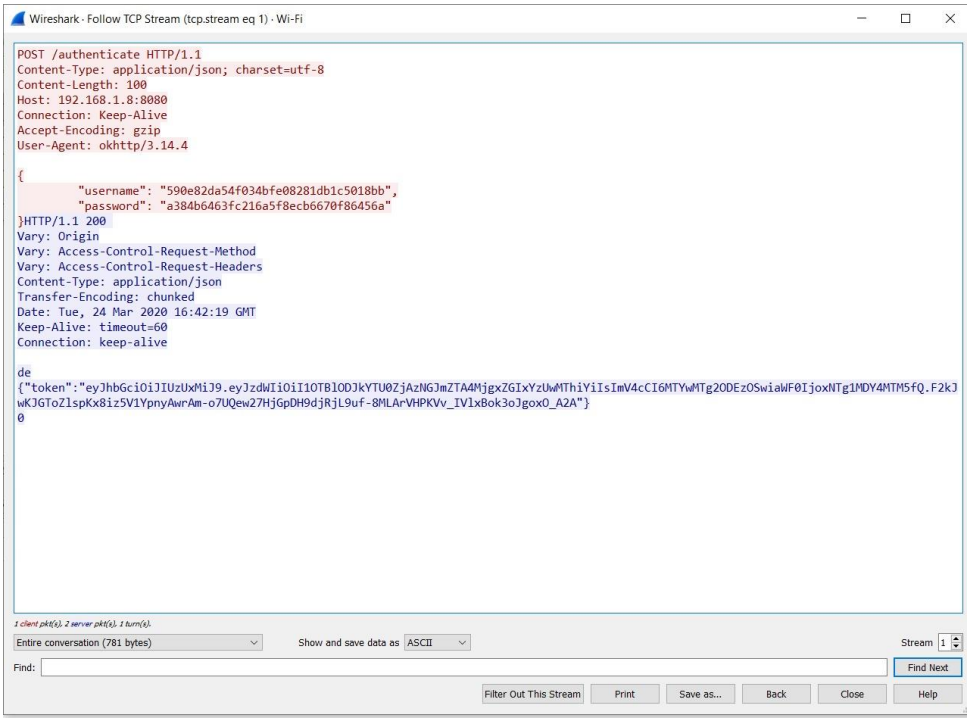
Nota. El gráfico muestra los paquetes transmitidos en la red local inalámbrica.

.

Al correr el programa y escuchar los paquetes transmitidos en la red local inalámbrica que se observa en la figura 25, procedemos a filtrar los paquetes que pertenecen a la comunicación entre el cliente y servidor por medio de sus respectivas direcciones ip. Para este proyecto el cliente y el servidor poseen las direcciones ip 192.168.1.7 y el 192.168.1.8 respectivamente.

Figura 25.

Paquetes transmitidos desde interfaz de usuario.

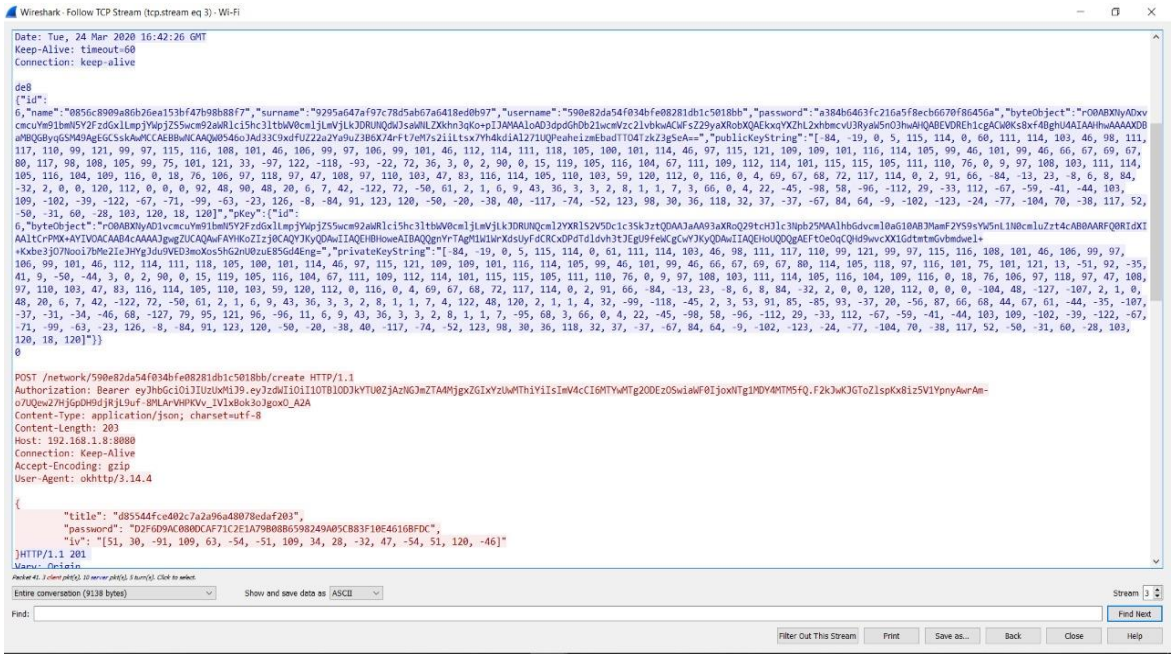


Nota. El gráfico muestra un paquete capturado en la transmisión de información por la red inalámbrica local.

Al validar tu usuario y contraseña para acceder al sistema se envían las credenciales del usuario para la respectiva autenticación y cómo respuesta se obtiene un token de inicio de sesión el cual será diferente cada vez que ingrese el usuario a la plataforma. La cadena de token la cual se puede visualizar en la figura 26, permite garantizar que el usuario que realiza una solicitud de servicio se encuentra autentificado de manera que, este proceso sea transparente para el usuario.

Figura 26.

Transmisión de información desde usuario a base de datos

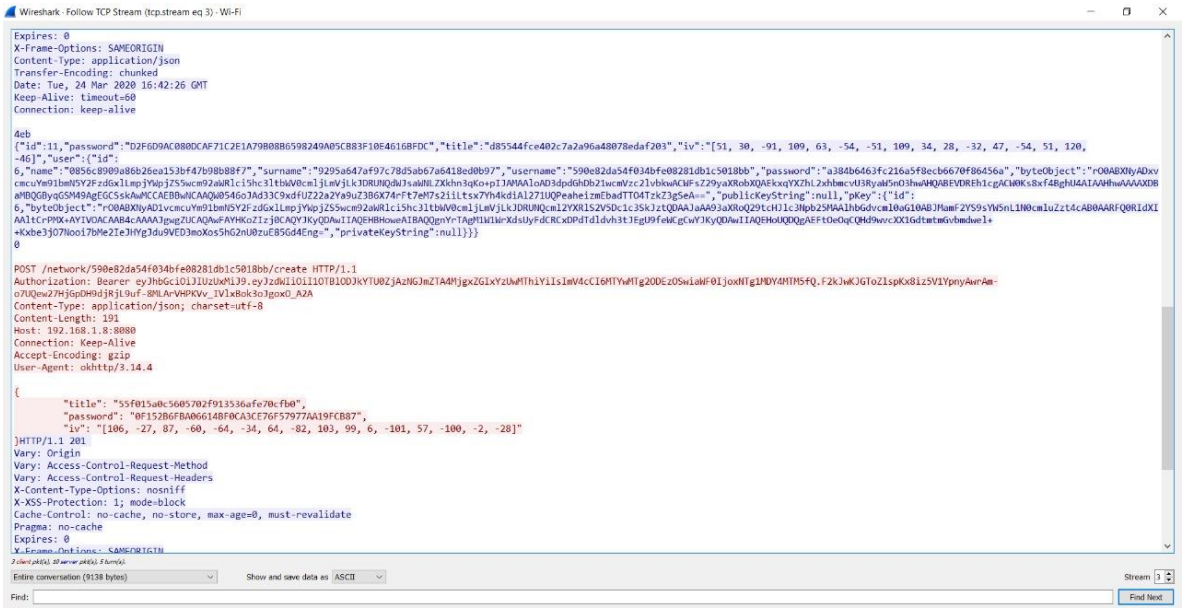


Nota. El gráfico muestra un paquete capturado en la transmisión de información por la red inalámbrica local hacia la base datos.

Cuando un usuario solicita guardar una contraseña de determinada plataforma como Facebook, se puede observar en la figura 27 que se transmiten los datos de autenticación, es decir, la información que el usuario registró anteriormente además de, la plataforma seleccionada y la contraseña respectiva para esta.

Figura 27.

Transmisión de información desde base de datos a usuario



Nota. El gráfico muestra un paquete capturado en la transmisión de información por la red inalámbrica local hacia el usuario desde la base datos.

Como se observa en figura 28, la misma información viaja cuando se realiza la transmisión de información desde la base de datos hacia el usuario para que pueda interpretar la información ya que se muestra de manera legible en la interfaz.

Desencriptación

Este proceso es necesario cuando el usuario o cliente requiera revisar la información que se encuentra almacenada, para este proceso se realizan los siguientes pasos:

- Obtener el punto Pc o la información encriptada desde la base de datos.

- Realizar la resta descrita en la ecuación 10 para obtener Pm, ya que el valor del secreto compartido se vuelve a calcular rápidamente con las claves privada y pública.
- Se procede a realizar el proceso inverso que se realizó en el mensaje original, convirtiendo el número obtenido de la resta con el secreto compartido que se encuentra en base 10 a base 65536.

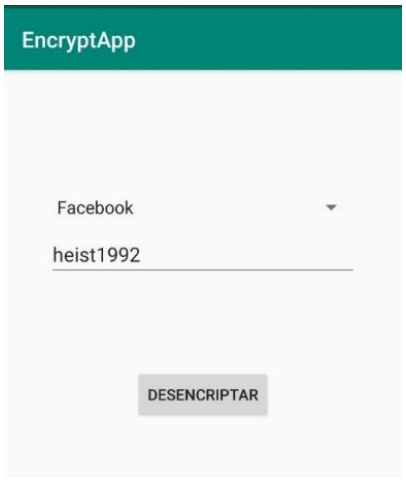
$$113999290923567984853125612857907836245105850253422_{10}$$

$$1079997115116105108108111_{65536}$$

- Cada número que resulta de la conversión a base 65536 son los valores ASCII representativos del mensaje original por ello se procede con la conversión a caracteres que corresponden al mensaje original.

Figura 28.

Obtención de información desencriptada



Nota. El gráfico muestra la visualización del usuario cuando consulta la contraseña de una de las plataformas que ya guardó anteriormente por medio de la interfaz.

Finalmente, este proceso se realiza únicamente para la información que se desea mostrar al cliente por medio de la interfaz como se observa figura 28.

Tiempo de Ataque

Los ataques más conocidos en ECC son el método Rho de Pollard y el método Lambda de Pollard. Se espera que el método Rho de Pollard encuentre la clave privada en pasos constantes de tiempo $\sqrt{n}$, donde n es el orden cíclico de la curva elíptica con G como generador. El método Lambda de Pollard es similar al método Rho de Pollard, pero utiliza muchos puntos de partida para encontrar una coincidencia. El método lambda de Pollard también espera encontrar la clave privada a lo sumo en un tiempo constante $\sqrt{n}$ pasos (Singh, 2015).

Si se implementa en paralelo, el tiempo de ejecución para encontrar la clave privada se puede reducir. Ambos métodos son probabilísticos, es decir, tienen una alta probabilidad pero no garantiza terminar en un tiempo constante de pasos $\sqrt{n}$.

Para una curva elíptica de 80 bits de longitud de clave el valor de n es,

$$1332297598440044874827085038830181364212942568457$$

$$\sqrt{n}\ pasos = 1.154\ x10^{24} pasos$$

Asumiendo que, cada paso dura 0.0000001 segundos, el tiempo de ataque sería aproximadamente $1.335x10^{19}$ días. En la tabla 4 del documento se registra el tiempo de ataque que tomaría vulnerar el algoritmo de curva elíptica según la longitud de la clave privada.

Tabla 4

Comparación de nivel de seguridad y longitud de clave entre algoritmos de encriptación.

Parámetro de Dominio	Longitud mínima clave privada[bits]	Valor de n	Tiempo de ataque[días]
brainpoolP160r1	80	1332297598440044874827085038830181364212942568457	$1.335x10^{19}$
brainpoolP192r1	96	4781668983906166242955001894269038308119863659119834868929	$8.00x10^{23}$
brainpoolP224r1	112	2272162293245435278755253799591092361256754634233075719139656096655 9	$5.517x10^{28}$
brainpoolP256r1	128	7688495639704534422080974662900164909273753178441452953875551906306353635907 9	$3.209x10^{33}$
brainpoolP320r1	160	1763593322239166354161909842446019520889512772717686063760686124016784784845843468355685258203921	$1.537x10^{43}$
brainpoolP384r1	192	2165927077011931617306923684233260497979611638701764860007564527482161150135851553796269511736890325222960171872394 1	$5.386x10^{52}$
brainpoolP512r1	256	8948962207650232551656602815159153422162609644098354511344597187200057010413418528378981730643524959857451398370029280583094215613882043973354392115544169	$1.094x10^{72}$

Nota. Esta tabla indica el tiempo de ataque que demoraría el método Rho de Pollard para encontrar la clave privada.

Se observa que el tiempo de ataque que demoraría un hacker usando el método Rho de Pollard para descifrar las contraseñas registradas en la base de datos es directamente proporcional al número de bits de clave de la clave privada utilizada en la encriptación de la información.

Conclusiones y Recomendaciones

- El problema del logaritmo discreto proporciona un reto computacionalmente difícil para un atacante, sin embargo, el algoritmo de la curva elíptica es conocido y para proporcionar un nivel de seguridad más alto se puede aumentar la longitud de la clave privada.
- La longitud de la clave privada usada en la encriptación de la información es proporcional al tiempo de ataque necesario para descifrar la clave privada del algoritmo de encriptación y vulnerar su seguridad, de manera que, un ataque por fuerza bruta para revelar la información almacenada dure años.
- Se recomienda no distribuir ni compartir la contraseña de usuario para el acceso a la aplicación con el fin de no exponer la clave privada ni el algoritmo de encriptación.
- El algoritmo de encriptación de curva elíptica permite una transferencia de segura de mensajes entre usuario por un canal inseguro mediante la generación de un secreto compartido entre las claves pública y privada, mas no está diseñado para realizar la encriptación de mensajes de un solo usuario a una fuente de almacenamiento de información.
- El algoritmo de encriptación de curva elíptica se basa en la transmisión de información entre dos usuarios, sin embargo, para el escenario que se plantea en este trabajo de grado donde se realiza la encriptación de la información en una base de datos donde el mismo usuario encripta y desencripta se descarta la existencia de otro usuario en la comunicación.

- Se evita la generación de claves públicas y privadas para la base de datos como otro usuario en base a la fórmula del secreto del compartido ya que genera más vulnerabilidad almacenar una clave privada de la comunicación en una base de datos.
- La representación de toda la información a ser encriptada en un solo punto de la curva mediante el pareo de los valores ASCII del mensaje es más eficiente con respecto al método de mapeo de caracteres del mensaje en puntos de la curva.
- Se recomienda asegurar que la generación de la clave privada sea aleatoria, ya que la mayor parte de los problemas de la seguridad del algoritmo que sean detectado son debido a una deficiencia al momento de generar la clave privada.
- Por motivo de seguridad se puede disminuir el tiempo de vida del token de autenticación de esta manera cada vez que este tiempo sea vencido, el usuario deberá volver a autenticarse.
- La implementación de la norma ISO27001 en el marco aplicativo a la encriptación de información, genera la implementación de políticas de seguridad que ayudan a reforzar la seguridad del sistema de encriptación enfocado en la gestión de claves.

Trabajos Futuros

- La creación de sistema que permita el cambio de clave de cada usuario, el proceso debe generar la desencriptación con la antigua clave privada del usuario y encriptar los datos de usuario con la nueva clave ingresada de manera automática.
- Desarrollar el sistema de seguridad de autenticación de acceso a la aplicación por medio del proceso de doble autenticación mediante un código al número de teléfono registrado o al correo electrónico del usuario.
- Implementar el algoritmo ECDSA (Algoritmo de Firma Digital de Curva Elíptica) para la verificación de firmas digitales de los usuarios, cuando se envía información a la base de datos.

Bibliografía

Alfred J. Menezes, P. C. (2018). *Handbook of Applied Cryptography.* Boca Raton: CRC Press.

Álvarez, S. C. (2017). *Implementción ISO 27001.* Catalunya.

Andress, J. (2014). *The Basics of Information Security.* Waltham: Syngress.

Aumasson, J. P. (2018). *Seriuos Cryptography.* San Francisco: No Starch Press.

Blancarte, O. (30 de Noviembre de 2018). *Software Arquitect*. Obtenido de Data Transfer Object (DTO) – Patrón de diseño: https://www.oscarblancarteblog.com/2018/11/30/data-transfer-object-dto-patron-diseno/

Burnett, M. M. (2004). *Hacking the Code.* Rockland: Syngress.

Camilo A. Barbosa, Y. A. (2011). Implementación del criptosistema de curva elíptica en un prototipo de aplicación móvil para E-Commerce. *Revista Tekhnê*, 12.

Candel, J. M. (2018). *Seguridad en aplicaciones Web Java.* Paracuellos de Jarama, Madrid: RA-MA Editorial .

Ceballos, F. J. (2010). *Java 2, Curso de programación* . España: Closas -Orcoyen, S.L.

Centro Criptológico Nacional. (Noviembre de 2012). Criptología de empleo en el esquema nacional de seguridad guía/norma de seguridad de las tic (ccn-stic-807). España.

Certicom Research. (2000). Recommended Elliptic Curve Domain Parameters. *Standards for Efficient Cryptography (SEC)*, 51.

Crespo, E. (11 de Noviembre de 2018). *Aprendiendo Arduino*. Obtenido de API REST: https://aprendiendoarduino.wordpress.com/category/api-rest/

Dante Ramírez. (Mayo de 2018). *Universidad Nacional Autónoma de México*. Obtenido de Revista de Seguridad: https://revista.seguridad.unam.mx/numero-10/el-cifrado-web-ssltls

Digital Guide IONOS. (03 de Septiembre de 2019). Obtenido de XAMPP: instalación y primeros pasos: https://www.ionos.es/digitalguide/servidores/herramientas/instala-tu-servidor-local-xampp-en-unos-pocos-pasos/

Fossati, M. (2014). *MYSQL.* Obtenido de Todo sobre MySQL: https://books.google.com.ec/books?id=GS3kAgAAQBAJ&printsec=frontcover&dq=mysql+base+de+datos&hl=es&sa=X&ved=0ahUKEwiH5sHf9vTkAhUkVd8KHeEjDqUQ6AEILzAB#v=onepage&q=mysql%20base%20de%20datos&f=true

Gustavo, R. J. (Agosto de 2012). *Universidad Nacional Autónoma de México*. Obtenido de Análisis de las ventajas de la criptografía de curva elíptica, frente a los sistemas criptográficos asimétricos actuales.: http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/2838/Tesis.pdf?sequence=1

Harinath, D. (2015). Enhancing Data Security Using Elliptic Curve . *Cryptography in Cloud Computing*. Hyderabad, Telangana, India: International Journal of Science and Research.

IBM Knowledge Center. (9 de Abril de 2018). *Cifrado*. Obtenido de https://www.ibm.com/support/knowledgecenter/es/SSFKSJ_9.0.0/com.ibm.mq.sec.doc/q009800_.htm

IMPERVA. (2017). *What is ISO/IEC 27001*. Obtenido de ISO/IEC 27001: https://www.imperva.com/learn/data-security/iso-27001/

ISOtools Excellence. (s.f.). *La norma 27001*. Obtenido de Aspectos clave de su diseño e implantación: https://www.isotools.org/pdfs-pro/iso-27001-sistema-gestion-seguridad-informacion.pdf

Jaiswal, S. (2015). *JavaTpoint*. Obtenido de Spring Boot Architecture: https://www.javatpoint.com/spring-boot-architecture

JSON ORG. (2012). *Introducción a JSON*. Obtenido de ECMA-404 The JSON Data Interchange Standard.: https://www.json.org/json-es.html

Kiennert, C. (2015). *Digital Identity Management.* London: ISTE Press.

Legion of Bouncy Castel Inc. (2013). *Legion of Bouncy Castel* . Obtenido de about the Legion of Bouncy Castle: htpps://www.bouncycastel.org/

Lloyd's Register. (2019). *Certificación ISO 27001 Sistemas de Gestión de Seguridad de la Información*. Obtenido de La norma ISO 27001: https://www.lr.org/es-es/iso-27001/

Lochter, M. (Marzo de 2010). *Elliptic Curve Cryptography (ECC) Brainpool Standard*. Obtenido de https://tools.ietf.org/html/rfc5639#section-2.1

Matthews, M. (2003). *MySQL and Java Developer's Guide.* Canada: Wiley Publishing Group. Obtenido de http://www.mysqltutorial.org/mysql-jdbc-tutorial/

Mintel. (s.f.).

Oracle. (Junio de 2017). *Oracle Database Security*. Obtenido de Oracle ha designado un directivo de seguridad de bases de datos : https://www.oracle.com/database/technologies/security/advanced-security.html

Oracle. (2018). *What Is a Database?* Obtenido de Database: https://www.oracle.com/database/what-is-database.html

Preukschat. (15 de Enero de 2014). *Oro y Finanzas, Diario Digital del dinero*. Obtenido de ¿Por qué se utiliza la Criptografía de Curva Elíptica en el Bitcoin?: https://www.oroyfinanzas.com/2014/01/criptografia-curva-eliptica-bitcoin-por-que-utiliza-ecdsa/

Reyad, O. (2018). Text Message Encoding Based on Elliptic Curve Cryptography and a Mapping Methodology. *Information Sciences Letters An International Journal*, 6.

Rules, H. (20 de Julio de 2016). *El protocolo HTTP*. Obtenido de https://www.youtube.com/watch?v=S975NVYbe2E

Security Site . (2018). Obtenido de Elliptic Curve Diffie Hellman (ECDH) with differing elliptic curves.: https://asecuritysite.com/encryption/ecdh3

Segovia. (16 de Octubre de 2017). *Advisera Expert Solutions Ltd.* Obtenido de ¿ Qué es la norma ISO 27001?: https://advisera.com/27001academy/es/que-es-iso-27001/

Sergio Navas, I. B. (2016). Acceso seguro a datos en Hadoop mediante criptografía de curva elíıptica. *Sesión I4: Privacidad / Control de Acceso*, 6.

Singh, L. D. (2015). Implementation of Text Encryption using Elliptic Curve Cryptography. *ScienceDirect*, 10.

Svetlin Nakov, P. (Noviembre de 2018). *ECDH Key Exchange*. Obtenido de https://cryptobook.nakov.com/asymmetric-key-ciphers/ecdh-key-exchange

Tomás, E. (25 de Abril de 2014). *https://desarrolloweb.com/*. Obtenido de Qué es REST - Manual de desarrollo de APIs: https://desarrolloweb.com/articulos/que-es-rest-caracteristicas-sistemas.html