UNIVERSIDAD TECNOLÓGICA DE LA HUASTECA HIDALGUENSE

Organismo Descentralizado de la Administración Pública Estatal

INGENIERÍA EN DESARROLLO Y GESTIÓN DE SOFTWARE

ACTIVIDAD:

PARCIA 2 ACTIVIDAD 1 TAREA 3

ASIGNATURA:

SEGURIDAD INFORMATICA

ELABORADO POR:

20221110 - DOMINGO HERNÁNDEZ GERARDO

20221058 - RODRÍGUEZ MARTÍNEZ EMMANUEL

CUATRIMESTRE: 7 **GRUPO: "B"**

DOCENTE:

MCE. ANA MARÍA FELIPE REDONDO

SEPTIEMBRE DE 2024 **HUEJUTLA, HIDALGO.**

Introducción

En esta actividad se utilizan dos frameworks diferentes, **React Native** y **Express.js**, para explorar cómo se integran los métodos de cifrado y hash en distintas plataformas, con el objetivo de comparar su rendimiento, facilidad de implementación y las características de seguridad que cada uno ofrece.

La seguridad de los datos es un aspecto crucial en el desarrollo de software, especialmente cuando se trabaja con información personal y sensible. Para esta actividad, se emplean tres técnicas criptográficas principales:

- **Cifrado Simétrico (Serpent)**: Se utiliza para asegurar datos sensibles localmente, protegiendo información como contraseñas y detalles personales.
- **Cifrado Asimétrico (NTRUEncrypt)**: Implementado para proteger la integridad y confidencialidad de los datos compartidos, asegurando que solamente los usuarios autorizados puedan acceder a la información.
- **Función Hash (GOST R 34.11-94)**: Utilizada para el almacenamiento seguro de datos críticos, como contraseñas, garantizando que estos no puedan ser revertidos a su forma original.

El uso de **React Native** y **Express.js** permite evaluar cómo se comportan estos algoritmos criptográficos en un entorno de desarrollo **móvil** y **web**, respectivamente. **React Native** se enfoca en la creación de aplicaciones móviles, facilitando la recolección de datos del usuario, mientras que **Express.js** se utiliza para aplicaciones web, proporcionando una plataforma robusta para la gestión y protección de la información.

Justificación de los Frameworks

Elección de React Native

React Native se eligió como framework para la parte móvil del proyecto debido a su capacidad de crear aplicaciones nativas de alto rendimiento utilizando JavaScript. Una de sus mayores ventajas es que permite compartir gran parte del código entre plataformas Android e iOS, reduciendo los tiempos de desarrollo y facilitando la entrega de una experiencia de usuario consistente. En el contexto de la seguridad de datos, React Native permite una integración directa de bibliotecas criptográficas mediante módulos nativos, lo cual es fundamental para garantizar que los datos sensibles, como las credenciales de los usuarios, se manejen adecuadamente dentro del dispositivo móvil.

Además, React Native permite una actualización rápida de la UI (interfaz de usuario) sin comprometer la lógica de seguridad en el backend. Esto hace que sea ideal para desarrollos donde se requiere flexibilidad y una alta capacidad de respuesta, a la vez que se garantiza la protección de los datos del usuario.

Elección de Express.js

Express.js fue seleccionado como framework para el desarrollo web por su simplicidad y eficiencia en la gestión de solicitudes HTTP, lo cual es fundamental cuando se desarrollan aplicaciones que requieren un backend robusto para la gestión de datos. Aunque en esta actividad no se conecta directamente con React Native, se eligió Express.js para demostrar cómo se pueden implementar técnicas de cifrado en una arquitectura web.

Express.js permite una fácil integración de bibliotecas criptográficas para la protección de datos a nivel del servidor. Su capacidad de manejar middleware de manera eficiente hace que sea ideal para la implementación de sistemas de seguridad, donde es necesario aplicar múltiples capas de protección antes de procesar las solicitudes. Esta flexibilidad es clave para implementar procesos como el cifrado simétrico, asimétrico, y el hash, que requieren ser ejecutados de manera segura y organizada antes de almacenar o transmitir datos.

Opinión sobre la Comparación de Frameworks

Ambos frameworks se utilizaron para proporcionar una visión clara de cómo los métodos de cifrado se integran en plataformas distintas —una para aplicaciones móviles y otra para aplicaciones web. La comparación también ayuda a entender cómo varían los desafíos de seguridad y las características de cada plataforma.

React Native se centra más en la seguridad del dispositivo, gestionando datos sensibles localmente antes de enviarlos o almacenarlos, mientras que Express.js se enfoca en la gestión de seguridad del lado del servidor. Cada uno tiene sus ventajas específicas; React Native tiene la capacidad de proteger los datos en el dispositivo móvil y garantizar que solo se envíe la información estrictamente necesaria, mientras que Express.js permite aplicar medidas de seguridad desde el inicio del proceso hasta la entrega final de la información al cliente.

Cifrado Simétrico vs. Asimétrico: Opinión sobre Eficiencia

Cifrado Simétrico (Serpent)

El cifrado simétrico, como Serpent, es conocido por ser más eficiente en términos de velocidad, ya que utiliza una única clave para el proceso de cifrado y descifrado. Esto lo hace particularmente adecuado cuando se necesita proteger una gran cantidad de datos de manera rápida y segura, sin incurrir en un alto costo computacional. En el caso de datos almacenados localmente o cuando se requieren operaciones rápidas, el cifrado simétrico es más eficiente.

Cifrado Asimétrico (NTRUEncrypt)

Por otro lado, el cifrado asimétrico, como NTRUEncrypt, emplea dos claves diferentes —una pública para cifrar y una privada para descifrar—, lo cual aporta un nivel adicional de seguridad, particularmente en escenarios donde se necesita transmitir datos de manera segura. Sin embargo, el cifrado asimétrico suele ser más lento y consume más recursos debido a la complejidad matemática implicada. Su principal ventaja radica en la seguridad del intercambio de claves; es ideal cuando se requiere establecer comunicaciones seguras entre distintas partes, pero no es tan eficiente como el cifrado simétrico para volúmenes de datos grandes.

Función Hash (GOST R 34.11-94)

La función hash, como GOST R 34.11-94, tiene un propósito diferente al de los cifrados simétricos y asimétricos. Su objetivo no es cifrar y descifrar datos, sino asegurar que los datos sensibles, como contraseñas, no puedan ser revertidos a su forma original. Los hashes son ideales para almacenar de forma segura contraseñas y otros datos críticos, ya que convierten los datos en una representación de longitud fija que es única y prácticamente irreversible.

Una ventaja significativa de los hashes es su eficiencia para la verificación de integridad y la protección de datos sensibles sin tener que preocuparse por la administración de claves, como en los métodos de cifrado. Sin embargo, a diferencia del cifrado, los hashes no permiten recuperar el valor original a partir del valor procesado, lo cual es fundamental en escenarios donde el objetivo es garantizar la privacidad y la no reversibilidad, como en el almacenamiento de contraseñas.

Conclusión sobre Eficiencia

En términos de eficiencia, el cifrado simétrico es generalmente superior cuando se necesita procesar datos de manera rápida y segura, gracias a su simplicidad y menor carga computacional.

El cifrado asimétrico proporciona una mayor seguridad para la transmisión de datos y el intercambio de claves, aunque su eficiencia es menor debido a la complejidad matemática involucrada.

La función hash es la opción más eficiente y segura para almacenar datos que no necesitan ser descifrados, como las contraseñas, ya que garantiza que estos datos no puedan ser revertidos, ofreciendo así una seguridad significativa en términos de protección frente a ataques.

Cada uno de estos métodos tiene su propósito específico en la protección de los datos, y su efectividad depende del contexto en el que se aplican. El cifrado simétrico es ideal para proteger datos a gran escala y con rapidez, el cifrado asimétrico es clave para la comunicación segura entre partes, y la función hash es fundamental para garantizar la integridad y seguridad de los datos sensibles de una manera irreversible**.**