Integridad

Garantizar que el 100% de la información transmitida por el paciente y el doctor no ha sido modificada o adulterada durante su transmisión.

Para dar cumplimiento a este escenario de calidad se aplicó inicialmente el protocolo seguro de transferencia https, esta táctica favorece la integridad y confidencialidad de la información a nivel de transporte creando un canal cifrado para la transmisión de información sensible.

**Implementación inicial: Https**

Se realizó mediante la creación de certificados digitales auto-firmados con el generador de claves públicas y privadas que ofrece el Java Development Kit y configurando en el servicio la conexión SSL/TLS (Secure Sockets Layer/Transmission Layer Security) que aplicando un cifrado con clave asimétrica para establecer un canal encriptado entre el cliente y el servidor.

Su implementación significó un costo alto en desempeño al disminuir en un 10% la cantidad de peticiones como se muestra en la tabla.

|  |  |  |
| --- | --- | --- |
| Escalabilidad | Experimento 1 | Aplicando https |
|  |  |  |
|  |  |  |

Debido a este tradeoff se analiza y detecta que se está teniendo en cuenta además de la integridad, la confidencialidad al cifrar el canal agregando costos en desempeño.

Se determina el uso de una táctica que garantice la integridad que es el atributo de interés y no el de confidencialidad, para esto se implementa un checkSum CRC32 el cual usa un algoritmo polinomico como técnica robusta para la detección de errores.

**Implementación final: CRC32**

Se creó una función que convierte el mensaje de solicitud más una llave privada simétrica que actúa como semilla, en un número determinado mediante el algoritmo CRC32. Este CheckSum es enviado en el header de la petición. Al llegar al servidor, este obtiene el número del checksum correspondiente al mensaje enviado con la clave privada y lo compara con el checksum enviado por el header. De esta manera determina la integridad del mensaje, si fue modificado tanto el cuerpo como el encabezado de la solicitud, esto se ve reflejado directamente en el checksum dando resultados diferentes al realizar el procedimiento en el servicio.

Esta solución ofrece un menor trade off con el desempeño debido a que no se realiza cifrado del mensaje y el CRC32 es potente y más rápido a diferentes técnicas de hash.

|  |  |  |
| --- | --- | --- |
| Escalabilidad | Experimento 1 | Aplicando CRC32 |
|  |  |  |
|  |  |  |

POSTMAN, JMETER GRAFICAS, TABLAS.