Planteamiento:

Haremos un total de dos scripts. El primero será para coger mensajes, encriptarlos, firmarlos y guardarlos en una carpeta con su fecha pertinente, listos para ser enviados al receptor. El segundo script será para coger los mensajes encriptados, comprobar que el mensaje proviene del emisor adecuado, y descifrarlo con la clave pública, guardando así el resultado en un archivo de texto con su fecha pertinente.

El script número 1 coge un fichero de texto (.txt) con un mensaje en su interior de la ruta que el usuario le indicará porque se la pedirá el script. Lo encripta con la clave pública del receptor, lo firma con la clave pública del emisor, y lo guarda en una carpeta que también le indicará el usuario, listo para ser enviado.

El script número 2 coge ese mensaje encriptado de la ruta que le introduce el usuario, comprueba que la clave con la que viene firmada, corresponde con la clave pública del emisor (que ya conoce), lo desencripta con la clave privada del receptor y guarda el descifrado en un fichero de texto plano (.txt) en la ruta que el usuario también le indica.

Orden correcto de la creación de claves:

Clave 1 emisor privada:
openssl genpkey -algorithm RSA -aes-256-cbc -pkeyopt rsa_keygen_bits:4096 -out /home/alumne/Escriptori/script/emisor_privada.pem

Passphrase: dhayan

Clave 2 emisor pública:
openssl pkey -in /home/alumne/Escriptori/script/emisor_privada.pem -pubout -out /home/alumne/Escriptori/script/emisor_publica.pem

Clave 3 receptor privada:
openssl genpkey -algorithm RSA -aes-256-cbc -pkeyopt rsa_keygen_bits:4096 -out /home/alumne/Escriptori/script/receptor_privada.pem

Passphrase: andres

Clave 4 receptor pública:
openssl pkey -in /home/alumne/Escriptori/script/receptor_privada.pem -pubout -out /home/alumne/Escriptori/script/receptor_publica.pem

Rutas y archivos:

/home/alumne/Escriptori/mensajes/manolito.txt

/home/alumne/Escriptori/mensajes_enviar

/home/alumne/Escriptori/mensajes_enviar/2025-11-12_22-34-59.txt.enc

/home/alumne/Escriptori/mensajes_desencriptados

/home/alumne/Escriptori/mensajes_enviar/2025-11-12_22-34-59.txt.sig

Script 1:

```
#!/bin/bash

# 1. Pedimos la ruta del archivo de entrada
echo "Introduce la ruta completa del archivo que deseas encriptar (por ejemplo:
/home/isard/archivo.txt):"
read ruta_archivo

# 2. Verificamos que el archivo exista
if [ ! -f "$ruta_archivo" ]; then
    echo "El archivo no existe. Por favor, asegúrate de que la ruta es correcta."
    exit 1
fi

# 3. Pedimos la ruta donde se guardarán los archivos encriptados y firmados
echo "Introduce la ruta completa donde quieras guardar los archivos encriptados y firmados
(por ejemplo: /home/isard/mensajes_enviar/):"
read ruta_destino

# 4. Verificamos si la ruta de destino existe, sino, la creamos
if [ ! -d "$ruta_destino" ]; then
    echo "La carpeta de destino no existe, creando la carpeta..."
    mkdir -p "$ruta_destino"
fi

# 5. Obtenemos la fecha y hora actual para nombrar los archivos de salida
fecha=$(date +"%Y-%m-%d_%H-%M-%S")

# 6. Realizamos el cifrado con la clave pública del receptor
openssl pkeyutl -encrypt -inkey receptor_publica.pem -pubin -in "$ruta_archivo" -out
"$ruta_destino/$fecha.txt.enc"

# 7. Firmamos el archivo con la clave privada del emisor
openssl dgst -sha256 -sign emisor_privada.pem -out "$ruta_destino/$fecha.txt.sig"
"$ruta_destino/$fecha.txt.enc"

# 8. El archivo cifrado y firmado ya está guardado en la carpeta de destino
```

```
echo "El archivo ha sido cifrado y firmado correctamente. Puedes encontrarlo en:
$ruta_destino"
```

Script 2

```
#!/bin/bash

# 1. Pedimos la ruta del archivo cifrado
echo "Introduce la ruta completa del archivo cifrado que quieres desencriptar (por ejemplo:
/home/isard/mensajes_enviar/2025-11-12_12-30-45.txt.enc):"
read ruta_archivo_cifrado

# 2. Verificamos si el archivo cifrado existe
if [ ! -f "$ruta_archivo_cifrado" ]; then
    echo "El archivo cifrado no existe. Por favor, asegúrate de que la ruta es correcta."
    exit 1
fi

# 3. Pedimos la ruta donde se guardará el archivo descifrado
echo "Introduce la ruta completa donde quieras guardar el archivo descifrado (por ejemplo:
/home/isard/mensajes_recibidos/):"
read ruta_destino

# 4. Verificamos si la ruta de destino existe, si no, la creamos
if [ ! -d "$ruta_destino" ]; then
    echo "La carpeta de destino no existe, creando la carpeta..."
    mkdir -p "$ruta_destino"
fi

# 5. Pedimos la ruta de la firma (que debe ser el archivo .sig)
echo "Introduce la ruta completa del archivo de la firma (por ejemplo:
/home/isard/mensajes_enviar/2025-11-12_12-30-45.txt.sig):"
read ruta_firma

# 6. Verificamos si el archivo de firma existe
if [ ! -f "$ruta_firma" ]; then
    echo "El archivo de firma no existe. Por favor, asegúrate de que la ruta es correcta."
    exit 1
fi

# 7. Comprobamos la firma con la clave pública del emisor
echo "Comprobando la firma con la clave pública del emisor..."
openssl dgst -sha256 -verify emisor_publica.pem -signature "$ruta_firma"
"$ruta_archivo_cifrado"
if [ $? -ne 0 ]; then
    echo "La firma no es válida. El mensaje puede haber sido alterado."
```

```
    exit 1
fi

# 8. Desencriptamos el archivo con la clave privada del receptor
echo "Desencriptando el archivo con la clave privada del receptor..."
openssl pkeyutl -decrypt -inkey receptor_privada.pem -in "$ruta_archivo_cifrado" -out
"$ruta_destino/$(basename "$ruta_archivo_cifrado" .enc).descifrado.txt"

# 9. Verificamos si la operación de desencriptado ha sido exitosa
if [ $? -eq 0 ]; then
    echo "El archivo ha sido desencriptado correctamente y guardado en: $ruta_destino"
else
    echo "Ha habido un error al desencriptar el archivo."
    exit 1
fi
```