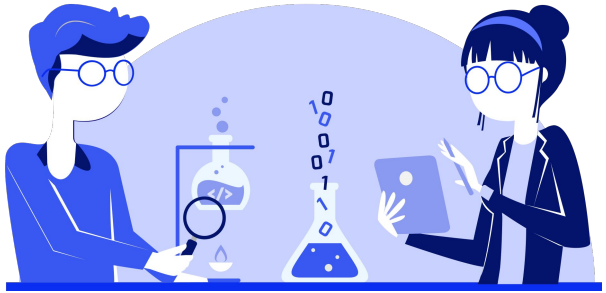
educación IT
CODE YOUR FUTURE

Criptografía y Blockchain

Módulo 2 - Laboratorio adicional

Para poder realizar este laboratorio, se recomienda:

- Revisar contenidos previos.
- Descargar los elementos necesarios.



Ejercicio 1: Criptoanálisis RSA

En el laboratorio de OpenSSL se vio cómo crear un par de claves RSA. Durante el tutorial, usamos *padding* OAEP. Sin embargo, algunas implementaciones por *default* continúan usando *padding* mediante PKCS#1 v1.5, que es vulnerable al ataque de Bleichenbacher.

El ataque de Bleichenbacher (1998) es un ataque de texto cifrado elegido contra un oráculo para romper el *padding.* Recuerda que un oráculo es una "*caja negra*" que responde ante un test que se le presente por verdadero o falso.

Este ataque ha sido muy real en el pasado, y aún continúa vigente contra servidores TLS mal configurados. Consiste en alimentar el oráculo con texto cifrado elegido y analizar su respuesta respecto a la validez del *padding.* Después de enviar miles de peticiones aleatorias, eventualmente el oráculo responderá `true`.

Este ataque, también conocido como el ataque del millón de mensajes, puede obtener una clave de sesión en un día, o tal vez menos. Con ligeras variaciones, subsiste bajo el nombre de "*Return Of Bleichenbacher's Oracle Threat*" (ROBOT, 2018).

1. Para este laboratorio, usa una máquina virtual de Kali Linux.

   En el escritorio, buscar la carpeta ***Cripto***, hacer clic derecho sobre ella y seleccionar ***Open Terminal Here***.

2. Ingresar en la carpeta ***Scripts***.

```
┌──(kali㉿kali)-[~/Desktop/cripto]
└─$ cd Scripts
```

3. Listar su contenido. Se mostrarán diversos *scripts* escritos en Python.

```
└─$ ls
blockchain.py  requirements.txt  rsa_oracle_attack.py  shor.py
```

4. Activar el entorno virtual para poder ejecutar los *scripts*.

```
┌──(kali㉿kali)-[~/Desktop/cripto/Scripts]
└─$ source ../bin/activate

┌──(cripto)─(kali㉿kali)-[~/Desktop/cripto/Scripts]
└─$
```

5. Ejecutar el *script*. Nos pedirá el tamaño de la clave a vulnerar. Seleccionar 512 para evaluar la velocidad de descifrado de la máquina.

```
┌──(cripto)─(kali㉿kali)-[~/Desktop/cripto/Scripts]
└─$ python rsa_oracle_attack.py
Ingrese la longitud de la clave (512, 1024, 2048): 512
```

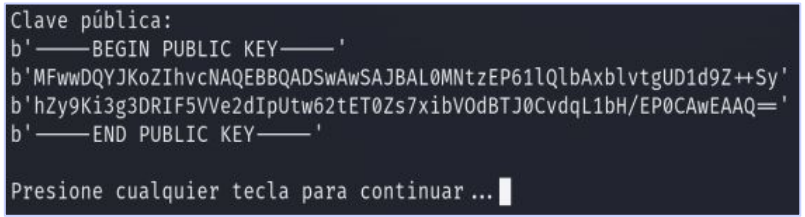
Se mostrará un mensaje explicativo advertencia.

```
Ejecutando 1 tests con clave de tamaño 512

        Este ataque consiste en enviar texto cifrado elegido a un
        oráculo. Este oráculo analizará la validez del padding y
        respondera casi siempre False y eventualmente, tras varios
        miles de peticiones, True.
        Se generará una clave privada que se cargará al falso
        oráculo para que éste pueda responder a las peticiones del
        atacante.
        A partir de esta clave privada, se generará su contraparte
        pública. Esta clave pública será utilzada por el atacante
        para intentar romper el cifrado
```

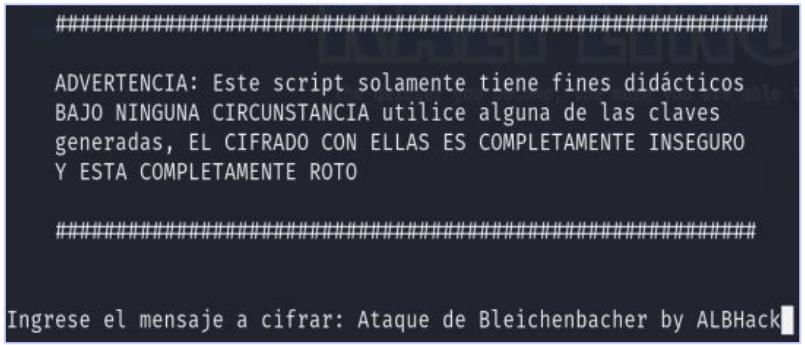
6. Posteriormente, se mostrará la clave vulnerable que genera. La misma será cargada en el falso oráculo a atacar.

```
Clave privada
b'-----BEGIN ENCRYPTED PRIVATE KEY-----'
b'MIIBvTBXBgkqhkiG9w0BBQ0wSjApBgkqhkiG9w0BBQwwHAQIBHuBIlyDGcsCAggA'
b'MAwGCCqGSIb3DQIJBQAwHQYJYIZIAWUDBAEqBBA5swzb7FGdXvRxBgFv3Y8NBIIB'
b'YK0Cr4nIhlacuH1aTuqfvHYydhpM9IKZgmTG6Asbra5ip1QIOY22GdH27C+efjVR'
b'HpHTL1F01Cfckl18ksrSWWsIOOotxOnf6pT7IAmAUMpvi6xt5k9Jl0RXkETL3fW6'
b'Zv4K/91Ou2ULunfD6rUs020bSnivKBra30u28ywhe8l7S57cJlcp1AFxt7CZygIg'
b'3zts/ZHavUXeDIuVc+c6k1LIRSuUB/1kA9lIPaAfrECDdi89UEvNhuYuRXKP4ej2'
b'HpEcI24b8m+vRhXvED3Qz68U5eZv+5Qd8wddou1YK7CG4WJipQA21QTx9KJqByqf'
b'cpv28HncIXp9S5uabJkh6i1VZ7ypHLOkRPd/fD/NoSbDoIc8ebcIYrpwYmYqUnTb'
b'Shfmo4uaAxyqPujlU7AJHWPAQOUym3bp61zHE9OhMgc3/Jk4blAL8YaMcWd5TTKY'
b'qWxz9ztE1Avj1PdX1cdMAjk='
b'-----END ENCRYPTED PRIVATE KEY-----'

Presione cualquier tecla para continuar ...
```

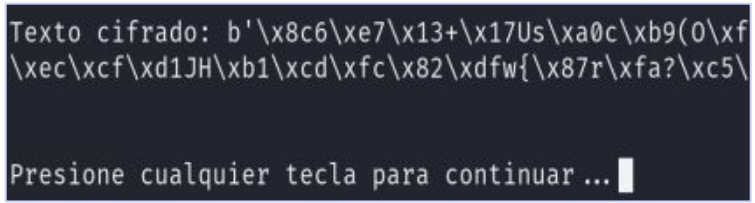
7. Presionar cualquier tecla. Se mostrará la clave pública.



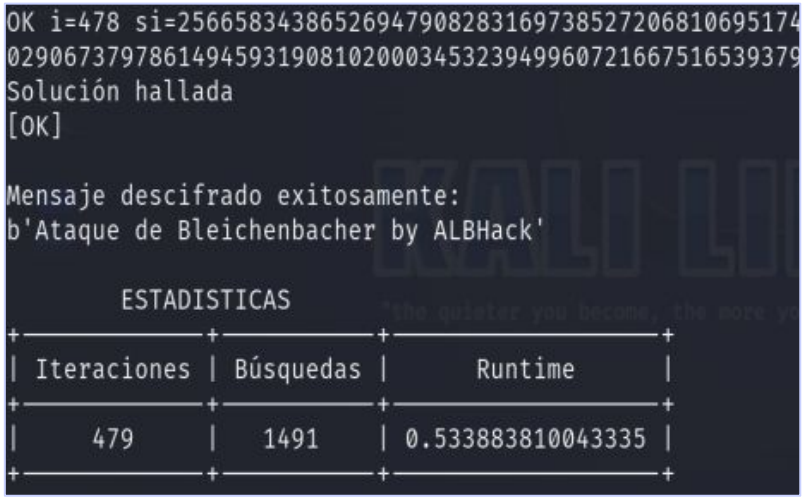
8. Tras un mensaje de advertencia, se nos pide ingresar una frase a encriptar.



9. Se mostrará el texto cifrado (se muestra un recorte).



Finalmente, el ataque resulta exitoso.



Se muestran las salidas para el ataque contra claves de 1024 y 2048 bits.

```
705727138150922492125141
Solución hallada
[OK]

Mensaje descifrado exitosamente:
b'Ataque de Bleichenbacher by ALBHack'

        ESTADISTICAS
+--------------+------------+--------------------+
| Iteraciones | Búsquedas |      Runtime       |
+--------------+------------+--------------------+
|     992      |    2773    | 23.663416385650635 |
+--------------+------------+--------------------+
```

Para un host Windows 10 con 8gb de RAM y un guest Kali Linux con 2gb de RAM, los tiempos de ejecución para 512, 1024 y 2048 bits resultaron ser 0.5 segs, 23 segs y 61 segs. respectivamente.

```
644275512177491050043245226482701640903607 6
Solución hallada
[OK]

Mensaje descifrado exitosamente:
b'Ataque de Bleichenbacher by ALBHack'

        ESTADISTICAS
+--------------+------------+--------------------+
| Iteraciones | Búsquedas |      Runtime       |
+--------------+------------+--------------------+
|     2017     |    3737    | 61.3011748790741 |
+--------------+------------+--------------------+
```

¡Sigamos trabajando!