Universidad del Valle de Guatemala
Facultad de Ingeniería
Departamento de Ciencias de la Computación

Christopher Garcia (20541)
Marco Jurado (20308)
Yong Bum Park (20117)
Gabriel Vicente (20498)

Proyecto 1 – Comunicaciones Seguras

Documento de diseño

CC 3078 CIFRADO DE INFORMACIÓN
Sección 11
Catedrático: CANO FUENTES, LUDWING OTTONIEL

Índice

Arquitectura del proyecto

Al acceder al repositorio, nos encontramos con tres rutas principales claramente definidas: api, db y frontend, además del archivo docker-compose.yml ubicado en la raíz del proyecto. Este archivo .yml simplifica enormemente la creación y configuración de los contenedores necesarios para las tres partes fundamentales del desarrollo. Cada ruta puede albergar estructuras más o menos complejas, adaptándose así a las necesidades específicas de cada componente. Por ejemplo, la ruta db cuenta con su propio Dockerfile y scripts SQL iniciales, mientras que las rutas API y Frontend contienen principalmente archivos .js (y en el caso de Frontend, también archivos html y ts). Al ejecutar estos contenedores, se establece una comunicación fluida entre ellos, lo que permite el funcionamiento integral y coordinado del proyecto en su totalidad.

Cifrado

En el desarrollo de la aplicación, se implementó un sistema de cifrado asimétrico utilizando la biblioteca node-forge. Este sistema se basa en la generación de un par de claves, una pública y una privada, utilizando el algoritmo RSA con una longitud de clave de 2048 bits y un exponente público de 0x10001. El código para generar este par de claves es let pair = Forge.pki.rsa.generateKeyPair(2048, 0x10001);. Una vez generadas las claves, se realizó un proceso de limpieza para eliminar los encabezados y los caracteres de nueva línea (\r y \n), dejando solo el texto de la clave como una cadena de caracteres.

La clave pública se almacena en la base de datos para su uso posterior, mientras que la clave privada se descarga y almacena localmente en el navegador del usuario. En particular, la clave privada se almacena en el almacenamiento local del navegador con la clave ‘privateKey’. Cuando se envía un mensaje, se utiliza la clave pública del destinatario, que se recupera de la base de datos, para cifrar el contenido del mensaje. El cifrado se realiza utilizando el algoritmo RSA-OAEP, que proporciona un alto nivel de seguridad. El código para cifrar el mensaje es let encryptedMessage = publicKey.encrypt(this.messageContent, 'RSA-OAEP');.

Para desencriptar el mensaje, se utiliza la clave privada del destinatario, que se recupera del almacenamiento local del navegador. Al igual que con el cifrado, la desencriptación se realiza utilizando el algoritmo RSA-OAEP. Esto asegura que solo el destinatario previsto, que posee la clave privada correspondiente, pueda leer el contenido del mensaje.