INGENIERÍA DE SEGURIDAD US
E.T.S. Ingeniería Informática. Universidad de Sevilla
Avda. Reina Mercedes s/n. 41012 Sevilla
Tlf/Fax 954 557 139 **E-mail** lsi@lsi.us.es **Web** www.lsi.us.es

PAI 1. INTEGRIDOS - VERIFICADORES DE INTEGRIDAD EN EL ALMACENAMIENTO Y TRANSMISIÓN PARA ENTIDAD FINANCIERA

Introducción

En este **Proyecto de Aseguramiento de la Información** usaremos técnicas para poder verificar **la integridad en el almacenamiento y en la transmisión de datos por redes públicas como Internet y evitar los diferentes tipos de posibles ataques.**

Asumamos el escenario donde diferentes usuarios se conectan al servidor de una entidad financiera. Por tanto, tendremos un mecanismo de credenciales (nombre de usuario y password) para poder comprobar que los usuarios pueden conectarse a dicha entidad. En este caso por simplificación vamos a suponer una arquitectura cliente-servidor con el esquema de la Figura 1.

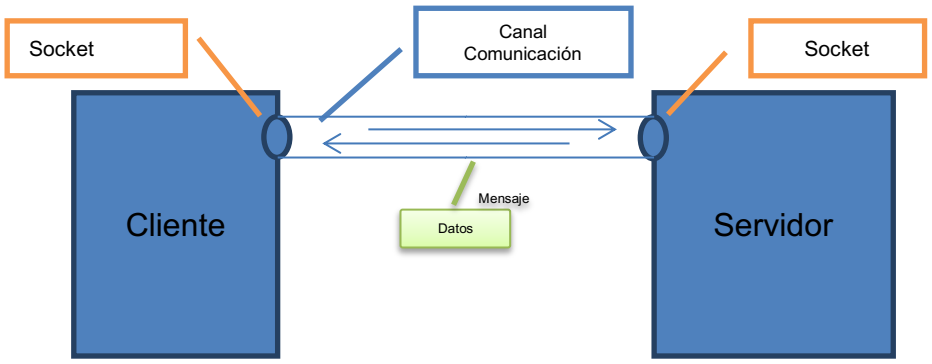


Figura 1: Arquitectura de la entidad financiera

Tras comprobar sus credenciales, el usuario sólo podrá enviar transacciones (una o varias) al servidor con el siguiente formato:

"***Cuenta origen, Cuenta destino, Cantidad transferida***"

Dichas transacciones no tendrán que pasar ningún tipo de validación previa y serán registradas en servidor sin más comprobación

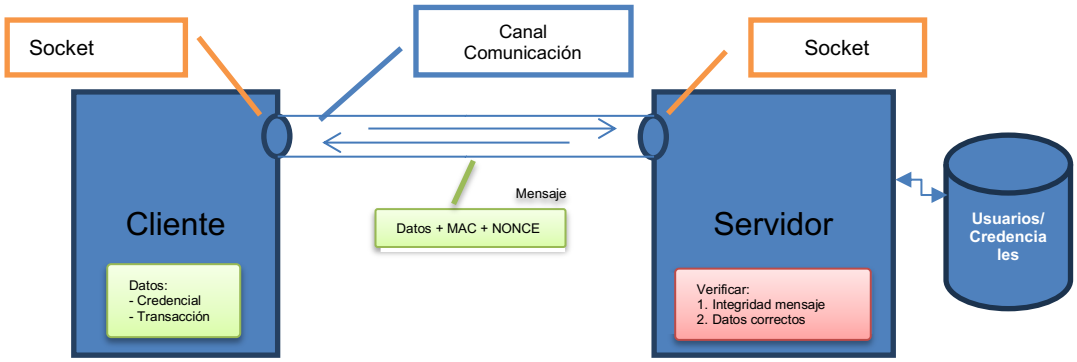
Política y Controles de Seguridad

En este **Proyecto de Aseguramiento de la Información** (PAI) se pretende comenzar a familiarizarse con del trabajo en el **gobierno/gestión/tecnologías de la seguridad de la información** y en este caso de la verificación de la integridad de datos en un sistema de

información. Por ello en este PAI se tiene **que dentro de una organización** se han definido las siguientes **Políticas de Seguridad**, que indica:

*"En todas las transferencias **por medios electrónicos no seguros se debe conservar la integridad** de las comunicaciones"*

"Debe preservar la integridad en el almacenamiento de credenciales de usuario y de las transacciones realizadas"



Objetivos del proyecto

A continuación, se propone a los equipos de trabajo los siguientes objetivos:

1. Desarrollar un sistema cliente-servidor usando sockets, que permita enviar datos de usuario, password, y mensajes de transferencias.
2. Desarrollar en servidor un mecanismo que permita almacenar y comprobar las credenciales de usuario preservando la integridad de la información.
3. Desarrollar el verificador de integridad para los mensajes de transferencia bancaria que se transmiten a través de las redes públicas evitando los ataques de *man-in-the-middle* y de *replay (tanto en el servidor como en el cliente)*, usando mecanismos de MAC y NONCE.
4. Desplegar un verificador de integridad en los sistemas cliente/servidor para llevar a cabo la realización de la verificación de forma práctica de los mensajes transmitidos entre un servidor y un cliente.

Recomendaciones de implementación

Para la comprobación de las credenciales se proponer seguir las siguientes recomendaciones:

1. Recopilación de Credenciales: El usuario introduce su nombre de usuario y contraseña en la aplicación.
2. Envió al Servidor: Las credenciales (usuario y contraseña) se envían al servidor. En la parte servidora, antes de almacenar una contraseña en la base de datos, esta no se guarda en texto plano, sino que se utiliza un algoritmo de hash. Cuando se recibe la contraseña del usuario, también se le aplica el mismo proceso de hash.
3. Comprobación en la Base de Datos: Una vez que el servidor recibe el nombre de usuario y la contraseña (transformada en hash), se compara el hash de la contraseña

recibida con el hash almacenado en la base de datos. Si los hashes coinciden, significa que la contraseña ingresada es correcta. Si las credenciales son correctas: se le pedirá al usuario que indique la transacción a realizar. Si las credenciales son incorrectas: Se envía una respuesta indicando que el nombre de usuario o la contraseña son incorrectos.

Para la comprobación de la integridad proponemos usar el siguiente esquema de la Figura 2.

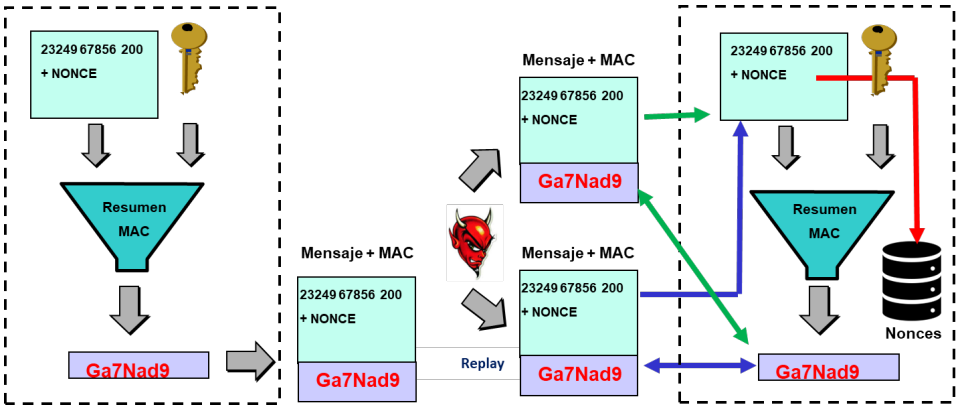


Figura 2: Esquema ejemplo de verificación de integridad de mensajes.

Consideraciones

- No es necesario desarrollar un sistema de alta de usuarios en el sistema.
- Será necesario implementar un sistema de almacenamiento de usuarios y credenciales, donde por defecto, el sistema ya deberá tener varios usuarios dados de alta.
- El sistema de almacenamiento puede ser cualquier tipo de base de datos que permita almacenar la información de nombres de usuarios y contraseñas.
- Se puede utilizar cualquier lenguaje de programación.
- Se debe utilizar sockets para la comunicación, y no será necesario implementar protocolo seguro.

Normas del entregable

- Cada Security Team debe entregar a través de la Plataforma de Enseñanza Virtual y en la **actividad** preparada para ello un archivo zip, nombrado ***PAI1-STXTrabajoY*.zip**, que deberá contener al menos los ficheros siguientes:
  - ✓ **Documento en formato PDF que contenga un informe/resumen del proyecto** con los detalles más importantes de las decisiones, soluciones adoptadas y/o implementaciones desarrolladas, así como el resultado y análisis de las pruebas realizadas (máximo 10 páginas).
  - ✓ **Código fuente de las posibles implementaciones o scripts desarrollados o configuraciones establecidas en herramientas ya disponibles.**
- El plazo de entrega de dicho proyecto finaliza el **día 7 de octubre a las 23:59 horas.**
- **Los proyectos entregados fuera del plazo establecidos serán considerados inadecuados por el cliente y por tanto entrarán en penalización por cada día de retraso entrega de 10% del total, hasta agotarse los puntos.**

- **El cliente no se aceptará envíos realizados por email, ni mensajes internos de la enseñanza virtual, ni correo interno de la enseñanza virtual. Toda entrega realizada por estos medios conllevará una penalización en la entrega del 10%.**

Métricas de valoración

Para facilitar el desarrollo de los equipos de trabajo el cliente ha decidido listar las métricas que se tendrán en cuenta para valorar los entregables de cada grupo de trabajo:

- **Documento (30%)**
  - Tamaño del informe
  - Calidad del informe aportado y justificaciones.
  - Calidad de pruebas presentadas y resultados
- **Código/Configuración aportada (70%)**
  - Cumplimiento de requisitos establecidos
  - Calidad del código entregado
  - Complejidad de la solución para almacenamiento y transmisión
  - Logs de pruebas realizas y entregadas