Tarea 3

Criptografía y seguridad 2020-2

Fecha límite de entrega: 15 de abril

Indicaciones

- Resuelve cada uno de los ejercicios, todos tienen el mismo valor.
- Puede hacerse de forma individual o en pareja, en caso de hacerla en pareja basta que se entregue una solución.
- **Sube tu tarea solo cuando estés completamente seguro de que es correcta, ya que solo se puede subir una vez.**
- Escribe los cálculos que realizaste, o en caso de haber usado otra herramienta (como un programa) indícalo en tu respuesta.
- Para los ejercicios que requieren programar utiliza un lenguaje de los siguientes: Python, Java, C o Haskell. Anexa tu código fuente.
- Organiza tus archivos en un archivo `.zip`, incluyendo los datos de los alumnos, y súbelo en `https://forms.gle/AwuaLmMR25HhtuyKA`

Ejercicios

1. Se puede aplicar un test para probar un generador de números aleatorios siguiendo el siguiente teorema:

   La probabilidad de que $\mathrm{mcd}(x, y) = 1$ *para dos enteros* x, y *escogidos al azar es* $6/\pi^2$.

   Usa este resultado para probar tres algoritmos generadores de números aleatorios:

   - La función `randint` del módulo `random` de Python.
   - El algoritmo RC4 usando semillas de 4 bytes (32 bits).
   - El archivo `/dev/urandom` de tu computadora.

   Para los tres casos usa enteros en un rango $[0, n]$, escogiendo n con un valor de al menos $2^{8 \cdot 7} - 1$ (¿cuántos bytes se necesitan para guardar este número?). Para cada algoritmo calcula el valor de π usando 100, 1000 y 10000 parejas (x, y), de forma que puedas llenar la siguiente tabla con los valores obtenidos

	randint	RC4	urandom
100			
1000			
10000			

2. Supongamos que Alicia y Bartolo se quieren mandar mensajes cifrados pero solo tienen una llave secreta compartida k de 128 bits. Para enviar un mensaje m hacen lo siguiente:

   - Se escoge una cadena aleatoria s de 48 bits.
   - Se obtiene el mensaje cifrado $c = \mathrm{RC4}(s \,\|\, k) \oplus m$.
   - Se manda la pareja (s, c).

   Responde lo siguiente.

   a) ¿Qué tiene que hacer Alicia para recuperar el mensaje claro cuando recibe (s, c)?

   b) Si un adversario puede ver una lista de mensajes $(s_1, c_1), (s_2, c_2), \ldots$ que fueron enviados, ¿cómo puede comprobar que dos mensajes c_i, c_j fueron cifrados con el mismo flujo generado por RC4?

   c) Usando la paradoja del cumpleaños calcula aproximadamente cuántos mensajes tendría que enviar Alicia para que se repita el flujo generado por RC4.

d) Con el método de Alicia y Bartolo y tomando en cuenta lo anterior, ¿cuántos mensajes pueden cifrarse de forma segura? ¿Cuántos serían si omiten la cadena s en todo el proceso?

3. El profesor Bartolo guarda las tareas de sus alumnos encriptadas con un cifrador de flujo. Cada tarea es un archivo de texto que comienza con lo siguiente

```
No. de cuenta: XXXXXXXXXX
Tarea N
Respuestas...
```

con los 10 dígitos del número de cuenta del alumno, y no se incluyen otros datos personales del alumno.

Como Carlos no entregó la última tarea, maliciosamente quiere entrar a la computadora del profesor e intercambiar el número de cuenta en la tarea de Alicia por el suyo, aunque no tiene forma de conseguir la llave que usa el profesor para encriptar los archivos.

a) Suponiendo que Carlos conoce el número de cuenta de Alicia y además tiene acceso al archivo cifrado, ¿qué debe hacerle a este archivo para intercambiar su número de cuenta por el de Alicia? Así cuando el profesor descifre el archivo que originalmente era de Alicia ahora tendrá el número de cuenta de Carlos.

b) Ejemplifica la situación usando RC4 para encriptar un archivo con número de cuenta 0123456789 y posteriormente hacer el cambio para que el número de cuenta sea 1231231231.

4. En la práctica es muy común usar la biblioteca OpenSSL para varias tareas relacionadas con criptografía, como generar llaves, cifrar, codificar, entre otras. También hay una aplicación que se ejecuta con el comando `openssl`, que normalmente se incluye en distribuciones de Linux y en MacOS.

Investiga cómo usar `openssl` para encriptar archivos con los cifradores de flujo RC4 y ChaCha20. ChaCha20 usa además una cadena llamada vector de inicialización (IV), para este ejercicio usa una cadena de 16 bytes cero. Obtén el cifrado de los siguientes mensajes con las respectivas llaves y escribe el resultado en Base64.

a) $m =$ `Este es un mensaje secreto`, $k =$ `Una llave muuy larga de 32 bytes` (ASCII).

b) $m =$ `060606060606` (son 6 bytes en hexadecimal), $k = \texttt{00}^{16}$ (llave de 16 bytes cero).

c) $m =$ `Este es un mensaje secreto060606060606` (ASCII, al final hexadecimal), $k =$ `Una llave muuy larga de 32 bytes`.