UNIVERSIDAD DEL VALLE DE GUATEMALA

Seguridad en Sistemas de Computación

Sección 20

Ing. Oscar Canek

Ejercicio 2

José Pablo Orellana - 21970

Diego Alberto Leiva - 21752

José Auyón - 201579

Renatto Guzmán - 21646

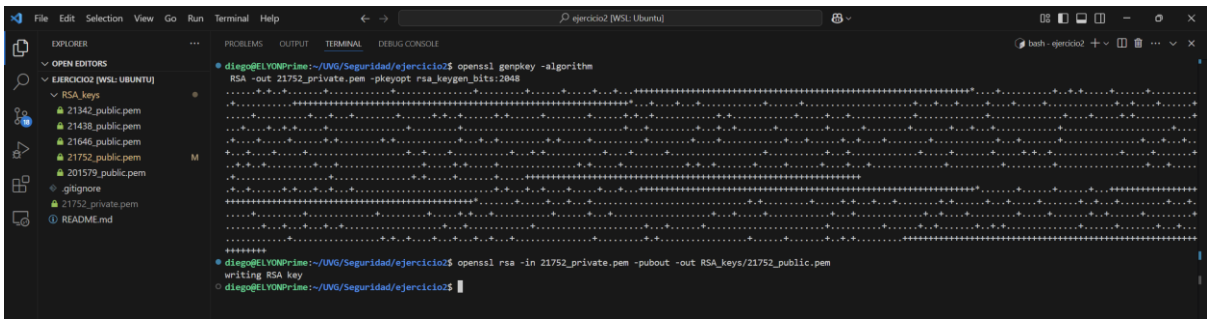
María Marta Ramírez - 21342

Gustavo González - 21438

Guatemala, 13 de febrero de 2025

1. Fase 1

Generacion de llaves

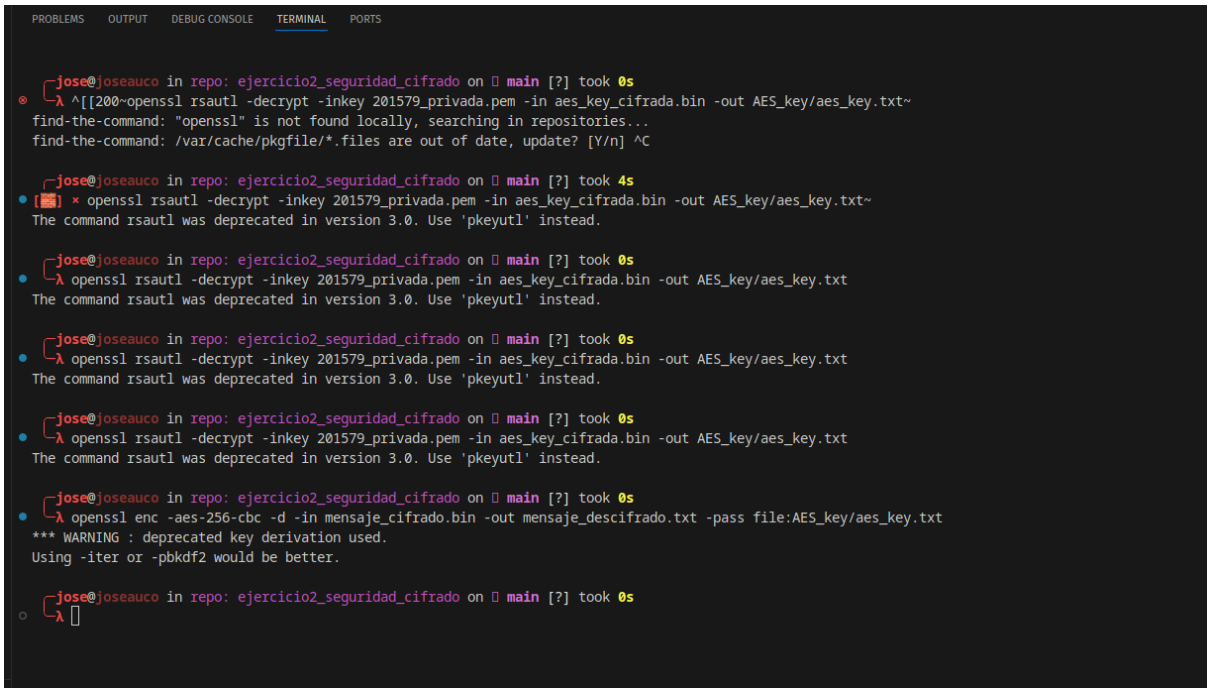


```
diego@ELYONPrime:~/UVG/Seguridad/ejercicio2$ openssl genpkey -algorithm
 RSA -out 21752_private.pem -pkeyopt rsa_keygen_bits:2048
```

```
+++++++++
diego@ELYONPrime:~/UVG/Seguridad/ejercicio2$ openssl rsa -in 21752_private.pem -pubout -out RSA_keys/21752_public.pem
writing RSA key
diego@ELYONPrime:~/UVG/Seguridad/ejercicio2$ ▌
```

2. Fase 2

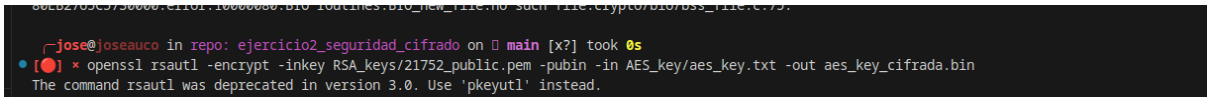
Cifrado de Mensaje con AES



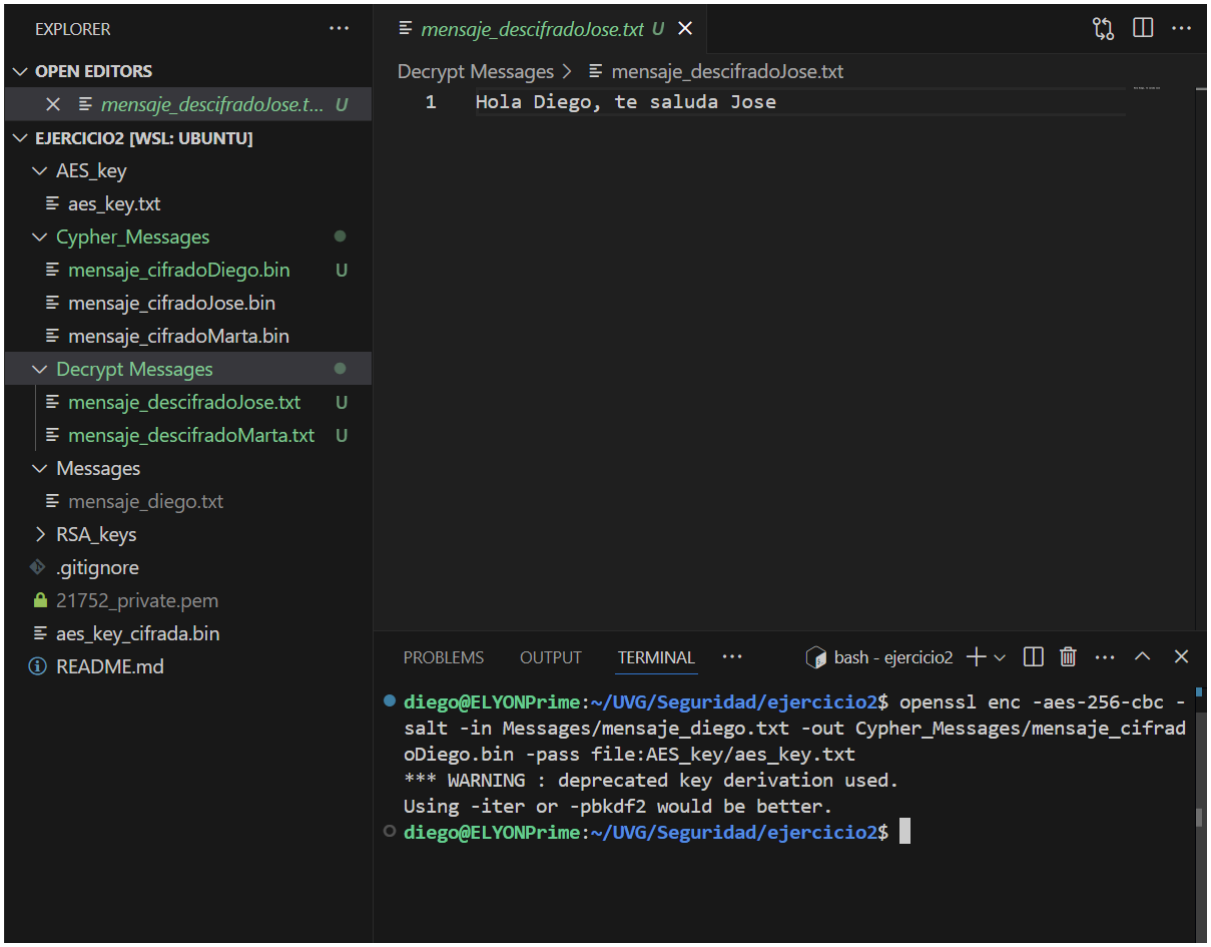
```
┌─jose@joseauco in repo: ejercicio2_seguridad_cifrado on  main [?] took 0s
└─λ ^[[200~openssl rsautl -decrypt -inkey 201579_privada.pem -in aes_key_cifrada.bin -out AES_key/aes_key.txt~
find-the-command: "openssl" is not found locally, searching in repositories...
find-the-command: /var/cache/pkgfile/*.files are out of date, update? [Y/n] ^C

┌─jose@joseauco in repo: ejercicio2_seguridad_cifrado on  main [?] took 4s
[🧱] × openssl rsautl -decrypt -inkey 201579_privada.pem -in aes_key_cifrada.bin -out AES_key/aes_key.txt~
The command rsautl was deprecated in version 3.0. Use 'pkeyutl' instead.

┌─jose@joseauco in repo: ejercicio2_seguridad_cifrado on  main [?] took 0s
└─λ openssl rsautl -decrypt -inkey 201579_privada.pem -in aes_key_cifrada.bin -out AES_key/aes_key.txt
The command rsautl was deprecated in version 3.0. Use 'pkeyutl' instead.

┌─jose@joseauco in repo: ejercicio2_seguridad_cifrado on  main [?] took 0s
└─λ openssl rsautl -decrypt -inkey 201579_privada.pem -in aes_key_cifrada.bin -out AES_key/aes_key.txt
The command rsautl was deprecated in version 3.0. Use 'pkeyutl' instead.

┌─jose@joseauco in repo: ejercicio2_seguridad_cifrado on  main [?] took 0s
└─λ openssl rsautl -decrypt -inkey 201579_privada.pem -in aes_key_cifrada.bin -out AES_key/aes_key.txt
The command rsautl was deprecated in version 3.0. Use 'pkeyutl' instead.

┌─jose@joseauco in repo: ejercicio2_seguridad_cifrado on  main [?] took 0s
└─λ openssl enc -aes-256-cbc -d -in mensaje_cifrado.bin -out mensaje_descifrado.txt -pass file:AES_key/aes_key.txt
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.

┌─jose@joseauco in repo: ejercicio2_seguridad_cifrado on  main [?] took 0s
└─λ ▯
```

3. Fase 3

Descifrado de mensajes

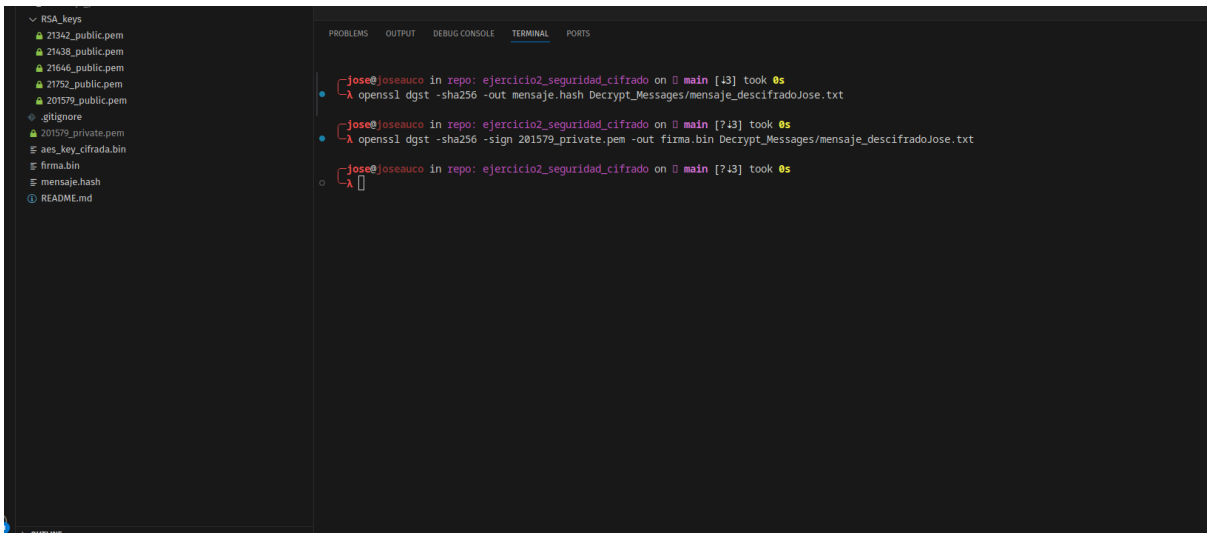


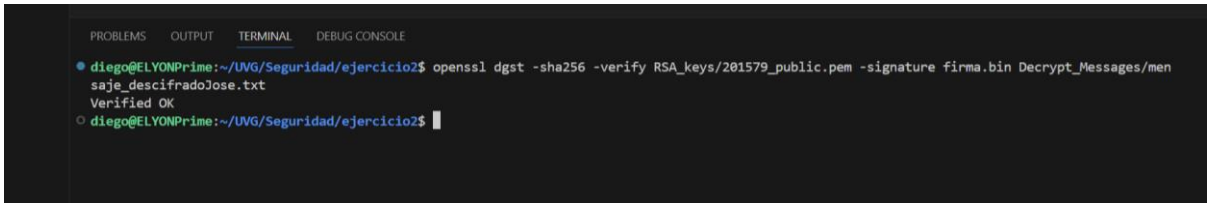
Mensaje cifrado para Diego



4. Fase 4

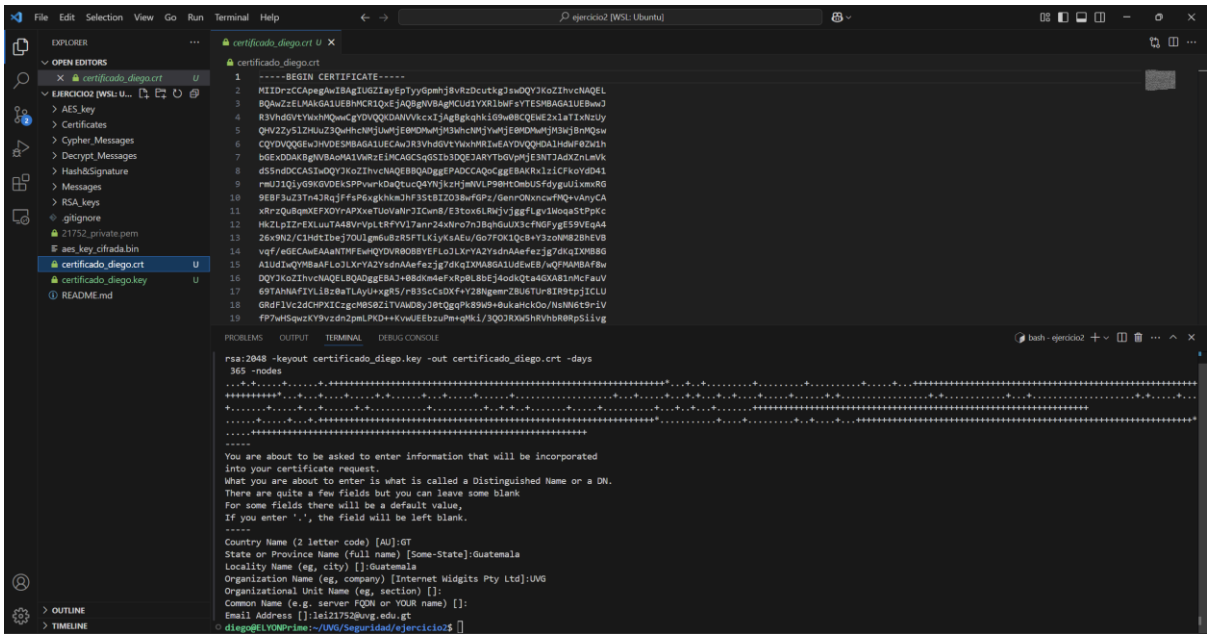
Generacion de Hashes y firmas digitales





5. Fase 5

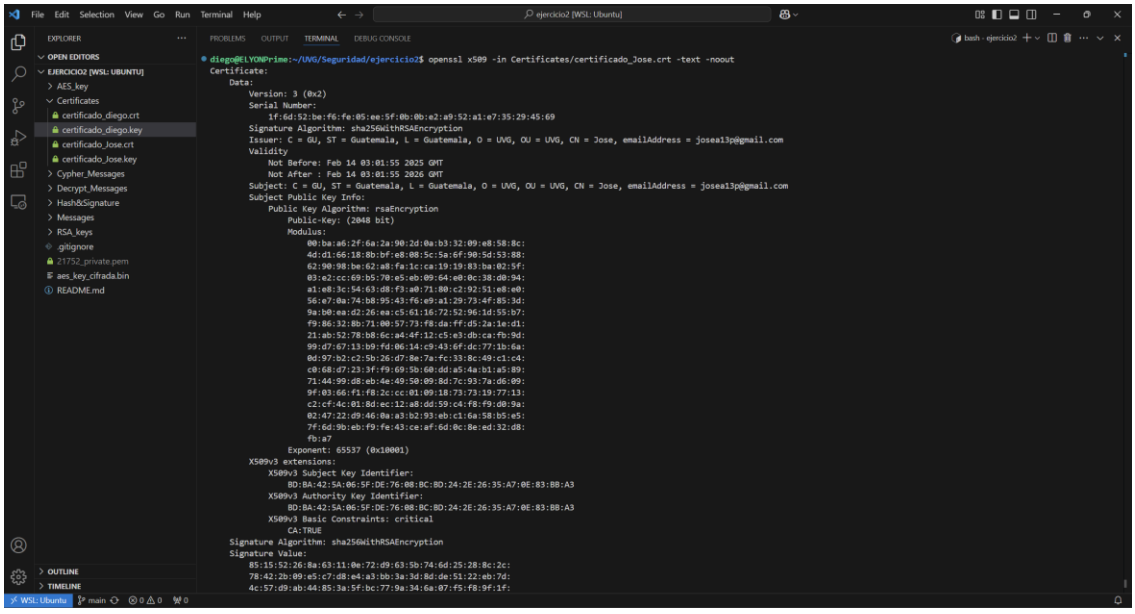
Generación de certificado de Diego



Certificado Diego:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            19:92:1a:c8:4a:53:cb:21:a9:9a:18:fc:bd:1c:c3:72:eb:64:80:9b
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=GT, ST=Guatemala, L=Guatemala, O=UVG, emailAddress=lei21752@uvg.edu.gt
        Validity
            Not Before: Feb 14 03:02:37 2025 GMT
            Not After : Feb 14 03:02:37 2026 GMT
        Subject: C=GT, ST=Guatemala, L=Guatemala, O=UVG, emailAddress=lei21752@uvg.edu.gt
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:a4:71:97:38:82:16:4a:18:74:3e:35:ae:65:09:
                    d5:08:b2:1b:d2:86:54:31:24:48:f3:ef:c2:b9:03:
                    69:0b:6e:71:0e:18:36:39:33:1e:39:8d:54:b3:fd:
                    d0:7b:4e:99:b5:12:7d:dc:a0:b9:48:b1:9b:14:46:
                    f4:40:45:de:e6:77:4e:7e:09:46:a8:c5:7e:c3:fa:
                    c6:09:21:92:62:61:17:74:ad:04:86:4e:df:cc:1f:
                    18:fc:ff:19:e9:eb:38:dc:67:73:07:cc:43:eb:c0:
                    9f:20:80:c5:1a:f3:42:e0:6a:99:71:05:5c:e6:2b:
                    00:f5:f1:79:35:28:55:a3:6b:24:80:b0:9f:cf:c4:
                    de:da:31:e8:b4:56:8e:f8:e0:81:f2:e0:bf:55:a8:
                    a9:a4:ad:3e:92:9c:1e:46:4b:a4:86:6b:11:72:ee:
                    b9:30:38:f1:5a:d5:a4:bb:51:7d:85:65:ed:a9:eb:
                    db:8c:4d:ae:8e:e7:24:1a:a1:1a:e5:17:dd:c7:cd:
                    18:5c:a0:13:9f:55:12:a0:38:db:ac:7d:37:6f:c2:
                    d4:77:6d:21:b7:a3:ec:e5:25:82:6e:ae:07:34:79:
                    15:32:ca:8b:22:ac:00:4b:bf:1a:8e:c5:38:ad:50:
                    70:1f:98:df:3a:0d:33:cd:81:84:45:41:be:a7:ff:
                    78:61
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                BA:09:2D:7A:D8:03:66:2C:76:70:00:79:F7:B3:8E:0E:DD:2A:A2:17
            X509v3 Authority Key Identifier:
                BA:09:2D:7A:D8:03:66:2C:76:70:00:79:F7:B3:8E:0E:DD:2A:A2:17
            X509v3 Basic Constraints: critical
                CA:TRUE
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        9f:b4:f1:d2:a6:e1:e1:71:46:9d:0b:f1:b1:23:e2:87:64:42:
        d6:b8:19:70:3c:d6:73:1c:15:ab:95:eb:d4:c0:84:d0:1f:21:
```

Certificado de Jose



6. Errores Encontrados y Cómo se Resolvieron

Durante la implementación del ejercicio con OpenSSL, se encontraron algunos problemas comunes que se resolvieron de la siguiente manera:

- Error: "unable to load Private Key" al generar la clave pública
- Causa: La clave privada RSA no se generó correctamente o no se encuentra en el formato esperado.
- Solución: Verificar que la clave privada fue generada correctamente con:

  openssl pkey -in mi_clave_privada.pem -noout –text

  Si hay problemas, regenerar la clave con: openssl genpkey -algorithm RSA -out mi_clave_privada.pem -pkeyopt rsa_keygen_bits:2048

- Error: "bad decrypt" al descifrar el mensaje con AES
- Causa: La clave de descifrado no coincide con la utilizada para cifrar el mensaje o hay un error en el formato de la clave.
- Solución: Asegurarse de que la clave AES fue correctamente descifrada antes de intentar descifrar el mensaje. Se puede verificar el contenido con:

  cat aes_key_descifrada.txt

  Además, verificar que el mensaje cifrado y la clave AES provienen de la misma sesión de cifrado.

- Error: "RSA operation error" al intentar descifrar la clave AES
- Causa: Se utilizó una clave pública o privada incorrecta, o la clave no estaba en el formato esperado.
- Solución: Confirmar que la clave pública del destinatario fue usada correctamente para cifrar la clave AES y que la clave privada correcta se está usando para descifrarla. Se puede probar con:

  openssl rsa -in mi_clave_privada.pem -check

- Error: "verification failure" al verificar la firma digital
- Causa: La firma pudo haber sido alterada, el archivo firmado no es el correcto o se utilizó una clave pública incorrecta para la verificación.
- Solución: Asegurar que el archivo verificado sea el mismo que se firmó. También verificar la clave pública con:

  openssl rsa -in clave_publica_remitente.pem -pubin -text -noout

- Problema: No se generó el certificado autofirmado correctamente
- Causa: Se omitió información requerida en el proceso de generación del certificado.
- Solución: Incluir -nodes para evitar la solicitud de una contraseña y proporcionar los datos necesarios cuando se pida. También se puede verificar el contenido del certificado con:

  openssl x509 -in mi_certificado.crt -text -noout

7. Lecciones Aprendidas sobre Criptografía y Seguridad

- La criptografía simétrica (AES) es eficiente para cifrar datos, pero necesita una forma segura de compartir la clave.
- La criptografía asimétrica (RSA) permite el intercambio seguro de claves, aunque es más lenta para cifrar grandes volúmenes de datos.
- Usar ambas técnicas en conjunto proporciona un equilibrio entre seguridad y rendimiento.
- No basta con cifrar los mensajes, también es necesario asegurarse de que el contenido no haya sido modificado y que provenga del remitente esperado.
- El uso de firmas digitales permite garantizar la autenticidad de los mensajes.
- La seguridad de la clave privada es fundamental, ya que comprometerla permite a un atacante descifrar mensajes y falsificar firmas.
- Los certificados autofirmados pueden proporcionar confianza en un grupo cerrado, pero en sistemas a gran escala se requiere una Autoridad de Certificación (CA) para validar la identidad de los participantes.
- La correcta ejecución de cada paso es clave. Errores en una fase pueden comprometer las siguientes.
- La verificación de claves, hashes, firmas y certificados antes de compartir archivos es esencial para evitar errores y malentendidos.